NON-TIDAL BENTHIC MONITORING DATABASE: Version 3.5

DATABASE DESIGN DOCUMENTATION AND DATA DICTIONARY

1 June 2013

Prepared for:
United States Environmental Protection Agency
Chesapeake Bay Program
410 Severn Avenue
Annapolis, Maryland 21403

Prepared By:
Interstate Commission on the Potomac River Basin
51 Monroe Street, PE-08
Rockville, Maryland 20850

Prepared for

United States Environmental Protection Agency
Chesapeake Bay Program
410 Severn Avenue
Annapolis, MD 21403

By

Jacqueline Johnson
Interstate Commission on the Potomac River Basin

To receive additional copies of the report please call or write:
The Interstate Commission on the Potomac River Basin
51 Monroe Street, PE-08
Rockville, Maryland 20850
301-984-1908

Funds to support the document The Non-Tidal Benthic Monitoring Database: Version 3.0; Database Design Documentation And Data Dictionary was supported by the US Environmental Protection Agency Grant CB-CBxxxxxxxxxx-x

Disclaimer

The opinion expressed are those of the authors and should not be construed as representing the U.S. Government, the US Environmental Protection Agency, the several states or the signatories or Commissioners to the Interstate Commission on the Potomac River Basin: Maryland, Pennsylvania, Virginia, West Virginia or the District of Columbia.

TABLE OF CONTENTS

BACKGROUND

In 1996, the Chesapeake Executive Counsel adopted the "Strategy for Increasing Basin-wide Public Access to Chesapeake Bay Information". This strategy calls for the Chesapeake Bay Program partners to develop the Chesapeake Bay Information Management System (CIMS). CIMS will electronically link a variety of information sources about the Bay and rivers and make this information available to anyone—from students, to scientists, to citizens groups—electronically through the Internet and World Wide Web. The information targeted by CIMS includes technical and public information, educational material, environmental indicators, policy documents and scientific data.

As a result of the CIMS initiative, the Bay Program is working to establish a system of distributed databases. In the ideal system, a CBP database would be created and managed by the data originator, reside with the data originator, and made directly available from the data originator's institution on an Internet server. This system has several advantages over the traditional single data repository. Primarily, the people with the most expertise and knowledge about the data, the originators, will manage the data. Additional advantages include reduced cost due to elimination of intermediate data handling at a central repository, and decreased time between collection and release of the data.

The key to the success of a distributed data management system lies in the willingness of the data generators to take responsibility for the quality and maintenance for their data as well as and in their adherence to the established data standards. As part of the implementation of CIMS, the Living Resources Biological Data Management program has chosen to design a series of relational database structures for managing various types of Chesapeake Bay related Monitoring data. Once developed, these database designs are populated with the existing data and were intended to be turned over to the data generators for long term maintenance. The advantage of this implementation scheme is that the data generators of like data types will be running databases of identical structure. The use of these identical database structures will facilitate implementing search engines and XML data exchanges between multiple sources. The design of these databases has been done as a joint effort between the data generators and the CBP technical staff. The participation of the data generators in this process has provided critical expertise about the data and its usage, producing a better database.

The original Non-Tidal Benthic monitoring data base was designed by Staff of the Interstate Commission on the Potomac River Basin in 2007 using Microsoft Access. The database design was based loosely on the Ecological Data Application System database developed by Tetra Tech, Inc. for EPA in 2000. The database design was modified to incorporate CIMS data standards migrated in to Microsoft SQL server in March 2010. During this migration, minor modifications were made to tables and fields to accommodate SQL Server and maintain continuity where possible with the Chesapeake Bay program Tidal Water Quality Monitoring Database. This updated document is not intended to provide a complete discussion of the concepts of a relational database. Instead, this document describes in detail the Non-Tidal Benthic Monitoring Database's revised structural design.

INTRODUCTION

THE NON-TIDAL BENTHIC MONITORING DATA

The non-tidal benthic database contains voluntarily reported in-stream macro-invertebrate survey data. The data were collected by natural resource agencies in Delaware, Maryland, New York, Pennsylvania, Virginia and West Virginia; Fairfax and Loudoun Counties in Virginia; Montgomery, Howard and Prince Georges counties in Maryland; the United States Forest Service, United States Environmental Protection Agency, United States Geological Survey, Virginia Commonwealth University, Maryland Volunteer Stream Waders Program, Interstate Commission on the Potomac River Basin and Susquehanna River Basin Commission. The benthic data derived from these sampling programs consists of sample enumerations for benthic species composition and abundances, ambient measurements of water quality parameters, and rapid assessments of habitat quality.

Data providers of non-tidal benthic data are responsible for the collection, processing, data compilation, and data quality assurance of biological monitoring data. Most sampling procedures are modifications of the Environmental Protection Agency's Rapid Bioassessment Protocols for Use in Stream and Wadeable Rivers (Plafkin et al. 1989). The majority of the data was collected under EPA approved quality assurance plans. The Living Resources Data Manger/Analyst is responsible for tracking receipt of data deliverables, final data QA checks, inclusion of data into long-term databases, and maintaining data user documentation. Non-tidal benthic data are received in varying increments in response to annual data calls in spring. Once data are received at the Data Center, they are inspected and undergo final processing and assurance checks before they are loaded into the long-term monitoring database.

RELATIONAL DATABASE CONCEPTS

In a relational database, data is stored in tables, which are linked to one another by common fields. Most tables are related to one another in a series of one to many relationships. In this type of arrangement, the one record in the "Parent" or one table is related to many records in the "child" or many table. The common fields are set as primary and/or foreign keys. The creation of relationships between tables using key fields allows for the enforcement of referential integrity. Referential integrity constraints prohibits the entering duplicate records or records into a child table containing a foreign key for which there is not an associated primary key in the parent table. This database also employs the use of auto-generated key field. An auto generated key field cannot be edited; it is a unique, sequential or random number automatically assigned to each new record added to the table. In the case of this database, auto-generated keys are assigned to unique records based on a combination of fields in the parent table. The auto-generated key is then added to a child table as part of its primary key. The principle advantage of an auto-generated key is that once assigned a table can be indexed and linked on one field instead of the combination of fields used to determine a unique record. This serves to increase the efficiency of the database and decrease data recovery time.

The following relational data structure for the Non-Tidal Benthic Monitoring data contains description of the primary data tables as well as the numerous lookup tables required to define in detail the codes contained in the primary tables. The primary tables contain the bulk of the data in the database and are related to one another by multiple key fields. The seven primary tables in the Non-Tidal Benthic database are TAB_EVENT, TAB_HABITAT_ASSESSMENT, TAB_IBI_METRICS, TAB_STATION, TAB_TAXONOMIC_COUNT TAB_TAXONOMIC_QUANTIFICATION TAB_WAREHOUSE AND TAB_WQ_DATA. The remaining associated look-up tables in the data base contain information supports the referential integrity of the database.

RELATIONAL DATABASE STRUCTURE

The following relational data structure for non-tidal benthic data contains descriptions of the primary data tables as well as the numerous lookup tables required to define in detail the codes contained in the primary tables. The table columns in this document used to describe the fields in the database tables are described below.

FIELD - This column contains the field name in the database table as well as the designation of the field as either a primary key (PK), a foreign key (FK), a not null (NN) field, a unique field (U) or a auto-generated key field (AK). Primary, foreign and auto-generated key fields, by definition, are not null fields. However, primary and foreign keys may contain zero length value fields. Fields, which are neither primary nor foreign key fields, but which have been designated as not null or unique are those fields deemed essential to certain applications of the database. In the case of auto generated primary keys there must be a unique clustered index (CI) for the table.

DESCRIPTION - This column contains a definition of the database table field.

TYPE (FORMAT) - This column specifies the field type as character, number, or date/time; it also includes the format of the field and the precision of the text value where appropriate. Currently accepted data types in Microsoft SQL server 2005 used in the CBP database include the following.

Exact Numerics

Type	From	To
BIGINT	-9,223,372,036,854,775,808	9,223,372,036,854,775,807
INT	-2,147,483,648	2,147,483,647
SMALLINT	-32,768	32,767
TINYINT	0	255
BIT	0	1
DECIMAL	-10^38 +1	10^38 –1
NUMERIC	-10^38 +1	10^38 –1
MONEY	-922,337,203,685,477.5808	+922,337,203,685,477,5807
SMALLMONEY	-214,748.3648	+214,748.3647

Numeric and decimal are fixed precision, scale data types, and are functionally equivalent.

Approximate Numeric

Type	From	To
FLOAT	-1.79E + 308	+1.79E + 308
REAL	-3.40E + 308	+3.40E + 308

Datetime and Smalldatetime

Type	From	To
DATETIME (3.33 milliseconds accuracy)	Jan 1, 1753	Dec 31, 9999
SMALLDATETIME (1 minute accuracy)	Jan 1, 1900	Jun 6, 2079

Character Strings

Type	Description
CHAR	Fixed-length non-Unicode character data with a maximum length of 8,000 characters.
VARCHAR	Variable-length non-Unicode data with a maximum of 8,000 characters.
VARCHAR(MAX)	Variable-length non-Unicode data with a maximum of 2^{31} characters.
TEXT	Variable-length non-Unicode data with a maximum length of 2,147,483,647 characters.

Unicode Character Strings

Type	Description
NCHAR	Fixed-length non-Unicode character data with a maximum length of 4,000 characters.
NVARCHAR	Variable-length non-Unicode data with a maximum of 4,000 characters.
NVARCHAR(MAX)	Variable-length non-Unicode data with a maximum of 2^{30} characters.
NTEXT	Variable-length non-Unicode data with a maximum length of 1,073,741,823 characters.

Binary Strings

Type	Description
binary	Fixed-length binary data with a maximum length of 8,000 bytes.
varbinary	Variable-length binary data with a maximum length of 8,000 bytes.
varbinary(max)	Variable-length binary data with a maximum length of 2^{31} bytes
image	Variable-length binary data with a maximum length of 2,147,483,647 bytes.

Other Data Types

Type	Description
sql_variant	Stores values of various SQL Server-supported data types, except text, ntext, and timestamp
timestamp	Stores a database-wide unique number that gets updated every time a row gets updated.
uniqueidentifier	Stores a globally unique identifier (GUID).
xml	Stores XML data. You can store xml instances in a column or a variable
cursor	A reference to a cursor.
table	Stores a result set for later processing.

LENGTH (BYTES) - This column specifies the maximum character length of a field as well as the internal database storage requirement.

NON-TIDAL BENTHIC DATABASE STRUCTURE

PRIMARY DATA TABLES

Within the current design, the primary tables are the TAB_EVENT, TAB_HABITAT_ASSESSMENT, TAB_IBI_METRICS, TAB_RAREFACTION_COUNTS, TAB_STATIONS, TAB_TAXONOMIC_COUNT, TAB_TAXONOMIC_QUANTIFICATION TAB_WQ_DATA and TAB_WAREHOUSE. The TAB_EVENT table contains all sampling event data for all types of sample collection events. The remaining table all store data of the type designated in the table name.

TAB_EVENT

Field Name	Description	Data Type	Length
EVENT_ID (AK,PK)	EVENT_ID KEY-(PROJECT_ID+STATION_ID+SAMPLE_DATE+SAMPLE_TIME) AN AUTO GENERATED NUMBER	int	
PROJECT_ID (CI,NN,FK,)	PROJECT_ID-A monitoring project identifier	tinyint	
STATION_ID (CI,NN,FK)	STATION_ID- Sampling station identifier	varchar	50
SAMPLE_DATE_TIME (CI,NN)	SAMPLE_DATE_TIME-Sampling date and time.	smalldatetime	
R_DATE (NN)	DATA VERSION DATE-Date denoting when data records were entered in to database	smalldatetime	
EVENT_TYPE (FK,NN)	EVENT_TYPE-Sampling event type code	tinyint	
EVENT_ LATITUDE (NN)	STATION LATITUDE-Reported Station Latitude in decimal degrees coordinates)	real	7
EVENT_ LONGITUDE (NN)	STATION LONGITUDE-Reported Station Longitude in negative decimal degrees coordinates)	real	7
SITE_TYPE _CODE (FK,NN)	SITETYPE-Sampling Site Selection Criteria Code	char	2
SAMPLE_DATE	SAMPLE_DATE- Sample date only a data base generated field	smalldatetime	
SAMPLE_TIME	SAMPLE_TIME- Sample time only a data base generated field	smalldatetime	
LL_DATUM (FK,NN)	LL_DATUM-Code specifying the associated datum of the latitude and longitude values	char	5

Notes:

1) GENERAL: Every event for which there were sample taken of any kind must have a record in this table. Event records must be loaded into the database first and all unique records are assigned a SURVEY_ID number. The EVENT_ID must then be merged onto all other data based on the key fields before data may be loaded into any other primary data tables.

2) EVENT_ID: The actual primary key for this table is a composite key base on the following fields: PROJECT_ID, STATION_ID, SAMPLE_DATE_TIME. An Auto-Key number is generated for each unique combination of these fields.

3) SAMPLE_TIME AND SAMPLE_DATE_TIME: Sampling events where sample collection time is missing, SAMPLE_TIME has been set to 00:00 (Mid-Night).

TAB_HABITAT_ASSESSMENT

Field Name	Description	Data Type	Length
EVENT_ID (PK,FK,NN)	EVENT_ID- Primary Key- an auto-generated field	int	
SAMPLE_ NUMBER (PK,NN)	SAMPLE NUMBER- Number of sample collected at Station(replicate number)	tinyint	6
HABITAT_ REPORTING_ PARAMETER (PK,FK,NN)	HABITAT_REPORTING_PARAMETER- Habitat monitoring parameter code	varchar	15
REPORTING_ PARAMETER_ VALUE (NN)	REPORTING_PARAMETER_VALUE-Numeric value of parameter	smallint	
HAB_METHOD (FK,NN)	HAB_METHOD CODE- Method code identifying field/laboratory analysis procedure	char	6

NOTES:

1) GENERAL: This table stores information relating to the assessment of stream habitat condition. This table assumes that all habitat information is collected using either EPA's RBP habitat assessment protocols or some modification of there of. All data is on a scale from 0 to 20 with 20 indicating the highest possible score.

2) EVENT_ID: The primary key for this table is a composite key base on the following fields: EVENT_ID, SAMPLE_NUMBER, HAB_METHOD and HABITAT_REPORTING_PARAMETER. The composite key of EVENT_ID is base on the combination following fields: PROJECT_ID, STATION_ID, and SAMPLE_DATE_TIME. EVENT_ID is an Auto-Key number is generated for each unique combination of these fields in the TAB _EVENT table and must be merged on to data before it can be loaded into this table.

TAB_IBI_METRICS

Field Name	Description	Data Type	Length
EVENT_ID (PK,FK,NN)	EVENT_ID- Primary Key- an auto-generated field	smallint	
SAMPLE_ NUMBER (PK,NN)	SAMPLE NUMBER-Number of sample collected at Station(replicate number)	tinyint	
IBI_ PARAMETER (PK,FK,NN)	IBI_REPORTING_PARAMETER-Index of biotic integrity (IBI)reporting metric code	varchar	30
IBI_VALUE (NN)	IBI _VALUE-Numeric value of IBI metric	real	
IBI_SCORE (NN)	IBI _SCORE-Numeric score of IBI metric	bigint	
R_DATE (NN)	R_DATA - Date denoting when data records were entered in to database	smalldatetime	
IBI_METHOD (PK,FK,NN)	IBI_METHOD-IBI calculation method code	char	5

NOTES:

1) GENERAL: This table stores calculated Index of Biotic Integrity (BIBI) metrics and scored values. The IBI values in this table may be from state protocols or from CBP protocols. Sites may have IBI values calculated under multiple methods.

2) EVENT_ID: The primary key for this table is a composite key base on the following fields: EVENT_ID, SAMPLE_NUMBER, IBI_METHOD and IBI_REPORTING_PARAMETER. The composite key of EVENT_ID is base on the combination following fields: PROJECT_ID, STATION_ID, and SAMPLE_DATE_TIME. EVENT_ID is an Auto-Key number is generated for each unique combination of these fields in the TAB _EVENT table and must be merged on to data before it can be loaded into this table.

TAB_STATIONS

Field Name	Description	Data Type	Length
STATION_ID (PK,NN)	STATION_ID- Sampling station identifier	varchar	50
WATERBODY_ NAME	WATER BODY_NAME-Water body name based on USGS HUC 12 designation where station is located	varchar	40
SITE_ LOCATION	SITE_LOCATION- data generator provided description of Water body (i.e. river, bay, creek, run) in which the station is located	varchar	255
STRAHLER_ STREAM_ ORDER	STRAHLER_STREAM_ORDER-Strahler Stream Order based on USGS NHDPlus Version 1.0, 1:100,000-scale NHD stream order assignments	smallint	
CATCHMENT_ AREA		bigint	
ECOREGION_ LEVEL_4 (FK, NN)	ECOREGION_LEVEL_4- EPA level 4 ecoregions	char	3
LATITUDE (NN)	STATION LATITUDE- Station Latitude rounded to 5 significant digits in decimal degrees	decimal	9,5
LONGITUDE (NN)	STATION LONGITUDE-Station Longitude rounded to 5 significant digits in negative decimal degrees	decimal	9,5
FIPS (FK,NN)	FIPS CODE-Federal Information Processing System code	char	5
HUC_12 (FK,NN)	12 DIGIT HUC CODE- 12-digit USGS hydrologic unit code	char	12
LL_DATUM (FK,NN)	LL_DATUM CODE Code specifying the associated geographic datum of the latitude and longitude values	char	5
R_DATE (NN)	R_DATA - Date denoting when data records were entered in to database	smalldatetime	
HABITAT_TYPE (FK,NN)	HABITAT_TYPE- Habitat Description Code	varchar	15
KARST (FK,NN)	KARST-Karst Designation	varchar	10
UTM_X	UTM_X- X Coordinate in UTM Zone 18 North Projection	integer	
UTM_Y	UTM_Y- Y Coordinate in UTM Zone 18 North Projection	integer	

NOTES:

1) GENERAL: The TAB_STATIONS table contains station names and associated geographic attributes. Note that station name must be unique in this database. The designated name is the first name associated with the first sampling event recorded a latitude and longitude position rounded to 5 significant digits. CBPO GIS specialist will determine Catchment_Area later based on standard CIMS protocol.

TAB_RAREFACTION_COUNTS

Field Name	Description	Data Type	Length
EVENT_ID (PK, FK, NN)	EVENT_ID- Primary Key- an auto-generated field	int	
SAMPLE_ NUMBER (PK,NN)	SAMPLE NUMBER-Number of sample collected at Station(replicate number)	tinyint	
TSN (PK,FK,NN)	TSN- Interagency Taxonomic Identification System (ITIS) Taxon Serial Numbers	char	7
LIFE_STAGE_ CODE (PK,FK,NN)	SPECIES LIFE STAGE CODE-Chesapeake bay program species life stage code	char	3
REPORTING_ PARAMETER (NN)	REPORTING_PARAMETER- Reporting Parameter	varchar	15
REPORTING_ VALUE	REPORTING _VALUE-Numeric value of parameter reported	real	
REPORTING_ UNITS (FK,NN)	REPORTING_UNITS- Reporting parameter units of measure	varchar	10
BIO_METHOD (FK,NN)	BIO_METHOD CODE- Code identifying the lab method used to enumerate biological sample	char	6
G_METHOD (PK,FK,NN)	G_ METHOD –CBP Code for Sampling Gear used for sample collection	char	3
SAMPLE_TYPE (FK,NN)	SAMPLE_TYPE- Code denoting type of sample collected for analysis	varchar	5
R_DATE (NN)	R_DATA - Date denoting when data records were entered in to database	smalldatetime	

1) GENERAL: This table stores processed information relating to measurements of benthic species abundance. Calculated benthic parameters such as species diversity indices which are computed directly from a sample's raw count increase in value up to an asymptote as the number of individuals counted per sample increases. Thus, count metrics from monitoring programs with different sample counting protocols should not be directly compared. In the current Non-Tidal Benthic database, methodologies for sample counting use subsample count ranging from 100 to 500 organism counts. The 100- and 200-count protocols are the commonly used among the most active data providers. Further complicating comparisons, some monitoring programs in the watershed perform genus-level counts of benthic macroinvertebrates while others perform family-level counts. In order to "standardized" the data collected under varying subsample and taxonomic identification level protocols. Counts are first "rolled up to the family-level" then data is rarafacted or randomly sub sampled to a achieve samples with a standard subsample size. The rarefaction data counts generated in the current data base are based on a 100 organism count at the family level.

2) GENERAL: This database uses the Interagency Taxonomic Identification System (ITIS) Taxon Serial Numbers (TSN) for species identification within the database. For species with no TSN values temporary Chesapeake Bay TSNs are generated until a species can be submitted to ITIS for recognition. If no taxonomic analysis was performed on a sample, there should be no records present for that sample in this table. Samples which were analyzed for biotic content but no organisms were found are denoted by the presence of a "Empty Sample Record" denoted by a TSN value of BAY0291.

3) EVENT_ID: The primary key for this table is a composite key base on the following fields:EVENT_ID, SAMPLE_NUMBER, HAB_METHOD and HABITAT_REPORTING_PARAMETER. The composite key of EVENT_ID is base on the combination following fields:PROJECT_ID, STATION_ID, and SAMPLE_DATE_TIME. EVENT_ID is an Auto-Key number is generated for each unique combination of these fields in the TAB _EVENT table and must be merged on to data before it can be loaded into this table.

TAB_TAXONOMIC_COUNT

Field Name	Description	Data Type	Length
EVENT_ID (PK, FK, NN)	EVENT_ID- Primary Key- an auto-generated field	int	
SAMPLE_ NUMBER (PK,NN)	SAMPLE NUMBER-Number of sample collected at Station(replicate number)	tinyint	
TSN (PK,FK,NN)	TSN- Interagency Taxonomic Identification System (ITIS) Taxon Serial Numbers	char	7
LIFE_STAGE_ CODE (PK,FK,NN)	SPECIES LIFE STAGE CODE-Chesapeake bay program species life stage code	char	3
REPORTING_ PARAMETER (NN)	REPORTING_PARAMETER- Reporting Parameter	varchar	15
REPORTING_ VALUE	REPORTING _VALUE-Numeric value of parameter reported	real	
REPORTING_ UNITS (FK,NN)	REPORTING_UNITS- Reporting parameter units of measure	varchar	10
BIO_METHOD (FK,NN)	BIO_METHOD CODE- Code identifying the lab method used to enumerate biological sample	char	6
G_METHOD (PK,FK,NN)	G_ METHOD –CBP Code for Sampling Gear used for sample collection	char	3
SAMPLE_TYPE (FK,NN)	SAMPLE_TYPE- Code denoting type of sample collected for analysis	varchar	5

NOTES:

1) GENERAL: This table stores information relating to measurements of benthic species abundance and composition. This database uses the Interagency Taxonomic Identification System (ITIS) Taxon Serial Numbers (TSN) for species identification within the database. For species with no TSN values temporary Chesapeake Bay TSNs are generated until a species can be submitted to ITIS for recognition. If no taxonomic analysis was performed on a sample, there should be no records present for that sample in this table. Samples which were analyzed for biotic content but no organisms were found are denoted by the presence of a “Empty Sample Record” denoted by a TSN value of BAY0291.

2) EVENT_ID: The primary key for this table is a composite key base on the following fields:EVENT_ID, SAMPLE_NUMBER, HAB_METHOD and HABITAT_REPORTING_PARAMETER. The composite key of EVENT_ID is base on the combination following fields:PROJECT_ID, STATION_ID, and SAMPLE_DATE_TIME. EVENT_ID is an Auto-Key number is generated for each unique combination of these fields in the TAB _EVENT table and must be merged on to data before it can be loaded into this table.

TAB_TAXONOMIC_QUATIFICATION

Field Name	Description	Data Type	Length
EVENT_ID (PK, FK, NN)	EVENT_ID- Primary Key- an auto-generated field	int	
SAMPLE_ NUMBER (PK,NN)	SAMPLE NUMBER-Number of sample collected at Station(replicate number)	tinyint	
TSN (PK,FK,NN)	TSN- Interagency Taxonomic Identification System (ITIS) Taxon Serial Numbers	char	7
REPORTING_ PARAMETER (NN)	REPORTING_PARAMETER- Reporting Parameter	varchar	15
REPORTING_ PARAMETER_ PREVALENCE	REPORTING_PARMETER_PREVALENCE- Prevalence category code for reported organism	char	2
BIO_METHOD (FK,NN)	BIO_METHOD CODE- Code identifying the lab method used to ennumerate biological sample	char	6
G_METHOD (PK,FK,NN)	G_ METHOD –CBP Code for Sampling Gear used for sample collection	char	3
SAMPLE_TYPE (FK,NN)	SAMPLE_TYPE- Code denoting type of sample collected for analysis	varchar	5

1) GENERAL: This table stores information relating to measurements of benthic species abundance and composition expressed in categories. This database uses the Interagency Taxonomic Identification System (ITIS) Taxon Serial Numbers (TSN) for species identification within the database. For species with no TSN values temporary Chesapeake Bay TSNs are generated until a species can be submitted to ITIS for recognition. If no taxonomic analysis was performed on a sample, there should be no records present for that sample in. this table. Samples which were analyzed for biotic content but no organisms were found are denoted bythe presence of a “Empty Sample Record” denoted by a TSN value of BAY0291.

2) EVENT_ID: The primary key for this table is a composite key base on the following fields: EVENT_ID, SAMPLE_NUMBER, HAB_METHOD and HABITAT_REPORTING_PARAMETER. The composite key of EVENT_ID is base on the combination following fields: PROJECT_ID, STATION_ID, and SAMPLE_DATE_TIME. EVENT_ID is an Auto-Key number is generated for each unique combination of these fields in the TAB _EVENT table and must be merged on to data before it can be loaded into this table.

TAB_WQ_DATA

Field Name	Description	Data Type	Length
EVENT_ID (PK, FK,NN)	EVENT_ID -The unique monitoring event identifier.	integer	
SAMPLE_ TYPE (PK, FK,NN)	SAMPLE_TYPE-Code identifying the how the sample or measurement was obtained.	varchar	5
SAMPLE_ REPLICATE_ TYPE (PK, FK, NN)	SAMPLE_REPLICATE_TYPE- Code identifying each sample when multiple samples or measurements are collected or a water sample is split into multiple samples.	varchar	7
LAYER (NN, FK)	LAYER-Code identifying the water column layer where the sample or measurement was obtained.	char	2
DEPTH (NN)	DEPTH-The depth (in meters) where the water sample or measurement was obtained.	real	
REPORTING_ PARAMETER (PK, FK, NN)	REPORTING_PARAMETER-Code identifying the parameter name.	varchar	15
REPORTED_ VALUE	REPORTED_VALUE-The reported value of the parameter.	real	
REPORTING _UNITS (FK, NN)	REPORTING_UNITS-Code identifying the units of measurement in which a parameter is reported.	varchar	10
QUALIFIER (FK)	QUALIFIER-Code that identifies if the reported value was outside of the method detection limit.	varchar	2
WQ_ METHOD (PK, FK,)	WQ_METHOD-Code identifying the field/laboratory test procedure used to measure the parameter value.	char	4
LAB (FK)	LAB-Code identifying the lab where the sample was analyzed.	varchar	10
PROBLEM_ CODE (FK)	PROBLEM_CODECode identifying any sample analysis problems.	varchar	2
DETAILS	DETAIL- Additional information about the data record.	varchar	375
D_FLAG	D_FLAG- Detection limit flag	bit	
B_FLAG	B_FLAG- Qualifier flag	bit	

1) GENERAL: This table stores water quality parameters collected along with benthic and habitat data. Data parameter names and reporting units are converted to CBP standard names and units. Method codes and quality assurance flags are assigned for each parameter based on data provider provided documentation and comply with CBP standard in the tidal and non-tidal water quality databases.

2) EVENT_ID: The primary key for this table is a composite key base on the following fields:EVENT_ID, SAMPLE_NUMBER, HAB_METHOD and HABITAT_REPORTING_PARAMETER. The composite key of EVENT_ID is base on the combination following fields:PROJECT_ID, STATION_ID, and SAMPLE_DATE_TIME. EVENT_ID is an Auto-Key number is generated for each unique combination of these fields in the TAB _EVENT table and must be merged on to data before it can be loaded into this table.

TAB_WAREHOUSE

Field Name	Description	Data Type	Length
EVENT_ID (PK,FK,NN)	EVENT_ID- Primary Key- an auto-generated field	smallint	
SAMPLE_ NUMBER (PK,NN)	SAMPLE NUMBER-Number of sample collected at Station(replicate number)	tinyint	
IBI_ PARAMETER (PK,FK,NN)	IBI_REPORTING_PARAMETER-Index of biotic integrity (IBI)reporting metric code	varchar	30
IBI_VALUE (NN)	IBI _VALUE-Numeric value of IBI metric	real	
R_DATE (NN)	R_DATA - Date denoting when data records were entered in to database	smalldatetime	

NOTES:

1) GENERAL: This table stores metrics cacluated on raw and rarefaction (n=100) data.

2) EVENT_ID: The primary key for this table is a composite key base on the following fields: EVENT_ID, SAMPLE_NUMBER, IBI_METHOD and IBI_REPORTING_PARAMETER. The composite key of EVENT_ID is base on the combination following fields: PROJECT_ID, STATION_ID, and SAMPLE_DATE_TIME. EVENT_ID is an Auto-Key number is generated for each unique combination of these fields in the TAB _EVENT table and must be merged on to data before it can be loaded into this table.

PRINCIPAL LOOK-UP TABLES

The primary tables also contain many fields containing codes that are described or defined in detail in related lookup tables. By creating one-to-many relationships between lookup tables and the primary data tables and enforcing referential integrity, data managers are restricted to entering only valid lookup table values into the primary data tables. Again, this provides an automatic layer of quality assurance that will improve the utility of the database for all users.

TAB_AGENCY

Field	Description	Type	Length
AGENCY_CODE (PK,NN)	AGENCY_CODE- Agency code.	char	7
AGENCY (NN)	AGENCY- Agency Name	varchar	100
DIVISION	DIVISION-Agency Division	varchar	100
STREET_ ADDRESS	STREET_ADDRESS-Agency street address	varchar	100
CITY	CITY-City	varchar	30
STATE	STATE- US Postal Service State code	char	
ZIP	ZIP-US Postal Service zip code	varchar	10
PHONE	PHONE-Agency phone number	char	14
DIRECTOR	DIRECTOR-Name of Agency Director	varchar	50
EMAIL	EMAIL- Email contact for Agency Director	varchar	100

1) GENERAL: This table stores information related exclusively to AGENCY_CODE in the TAB_PROGRAM table. The agency code identifies the organization responsible for processing and storing collected data. These codes, taken directly from the currently in the water quality, and living resources databases. As data from other sources (e.g. DCRA, AACO) is added, their AGENCY codes must first be added to this table. Currently accepted AGENCY designations are as follows:

AGENCY_CODE	AGENCY
DNREC	Deleware Department of Natural Resources and Environmental Control
FC-DPW	Frederick County Department of Public Works
FC-SPS	Fairfax County Department of Public Works and Environmental Services
HC_DPW	Howard County Department of Public Works
LC-DBD	Louden County Department Of Building And Development
MC-SPS	Montgomery County Department of Environmental Protection
MDDNR	Maryland Department of Natural Resources
NYDEC	New York Department Of Environmental Conservation
PADEP	Pennsylvania Department of Environmental Protection
PGC-DER	Prince George's County Department of Environmental Resources
SRBC	Susquehanna River Basin Commission
USEPA	United States Environmental Protection Agency
USFS	United States Forest Service
USGS	United States Geological Survey
VADEQ	Virginia Department of Environmental Quality
VCU	Virginia Commonwelth University
WVDEP	West Virginia Deparment of Environmental Protection

TAB_CBP_MSTR

Field Name	Description	Data Type	Length
TSN_NUM	TAXON SERIAL NUMBER- ITIS Serial Number for Species Identification (defined as a numeric value)	int	
TSN (PK)	TAXON SERIAL NUMBER- ITIS Serial Number for Species Identification (defined as a fixed 7 character value with leading zeros)	char	7
NODC_CODE	NODC_CODE-National Oceanographic Data Center Taxonomic Codes-Version 8	varchar	12
SYN	SYNONYM FLAG- CIMS flag denoting species with synonymous name	varchar	2
LATIN_NAME (NN)	LATIN_NAME-Species Latin/Scientific Name, spelling based on ITIS standard	varchar	45
TAXON_LEVEL	PHYLOGENIC CLASSIFICATION-Denotes Phylogenic Level (phylum, class, order, etc)	varchar	6
COMMON_ NAME	COMMON NAME- Species Common Name	varchar	40
R_DATE (NN)	DATA VERSION DATE-Date denoting when data records were entered in to database	smalldate time	

1) General: This table stores information in relating to the identification of species in the TAB_TAXONOMIC_COUNT and TAB_TAXONOMIC_QUANTIFICATION tables. The list includes listings for all types of organisms including phytoplankton, zooplankton, fish and benthos. This database uses the Interagency Taxonomic Identification System (ITIS) Taxon Serial Numbers (TSN) for species identification within the database. For species with no TSN values, temporary Chesapeake Bay TSN's are generated until a species can be submitted to ITIS for recognition. The use of the standardized TSN codes among all Bay Program databases will allows for queries by species from multiple State and National biological databases.

2) TSN: Each species has been given its ITIS Taxonomic Serial Number (TSN). The ITIS (Interagency Taxonomic Information System) is a partnership of federal agencies working together to improve the organization of, and access to, standardized nomenclature. As part of this system a national, easily accessible database with reliable information on species names and their hierarchical classification has been established. The database is reviewed periodically to ensure high quality with valid classifications, revisions, and additions of newly described species. As part of this effort all Federal agencies have been asked to adopt the use of TSN code which assigns each recognized species a permanent number. The TSN allows a species to be tracked over time regardless of changes in name and taxonomic classification. TSN also provides a uniform key field for database development and species identification across multiple organizations. When used in conjunction with the NODC, the TSN overcomes the problem of numeric changes in the NODC code whenever species are reclassified. Temporary codes are assigned to taxa that are recognized in the scientific literature but have not been assigned an NODC Code and a TSN. The value BAYxxxx has been assigned to all taxa without TSN. A temporary NODC code is developed for each unassigned taxon based on its known taxonomy and its species name. For example, the beginning couplets of the NODC code which reflect the known phylogeny of an unassigned taxon are combined with letters from its species name to form a temporary code. The most up to date TSN numbers are available at http://www.itis.gov/. The most recent *Comprehensive List of Chesapeake Bay Basin Species 2007* can be found at https://archive.chesapeakebay.net/pub/Living_Resources/species2007.pdf .

3) NODCCODE: All species on the list have been assigned at least partial National Oceanographic Data Center (NODC) Taxon Codes (Version 8.0). The NODC Taxon Code is a hierarchical system of numerical

codes used to represent the scientific names and phylogeny of organisms. The code links the Linnean system of biological nomenclature to a numerical schema that facilitates modern methods of computerized data storage and retrieval. An NODC code contains a maximum of 12 digits partitioned into 2-digit couplets. Each couplet represents one or more levels of the taxonomic hierarchy. For example,

Digit	Represents
1-2	Phylum
3-4	Class and/or Order
5-6	Family
7-8	Genus
9-10	Species
11-12	Subspecies

One drawback of the NODC code is it changes over time to reflect current changes in taxonomic classifications. However, it provides data analysts with a very useful tool for sorting organisms into taxonomic groups.

4) SYN: Synonymous species are denoted in the table TAB_CBP_MSTR by a flag field named SYN. A code of S means a name is an ITIS recognized synonym and SA indicates the name is the accepted name for the taxa. Synonymous species will have identical NODC Taxon Codes.

5) TAXON_LEVEL: **T**he phylogentic levels for all taxa in the TAB_CBP_MSTR are denoted not only by NODC_CODE but also a TAXON_LEVEL code. Taxon levels are assigned through the Linnean system of biological nomenclature as implemented in ITIS. Currently accepted TAXON_LEVELS and DESCRIPTION designations are as follows:

TAXON_LEVEL	DESCRIPTION
CLS	CLASS
DIV	DIVISION
FAM	FAMILY
GEN	GENUS
GRP	GROUP
HYB	HYBRED
IFC	INFRA-CLASS
IFO	INFRA-ORDER
NON	NON SPECIFIC LEVEL
ORD	ORDER
PHY	PHYLUM
SBC	SUB-CLASS
SBF	SUB-FAMILY
SBO	SUB-ORDER
SBP	SUB-PHYLUM
SGEN	SUB-GENUS
SPC	SUPER-CLASS
SPE	SPECIES
SPO	SUPER-ORDER
SSP	SUB-SPECIES
TRI	TRIBE
VAR	VARIETY

TAB_CONVERT_TAXA

Field Name	Description	Data Type	Length
TSN	TAXON SERIAL NUMBER- ITIS Serial Number for Species Identification (defined as a fixed 7 character value with leading zeros)	char	7
NODC_CODE	NODC_CODE-National Oceanographic Data Center Taxonomic Codes-Version 8	nvarchar	12
LEVEL	PHYLOGENIC CLASSIFICATION-Denotes Phylogenic Level (phylum, class, order, etc)	nvarchar	6
LATIN_NAME	LATIN_NAME-Species Latin/Scientific Name, spelling based on ITIS standard associated with TSN	nvarchar	45
Family	FAMILY LATIN_NAME-Latin/Scientific Name of Family associated with TSN, spelling based on ITIS standard	nvarchar	45
parent_tsn	PARENT TAXON SERIAL NUMBER- ITIS Serial Number for Species Identification (defined as a fixed 7 character value with leading zeros)	nvarchar	7
2_order_tsn	SECOND ORDER PARENT TAXON SERIAL NUMBER- ITIS Serial Number for Species Identification (defined as a fixed 7 character value with leading zeros)	nvarchar	7
family_tsn	FAMILY LEVEL TAXON SERIAL NUMBER- ITIS Serial Number for Species Identification (defined as a fixed 7 character value with leading zeros)	nvarchar	7
R_date	DATA VERSION DATE-Date denoting when data records were entered in to database	datetime	8

1) GENERAL: This table contain information to summarize genus and species level taxonomic identifications into family-level groupings. Data in this table is used by the two stored procedures in the database to generate calculated metrics which populate the TAB_WAREHOUSE table and the prep_rarefaction_data_v2.mdb database application. This table must be updated when ever new taxa are reported in the taxonomic data. This database uses the Interagency Taxonomic Identification System (ITIS) Taxon Serial Numbers (TSN) for species identification within the database. For species with no TSN values, temporary Chesapeake Bay TSN's are generated until a species can be submitted to ITIS for recognition. The use of the standardized TSN codes among all Bay Program databases will allows for queries by species from multiple State and National biological databases. For extended details please in regards to the summarization of data to the family level please see:

Buchanan, C., K. Foreman, J. Johnson, and A. Griggs. 2011. Development of a Basin-wide Benthic Index of Biotic Integrity for Non-Tidal Streams and Wadeable Rivers in the Chesapeake Bay Watershed: Final Report to the Chesapeake Bay Program Non- Tidal Water Quality Workgroup. ICPRB Report 11-1. Report prepared for the US Environmental Protection Agency, Chesapeake Bay Program.

TAB_ECO_REGIONS

Field Name	Description	Data Type	Length
SUB-ECOREGION _LEVEL4 (PK,FK)	SUBECOREGION_LEVEL4- Denotes level 4 EPA Ecoregion Code	char	3
SUB-ECOREGION_ LEVEL4_NAME (NN)	SUBECOREGION_LEVEL4_NAME- Denotes level 4 EPA Ecoregion Name	varchar	90
SUB-ECOREGION _LEVEL3 (NN)	SUBECOREGION_LEVEL3- Denotes level 3 EPA Ecoregion Code	char	5
SUB-ECOREGION _LEVEL3_NAME (NN)	SUBECOREGION_LEVEL3_NAME- Denotes level 3 EPA Ecoregion Name	varchar	90
CBP_IBI_ REGION	CBP_IBI_REGION- Cheapeake Bay Program Non-Tidal IBI Region Code	varchar	10
CBP_IBI_ REGION_NAME	CBP_IBI_REGION_NAME- Cheapeake Bay Program Non-Tidal IBI Region Name	varchar	50

1) GENERAL: This table stores geographic information related exclusively to SUBECOREGION_LEVEL4 codes in the TAB_STATIONS TABLE. This table contains information to type of sample collected during an event. The following list of data types represent those which either directly measured in the field or analyzed in the laboratory. Additional codes may be added as needed. Currently accepted DATA_TYPE and DESCRIPTION designations are as follows:

SUBECOREGION_ LEVEL4	SUBECOREGION_ LEVEL4_NAME	ECOREGION_ LEVEL3	ECOREGION_ LEVEL3_NAME	CBP_IBI_ REGION	CBP_IBI_ REGION_NAME
45c	Carolina Slate Belt	45	Piedmont	45_64	Piedmont
45e	Northern Inner Piedmont	45	Piedmont	45_64	Piedmont
45f	Northern Outer Piedmont	45	Piedmont	45_64	Piedmont
45g	Triassic Basins	45	Piedmont	45_64	Piedmont
58h	Reading Prong	58	Northeastern Highlands	60plus	Northern Appalachians
60a	Glaciated Low Plateau	60	Northern Appalachian Plateau and Uplands	60plus	Northern Appalachians
60b	Northeastern Uplands	60	Northern Appalachian Plateau and Uplands	60plus	Northern Appalachians
60D	Finger Lakes Uplands and Gorges	60	Northern Allegheny Plateau	60plus	Northern Appalachians
60e	Glaciated Allegheny Hills	60	Northern Allegheny Plateau	60plus	Northern Appalachians
61b	Mosquito Creek/Pymatuning Lowlands	61	Erie Drift Plain	60plus	Northern Appalachians
61c	Low Lime Drift Plain	61	Erie Drift Plain	60plus	Northern Appalachians
62a	Pocono High Plateau	62	North Central Appalachians	62	Northern Appalachians
62b	Low Poconos	62	North Central Appalachians	62	Northern Appalachians
62c	Glaciated Allegheney High Plateau	62	North Central Appalachians	62	Northern Appalachians
62d	Unglaciated Allegheney High Plateau	62	North Central Appalachians	62	Northern Appalachians
62e	Low Catskills	62	North Central Appalachians	62	Northern Appalachians
63a	Delaware River Terraces and Uplands	63	Middle Atlantic Coastal Plain	63	Middle Atlantic Coastal Plain

SUBECOREGION_ LEVEL4	SUBECOREGION_ LEVEL4_NAME	ECOREGION_ LEVEL3	ECOREGION_ LEVEL3_NAME	CBP_IBI_ REGION	CBP_IBI_ REGION_NAME
63b	Chesapeake-Pamlico Lowlands and Tidal Marshes	63	Middle Atlantic Coastal Plain	63	Middle Atlantic Coastal Plain
63c	Swamps and Peatlands	63	Middle Atlantic Coastal Plain	63	Middle Atlantic Coastal Plain
63d	Virginian Barrier Islands and Coastal Marshes	63	Middle Atlantic Coastal Plain	63	Middle Atlantic Coastal Plain
63e	Mid-Atlantic Flatwoods	63	Middle Atlantic Coastal Plain	63	Middle Atlantic Coastal Plain
63f	Delmarva Uplands	63	Middle Atlantic Coastal Plain	63	Middle Atlantic Coastal Plain
64a	Triassic Lowlands	64	Northern Piedmont	45_64	Piedmont
64b	Trap Rock and Conglomerate Uplands	64	Northern Piedmont	45_64	Piedmont
64c	Piedmont Uplands	64	Northern Piedmont	45_64	Piedmont
64d	Piedmont Limestone/Dolomite Lowlands	64	Northern Piedmont	45_64	Piedmont
65m	Rolling Coastal Plain	65	Southeastern Plains	65	Southeastern Plains
65n	Chesapeake Rolling Coastal Plain	65	Southeastern Plains	65	Southeastern Plains
66a	Northern Igneous Ridges	66	Blue Ridge	6_679Ridge	Highlands
66b	Northern Sedimentary and Metasedimentary Ridges	66	Blue Ridge	6_679Ridge	Highlands
66c	New River Plateau	66	Blue Ridge	6_679Ridge	Highlands
66d	Southern Crystaline Ridges and Mountains	66	Blue Ridge	6_679Ridge	Highlands
66e	Southern Sedimentary Ridges	66	Blue Ridge	6_679Ridge	Highlands
66f	Limestone Valleys and Coves	66	Blue Ridge	6_679Ridge	Highlands
67a	Northern Limestone/Dolomite Valleys	67	Ridge and Valley	67Valleys	Valleys
67b	Northern Shale Valleys	67	Ridge and Valley	67Valleys	Valleys
67c	Northern Sandstone Ridges	67	Ridge and Valley	6_679Ridge	Highlands
67d	Northern Dissected Ridges and Knobs	67	Ridge and Valley	6_679Ridge	Highlands
67e	Anthracite Subregion	67	Ridge and Valley	67Valleys	Valleys
67f	Southern Limestone/Dolomite Valleys and Low Rolling Hills	67	Ridge and Valley	67Valleys	Valleys
67g	Southern Shale Valleys	67	Ridge and Valley	67Valleys	Valleys
67h	Southern Sandstone Ridges	67	Ridge and Valley	6_679Ridge	Highlands
67i	Southern Dissected Ridges and Knobs	67	Ridge and Valley	6_679Ridge	Highlands
69a	Forested Hills and Mountains	69	Central Appalachians	6_679Ridge	Highlands
69b	Uplands and Valleys of Mixed Land Use	69	Central Appalachians	6_679Ridge	Highlands
69c	Greenbriar Karst	69	Central Appalachians	6_679Ridge	Highlands
69d	Dissected Appalachian Plateau	69	Central Appalachians	6_679Ridge	Highlands
69e	Cumberland Mountain Thrust Block	69	Central Appalachians	6_679Ridge	Highlands
70a	Permian Hills	70	Western Allegheny Plateau	60plus	Northern Appalachians
70b	Monongahela Transition Zone	70	Western Allegheny Plateau	60plus	Northern Appalachians
70c	Pittsburgh Low Plateau	70	Western Allegheny Plateau	60plus	Northern Appalachians
70f	Ohio/Kentucky Carboniferous Plateau	70	Western Allegheny Plateau	60plus	Northern Appalachians

TAB_EVENT_TYPE

Field Name	Description	Data Type	Length
EVENT_ TYPE_ID (PK)	EVENT_TYPE_ID- Primary Key (EVENT_TYPE_CODE) AN AUTO GENERATED NUMBER	int	
EVENT_TYPE (NN,U)	EVENT_TYPE_CODE- Code denotes type of sample collected	char	2
EVENT_TYPE_ DESCRIPTION (NN)	EVENT_TYPE_DESCRIPTION-Detailed description of sampling event.	varchar	50

1) GENERAL: This table stores information related exclusively to EVENT_TYPE codes in the TAB_EVENT table. This table contains information to type of sample collected during an event. The following list of data types represent those which either directly measured in the field or analyzed in the laboratory. Additional codes may be added as needed. Currently accepted EVENT_TYPE and EVENT_TYPE_DESCRIPTION designations are as follows:

EVENT_TYPE	EVENT_TYPE_DESCRIPTION
BEN	BENTHIC ONLY
BEN\HAB	BENTHIC\HABITAT
HAB	HABITAT ONLY
QBEN	QUALITATIVE BENTHIC ONLY
QBEN\HAB	QUALITATIVE BENTHIC\HABITAT
WQ	WATER QUALITY ONLY
WQ\BEN	WATER QUALITY\BENTHIC
WQ\BEN\HAB	WATER QUALITY\BENTHIC\HABITAT
WQ\HAB	WATER QUALITY\HABITAT
WQ\QBEN	WATER QUALITY\QUALITATIVE BENTHIC
WQ\QBEN\HAB	WATER QUALITY\QUALITATIVE BENTHIC\HABITAT

TAB_FAMILY

Field Name	Description	Data Type	Length
TSN	TAXON SERIAL NUMBER- ITIS Serial Number for Species Identification (defined as a fixed 7 character value with leading zeros)	Char	7
Family	LATIN_FAMILY NAME-Species Latin/Scientific Name, spelling based on ITIS standard	varchar	45
TOLARANCE_VALUE	FAMILY LEVEL TOLARANCE SCORE-Values based upon Merritt, R. W., K. W. Cummins, and M. B. Berg. 2008. An Introduction to the Aquatic Insects of North America, Fourth Edition. Kendall Hunt Publishing Co., Iowa, pp. 1158.	Float	8
GUILD	FAMILY FEEDING GUILD-Family Feeding Guild Designation	varchar	10
EPTD	EPTD TAXA DESIGNATION-Ephemeroptera, Plecoptera, Trichoptera or Diptera Family Designation	varchar	1
HABITAT	Habitat Designation-Family Habitat Guild Designation	varchar	10
BECK_CLASS	BECK'S TOLARANCE-Beck's Index group Designation	Float	8
INTOERANT_URBAN	MBSS URBAN TOLARANCE INDEX Score	nvarchar	10
EXCLUSION_TAXA	EXCLUSION TAXA FLAG-Denotes Organisms not deemed to be true infaunal organisms	nvarchar	1
NOTES	NOTES-General comments on family	Memo	
R_DATE	DATA VERSION DATE-Date denoting when data records were entered in to database	datetime	8
Tolarance_class	TOLARANCE_CLASS-Denotes Tolarance score >=7 (T) or <=3 (I)	nvarchar	1
ORTHOCLADIINAE	ORTHOCLADIINAE FLAG-Denotes Organism belonging to an Orthocladine Sub Family	nvarchar	1
CHRONONMIDAE	CHRONONMIDAE FLAG-Denotes Organism belonging to an CHRONONMIDAE Sub Family	nvarchar	1
SIMULIIDAE	SIMULIIDAE FLAG-Denotes Organism belonging to an SIMULIIDAE Sub Family	nvarchar	1
GOLD_TAXA	GOLD INDEX FLAG- Denotes gastropods, oligochaetes and Diptera individuals	nvarchar	1
SEL_EPTD	SELECT EPTD TAXA-Select EPTD taxa used to caclulate	nvarchar	1

Field Name	Description	Data Type	Length
BMWP	BMWP SCORE-Biological monitoring working party Tolerance Score	float	8
NON_INSECT	NON INSECT FLAG-Non insect family flag	varchar	1
AMPHOPOD/ISOPOD	AMPHOPOD OR ISOPOD FLAG-Amphopod or Isopod family flag	varchar	1
NCO	NCO FLAG-Denotes no chironomidae and oligochaeta family	varchar	1

1) GENERAL: This table contain family-level tolerance values, functional feeding groups, and habit characteristics assigned from a combination of sources. Data in this table is used by the two stored procedures in the database to generate calculated metrics which populate the TAB_WAREHOUSE table. This table must be updated when ever new families are reported in the taxonomic data. This database uses the Interagency Taxonomic Identification System (ITIS) Taxon Serial Numbers (TSN) for species identification within the database. For species with no TSN values, temporary Chesapeake Bay TSN's are generated until a species can be submitted to ITIS for recognition. The use of the standardized TSN codes among all Bay Program databases will allows for queries by species from multiple State and National biological databases. For extended details please see:

Buchanan, C., K. Foreman, J. Johnson, and A. Griggs. 2011. Development of a Basin-wide Benthic Index of Biotic Integrity for Non-Tidal Streams and Wadeable Rivers in the Chesapeake Bay Watershed: Final Report to the Chesapeake Bay Program Non- Tidal Water Quality Workgroup. ICPRB Report 11-1. Report prepared for the US Environmental Protection Agency, Chesapeake Bay Program.

TAB_FIPS

Field Name	Description	Data	Length
FIPS (PK)	FIPS CODE-Federal Information Processing System code	char	5
STATE_INITIALS	STATE INITIAL DESIGNATION-Federal Information Processing System code Two-letter state postal abbreviation	char	2
COUNTY_NAME	COUNTY DESIGNATION-County name	varchar	30

1) GENERAL: This table contains (FIPS) Federal Information Processing System codes identifying state and county type of field samples taken at given site. This code is used in the TAB_STATION tables. Additional codes may be added as needed. Currently accepted FIPS CODES, STATE AND COUNTY designations are as follows:

11001	DC	WASHINGTON
10001	DE	KENT
10003	DE	NEW CASTLE
10005	DE	SUSSEX
24001	MD	ALLEGANY
24003	MD	ANNE ARUNDEL
24005	MD	BALTIMORE
24510	MD	BALTIMORE CITY
24009	MD	CALVERT
24011	MD	CAROLINE
24013	MD	CARROLL
24015	MD	CECIL
24017	MD	CHARLES
24019	MD	DORCHESTER
24021	MD	FREDERICK
24023	MD	GARRETT
24025	MD	HARFORD
24027	MD	HOWARD
24029	MD	KENT
24031	MD	MONTGOMERY
24033	MD	PRINCE GEORGES
24035	MD	QUEEN ANNES
24039	MD	SOMERSET
24037	MD	ST MARYS
24041	MD	TALBOT
24043	MD	WASHINGTON
24045	MD	WICOMICO
24047	MD	WORCESTER
36003	NY	ALLEGANY
36007	NY	BROOME
36015	NY	CHEMUNG
36017	NY	CHENANGO
36023	NY	CORTLAND
36025	NY	DELAWARE
36043	NY	HERKIMER
36051	NY	LIVINGSTON
36053	NY	MADISON
36065	NY	ONEIDA
36067	NY	ONONDAGA
36069	NY	ONTARIO
36077	NY	OTSEGO
36095	NY	SCHOHARIE
36097	NY	SCHUYLER
36101	NY	STEUBEN
36107	NY	TIOGA
36109	NY	TOMPKINS
36123	NY	YATES
42001	PA	ADAMS
42009	PA	BEDFORD
42011	PA	BERKS
42013	PA	BLAIR
42015	PA	BRADFORD
42021	PA	CAMBRIA
42023	PA	CAMERON
42027	PA	CENTRE
42029	PA	CHESTER
42033	PA	CLEARFIELD
42035	PA	CLINTON
42037	PA	COLUMBIA
42041	PA	CUMBERLAND
42043	PA	DAUPHIN
42047	PA	ELK
42055	PA	FRANKLIN
42057	PA	FULTON
42061	PA	HUNTINGDON
42063	PA	INDIANA
42067	PA	JUNIATA
42069	PA	LACKAWANNA
42071	PA	LANCASTER
42075	PA	LEBANON
42079	PA	LUZERNE
42081	PA	LYCOMING
42083	PA	MCKEAN
42087	PA	MIFFLIN
42093	PA	MONTOUR
42097	PA	NORTHUMBERLAND
42099	PA	PERRY
42105	PA	POTTER
42107	PA	SCHUYLKILL
42109	PA	SNYDER
42111	PA	SOMERSET
42113	PA	SULLIVAN
42115	PA	SUSQUEHANNA
42117	PA	TIOGA
42119	PA	UNION
42127	PA	WAYNE
42131	PA	WYOMING
42133	PA	YORK
51001	VA	ACCOMACK
51003	VA	ALBEMARLE
51510	VA	ALEXANDRIA
51005	VA	ALLEGHANY
51007	VA	AMELIA
51009	VA	AMHERST
51011	VA	APPOMATTOX
51013	VA	ARLINGTON
51015	VA	AUGUSTA
51017	VA	BATH
51019	VA	BEDFORD
51023	VA	BOTETOURT
51029	VA	BUCKINGHAM
51031	VA	CAMPBELL
51033	VA	CAROLINE
51036	VA	CHARLES CITY
51550	VA	CHESAPEAKE CITY
51041	VA	CHESTERFIELD
51043	VA	CLARKE
51570	VA	COLONIAL HEIGHTS
51045	VA	CRAIG
51047	VA	CULPEPER
51049	VA	CUMBERLAND
51053	VA	DINWIDDIE
51057	VA	ESSEX
51059	VA	FAIRFAX
51610	VA	FALLS CHURCH
51061	VA	FAUQUIER
51065	VA	FLUVANNA
51069	VA	FREDERICK
51630	VA	FREDERICKSBURG
51071	VA	GILES
51073	VA	GLOUCESTER
51075	VA	GOOCHLAND
51079	VA	GREENE
51650	VA	HAMPTON
51085	VA	HANOVER
51087	VA	HENRICO
51091	VA	HIGHLAND
51093	VA	ISLE OF WIGHT
51095	VA	JAMES CITY
51097	VA	KING AND QUEEN
51099	VA	KING GEORGE
51101	VA	KING WILLIAM
51103	VA	LANCASTER
51107	VA	LOUDOUN

51109	VA	LOUISA
51680	VA	LYNCHBURG
51113	VA	MADISON
51115	VA	MATHEWS
51119	VA	MIDDLESEX
51121	VA	MONTGOMERY
51125	VA	NELSON
51127	VA	NEW KENT
51700	VA	NEWPORT NEWS
51710	VA	NORFOLK
51131	VA	NORTHAMPTON
51133	VA	NORTHUMBERLAND
51135	VA	NOTTOWAY
51137	VA	ORANGE
51139	VA	PAGE
51730	VA	PETERSBURG
51740	VA	PORTSMOUTH
51145	VA	POWHATAN
51147	VA	PRINCE EDWARD
51149	VA	PRINCE GEORGE
51153	VA	PRINCE WILLIAM
51157	VA	RAPPAHANNOCK
51159	VA	RICHMOND
51760	VA	RICHMOND CITY
51161	VA	ROANOKE
51163	VA	ROCKBRIDGE
51165	VA	ROCKINGHAM
51171	VA	SHENANDOAH
51177	VA	SPOTSYLVANIA
51179	VA	STAFFORD
51800	VA	SUFFOLK
51181	VA	SURRY
51810	VA	VIRGINIA BEACH
51187	VA	WARREN
51193	VA	WESTMORELAND
51830	VA	WILLIAMSBURG
51199	VA	YORK
54003	WV	BERKELEY
54023	WV	GRANT
54027	WV	HAMPSHIRE
54031	WV	HARDY
54037	WV	JEFFERSON
54057	WV	MINERAL
54063	WV	MONROE
54065	WV	MORGAN
54071	WV	PENDLETON

TAB_G_METHOD

Field Name	Description	Data Type	Length
G_METHOD (PK,FK)	GEAR METHOD CODE- CBP Code of Sampling Gear used for sample collection	char	3
G_METHOD_ DESCRIPTION (NN)	GEAR DESCRIPTION-CBP biological field sampling gear descriptions	varchar	30
G_METHOD_ DETAILS	DETAILED DESCRIPTION- Detailed Description of Sampling Gear Including Dimensions	varchar	50

1) GENERAL: This table stores information relating to the type of gear used to collect samples for all analysis. This table stores identification codes for sampling gear used primary in the TAB_BIOTA_EVENT TABLE. The primary key in this table is defined by G_METHOD. Please note that some gears appear in table multiple times due to gear dimentions see d Additional codes may be added as needed. Currently accepted G_METHODS designations are as follows:

G_METHOD	G_METHOD_DESCRIPTION	G_METHOD_DETAILS
01	HAND DREDGE	
02	DREDGE	
03	ARTIFICIAL SUBSTRAIT	UNSPECIFIED
04	DIATOMER SLIDES	
05	CLARKE-BUMPUS SAMPLER	
06	PLANKTON TRAP	UNSPECIFIED
07	PLANKTON PUMP	UNSPECIFIED
08	PLANKTON NET	UNSPECIFIED
09	PLANKTON NET	500µ MESH
10	PLANKTON NET	NO. 20, 80µ MESH
11	PLANKTON NET	10µ MESH
12	BEAM PLANKTON LINE	
13	ANCHOR DREDGE	

G_METHOD	G_METHOD_DESCRIPTION	G_METHOD_DETAILS
14	HYDRAULIC GRAB	1200 SQ. CM
15	HAND CORE	45 SQ. CM
16	POST-HOLE DIGGER	200 SQ. CM
17	PONAR GRAB	200 SQ. CM
18	PONAR GRAB	1000 SQ. CM
19	PONAR GRAB	50 SQ. CM
20	BOX CORE GRAB	0.018 M2
21	VAN VEEN GRAB	0.07 M2
22	SHIPEK GRAB	0.04 M2
23	SEINE HAUL	UNSPECIFIED
24	SMITH-MACINTIRE GRAB	1000 SQ CM
25	SEINE NET	15 FT, 1/8 IN STRECH MESH
26	SEINE NET	50 FT, 1/2 IN STRECH MESH

G_METHOD	G_METHOD_DESCRIPTION	G_METHOD_DETAILS
27	SEINE NET	50 FT, 1/4 IN STRECH MESH
28	SEINE NET	200 FT , 1/2 IN STRECH MESH, NET 200X 20
29	SEINE NET	10 FT , 1/4 IN STRECH MESH, NET 10X4
30	TRAWL	UNSPECIFIED
31	OTTER TRAWL	6 FT, 1 IN. MESH, W/ 1/2 IN INNER LINER
32	OTTER TRAWL	25 FT, 1.24 IN. MESH, W/ 1/2 IN INNER LINER
33	TRAWL	15 FT SEMI-BALLON
34	TUCKER TRAWL	2 MM . MESH, 1 SQ. METER
35	CARGO_SLED	Cargo jellyfish sled
36	TRAWL	16 FT SEMI-BALLON, 1/2 IN MESH
37	OTTER TRAWL	10 FT, 1/4 IN. MESH, W/500 µ IN INNER LINER
38	MID-WATER TRAWL	5 FT, 1/4 IN. MESH, W/500 µ IN INNER LINER
39	VISUAL COUNT	
40	TRAP NET	3 x 6 FT, 1/2 IN MESH, 50 FT LEAD
41	ELECTROSHOCKER	
42	ECKMAN CAGE	
43	CAGE	
44	CATFISH TRAP	
45	CRAYFISH TRAP	
46	CRAB TRAP	
47	ANIMAL TRAP	
48	HOOK AND LINE FISHING	
49	DIP NET	
50	DIVER	
51	DIAPHRAGM PUMP	
52	CENTRIFUGAL PUMP	
53	RESERVED	
54	POUND NET	
55	EPIFAUNA PANELS	
56	PONAR GRAB	UNSPECIFIED
57	D-FRAME NET	500 MICRON MESH, 12 INCH DIAMETER
58	RETANGULAR DIP NET	0.5 METER BY 0.5 METERS
59	HAND PICK	
60	ENDICO CURRENT METER	
61	BRAINCON CURRENT METER	
62	SEDIMENT TRAP ARRAY	6- 3"X30" CUPS {BOYTON-CBL}

G_METHOD	G_METHOD_DESCRIPTION	G_METHOD_DETAILS
63	SEINE NET	50 FT, 1/4 IN MESH, NET 100X4 FT
64	BONGO NET	UNSPECIFIED
65	PURSE SEINE	
66	FYKE AND HOOP NETS	
67	POTS	
68	BOX TRAP	
69	PUSH NET	
70	GREAT LAKE SHOAL	1-2 INCHES
71	GREAT LAKE SHOAL	2-4 INCHES
72	GREAT LAKE SHOAL	4-7 INCHES
73	GREAT LAKE SHOAL	7-14 INCHES
74	BEAM TRAWL	
75	BONGO NET	202 µ, 20 CM OPENING, 0.76 M LENGTH
76	BONGO NET	202 µ, 50 CM OPENING, 4 M LENGTH
77	RESERVED	
78	SLAT TRAP	
79	RESERVED	
80	GIL NETS	
81	USNOL SPADE CORE	0.06 M2 SPADE BOX CORE
82	PONAR GRAB-ODU	
83	DOUBLE PONAR GRAB-VA DEQ	50 SQ CM
84	RESERVED	
85	MID-WATER TRAWL	
86	KICK NET	23 CM x 46 CM, MESH OPEN SIZE 0.8MM BY 0.9 MM
87	KICK NET	UNSPECIFIED
88	RESERVED	
89	D-FRAME NET	UNSPECIFIED
90	HESTER DENDY MULTIPLATE SAMPL	
91	SURBER SAMPLER	
92	KICK SEINE	
93	D-FRAME NET	600 MICRON, 12 INCH DIAMETER
94	KICK NET	600 MICRON, 1 SQUARE METER KICK SCREEN
96	HYDROLIC VAN VEEN GRAB	0.1 SQUARE METERS
97	YOUNG MODIFIED VAN VEEN GRAB	0.04 SQ M
98	PETITE PONAR GRAB	25 SQUARE CM
99	SMITH-MACINTIRE GRAB	0.3 SQUARE METER
100	SMITH-MACINTIRE GRAB	0.2 SQUARE METER

TAB_HABITAT_PARAMETERS

Field Name	Description	Data	Length
HABITAT_ REPORTING_ PARAMETER (PK)	HABITAT_REPORTING_PARAMETER- Habitat condition parameter name	varchar	15
HABITAT_ PARAMETER_ DESCRIPTION (NN)	HABITAT_REPORTING_PARAMETER_DESCRIPTION- Habitat condition parameter description	varchar	50

1) General: This table stores information related exclusively to the TAB_HABITAT_REPORTING_PARAMETER codes in codes in the TAB_HABITAT_ASSESSMENT table. This table contains information to parameter names. Currently accepted TAB_HABITAT_REPORTING_PARAMETER and HABITAT_REPORTING_PARAMETER_DESCRIPTION designations are as follows:

HABITAT_REPORTING _PARAMETER	HABITAT_REPORTING_ PARAMETER_DESCRIPTION
AESTH	AESTHETICS
BANKS	BANK STABILITY
BANKV	BANK VEGETATION
CH_ALT	CHANNEL ALTERATION
COVER	BOTTOM SUBSTRATE & AVAILABLE COVER
EMBED	EMBEDDEDNESS
EPI_SUB	EPIFAUNAL SUBSTRATE
FLOW	FLOW
GRAZE	GRAZING
INSTR	INSTREAM HABITAT
P_SUB	POOL SUBSTRATE
POOL	POOL/GLIDE QUALITY
REMOT	REMOTENESS
RIFF	RIFFLE/RUN/POOL RATIO
RIP_SC	RIPARIAN VEGETATION SCORE
RIP_W	RIPARIAN VEGETATION ZONE WIDTH
ROOT	NUMBER OF ROOTWADS
SED	SEDIMENTATION
SHAD	SHADING
SINU	SINUOSITY
THAL	THALWEG DEPTH
VEL_D	VELOCITY/DEPTH RATIO
WWID	WETTED WIDTH

TAB_HABITAT_TYPE

Field Name	Description	Data Type	Length
HABITAT_TYPE (PK)	HABITAT_TYPE- Habitat type code.	varchar	15
HABITAT_TYPE_ DESCRIPTION (NN)	HABITAT_TYPE_DESCRIPTION- Habitat type code description	varchar	50

1) General: This table stores information related exclusively to the TAB_HABITAT_REPORTING_PARAMETER codes in codes in the TAB_STATION table. This table contains information describing the habitat type of a sampling station. Currently accepted HABITAT_TYPE and HABITAT_TYPE_DESCRIPTION designations are as follows:

HABITAT_TYPE	HABITAT_TYPE_DESCRIPTION
G/P	Glide/pool
H/LGS	Mixed High & Low gradient stream
HGS	High gradient stream
LGS	Low gradient stream
NA	Not Available
POOL	Pool
RIF	Riffle
RIF/CPOM	Riffle/Coarse particulate organic matter
RIF/RUN	Riffle/Run
S	SPECIAL

TAB_HUCS_12

Field Name	Description	Data Type	Length
HUC12 (PK)	12 DIGIT HUC CODE- Subwatershed 12-digit USGS hydrologic unit code	char	12
HUC_2 (NN)	2 DIGIT HUC CODE-Region two digits of HUC_12	char	2
HUC_4 (NN)	4 DIGIT HUC CODE- Sub-region associated with the first four digits of HUC_12	char	4
HUC_6 (NN)	6 DIGIT HUC CODE- Basin associated with the first six digits of HUC_12	char	6
HUC_8 (NN)	8 DIGIT HUC CODE- Sub-Basin unit associated with the first eight digits of HUC_12	char	8
HUC_10 (NN)	10 DIGIT HUC CODE- Watershed unit associated with the first ten digits of HUC_12	char	10
ACRES (NN)	ACRES-Area of 12 digit HUC watershed in acres	real	
SQ_KM (NN)	SQ_KM- Area of 12 digit HUC watershed in square kilometers	real	
REGION_ DESCRIPTION (NN)	REGION_DESCRIPTION-Detailed Description of Region described by first two digits of HUC code	varchar	80
SUBREGION_ DESCRIPTION (NN)	SUBREGION_DESCRIPTION-Detailed Description of Region described by first four digits of HUC code	varchar	80
BASIN_ DESCRIPTION (NN)	BASIN_DESCRIPTION- Detailed Description of Region described by first six digits of HUC code	varchar	80
SUBBASIN_ DESCRIPTION (NN)	SUBBASIN_DESCRIPTION_ Detailed Description of Region described by first eight digits of HUC code	varchar	80
WATERSHED_ DESCRIPTION (NN)	WATERSHED_DESCRIPTION- Detailed Description of Region described by first ten digits of HUC code	varchar	80
SUBWATERSHED_ DESCRIPTION (NN)	SUBWATERSHED_DESCRIPTION- Detailed Description of Region described by twelve digits of HUC code	varchar	80

1) GENERAL: The TAB_HUCS8 TABLE contains 12-digit USGS hydrologic unit codes and descriptions The HUC8 code is the 12-digit USGS hydrologic unit code in which the station is located. The list that follows contains only the HUC and the associated cataloging unit description. Additional lookup tables related to this table may or may not be included in the final database design. These tables contain specific information related to the REGION, SUBREGION, ACCOUNTING_UNIT, and CATALOGING_UNIT fields (i.e. detailed description, states covered, and area in square miles). The currently accepted 12 digit HUC and SUBWATERSHED_DESCRIPTIONS are as follows:

HUC_12	SUBWATERSHED_DESCRIPTION
020402050502	Upper Christina River
020501010101	Ocquionis Creek
020501010102	Herkimer Creek-Canadarago Lake
020501010103	Oaks Creek
020501010201	Pleasant Brook
020501010202	Upper Cherry Valley Creek
020501010203	Middle Cherry Valley Creek
020501010204	Lower Cherry Valley Creek
020501010301	Upper Schenevus Creek
020501010302	Elk Creek
020501010303	Middle Schenevus Creek
020501010304	Lower Schenevus Creek
020501010401	Center Brook
020501010402	Middle Brook
020501010403	Upper Charlotte Creek
020501010404	Kortright Creek
020501010405	Middle Charlotte Creek
020501010406	Lower Charlotte Creek
020501010501	West Branch Otego Creek
020501010502	Upper Otego Creek
020501010503	Middle Otego Creek
020501010504	Lower Otego Creek
020501010601	Cripple Creek
020501010602	Shadow Brook
020501010603	Hayden Creek-Ostego Lake
020501010604	Red Creek-Susquehanna River
020501010605	Goodyear Lake-Susquehanna River
020501010606	Oneonta Creek-Susquehanna River
020501010701	Upper Wharton Creek
020501010702	Middle Wharton Creek
020501010703	Lower Wharton Creek
020501010801	Upper Butternut Creek
020501010802	Middle Butternut Creek
020501010803	Lower Butternut Creek
020501010901	North Winfield Creek
020501010902	West Branch Unadilla River
020501010903	Headwaters Unadilla River
020501010904	Beaver Creek
020501010905	Upper Unadilla River
020501010906	Center Brook
020501010907	Great Brook
020501010908	Middle Unadilla River
020501010909	Guilford Creek
020501010910	Lower Unadilla River
020501011001	Upper Ouleout Creek
020501011002	Treadwell Creek
020501011003	Middle Ouleout Creek
020501011004	Handsome Brook

HUC_12	SUBWATERSHED_DESCRIPTION
020501011005	Lower Ouleout Creek
020501011101	Otsdawa Creek
020501011102	Brier Creek-Susquehanna River
020501011103	Sand Hill Creek-Susquehanna River
020501011104	Carrs Creek
020501011105	Martin Brook-Susquehanna River
020501011201	Bennettsville Creek
020501011202	Kelsey Brook
020501011203	Yaleville Brook-Susquehanna River
020501011204	Wylie Brook
020501011205	Cornell Creek-Susquehanna River
020501011206	Belden Brook-Susquehanna River
020501011207	Ouaquaga Creek-Susquehanna River
020501011208	Occanum Creek-Susquehanna River
020501011209	Cascade Creek-Susquehanna River
020501011301	Shadigee Creek
020501011302	Upper Starrucca Creek
020501011303	Middle Starrucca Creek
020501011304	Lower Starrucca Creek
020501011305	Canawacta Creek-Susquehanna River
020501011306	Salt Lick Creek
020501011307	Trowbridge Creek
020501011308	Mitchell Creek-Susquehanna River
020501011309	Silver Creek
020501011310	Snake Creek
020501011311	Little Snake Creek
020501011312	Carlin Creek
020501011313	Park Creek-Susquehanna River
020501020101	Fabius Brook-West Branch Tioughnioga Creek
020501020102	Upper East Branch Tioughnioga Creek
020501020103	Labrador Creek
020501020104	Middle East Branch Tioughnioga Creek
020501020105	Cheningo Creek
020501020106	Lower East Branch Tioughnioga Creek
020501020201	Cold Brook
020501020202	Upper Little York Lake-West Br. Tioughnioga River
020501020203	Factory Creek
020501020204	Otter Creek
020501020205	Dry Creek-West Branch Tioughnioga River
020501020301	Headwaters Otselic River
020501020302	Mud Creek
020501020303	Upper Otselic River
020501020304	Brakel Creek
020501020305	Middle Otselic River
020501020306	Merrill Creek
020501020307	Lower Otselic River
020501020401	Trout Brook
020501020402	Gridley Creek

HUC_12	SUBWATERSHED_DESCRIPTION
020501020403	Upper Tioughnioga River
020501020404	Jennings Creek
020501020405	Culver Creek-Dudley Creek
020501020406	Middle Tioughnioga River
020501020407	Halfway Brook
020501020408	Lower Tioughnioga River
020501020501	Upper Sangerfield River
020501020502	Middle Sangerfield River
020501020503	Lower Sangerfield River
020501020504	Callahan Brook-Chenango River
020501020505	Payne Brook
020501020506	Eaton Brook-Chenango River
020501020507	South Lebanon Brook-Cold Spring Brook
020501020508	Crooked Brook-Pleasant Brook
020501020509	Stone Mill Brook-Chenango River
020501020601	Pleasant Brook
020501020602	Handsome Brook
020501020603	Mad Brook-Chenango River
020501020604	East Branch Canasawacta Creek
020501020605	Fly Creek-Chenango River
020501020606	Canasawacta Creek
020501020607	Thompson Creek-Chenango River
020501020608	Gilmore Brook-Chenango River
020501020609	Turner Creek-Fly Meadow Creek
020501020610	Lyon Brook-Chenango River
020501020701	Headwaters Genegantslet Creek
020501020702	Upper Genegantslet Creek
020501020703	Middle Genegantslet Creek
020501020704	Lower Genegantslet Creek
020501020801	Bowman Creek
020501020802	Mill Brook-Chenango River
020501020803	Padget Brook-Bear Brook
020501020804	Wheeler Brook-Chenango River
020501020805	Spring Brook-Chenango River
020501020806	Ockerman Brook-Chenango River
020501020807	Page Brook
020501020808	Osborne Creek
020501020809	Castle Creek
020501020810	Thomas Creek-Chenango River
020501030101	Upper Nanticoke Creek
020501030102	East Branch Nanticoke Creek
020501030103	Crocker Creek
020501030104	Middle Nanticoke Creek
020501030105	Lower Nanticoke Creek
020501030201	Little Choconut Creek-Susquehanna River
020501030202	Upper Chocohut Creek
020501030203	Middle Chocohut Creek
020501030204	Lower Chocohut Creek
020501030205	Patterson Creek-Susquehanna River

HUC_12	SUBWATERSHED_DESCRIPTION
020501030206	Tracy Creek-Susquehanna River
020501030301	Upper Catatonk Creek
020501030302	South Branch Catatonk Creek
020501030303	Willseyville Creek
020501030304	Middle Catatonk Creek
020501030305	Lower Catatonk Creek
020501030401	Headwaters East Branch Owego Creek
020501030402	Upper East Branch Owego Creek
020501030403	Wilson Creek
020501030404	Middle East Branch Owego Creek
020501030405	Upper West Branch Owego Creek
020501030406	Middle West Branch Owego Creek
020501030407	Doolittle Creek
020501030408	Lower West Branch Owego Creek
020501030409	Lower East Branch Owego Creek-Owego Creek
020501030501	Apalachin Creek
020501030502	Little Nanticoke Creek
020501030503	Pumpelly Creek-Susquehanna River
020501030504	Chambers Creek-Pipe Creek
020501030505	Hunts Creek-Susquehanna River
020501030601	Pony Hollow Creek
020501030602	Headwaters Cayuta Creek
020501030603	Upper Cayuta Creek
020501030604	Middle Cayuta Creek
020501030605	Lower Cayuta Creek
020501030701	Upper Wappasening Creek
020501030702	Middle Wappasening Creek
020501030703	Lower Wappasening Creek
020501030704	Sackett Creek-Susquehanna River
020501030705	Ellis Creek
020501030706	Parks Creek-Susquehanna River
020501030707	Satterlee Creek-Susquehanna River
020501040101	McHenry Valley Creek
020501040102	Karr Valley Creek
020501040103	Upper Canacadea Creek
020501040104	Lower Canacadea Creek
020501040201	Woodward Hollow
020501040202	Upper Bennetts Creek
020501040203	Middle Bennetts Creek
020501040204	Purdy Creek
020501040205	Lower Bennetts Creek
020501040301	South Branch Tuscarora Creek
020501040302	Upper Tuscarora Creek
020501040303	North Branch Tuscarora Creek
020501040304	Middle Tuscarora Creek
020501040305	Elk Creek
020501040306	Lower Tuscarora Creek
020501040401	Lime Kiln Creek
020501040402	Headwaters Canisteo River

HUC_12	SUBWATERSHED_DESCRIPTION
020501040403	Seeley Creek
020501040404	Big Creek
020501040405	Crosby Creek
020501040406	Upper Canisteo River
020501040407	Colonel Bills Creek
020501040408	Middle Canisteo River
020501040409	Tracy Creek
020501040410	Goodhue Creek
020501040411	Lower Canisteo River
020501040501	Young Hickory Hollow
020501040502	Upper Troups Creek
020501040503	Middle Troups Creek
020501040504	Lower Troups Creek
020501040601	Catlin Hollow
020501040602	Norris Brook
020501040603	Losey Creek
020501040604	Upper Crooked Creek
020501040605	Hills Creek
020501040606	Middle Crooked Creek
020501040607	Elkhorn Creek
020501040608	Lower Crooked Creek
020501040701	Gaffers Creek-Elk Run
020501040702	Bailey Creek-Mill Creek
020501040703	Painter Run-Mill Creek
020501040801	Headwaters Cowanesque River
020501040802	North Fork
020501040803	Elklick Run-Mill Creek
020501040804	Jemison Creek
020501040805	Upper Cowanesque River
020501040806	Holden Creek
020501040807	Thornbottom Creek
020501040808	MIddle Cowanesque River
020501040809	Lower Cowanesque River
020501040901	Headwaters Tioga River
020501040902	Johnson Creek
020501040903	Elk Run
020501040904	Upper Tioga River
020501040905	Canoe Camp Creek
020501040906	Corey Creek
020501040907	Middle Tioga River
020501040908	Mutton Lane Creek-Tioga River
020501040909	Glendening Creek
020501040910	Lower Tioga River
020501050101	Punky Hollow-Cohocton River
020501050102	Twelvemile Creek
020501050103	Neils Creek
020501050104	Reynolds Creek-Cohocton River
020501050105	Goff Creek
020501050106	Tenmile Creek-Cohocton River

HUC_12	SUBWATERSHED_DESCRIPTION
020501050201	Upper Fivemile Creek
020501050202	Middle Fivemile Creek
020501050203	Lower Fivemile Creek
020501050204	Campbell Creek
020501050205	Stocking Creek
020501050206	Knight Creek-Cohocton River
020501050207	Smith Run-Cohocton River
020501050301	Tobehanna Creek
020501050302	Lamoka Lake-Mud Creek
020501050303	Mud Creek
020501050304	Michigan Creek
020501050305	Dry Run
020501050306	Meads Creek
020501050307	Curtis Creek-Cohocton River
020501050308	Hodgmans Creek-Cohocton River
020501050401	Post Creek
020501050402	Cutler Creek-Chemung River
020501050403	Sing Sing Creek
020501050404	Whisky Creek-Chemung River
020501050405	Hendy Creek-Chemung River
020501050501	North Branch Newtown Creek
020501050502	Upper Newtown Creek
020501050503	Lower Newtown Creek
020501050504	Hammond Creek
020501050505	Mudlick Creek
020501050506	South Creek
020501050507	Seeley Creek
020501050508	Coldbrook Creek-Chemung River
020501050601	Upper Bentley Creek
020501050602	Middle Bentley Creek
020501050603	Lower Bentley Creek
020501050604	Baldwin Creek
020501050605	Shoemaker Mountain-Chemung River
020501050606	Wyncoop Creek
020501050607	Dry Brook-Chemung River
020501050608	Wolcott Creek-Chemung River
020501060101	North Branch Sugar Creek
020501060102	South Branch Sugar Creek
020501060103	Leonard Creek
020501060104	Mill Creek-Sugar Creek
020501060105	Tomjack Creek
020501060106	Browns Creek
020501060107	Bailey Run-Sugar Creek
020501060201	Little Schrader Creek
020501060202	Millstone Creek-Schrader Creek
020501060301	Headwaters Towanda Creek
020501060302	North Branch Towanda Creek
020501060303	South Branch Towanda Creek
020501060304	Towanda Creek-Susquehanna River

HUC_12	SUBWATERSHED_DESCRIPTION
020501060401	Bullard Creek
020501060402	Johnson Creek
020501060403	Parks Creek-Wysox Creek
020501060501	Spaulding Creek-Susquehanna River
020501060502	Laning Creek-Upper Susquehanna River
020501060503	Rummerfield Creek-Susquehanna River
020501060601	Lake Stream
020501060602	Deer Lick Creek-East Branch Wyalusing Creek
020501060701	Middle Branch Wyalusing Creek
020501060702	Gaylord Creek
020501060703	North Branch Wyalusing Creek
020501060704	Rockwell Creek-Wyalusing Creek
020501060705	Cold Creek-Wyalusing Creek
020501060801	Riley Creek
020501060802	West Branch Meshoppen Creek
020501060803	Thomas Creek-Meshoppen Creek
020501060804	Little Meshoppen Creek-Meshoppen Creek
020501060901	Upper Mehoopany Creek
020501060902	North Branch Mehoopany Creek
020501060903	Lower Mehoopany Creek
020501061001	Upper East Branch Tunkhannock Creek
020501061002	Lower East Branch Tunkhannock Creek
020501061101	Upper South Branch Tunkhannock Creek
020501061102	Lower South Branch Tunkhannock Creek
020501061201	Upper Tunhannock Creek
020501061202	Butler Creek
020501061203	Nine Partners Creek
020501061204	Middle Tunkhannock Creek
020501061205	Hop Bottom Creek
020501061206	Martins Creek
020501061207	Horton Creek
020501061208	Lower Tunkhannock Creek
020501061301	Leonard Creek
020501061302	Upper Bowman Creek
020501061303	Lower Bowman Creek
020501061401	Sugar Run
020501061402	Sugar Run Creek
020501061403	Tuscarora Creek
020501061404	Little Tuscurora Creek-Lower Susquehanna River
020501061405	Little Mehoopany Creek-Lower Susquehanna River
020501061406	Mill Run-Lower Susquehanna River
020501061407	Buttermilk Creek
020501061408	Gardner Creek
020501061409	Obendoffers Creek-Susquehanna River
020501070101	West Branch Lackawanna River
020501070102	East Branch Lackawanna River
020501070103	Lees Creek-Lackawanna River
020501070104	Rush Brook-Lackawanna River
020501070105	Leggetts Creek

HUC_12	SUBWATERSHED_DESCRIPTION
020501070106	Grassy Island Creek-Lackawanna River
020501070107	Roaring Brook
020501070108	Spring Brook
020501070109	City of Scranton-Lackawanna River
020501070110	Lackawanna River-Susquehanna River
020501070201	Abrahams Creek
020501070202	City of Wilkes-Barre-Mill Creek
020501070203	Toby Creek
020501070204	Sugar Notch Run-Solomon Creek
020501070205	City of Wilkes-Barre-Susquehanna River
020501070206	Nanticoke Creek
020501070207	Newport Creek
020501070208	Warrior Creek-Susquehanna River
020501070301	Harveys Lake-Harveys Creek
020501070302	Hunlock Creek
020501070303	Little Shickshinny Creek-Shickshinny Creek
020501070304	Little Wapwallopen Creek
020501070305	Big Wapwallopen Creek
020501070306	Wapwallopen Creek
020501070307	City of Berwick-Susquehanna River
020501070401	Little Nescopeck Creek-Nescopeck Creek
020501070402	Black Creek
020501070403	Nescopeck Creek-Susquehanna River
020501070501	Headwaters Huntington Creek
020501070502	Kitchen Creek
020501070503	Pine Creek
020501070504	Huntington Creek-Fishing Creek
020501070601	Kline Hollow Run-Little Fishing Creek
020501070602	Little Fishing Creek-Fishing Creek
020501070701	East Branch Fishing Creek
020501070702	West Branch Fishing Creek
020501070703	West Creek
020501070704	Raven Creek
020501070705	Mud Run-Green Creek
020501070706	Hemlock Creek
020501070707	Fishing Creek-Susquehanna River
020501070801	Little Catawissa Creek
020501070802	Tomicken Creek
020501070803	Messers Run-Catawissa Creek
020501070804	Beaver Run-Catawissa Creek
020501070805	Catawissa Creek-Susquehanna River
020501070901	Mugser Run-South Branch Roaring Creek
020501070902	Roaring Creek-Susquehanna River
020501071001	Briar Creek
020501071002	Tenmile Creek-Susquehanna River
020501071003	Logan Run
020501071004	Sechler Run
020501071005	Mahoning Creek
020501071006	City of Sunbury-Susquehanna River

HUC_12	SUBWATERSHED_DESCRIPTION
020502010101	Upper Chest Creek
020502010102	Middle Chest Creek
020502010103	Lower Chest Creek
020502010201	Upper Anderson Creek
020502010202	Lower Anderson Creek
020502010301	Headwaters Clearfield Creek
020502010302	Slate Lick Run
020502010303	Glendale Dam-Beaverdam Run
020502010304	Upper Clearfield Creek
020502010305	South Witmer Run-North Witmer Run
020502010306	Muddy Run
020502010307	Middle Clearfield Creek
020502010308	Lower Clearfield Creek
020502010309	Little Clearfield Creek
020502010310	Morgan Run-Lower Clearfield Creek
020502010401	Headwaters West Branch Susquehanna River
020502010402	Cush Creek
020502010403	Beaver Run-West Branch Susquehanna River
020502010404	Bear Run
020502010405	Bell Run
020502010406	Deer Run-West Branch Susquehanna River
020502010407	Montgomery Creek
020502010408	Curwensville Dam-West Branch Susquehanna River
020502010501	Beaver Run
020502010502	Upper Moshannon Creek
020502010503	Laurel Run
020502010504	Cold Stream
020502010505	Sixmile Run
020502010506	Middle Moshannon Creek
020502010507	Black Moshannon Creek
020502010508	Lower Moshannon Creek
020502010601	Headwaters Mosquito Creek
020502010602	Gifford Run-Mosquito Creek
020502010701	Lick Run
020502010702	Trout Run
020502010703	Moravian Run
020502010704	Deer Creek
020502010705	Sandy Creek
020502010706	Alder Run
020502010707	Millstone Run-West Branch Susquehanna River
020502010708	Upper Three Runs
020502010709	Saltlick Run-West Branch Susquehanna River
020502010710	Sterling Run
020502010711	Birch Island Run
020502010712	Lower Three Runs-West Branch Susquehanna River
020502010713	Burns Run-West Branch Susquehanna River
020502020101	Cowley Run
020502020102	Sinnemahoning Portage Creek-Driftwood Branch Sinnemahoning Creek

HUC_12	SUBWATERSHED_DESCRIPTION
020502020201	Elk Fork-Driftwood Branch Sinnemahoning Creek
020502020202	Clear Creek
020502020203	North Creek
020502020204	West Creek
020502020205	Hunts Run
020502020206	Sterling Run
020502020207	Driftwood Branch Sinnemahoning Creek-Sinnemahoning Creek
020502020301	Upper Bennett Branch Sinnemahoning Creek
020502020302	Kersey Run
020502020303	Laurel Run
020502020304	Medix Run
020502020305	Middle Bennett Branch Sinnemahoning Creek
020502020306	Spring Run
020502020307	Trout Run
020502020308	Dents Run
020502020309	East Branch Hicks Run
020502020310	Hicks Run
020502020311	Mix Run
020502020312	Lower Bennett Branch Sinnemahoning Creek
020502020401	Big Moores Run
020502020402	South Woods Branch
020502020403	Upper First Fork Sinnemahoning Creek
020502020404	Freeman Run
020502020405	East Fork Sinnemahoning Creek
020502020406	Middle First Fork Sinnemahoning Creek
020502020407	Lower First Fork Sinnemahoning Creek
020502020501	Wykoff Run
020502020502	Sinnemahoning Creek-West Branch Susquehanna River
020502030101	Little Kettle Creek
020502030102	Upper Kettle Creek
020502030103	Cross Fork
020502030104	Hammersley Fork
020502030105	Middle Kettle Creek
020502030106	Lower Kettle Creek
020502030201	Cooks Run
020502030202	Fish Dam Run-West Branch Susquehanna River
020502030203	Drury Run
020502030204	Paddy Run
020502030205	Hall Run-West Branch Susquehanna River
020502030301	Left Branch Young Womans Creek
020502030302	Young Womans Creek-West Branch Susquehanna River
020502030401	Hyner Run
020502030402	Rattlesnake Run-West Branch Susquehanna River
020502030403	Clendenin Branch Baker Run
020502030404	Baker Run
020502030405	North Fork Tangascootack Creek
020502030406	Tangascootack Creek
020502030407	Lick Run

HUC_12	SUBWATERSHED_DESCRIPTION
020502030408	Ferney Run-West Branch Susquehanna River
020502030409	Queens Run
020502030410	McElhattan Creek
020502030411	Chatham Run
020502030412	Reeds Run-West Branch Susquehanna River
020502040101	Cedar Run
020502040102	Slab Cabin Run
020502040103	Big Hollow
020502040104	Logan Branch
020502040105	Buffalo Run
020502040106	Spring Creek-Bald Eagle Creek
020502040201	South Fork Beach Creek
020502040202	North Fork Beach Creek
020502040203	Sandy Run-Beech Creek
020502040204	Big Run
020502040205	Beech Creek-Bald Eagle Creek
020502040301	Bull Run-Fishing Creek
020502040302	Little Fishing Creek
020502040303	Cedar Run
020502040304	Long Run
020502040305	Cherry Run-Fishing Creek
020502040401	Laurel Run-Bald Eagle Creek
020502040402	Dicks Run-Bald Eagle Creek
020502040403	Wallace Run
020502040404	Nittany Creek
020502040405	Lick Run-Bald Eagle Creek
020502040406	Marsh Creek
020502040407	Bald Eagle Creek-West Branch Susquehanna River
020502050101	Lyman Run
020502050102	Wetmore Run-West Branch Pine Creek
020502050201	Ninemile Run
020502050202	Headwaters Pine Creek
020502050203	Genesee Forks
020502050204	West Branch Pine Creek-Pine Creek
020502050205	Phoenix Run
020502050206	Elk Run
020502050207	Long Run
020502050208	Lick Run-Pine Creek
020502050301	Charleston Creek
020502050302	Asaph Run
020502050303	Marsh Creek-Pine Creek
020502050401	Headwaters Babb Creek
020502050402	Wilson Creek
020502050403	East Branch Stony Fork
020502050404	Stony Fork
020502050405	Long Run-Babb Creek
020502050501	Zimmerman Creek
020502050502	Texas Creek
020502050503	Blacks Creek

HUC_12	SUBWATERSHED_DESCRIPTION
020502050504	Blockhouse Creek
020502050505	Otter Run
020502050506	Little Pine Creek-Pine Creek
020502050601	Trout Run-Pine Creek
020502050602	Cedar Run
020502050603	Slate Run
020502050604	Mill Run-Pine Creek
020502050605	Trout Run
020502050606	Upper Pine Bottom Run-Pine Creek
020502050607	Pine Creek-West Branch Susquehanna River
020502060101	Second Fork Larrys Creek
020502060102	First Fork Larrys Creek
020502060103	Larrys Creek-West Branch Susquehanna River
020502060201	Roaring Branch
020502060202	Mill Creek-Lycoming Creek
020502060203	Rock Run
020502060204	Pleasant Stream
020502060205	Grays Run
020502060206	Trout Run-Lycoming Creek
020502060207	Hoagland Run
020502060208	Lycoming Creek-West Branch Susquehanna River
020502060301	Lopez Creek
020502060302	Glass Creek-Loyalsock Creek
020502060303	Birch Creek
020502060304	Little Loyalsock Creek-Loyalsock Creek
020502060401	Lick Creek
020502060402	Black Creek-Little Loyalsock Creek
020502060501	Porter Creek-Hoagland Branch
020502060502	Elk Creek
020502060503	Ogdonia Creek-Loyalsock Creek
020502060504	Plunketts Creek
020502060505	Bear Creek
020502060506	Wallis Run
020502060507	Mill Creek-East Side of Loyalsock Creek
020502060508	Mill Creek-West Side of Loyalsock Creek
020502060509	Little Bear Creek-Loyalsock Creek
020502060601	Antes Creek
020502060602	Quenshukeny Run
020502060603	Mosquito Creek
020502060604	Millers Run
020502060605	Wolf Run
020502060701	Beaver Run
020502060702	Big Run
020502060801	Big Run-Muncy Creek
020502060802	Rock Run-Muncy Creek
020502060803	Gregs Run-Muncy Creek
020502060901	Spring Creek
020502060902	White Deer Hole Creek-West Branch Susquehanna River
020502061001	North Branch Buffalo Creek

HUC_12	SUBWATERSHED_DESCRIPTION
020502061002	Rapid Run
020502061003	Spruce Run
020502061004	Little Buffalo Creek
020502061005	Buffalo Creek-West Branch Susquehanna River
020502061101	Mud Creek
020502061102	Upper Branches Chillisquaque Creek
020502061103	Chillisquaque Creek-West Branch Susquehanna River
020502061201	White Deer Creek-Lower West Branch Susquehanna River
020502061202	Delaware Run-Lower West Branch Susquehanna River
020502061203	Warrior Run
020502061204	Limestone Run-Northumberland County
020502061205	Muddy Run-Lower West Branch Susquehanna River
020502061206	Limestone Run-Union County
020502061207	West Branch Susquehanna River-Susquehanna River
020503010101	Shamokin Creek-City of Shamokin
020503010102	Carbon Run-Shamokin Creek
020503010103	Little Shamokin Creek
020503010104	Shamokin Creek-Susquehanna River
020503010201	Elk Creek
020503010202	Voneida Run-Pine Creek
020503010301	Faylor Lake Dam-South Branch Middle Creek
020503010302	North Branch Middle Creek-Walker Lake Dam
020503010303	North Branch Middle Creek
020503010304	Beaver Creek-Middle Creek
020503010305	Middle Creek-Penns Creek
020503010401	Colyer Lake-Sinking Creek
020503010402	Headwaters Penns Creek
020503010403	Upper Penns Creek
020503010404	Laurel Run
020503010405	Middle Penns Creek
020503010406	Lower Penns Creek-Susquehanna River
020503010501	Upper Mahanoy Creek
020503010502	Schwaben Creek
020503010503	Lower Mahanoy Creek-Susquehanna River
020503010601	North Branch Mahantango Creek
020503010602	Upper West Branch Mahantango Creek
020503010603	Lower West Branch Mahantango Creek
020503010701	Hans Yost Creek-Deep Creek
020503010702	Rausch Creek-Pine Creek
020503010801	Upper Mahantango Creek
020503010802	Lower Mahantango Creek
020503010901	Upper Wiconisco Creek
020503010902	Rattling Creek
020503010903	Little Wiconisco Creek
020503010904	Lower Wiconisco Creek
020503011001	Hallowing Run-Susquehanna River
020503011002	Fidlers Run-Susquehanna River
020503011003	Bargers Run-Susquehanna River
020503011004	Armstrong Creek

HUC_12	SUBWATERSHED_DESCRIPTION
020503011005	Powell Creek
020503011006	Haldeman Island-Susquehanna River
020503020101	Beaverdam Creek
020503020102	South Poplar Run-Frankstown Branch Juniata River
020503020103	Plum Creek
020503020104	Halter Creek
020503020105	Poplar Run
020503020106	Oldtown Run-Frankstown Branch Juniata River
020503020201	Blair Gap Run
020503020202	Mill Run-Beaverdam Branch
020503020301	Canoe Creek
020503020302	Robinson Run-Frankstown Branch Juniata River
020503020303	Piney Creek
020503020304	Clover Creek
020503020305	Frankstown Branch Juniata River-Juniata River
020503020401	Beaver Branch
020503020402	Halfmoon Creek
020503020403	Warriors Mark Run
020503020404	Spruce Creek-Little Juniata River
020503020501	Bells Gap Run
020503020502	Upper Little Juniata River
020503020503	Tipton Run
020503020504	Bald Eagle Creek
020503020505	Sinking Run
020503020506	Lower Little Juniata River
020503020601	Upper Shaver Creek
020503020602	Lower Shaver Creek
020503020701	Laurel Run
020503020702	Upper Standing Stone Creek
020503020703	East Branch Standing Stone Creek
020503020704	Lower Standing Stone Creek
020503020801	Juniata River-City of Huntingdon
020503020802	Snyders Run-Juniata River
020503020803	Crooked Creek-Juniata River
020503030101	Shawnee Branch-Shawnee Lake
020503030102	Headwaters Raystown Branch Juniata River
020503030103	Buffalo Run
020503030104	Shobers Run
020503030105	Cumberland Valley Run-Raystown Branch Juniata River
020503030201	Scrubgrass Creek
020503030202	Bobs Creek-Dunning Creek
020503030301	Upper Dunning Creek
020503030302	Georges Creek-Dunning Creek
020503030303	Lower Dunning Creek
020503030401	Little Brush Creek
020503030402	Shaffer Creek
020503030403	Brush Creek-Raystown Branch Juniata River
020503030501	Cove Creek
020503030502	Snake Spring Valley Run

HUC_12	SUBWATERSHED_DESCRIPTION
020503030503	Clear Creek
020503030504	Tub Mill Run-Raystown Branch Juniata River
020503030505	Sandy Run-Raystown Branch Juniata River
020503030601	Beaver Creek
020503030602	Upper Yellow Creek
020503030603	Lower Yellow Creek
020503030701	Little Trough Creek
020503030702	Great Trough Creek
020503030801	Shoup Run
020503030802	Sixmile Run-Raystown Branch Juniata River
020503030803	Shy Beaver Creek-Raystown Lake
020503030804	Raystown Lake-Raystown Branch Juniata River-Juniata River
020503040101	Saddler Creek
020503040102	Mill Creek
020503040103	Hares Valley Creek-Juniata River
020503040201	Upper Sideling Hill Creek
020503040202	Wooden Bridge Creek
020503040203	Lower Sideling Hill Creek
020503040301	Shade Creek
020503040302	Blacklog Creek
020503040401	North Branch Little Aughwick Creek
020503040402	Little Aughwick Creek
020503040403	Three Springs Creek
020503040404	Aughwick Creek-Juniata River
020503040501	West Licking Creek-Juniata River
020503040502	Musser Run-Juniata River
020503040503	Strodes Run-Juniata River
020503040601	Treaster Run
020503040602	Laurel Creek
020503040603	Honey Creek-Kishacoquillas Creek
020503040701	Upper Kishacoquillas Creek
020503040702	Lower Kishacoquillas Creek
020503040801	Meadow Creek-Jacks Creek
020503040802	Little Lost Creek-Lost Creek
020503040803	Horning Creek-Juniata River
020503040901	Narrows Branch Tuscarora Creek
020503040902	Trough Spring Branch-Tuscarora Creek
020503040903	Horse Valley Run
020503040904	Willow Run
020503040905	Lick Run-Tuscarora Creek
020503040906	East Licking Creek
020503040907	Tuscarora Creek-Juniata River
020503041001	Upper Cocolamus Creek
020503041002	Lower Cocolamus Creek
020503041101	Upper Buffalo Creek
020503041102	Lower Buffalo Creek
020503041201	Doe Run-Juniata River
020503041202	Raccoon Creek

HUC_12	SUBWATERSHED_DESCRIPTION
020503041203	Little Buffalo Creek
020503041204	Juniata River-Susquehanna River
020503050101	Shultz Creek-Sherman Creek
020503050102	Bull Run
020503050103	Bixler Run
020503050104	Upper Sherman Creek
020503050105	Laurel Run
020503050106	Middle Sherman Creek
020503050107	Lower Sherman Creek
020503050201	Rowe Run
020503050202	Lehman Run-Muddy Run
020503050203	Trout Run-Conodoguinet Creek
020503050301	Thompson Creek-Burd Run
020503050302	Middle Spring Creek
020503050303	Laughlin Run-Paxton Run
020503050304	Bulls Head Branch
020503050305	Green Spring Creek
020503050306	Three Square Hollow Run-Conodoguinet Creek
020503050307	Doubling Gap Creek
020503050308	Big Spring Creek-Conodoguinet Creek
020503050401	Mount Rock Spring Creek
020503050402	Alexanders Spring Creek
020503050403	Wertz Run-Conodoguinet Creek
020503050404	Letort Spring Run
020503050405	Hogestown Run
020503050406	Simmons Creek-Conodoguinet Creek
020503050407	Trindle Spring Run
020503050408	Conodoguinet Creek-Susquehanna River
020503050501	Headwaters Yellow Breeches Creek
020503050502	Upper Yellow Breeches Creek
020503050503	Mountain Creek
020503050504	Middle Yellow Breeches Creek
020503050505	Lower Yellow Breeches Creek
020503050601	Upper Little Swartara Creek
020503050602	Good Spring Creek-Upper Swatara Creek
020503050603	Lower Little Swatara Creek
020503050604	Mill Creek
020503050605	Middle Swatara Creek
020503050606	Lower Swatara Creek
020503050701	Crosskill Creek
020503050702	Upper Little Swatara Creek
020503050703	Lower Little Swatara Creek
020503050801	Killinger Creek
020503050802	Snitz Creek-Quittapahilla Creek
020503050901	Reeds Run-Swatara Creek
020503050902	Bow Creek-Swatara Creek
020503050903	Manada Creek
020503050904	Spring Creek
020503050905	Beaver Creek

HUC_12	SUBWATERSHED_DESCRIPTION
020503050906	Swatara Creek-Susquehanna River
020503051001	Little Juniata Creek
020503051002	Clark Creek
020503051003	Stony Creek
020503051004	Fishing Creek-Dauphin County
020503051005	Fishing Creek-Perry County
020503051006	Cove Creek-Susquehanna River
020503051007	Paxton Creek
020503051008	Spring Creek
020503051009	Fishing Creek-York County
020503051010	Conewago Creek
020503051011	Laurel Run-Susquehanna River
020503060101	Headwaters South Branch Conewago Creek
020503060102	Plum Creek-South Branch Conewago Creek
020503060201	Opossum Creek
020503060202	Headwaters Conewago Creek
020503060203	Swift Run-Conewago Creek
020503060204	Boro of East Berlin-Conewago Creek
020503060301	Latimore Creek
020503060302	North Branch Bermudian Creek
020503060303	Mud Run-Bermudian Creek
020503060401	Upper Little Conewago Creek
020503060402	Lower Little Conewago Creek
020503060501	Beaver Creek
020503060502	Davidsburg Run-Conewago Creek
020503060503	Conewago Lake-Beaver Creek
020503060504	Conewago Creek-Susquehanna River
020503060601	Upper South Branch Codorus Creek
020503060602	Lake Redman-Lake Williams-East Branch Codorus Creek
020503060603	Lower South Branch Codorus Creek
020503060701	Lake Marburo-West Branch Codorus Creek
020503060702	Oil Creek
020503060703	Headwaters Codorus Creek
020503060704	Stoverstown Branch-Codorus Creek
020503060705	Willis Run-Codorus Creek
020503060706	Mill Creek
020503060707	Codorus Creek-Susquehanna River
020503060801	Upper Chickies Creek
020503060802	Little Chickies Creek
020503060803	Donegal Creek
020503060804	Lower Chickies Creek
020503060901	Little Cocalico Creek-Cocalico Creek
020503060902	Middle Creek
020503060903	Hammer Creek
020503060904	Cocalico Creek-Conestoga River
020503061001	Millers Run-Little Conestoga Creek
020503061002	West Branch Little Conestoga Creek-Little Conestoga Creek
020503061101	Little Muddy Creek

HUC_12	SUBWATERSHED_DESCRIPTION
020503061102	Muddy Creek
020503061103	Upper Conestoga River
020503061104	Middle Conestoga River
020503061105	Lititz Run
020503061106	Muddy Run-Mill Creek
020503061107	Lower Conestoga River
020503061201	Headwaters Pequea Creek
020503061202	Eshleman Run-Pequea Creek
020503061203	Big Beaver Creek
020503061204	Climbers Run-Pequea Creek
020503061301	North Branch Muddy Creek
020503061302	South Branch Muddy Creek
020503061303	Bald Eagle Creek-Muddy Creek
020503061304	Fishing Creek-Muddy Creek
020503061401	Pine Creek
020503061402	Valley Creek-East Branch Octoraro Creek
020503061403	Muddy Run-East Branch Octoraro Creek
020503061501	West Branch Octoraro Creek
020503061502	Tweed Creek-Octoraro Creek
020503061503	Basin Run-Octoraro Creek
020503061601	Headwaters Deer Creek
020503061602	Upper Deer Creek
020503061603	Middle Deer Creek
020503061604	Lower Deer Creek
020503061701	Conoy Creek
020503061702	Hartman Run-Susquehanna River
020503061703	Kreutz Creek
020503061704	Cabin Creek-Susquehanna River
020503061705	Fishing Creek
020503061706	Green Branch-Susquehanna River
020503061707	Otter Creek
020503061708	Muddy Run-Susquehanna River
020503061709	Fishing Creek-Susquehanna River
020503061710	Broad Creek
020503061711	Conowingo Creek
020503061712	Conowingo Dam-Susquehanna River
020503061713	Rock Run-Susquehanna River
020600010000	Upper Chesapeake Bay
020600020101	Little North East Creek
020600020102	North East Creek
020600020103	Mill Creek-Furnace Bay
020600020104	Hance Point Creek-North East River
020600020105	Elk Neck-Upper Chesapeake Bay
020600020201	East Branch Big Elk Creek
020600020202	Little Elk Creek
020600020203	Big Elk Creek
020600020204	C&D Canal West-Back Creek
020600020205	Upper Elk River
020600020206	Bohemia River

HUC_12	SUBWATERSHED_DESCRIPTION
020600020207	Lower Elk River
020600020301	Upper Sassafras River
020600020302	Lower Sassafras River
020600020401	Cypress Branch
020600020402	Andover Branch
020600020403	Unicorn Branch
020600020404	Red Lion Branch
020600020405	Morgan Creek
020600020406	Upper Chester River
020600020407	Southeast Creek
020600020408	Middle Chester River
020600020409	Corsica River
020600020410	Langford Creek
020600020411	Lower Chester River
020600020501	Still Pond Creek-Upper Chesapeake Bay
020600020502	Fairlee Creek-Upper Chesapeake Bay
020600020503	Swan Creek-Upper Chesapeake Bay
020600020504	Craney Creek-Upper Chesapeake Bay
020600020601	Skipton Creek
020600020602	Upper Wye East River
020600020603	Lower Wye East River
020600020604	Wye River
020600020605	Miles River
020600020606	Prospect Bay-Eastern Bay
020600020607	Cox Creek-Eastern Bay
020600020608	Tilghman Creek-Eastern Bay
020600020609	Eastern Bay Deep
020600030101	Upper Winters Run
020600030102	Lower Winters Run
020600030104	Bynum Run-Bush Creek
020600030105	Grays Run-Church Creek
020600030106	Bush River
020600030201	Swan Creek-Chesapeake Bay
020600030202	Mosquito Creek-Chesapeake Bay
020600030203	Upper Romney Creek
020600030204	Lower Romney Creek-Chesapeake Bay
020600030301	South Branch Gunpowder Falls-Gunpowder Falls
020600030302	Prettyboy Reservoir-Gunpowder Falls
020600030401	Little Falls
020600030402	Piney Creek-Gunpowder Falls
020600030403	Piney Run-Western Run
020600030404	Blackrock Run-Western Run
020600030405	Beaverdam Run-Loch Raven Reservoir
020600030406	Lock Raven Reservoir-Gunpowder Falls
020600030501	Little Gunpowder Falls
020600030502	Long Green Creek
020600030601	Whitemarsh Run-Bird River
020600030602	Saltpeter Creek-Gunpowder River
020600030701	Middle River

HUC_12	SUBWATERSHED_DESCRIPTION
020600030702	Redhouse Creek-Back River
020600030703	Back River-Hawk Cove-Chesapeake Bay
020600030801	East Branch of North Branch Patapsco River
020600030802	Headwaters North Branch Patapsco River
020600030803	Beaver Run-Liberty Lake
020600030804	Morgan Run-Liberty Lake
020600030805	Deep Run-Liberty Lake-North Branch Patapsco River
020600030806	Falls Run-Liberty Lake-North Branch Patapsco River
020600030901	Red Run-Gwynns Falls
020600030902	Dead Run-Gywnns Falls
020600031001	Gillis Falls
020600031002	Hay Meadow Branch-South Branch Patapsco River
020600031003	Piney Run
020600031004	Piney Branch-South Branch Patapsco River
020600031101	Brice Run-Patapsco River
020600031102	Deep Run-Patapsco River
020600031103	Herbert Run-Patapsco River
020600031201	Jones Falls
020600031202	Curtis Creek-Curtis Bay
020600031203	Northwest Harbor-Patapsco River
020600031204	Stoney Creek-Patapsco River-Chesapeake Bay
020600040101	Cattail Creek-Magothy River
020600040102	Sillery Bay-Chesapeake Bay
020600040201	Severn Run
020600040202	Round Bay-Severn River
020600040203	Whitehall Creek-Severn River-Chesapeake Bay
020600040301	Beacon Ridge Branch-North River
020600040302	Beards Creek-South River
020600040401	Rhode River-West River
020600040402	Tracys Creek-Herring Bay
020600040403	Parker Creek-Chesapeake Bay
020600040404	Saint Jerome Creek-Chesapeake Bay
020600050101	Upper Mason Branch
020600050102	Lower Mason Branch
020600050103	German Branch
020600050104	Norwich Creek-Tuckahoe Creek
020600050105	Jadwins Creek-Tuckahoe Creek
020600050201	Cow Marsh Creek
020600050202	Tappahanna Ditch-Choptank River
020600050203	Gravelly Branch-Choptank River
020600050204	Forge Branch-Choptank River
020600050205	Chapel Branch-Choptank River
020600050206	Fowling Creek-Choptank River
020600050301	Kings Creek
020600050302	Williams Creek-Choptank River
020600050303	Hunting Creek
020600050304	Marsh Creek-Choptank River
020600050401	Fishing Creek-Little Choptank River
020600050402	Slaughter Creek-Little Choptank River

HUC_12	SUBWATERSHED_DESCRIPTION
020600050501	Warwick River-Choptank River
020600050502	Bolingbroke Creek-Choptank River
020600050503	LaTrappe Creek-Choptank River
020600050504	Tred Avon River-Choptank River
020600050505	Broad Creek-Choptank River
020600050506	Harris Creek-Coptank River
020600050507	Brannock Bay-Choptank River
020600050508	Choptank River-Deep
020600050509	Poplar Harbor-Chesapeake Bay
020600050601	Saint John Creek-Chesapeake Bay
020600050602	Charles Creek-Honga River
020600050603	Fox Creek-Honga River
020600050604	Tar Bay-Chesapeake Bay
020600050605	Holland Straits-Chesapeake Bay
020600060101	Cabin Branch-Patuxent River
020600060102	Cattail Creek
020600060103	Triadelphia Reservoir-Patuxent River
020600060104	Hawlings River
020600060105	Rocky Gorge Reservoir-Patuxent River
020600060201	Benson Branch-Middle Patuxent River
020600060202	Dorsey Run-Little Patuxent River
020600060203	Towsers Branch-Little Patuxent River
020600060301	Southwest Branch of the Western Branch Patuxent River
020600060302	Northwest Branch of the Western Branch Patuxent River
020600060303	Collington Branch
020600060304	Charles Branch-Western Branch Patuxent River
020600060401	Horsepen Branch-Patuxent River
020600060402	Stocketts Run-Patuxent River
020600060403	Wilson Owens Branch-Patuxent River
020600060501	Lyons Creek
020600060502	Mataponi Creek-Patuxent River
020600060503	Hall Creek
020600060504	Chew Creek-Patuxent River
020600060505	Hunting Creek
020600060506	Tucker Creek-Patuxent River
020600060507	Swanson Creek-Patuxent River
020600060601	Indian Creek-Patuxent River
020600060602	Battle Creek-Patuxent River
020600060603	Saint Leonard Creek-Patuxent River
020600060604	Mill Creek-Patuxent River
020700010101	Laurel Fork-North Fork South Branch Potomac River
020700010102	Big Run
020700010103	Red Lick Run-North Fork South Branch Potomac River
020700010104	Headwaters Seneca Creek
020700010105	Outlet Seneca Creek
020700010106	Mill Creek-North Fork South Branch Potomac River
020700010107	Zeke Run-North Fork South Branch Potomac River
020700010108	Jordan Run-North Fork South Branch Potomac River
020700010201	Headwaters Lunice Creek

HUC_12	SUBWATERSHED_DESCRIPTION
020700010202	Outlet Lunice Creek
020700010301	Frank Run-South Branch Potomac River
020700010302	Strait Creek
020700010303	East Dry Run-South Branch Potomac River
020700010304	Whitethorn Creek-Thorn Creek
020700010305	Smith Creek-South Branch Potomac River
020700010306	Hayes Gap Run-South Branch Potomac River
020700010307	Reeds Creek
020700010308	Mill Run-South Branch Potomac River
020700010309	Briggs Run-South Branch Potomac River
020700010310	Hoglan Run-South Branch Potomac River
020700010401	South Mill Creek
020700010402	Johnson Run-Mill Creek
020700010501	Brushy Fork-South Fork South Branch Potomac River
020700010502	Little Fork-South Fork South Branch Potomac River
020700010503	Miller Run-South Fork South Branch Potomac River
020700010504	Hawes Run-South Fork South Branch Potomac River
020700010505	Rough Run-South Fork South Branch Potomac River
020700010506	Kettle Creek-South Fork South Branch Potomac River
020700010507	Rohrbaugh Run-South Fork South Branch Potomac River
020700010508	Stump Run-South Fork South Branch Potomac River
020700010509	Stony Run-South Fork South Branch Potomac River
020700010601	Hutton Run-South Branch Potomac River
020700010602	Anderson Run
020700010603	Fort Run-South Branch Potomac River
020700010604	Stony Run-South Branch Potomac River
020700010605	Sawmill Run-South Branch Potomac River
020700010606	Mill Creek
020700010607	McDowell Run-South Branch Potomac River
020700010608	Fox Run-South Branch Potomac River
020700010609	Abernathy Run-South Branch Potomac River
020700020101	Upper Savage River
020700020102	Crabtree Creek
020700020103	Lower Savage River
020700020201	Shields Run-North Branch Potomac River
020700020202	Mount Storm Lake-Stony River
020700020203	Buffalo Creek-North Branch Potomac River
020700020204	Abram Creek
020700020205	Lostland Run-North Branch Potomac River
020700020206	Bloomington Lake-North Branch Potomac River
020700020207	Piney Swamp Run-North Branch Potomac River
020700020301	Upper Georges Creek
020700020302	Lower Georges Creek
020700020401	New Creek
020700020402	Limestone Run-North Branch Potomac River
020700020403	Mill Run-North Branch Potomac River
020700020501	Brush Creek
020700020502	Laurel Run
020700020503	Little Wills Creek

HUC_12	SUBWATERSHED_DESCRIPTION
020700020504	Gladdens Run
020700020505	Jennings Run
020700020506	Shaffers Run-Wills Creek
020700020507	Braddock Creek-Wills Creek
020700020601	Upper Evitts Creek
020700020602	Rocky Gap Run-Evitts Creek
020700020603	Lower Evitts Creek
020700020701	North Fork Patterson Creek
020700020702	Middle Fork Patterson Creek-Patterson Creek
020700020703	Mikes Run
020700020704	Rosser Run-Patterson Creek
020700020705	Mill Creek-Patterson Creek
020700020706	Cabin Run
020700020707	Beaver Run-Patterson Creek
020700020708	Horseshoe Creek-Patterson Creek
020700020709	Keller Run-Patterson Creek
020700020801	Collier Run-North Branch Potomac River
020700020802	Mill Run
020700020803	Green Spring Run-North Branch Potomac River
020700030101	Wilson Run-Elk Lick Creek
020700030102	Flintstone Creek
020700030103	Murley Branch
020700030104	Sweet Root Creek-Town Creek
020700030105	Peters Run-Town Creek
020700030106	Sawpit Run-Town Creek
020700030201	North Fork-Little Cacapon River
020700030202	Shawan Run-Little Cacapon River
020700030203	Three Churches Run-Little Cacapon River
020700030204	Crooked Run-Little Cacapon River
020700030205	Dug Hill Run-Little Cacapon River
020700030301	Upper Fifteenmile Creek
020700030302	Lower Fifteenmile Creek
020700030304	Crooked Run-Sideling Hill Creek
020700030305	Bear Creek-Sideling Hill Creek
020700030401	East Branch Sideling Hill Creek
020700030402	West Branch Sideling Hill Creek
020700030403	Piney Creek
020700030501	Cullers Run-Lost River
020700030502	Upper Cove Run-Lost River
020700030503	Baker Run
020700030504	Kimsey Run-Lost River
020700030505	Three Springs Run-Lost River
020700030601	Meadow Run-North River
020700030602	Sperry Run-North River
020700030603	Tear Coat Creek
020700030604	Pine Draft Run-North River
020700030605	Hiett Run-North River
020700030606	Crooked Run-North River
020700030701	Trout Run

HUC_12	SUBWATERSHED_DESCRIPTION
020700030702	Waites Run-Cacapon River
020700030703	Capon Springs Run-Cacapon River
020700030704	Dillons Run
020700030705	Mill Branch-Cacapon River
020700030706	Bloomery Run-Cacapon River
020700030707	Critton Run-Cacapon River
020700030708	Connor Hollow-Cacapon River
020700030801	Purslane Run-Potomac River
020700030802	Rockwell Run-Potomac River
020700030803	Willett Run-Potomac River
020700040101	Little Tonoloway Creek(PA)
020700040102	Barnetts Run-Tonoloway Creek
020700040201	Upper Sleepy Creek
020700040202	Middle Fork Sleepy Creek
020700040203	Middle Sleepy Creek
020700040204	Meadow Branch
020700040205	Lower Sleepy Creek
020700040301	Patterson Run-Licking Creek
020700040302	Big Cove Creek
020700040303	Little Cove Creek
020700040304	Owl Creek-Licking Creek
020700040305	Lanes Run-Licking Creek
020700040401	Mine Spring Run-Back Creek
020700040402	Isaacs Creek-Back Creek
020700040403	Hogue Creek
020700040404	Brush Creek-Back Creek
020700040405	Babbs Run
020700040406	Warm Springs Hollow-Back Creek
020700040407	Elk Branch-Back Creek
020700040408	Tilhance Creek
020700040409	Outlet Back Creek
020700040501	Minnow Run-Little Tonoloway Creek(MD)
020700040502	Sir Johns Run-Potomac River
020700040503	Warm Spring Run
020700040504	Ditch Run-Potomac River
020700040505	Cherry Run-Potomac River
020700040601	Headwaters West Branch Conococheague Creek
020700040602	Upper West Branch Conococheague Creek
020700040603	Middle West Branch Conococheague Creek
020700040604	Licking Creek
020700040605	Lower West Branch Conococheague Creek
020700040701	Rocky Spring Branch
020700040702	Dennis Creek-Back Creek
020700040703	Campbell Run-Back Creek
020700040801	Rocky Mountain Creek
020700040802	Headwaters Conococheague Creek
020700040803	Mountain Creek-Conococheague Creek
020700040804	Muddy Run
020700040805	Falling Spring Branch-Conococheague Creek

HUC_12	SUBWATERSHED_DESCRIPTION
020700040806	Rockdale Run-Conococheague Creek
020700040807	Meadow Brook-Conococheague Creek
020700040901	Sulphur Spring Run-Opequon Creek
020700040902	Abrams Creek
020700040903	Redbud Run-Opequon Creek
020700040904	Turkey Run-Opequon Creek
020700040905	Mill Creek
020700040906	Middle Creek-Opequon Creek
020700040907	Tuscarora Creek
020700040908	Evans Run-Opequon Creek
020700040909	Hoke Run-Opequon Creek
020700041001	Red Run
020700041002	East Branch Antietam Creek
020700041003	West Branch Antietam Creek
020700041004	Little Antietam Creek
020700041005	West Branch Marsh Run-Marsh Run
020700041006	Middle Antietam Creek
020700041007	Beaver Creek
020700041008	Dog Creek-Little Antietam Creek
020700041009	Sharmans Branch-Antietam Creek
020700041101	Harlan Run
020700041102	Little Conococheague Creek
020700041103	Camp Spring Run-Potomac River
020700041104	Saint James Run-Marsh Run
020700041105	Rockymarsh Run
020700041106	Rattlesnake Run-Potomac River
020700041107	Elks Run
020700041108	Harpers Ferry-Potomac River
020700050101	Edison Creek-Middle River
020700050102	Buffalo Branch-Middle River
020700050103	Jennings Branch
020700050104	Bell Creek-Middle River
020700050105	Moffett Creek
020700050201	Folly Mills Creek-Christians Creek
020700050202	Barterbrook Branch-Christians Creek
020700050203	Meadow Run
020700050301	Lewis Creek
020700050302	Falling Spring Run-Middle River
020700050303	Broad Run-Middle River
020700050401	Skidmore Fork-North River
020700050402	Little River
020700050403	Briery Branch
020700050404	Mossy Creek
020700050405	Thorny Branch-North River
020700050501	Skidmore Fork-Dry River
020700050502	Black Run-Dry River
020700050503	Muddy Creek
020700050504	Honey Run-Dry River
020700050601	Long Glade Creek

HUC_12	SUBWATERSHED_DESCRIPTION
020700050602	Blacks Run
020700050603	Cooks Creek
020700050604	North Fork Naked Creek-Naked Creek
020700050605	Pleasant Run-North River
020700050606	Mill Creek-North River
020700050701	Stony Run-South River
020700050702	Canada Run-South River
020700050703	Inch Branch-Back Creek
020700050704	Porterfield Run-South River
020700050705	Paine Run-South River
020700050801	Big Run-South Fork Shenandoah River
020700050802	Keezletown-Cub Run
020700050803	Hawksbill Creek-South Fork Shenandoah River
020700050804	Boone Run-Elk Run-South Fork Shenandoah River
020700050805	South Branch-Naked Creek
020700050901	Fultz Run-South Fork Shenandoah River
020700050902	Pitt Spring Run-Cub Run
020700050903	Stony Run-South Fork Shenandoah River
020700050904	Hawksclaw Creek-South Fork Shenandoah River
020700050905	Mill Creek-South Fork Shenandoah River
020700050906	East Hawksbill Creek-Hawksbill Creek
020700050907	Pass Run-Hawksbill Creek
020700051001	Jeremys Run-South Fork Shenandoah River
020700051002	Brown Hollow Run-South Fork Shenandoah River
020700051003	Gooney Run
020700051004	Punches Run-South Fork Shenandoah River
020700051005	Happy Creek
020700060101	German River
020700060102	Crab Run
020700060103	Capon Run-North Fork Shenandoah River
020700060104	Little Dry River
020700060105	Shoemaker River
020700060106	Runion Creek-North Fork Shenandoah River
020700060201	Dry Fork
020700060202	Mountain Run-Smith Creek
020700060203	War Branch-Smith Creek
020700060204	Gap Creek-Smith Creek
020700060301	Turley Creek-North Fork Shenandoah River
020700060302	Linville Creek
020700060303	Long Meadow-North Fork Shenandoah River
020700060304	Holmans Creek-North Fork Shenandoah River
020700060305	Crooked Run-Mill Creek
020700060306	Mt Jackson-North Fork Shenandoah River
020700060401	Riles Run-Stony Creek
020700060402	Yellow Spring Run-Stony Creek
020700060403	Painter Run-Stony Creek
020700060501	Narrow Passage Creek-North Fork Shenandoah River
020700060502	Toms Brook-North Fork Shenandoah River
020700060503	Tumbling Run-North Fork Shenandoah River

HUC_12	SUBWATERSHED_DESCRIPTION
020700060601	Paddy Run-Cedar Creek
020700060602	Duck Run-Cedar Creek
020700060603	Fall Run
020700060604	Froman Run-Cedar Creek
020700060605	Meadow Brook-Cedar Creek
020700060701	Upper Passage Creek
020700060702	Lower Passage Creek
020700060703	Molly Booth Run-North Fork Shenandoah River
020700070101	Crooked Run
020700070102	Manassas Run-Shenandoah River
020700070103	Borden Marsh Run-Shenandoah River
020700070104	Long Branch-Shenandoah River
020700070105	Spout Run
020700070201	Chapel Run-Shenandoah River
020700070202	Dog Run-Shenandoah River
020700070203	Long Marsh Run
020700070301	Bullskin Run
020700070302	Evitts Run
020700070303	Furnace Run-Shenandoah River
020700070304	Flowing Springs Run-Shenandoah River
020700080101	Upper Catoctin Creek
020700080102	Middle Catoctin Creek
020700080103	Lower Catoctin Creek
020700080201	Israel Creek
020700080202	Piney Run-Potomac River
020700080301	South Fork Catoctin Creek
020700080302	Catoctin Creek
020700080401	Tuscarora Creek-Potomac River
020700080402	Little Monocacy River
020700080403	Limestone Branch-Potomac River
020700080501	Mitchells Branch-Goose Creek
020700080502	Crooked Run-Goose Creek
020700080503	Panther Skin Creek
020700080504	Cromwells Run
020700080505	Wancopin Creek-Goose Creek
020700080601	Beaverdam Creek
020700080602	North Fork Goose Creek
020700080701	Little River
020700080702	Big Branch-Goose Creek
020700080703	Sycolin Creek
020700080704	Cattail Branch-Goose Creek
020700080801	Little Seneca Creek
020700080802	Great Seneca Creek
020700080803	Dry Seneca Creek-Seneca Creek
020700080901	Lenah Run-Broad Run
020700080902	Horsepen Run
020700080903	Beaverdam Run-Broad Run
020700080904	Selden Island-Potomac River
020700080905	Sugarland Run

HUC_12	SUBWATERSHED_DESCRIPTION
020700081001	Muddy Branch
020700081002	Watts Branch
020700081003	Cabin John Creek
020700081004	Difficult Run
020700081005	Nichols Run-Potomac River
020700090101	Upper Rock Creek
020700090102	Lower Rock Creek
020700090201	Little Marsh Creek
020700090202	Upper Marsh Creek
020700090203	Lower Marsh Creek
020700090301	Upper Toms Creek
020700090302	Middle Creek
020700090303	Lower Toms Creek
020700090401	Upper Little Pipe Creek
020700090402	Middle Little Pipe Creek
020700090403	Lower Little Pipe Creek
020700090404	Upper Big Pipe Creek
020700090405	Lower Big Pipe Creek-Double Pipe Creek
020700090501	Alloway Creek
020700090502	Piney Creek
020700090503	Cattail Branch-Monocacy River
020700090504	Owens Creek
020700090505	Hunting Creek
020700090506	Broad Run-Monocacy River
020700090601	Tuscarora Creek-Monocacy River
020700090602	South Fork Linganore Creek
020700090603	North Fork Linganore Creek
020700090604	Upper Linganore Creek
020700090605	Lower Linganore Creek
020700090606	Carroll Creek-Monocacy River
020700090701	Bush Creek
020700090702	Little Bennett Creek
020700090703	Bennett Creek
020700090704	Ballenger Creek-Monocacy River
020700100101	Upper Rock Creek
020700100102	Lower Rock Creek
020700100103	Pimmit Run-Potomac River
020700100201	Northwest Branch Anacostia River
020700100202	Paint Branch
020700100203	Upper Anacostia River
020700100204	Lower Anacostia River
020700100301	Fourmile Run-Potomac River
020700100302	Cameron Run
020700100303	Henson Creek
020700100304	Tinkers Creek
020700100305	Piscataway Creek
020700100306	Dogue Creek
020700100307	Little Hunting Creek-Potomac River
020700100401	Pohick Creek

HUC_12	SUBWATERSHED_DESCRIPTION
020700100402	Accotink Creek
020700100501	Trapp Branch-Broad Run
020700100502	Catletts Branch-Broad Run
020700100503	Kettle Run
020700100504	Rocky Branch-Broad Run
020700100601	Mill Run-Cedar Run
020700100602	Owl Run-Cedar Run
020700100603	Licking Run
020700100604	Walnut Branch-Cedar Run
020700100605	Town Run
020700100606	Slate Run-Cedar Run
020700100701	Upper Bull Run
020700100702	Little Bull Run
020700100703	Middle Bull Run
020700100704	Cub Run
020700100705	Lower Bull Run
020700100801	Lake Jackson-Occoquan River
020700100802	Occoquan Reservoir-Occoquan River
020700100803	Belmont Bay-Occoquan River
020700100804	Neabsco Creek
020700100805	Occoquan Bay-Potomac River
020700110101	Piney Branch-Mattawoman Creek
020700110102	Marbury Run-Mattawoman Creek
020700110103	Powells Creek
020700110104	Quantico Creek
020700110105	Chopawamsic Creek
020700110106	Tank Creek-Potomac River
020700110201	Beaverdam Run
020700110202	Upper Aquia Creek
020700110203	Lower Aquia Creek
020700110204	Accokeek Creek
020700110205	Long Branch-Potomac Creek
020700110206	Beaverdam Creek-Potomac Creek
020700110207	Passapatanzy Creek-Potomac River
020700110301	Hancock Run-Nanjemoy Creek
020700110302	Burgess Creek-Nanjemoy Creek
020700110303	Chotank Creek-Potomac River
020700110304	Port Tobacco River
020700110305	Gambo Creek-Potomac River
020700110306	Piccowaxen Creek
020700110401	Jordan Swamp-Zekiah Swamp Run
020700110402	Kerrick Swamp-Zekiah Swamp Run
020700110403	Clark Run-Zekiah Swamp Run
020700110501	Trinity Church Run-Wicomico River
020700110502	Budds Creek-Wicomico River
020700110503	Chaptico Creek
020700110504	Charleston Creek-Wicomico River
020700110505	Whites Neck Creek-Saint Catherine Sound
020700110601	Upper Machodoc Creek

HUC_12	SUBWATERSHED_DESCRIPTION
020700110602	Rosier Creek-Potomac River
020700110603	Mattox Creek
020700110604	Popes Creek-Potomac River
020700110701	Saint Clements Creek-Saint Clements Bay
020700110702	McIntosh Run
020700110703	Glebe Run-Breton Bay
020700110704	Poplar Hill Creek
020700110801	Nomini Creek
020700110802	Nomini Bay-Potomac River
020700110803	Lower Machodoc Creek-Potomac River
020700110804	Yeocomico River
020700110805	Coan River
020700110806	Hull Creek-Potomac River
020700110901	Western Branch-Saint Marys River
020700110902	Eastern Branch-Saint Marys River
020700110903	Saint George Creek-Saint Marys River
020700110904	Jutland Creek
020700111001	Potomac River Channel
020801010000	Lower Chesapeake Bay
020801020101	Little Wicomico River
020801020102	Great Wicomico River
020801020103	Cockrell Creek-Lower Chesapeake Bay
020801020104	Dividing Creek-Lower Chesapeake Bay
020801020105	Fleets Bay-Lower Chesapeake Bay
020801020201	Dragon Run-Dragon Swamp
020801020202	Exol Swamp
020801020203	Timber Branch Swamp-Dragon Swamp
020801020204	Meggs Bay-Dragon Swamp
020801020301	Carvers Creek-Piankatank River
020801020302	Hills Bay-Piankatank River
020801020303	Milford Haven-Lower Chesapeake Bay
020801020401	Beaverdam Swamp
020801020402	Crany Creek-Fox Mill Run
020801020403	Ware River
020801020404	North River
020801020405	East River
020801020406	Winter Harbor-Lower Chesapeake Bay
020801020407	Severn River
020801020408	Monday Creek-Mobjack Bay
020801030101	Buck Run-Rappahannock River
020801030102	Jordan River
020801030103	Lake Mosby-Rappahannock River
020801030104	Thumb Run
020801030201	Glascock Run-Rappahannock River
020801030202	Carter Run
020801030203	Great Run-Rappahannock River
020801030301	Piney River-Thornton River
020801030302	Covington River
020801030303	Battle Run

HUC_12	SUBWATERSHED_DESCRIPTION
020801030304	Mill Run-Thornton River
020801030401	Hughes River
020801030402	Sams Run-Hazel River
020801030403	Devils Run-Hazel River
020801030404	Muddy Run
020801030405	Indian Run-Hazel River
020801030501	Hiders Branch-Mountain Run
020801030502	Jonas Run
020801030503	Flat Run-Mountain Run
020801030601	Marsh Run
020801030602	Ruffans Run-Rappahannock River
020801030603	Rock Run-Rappahannock River
020801030604	Deep Run-Rappahannock River
020801030701	Garth Run-Rapidan River
020801030702	Conway River
020801030703	South River-Rapidan River
020801030801	Marsh Run-Rapidan River
020801030802	Blue Run
020801030803	Beautiful Run
020801030804	Poplar Run-Rapidan River
020801030901	Rose River-Robinson River
020801030902	Leathers Run-Robinson River
020801030903	White Oak Run
020801030904	Deep Run-Robinson River
020801030905	Crooked Run
020801030906	Great Run-Robinson River
020801031001	Rapidan-Rapidan River
020801031002	Cedar Run
020801031003	Potato Run-Rapidan River
020801031101	Mill Run-Mountain Run
020801031102	Mine Run
020801031103	Fields Run-Rapidan River
020801031104	Wilderness Run
020801031105	Hazel Run-Rapidan River
020801040101	Motts Run-Rappahannock River
020801040102	Hazel Run-Rappahannock River
020801040103	Massaponax Creek
020801040104	Muddy Creek-Rappahannock River
020801040201	Mount Creek-Rappahannock River
020801040202	Mill Creek
020801040203	Goldenvale Creek-Rappahannock River
020801040301	Portobago Creek-Rappahannock River
020801040302	Elmwood Creek
020801040303	Peedee Creek-Rappahannock River
020801040304	Occupacia Creek
020801040305	Brockenbrough Creek-Rappahannock River
020801040401	The Big Swamp-Cat Point Creek
020801040402	Menokin Bay-Cat Point Creek
020801040403	Mount Landing Creek

HUC_12	SUBWATERSHED_DESCRIPTION
020801040404	Hoskins Creek
020801040405	Piscataway Creek
020801040406	Little Carter Creek-Rappahannock River
020801040501	Little Totuskey Creek
020801040502	Totuskey Creek
020801040503	Farnham Creek
020801040504	Cedar Creek-Rappahannock River
020801040601	Lancaster Creek
020801040602	Parrotts Creek-Rappahannock River
020801040603	Lagrange Creek-Rappahannock River
020801040701	Western Branch Corrotoman River
020801040702	Eastern Branch Corrotoman River
020801040703	Taylor Creek-Corrotoman River
020801040704	Carter Creek-Rappahannock River
020801040705	Locklies Creek-Rappahannock River
020801050101	Ni River
020801050102	Robertson Run-Po River
020801050103	Glady Run
020801050104	Lake Pocahontas-Po River
020801050105	Poni River
020801050201	Mat River
020801050202	Ta River
020801050203	Matta River
020801050204	South River
020801050205	Campbell Creek-Mattaponi River
020801050301	Polecat Creek
020801050302	Reedy Creek
020801050303	Union Swamp-Mattaponi River
020801050401	Jacks Creek-Maracossic Creek
020801050402	Beverly Run
020801050403	Doctors Creek-Maracossic Creek
020801050501	Chapel Creek
020801050502	Gravel Run-Mattaponi River
020801050503	Herring Creek
020801050504	Aylett Creek-Mattaponi River
020801050601	Garnetts Creek
020801050602	Courthouse Creek-Mattaponi River
020801050603	Heartquake Creek-Mattaponi River
020801050604	Cabin Creek-Mattaponi River
020801060101	Dove Fork-South Anna River
020801060102	Wheeler Creek
020801060103	Roundabout Creek-South Anna River
020801060201	Harris Creek-South Anna River
020801060202	Fork Creek-South Anna River
020801060203	Cub Creek
020801060204	Owens Creek-South Anna River
020801060301	Taylors Creek
020801060302	Turkey Creek-South Anna River
020801060303	Newfound River

HUC_12	SUBWATERSHED_DESCRIPTION
020801060304	Cedar Creek-South Anna River
020801060401	Mountain Run-North Anna River
020801060402	Hickory Creek
020801060403	Gold Mine Creek
020801060404	Christopher Creek-North Anna River
020801060501	Clear Creek-Pamunkey Creek
020801060502	Terrys Run
020801060503	Plentiful Creek-Pamunkey Creek
020801060601	Contrary Creek
020801060602	Pigeon Run-Lake Anna
020801060603	Elk Creek-Lake Anna
020801060701	Upper Little River
020801060702	Lower Little River
020801060801	Northeast Creek
020801060802	Hawkins Creek-North Anna River
020801060803	Long Creek-North Anna River
020801060901	Mechumps Creek-Pamunkey River
020801060902	Crump Creek
020801060903	Judy Swamp-Pamunkey River
020801060904	Totopotomoy Creek
020801061001	Hollyfield Pond-Pamunkey River
020801061002	Moncuin Creek
020801061003	Black Creek
020801061004	Montague Creek-Pamunkey River
020801061005	Jacks Creek
020801061101	Cohoke Mill Creek-Pamunkey River
020801061102	Mill Creek-Pamunkey River
020801070101	Ware Creek
020801070102	Philbates Creek-York River
020801070103	Poropotank River
020801070104	Skimino Creek-York River
020801070201	Jones Creek-York River
020801070202	Queen Creek
020801070203	Carter Creek-York River
020801070204	Sarah Creek-York River
020801080101	Poquoson River-Lower Chesapeake Bay
020801080102	Northwest Branch Back River
020801080103	Southwest Branch Back River
020801080104	Back River-Lower Chesapeake Bay
020801080201	Lynnhaven River
020801080202	Little Creek-Lower Chesapeake Bay
020801090101	Upper Deep Creek
020801090102	Lower Deep Creek
020801090201	Hitch Pond Branch
020801090202	James Branch
020801090203	Elliott Pond Branch
020801090204	Little Creek-Broad Creek
020801090205	Tussocky Branch-Broad Creek
020801090301	Headwaters Marshyhope Creek

HUC_12	SUBWATERSHED_DESCRIPTION
020801090302	Saulsbury Creek-Marshyhope Creek
020801090303	Tommy Wright Branch-Marshyhope Creek
020801090304	Sullivan Branch-Marshyhope Creek
020801090305	Faulkner Branch-Marshyhope Creek
020801090306	Puckum Branch-Marshyhope Creek
020801090307	Stony Bar Creek-Marshyhope Creek
020801090401	Gum Branch
020801090402	Headwaters Nanticoke River
020801090403	Gravelly Branch
020801090404	Clear Brook-Nanticoke River
020801090405	Butler Mill Branch-Nanticoke River
020801090406	Gales Creek-Nanticoke River
020801090501	Chicone Creek-Nanticoke River
020801090502	Rewastico Creek
020801090503	Quantico Creek
020801090504	Barren Creek-Nanticoke River
020801090505	Wetipquin Creek-Nanticoke River
020801100101	Middletown Branch-Transquaking River
020801100102	Chicamacomico River
020801100103	Doctors Creek-Transquaking River
020801100201	Buttons Creek-Blackwater River
020801100202	Little Blackwater River
020801100203	Coles Creek-Blackwater River
020801100204	Island Creek
020801100205	Upper Fishing Bay
020801100206	Lower Fishing Bay
020801100301	North Prong Wicomico River
020801100302	South Prong Wicomico River
020801100303	Tonytank Creek-Wicomico River
020801100304	Wicomico Creek
020801100305	Shiles Creek-Wicomico River
020801100306	Monie Bay
020801100307	Ellis Bay-Wicomico River
020801100401	Kings Creek
020801100402	Back Creek
020801100403	Taylor Branch-Manokin River
020801100404	Broad Creek-Manokin River
020801100405	Big Annemessex River
020801100501	Deal Island-Tangier Sound
020801100502	Kedges Straits-Tangier Sound
020801100503	Upper Tangier Sound Channel
020801100601	Little Annemessex River-Tangier Sound
020801100602	Smith Island
020801100603	Lower Tangier Sound Channel
020801110101	Upper Nassawango Creek
020801110102	Lower Nassawango Creek
020801110201	Careytown Branch-Pocomoke River
020801110202	Whaleyville Branch-Pocomoke River
020801110203	Ninepin Branch-Pocomoke River

HUC_12	SUBWATERSHED_DESCRIPTION
020801110204	Old Mill Branch-Pocomoke River
020801110205	Purnell Branch-Pocomoke River
020801110301	Dividing Creek
020801110302	Corkers Creek
020801110303	Cypress Swamp-Pocomoke River
020801110401	Pitts Creek
020801110402	Bullbegger Creek-Pocomoke River
020801110501	Marumsco Creek-Pocomoke Sound
020801110502	East Creek-Pocomoke Sound
020801110601	Beasley Bay-Messongo Creek
020801110602	Guilford Creek-Beasley Bay
020801110701	The Prong-Pocomoke Sound
020801110702	Deep Creek-The Thorofare
020801110703	Chesconessex Creek-Onancock Creek
020801110704	Pocomoke Sound Channel
020801110801	Pungoteague Creek-Lower Chesapeake Bay
020801110802	Nandua Creek-Lower Chesapeake Bay
020801110803	Occohannock Creek-Lower Chesapeake Bay
020801110804	Warehouse Creek-Nassawadox Creek
020801110901	Hungars Creek-Lower Chesapeake Bay
020801110902	Cherrystone Inlet-Lower Chesapeake Bay
020802010101	Dry Branch-Jackson River
020802010102	Bolar Run-Jackson River
020802010103	Warm Springs Run-Jackson River
020802010201	East Back Creek-Back Creek
020802010202	Jim Dave Run-Back Creek
020802010203	Little Back Creek
020802010204	Cummings Run-Back Creek
020802010205	Lake Moomaw-Jackson River
020802010301	Sweet Springs Creek-Cove Creek
020802010302	Cove Run-Dunlap Creek
020802010303	Johnsons Creek-Ogle Creek
020802010304	Jerrys Run-Dunlap Creek
020802010401	South Fork Potts Creek-North Fork Potts Creek
020802010402	Trout Branch-Potts Creek
020802010403	Mill Branch-Potts Creek
020802010404	Cast Steel Run-Potts Creek
020802010405	Hays Creek-Potts Creek
020802010501	Hot Springs Run-Cedar Creek
020802010502	Falling Spring Creek-Jackson River
020802010503	Indian Draft-Jackson River
020802010504	Pounding Mill Creek-Jackson River
020802010505	Karnes Creek-White Rock Creek
020802010506	Wilson Creek
020802010507	Smith Creek-Jackson River
020802010601	Wolfe Draft-Cowpasture River
020802010602	Shaws Fork
020802010603	Benson Run-Cowpasture River
020802010604	Daves Run-Bullpasture River

HUC_12	SUBWATERSHED_DESCRIPTION
020802010605	Crab Run-Bullpasture River
020802010701	Scotchtown Draft-Cowpasture River
020802010702	Dry Run
020802010703	Thompson Creek-Cowpasture River
020802010704	Lick Run-Stuart Run
020802010801	Mill Creek-Cowpasture River
020802010802	Pads Creek
020802010803	Simpson Creek-Cowpasture River
020802010901	Big Creek-James River
020802010902	Sinking Creek
020802010903	Smith Branch-Mill Creek
020802010904	Black Lick-James River
020802011001	Trout Creek-Craig Creek
020802011002	Meadow Creek
020802011003	Broad Run-Craig Creek
020802011101	Upper Johns Creek
020802011102	Lower Johns Creek
020802011201	Rolands Run Branch-Craig Creek
020802011202	Barbours Creek
020802011203	Mill Creek-Craig Creek
020802011204	Patterson Creek
020802011205	Roaring Run-Craig Creek
020802011301	Little Catawba Creek-Catawba Creek
020802011302	Town Branch-Catawba Creek
020802011401	Lapsley Run-James River
020802011402	Hickory Hollow Branch-James River
020802011403	Mill Creek-Looney Creek
020802011501	Purgatory Creek-James River
020802011502	North Creek-Jennings Creek
020802011503	Roaring Run-James River
020802011504	Spring Gap Creek-Cedar Creek
020802011505	Elk Creek-James River
020802020101	Chair Draft-Calfpasture River
020802020102	Ramseys Draft
020802020103	Holloway Draft-Calfpasture River
020802020104	Hamilton Branch
020802020105	Fridley Branch-Calfpasture River
020802020106	Cabin Creek-Mill Creek
020802020107	Brattons Run
020802020108	Guys Run-Calfpasture River
020802020201	Upper Little Calfpasture River
020802020202	Lower Little Calfpasture River
020802020203	Taylor Branch-Maury River
020802020204	Walker Creek
020802020205	Hays Creek
020802020301	Alone Mill Creek-Maury River
020802020302	Kerrs Creek
020802020303	Mill Creek-Maury River
020802020401	Saint Marys River

HUC_12	SUBWATERSHED_DESCRIPTION
020802020402	Upper South River
020802020403	Irish Creek
020802020404	Lower South River
020802020501	Bennetts Run-Maury River
020802020502	South Buffalo Creek
020802020503	North Buffalo Creek
020802020504	Colliers Creek
020802020505	Buffalo Creek
020802020506	Poague Run-Maury River
020802030101	Otter Creek-James River
020802030102	Reed Creek
020802030103	Thomas Mill Creek-James River
020802030201	Lynchburg Reservoir-Pedlar River
020802030202	Browns Creek-Pedlar River
020802030203	Horsley Creek-Pedlar River
020802030301	Judith Creek-James River
020802030302	Harris Creek
020802030303	Cheese Creek-Ivy Creek
020802030304	Blackwater Creek
020802030305	Opossum Creek-James River
020802030306	Beaver Creek
020802030401	Beck Creek-James River
020802030402	Stonewall Creek-James River
020802030403	Christian Mill Creek-James River
020802030404	Wreck Island Creek
020802030405	Allens Creek-James River
020802030406	Bent Creek
020802030501	South Fork Tye River-North Fork Tye River
020802030502	Cub Creek-Tye River
020802030503	Hat Creek
020802030504	Black Creek-Tye River
020802030505	Little Piney River-Piney River
020802030506	Naked Creek-Piney River
020802030601	North Fork Buffalo River-Buffalo River
020802030602	Stonewall Creek-Buffalo River
020802030603	Rutledge Creek
020802030604	Rocky Creek-Buffalo River
020802030701	Brown Creek-Tye River
020802030702	Rucker Run
020802030703	Joe Creek-Tye River
020802030801	David Creek
020802030802	Alabama Creek-James River
020802030803	Mallorys Creek-James River
020802030804	Sycamore Creek-James River
020802030901	North Fork Rockfish River
020802030902	South Fork Rockfish River
020802030903	Buck Creek-Rockfish River
020802031001	Hickory Creek-Cove Creek
020802031002	Dutch Creek-Rockfish River

HUC_12	SUBWATERSHED_DESCRIPTION
020802031003	Beaver Creek-Rockfish River
020802031101	Ballinger Creek-James River
020802031102	Rock Island Creek-James River
020802031103	Totier Creek
020802031104	Little George Creek-James River
020802031201	North Fork Hardware River
020802031202	South Fork Hardware River
020802031203	Turkey Run-Hardware River
020802031204	Woodson Creek-Hardware River
020802031301	Grease Creek-Slate River
020802031302	Meadow Creek-North River
020802031303	Horsepen Creek-Slate River
020802031304	Ripley Creek-Walton Fork
020802031401	Joshua Creek-Slate River
020802031402	Sharps Creek-Slate River
020802031403	Hunts Creek-Slate River
020802031501	Bremo Creek-James River
020802031502	Bear Garden Creek-James River
020802040101	Stockton Creek-Mechums River
020802040102	Beaver Creek-Mechums River
020802040103	North Moormans River-Moormans River
020802040104	Doyles River
020802040105	Wards Creek-Moormans River
020802040201	Buck Mountain Creek
020802040202	Little Ivy Creek-Ivy Creek
020802040203	South Fork Rivanna River
020802040301	Lynch River-North Fork Rivanna River
020802040302	Swift Run
020802040303	Jacobs Run-North Fork Rivanna River
020802040304	Preddy Creek
020802040305	Flannigan Branch-North Fork Rivanna River
020802040401	Meadow Creek-Rivanna River
020802040402	Moores Creek
020802040403	Buck Island Creek
020802040404	Carroll Creek-Rivanna River
020802040405	Mechunk Creek
020802040501	Stigger Creek-Rivanna River
020802040502	Cunningham Creek
020802040503	Ballinger Creek
020802040504	Carys Creek-Rivanna River
020802050101	Upper Byrd Creek
020802050102	Middle Byrd Creek
020802050103	Lower Byrd Creek
020802050201	Bishop Creek-Willis River
020802050202	Whispering Creek-Willis River
020802050203	Little Willis River
020802050204	Buffalo Creek-Willis River
020802050205	Hatcher Creek
020802050301	Bonbrook Creek-Willis River

HUC_12	SUBWATERSHED_DESCRIPTION
020802050302	Randolph Creek
020802050303	Trice Lake-Willis River
020802050401	Hooper Rock Creek-James River
020802050402	Muddy Creek
020802050403	Picketts Creek-James River
020802050404	Maxey Mill Creek-Deep Creek
020802050405	Sallee Creek-Deep Creek
020802050501	Solomons Creek-James River
020802050502	Little Lickinghole Creek
020802050503	Big Lickinghole Creek
020802050504	Mohawk Creek-James River
020802050505	Beaverdam Creek
020802050601	Fine Creek-James River
020802050602	Norwood Creek
020802050603	Little River-James River
020802050604	Bernards Creek-James River
020802050605	Tuckahoe Creek
020802050606	East Branch Tuckahoe Creek-James River
020802050607	Little Westham Creek-James River
020802060101	Almond Creek-James River
020802060102	Falling Creek
020802060103	Proctors Creek-James River
020802060104	Fourmile Creek
020802060105	Turkey Island Creek
020802060106	Curles Creek-James River
020802060201	Bailey Creek-James River
020802060202	Powell Creek
020802060203	Herring Creek
020802060204	Courthouse Creek-Queens Creek
020802060205	Flowerdew Hundred Creek-James River
020802060301	Wards Creek
020802060302	Kittewan Creek-James River
020802060303	Upper Chippokes Creek
020802060304	Sunken Meadow Pond-James River
020802060401	Grassy Swamp Creek-Chickahominy River
020802060402	Stony Run-Chickahominy River
020802060403	Upham Brook
020802060501	Powhite Creek-Chickahominy River
020802060502	Higgins Swamp-Chickahominy River
020802060503	White Oak Swamp
020802060504	Toe Ink Swamp-Chickahominy River
020802060505	Rumley Marsh-Chickahominy River
020802060506	Big Swamp-Chickahominy River
020802060601	Barrows Creek-Chickahominy River
020802060602	Diascund Creek Reservoir-Diascund Creek
020802060603	Mill Creek-Diascund Creek
020802060604	Yarmouth Creek-Chickahominy River
020802060605	Morris Creek-Chickahominy River
020802060701	Broad Swamp-James River

HUC_12	SUBWATERSHED_DESCRIPTION
020802060702	Powhatan Creek
020802060703	Grays Creek
020802060704	Lower Chippokes Creek-James River
020802060801	College Creek
020802060802	Skiffes Creek-James River
020802060803	Lawnes Creek
020802060804	Morrisons Creek-James River
020802060901	Warwick River
020802060902	Warren Creek-Pagan River
020802060903	Cypress Creek
020802060904	Jones Creek-Pagan River
020802060905	Chuckatuck Creek
020802060906	Cooper Creek-James River
020802070101	Spring Creek
020802070102	Little Buffalo Creek-Buffalo Creek
020802070103	Locket Creek-Buffalo Creek
020802070201	Wolf Creek-Appomattox River
020802070202	Suanee Creek-Appomattox River
020802070203	Fishpond Creek-Appomattox River
020802070204	Vaughans Creek
020802070205	Ducker Creek-Appomattox River
020802070206	Bad Luck Branch-Appomattox River
020802070301	Briery Creek
020802070302	Evans Creek-Bush River
020802070303	Mountain Creek
020802070304	Millers Creek-Bush River
020802070305	Sandy River
020802070401	Saylers Creek
020802070402	Angola Creek-Appomattox River
020802070403	Sandy Creek-Appomattox River
020802070404	Big Guinea Creek
020802070405	Little Guinea Creek-Appomattox River
020802070406	Stock Creek
020802070501	Little Creek-Flat Creek
020802070502	Beaverpond Creek-Flat Creek
020802070503	Nibbs Creek
020802070504	Haw Branch-Flat Creek
020802070601	Bent Creek-Appomattox River
020802070602	Rocky Ford Creek
020802070603	Skinquarter Creek-Appomattox River
020802070604	Smacks Creek-Appomattox River
020802070701	Little Creek-Deep Creek
020802070702	Cellar Creek
020802070703	West Creek
020802070704	Sweathouse Creek-Deep Creek
020802070705	Beaverpond Creek-Deep Creek
020802070801	Winticomack Creek-Appomattox River
020802070802	Winterpock Creek
020802070803	Nooning Creek-Appomattox River

HUC_12	SUBWATERSHED_DESCRIPTION
020802070804	Namozine Creek
020802070805	Whipponock Creek
020802070806	Cattle Creek-Lake Chesdin
020802070901	Swift Creek Reservoir-Swift Creek
020802070902	Third Branch-Swift Creek
020802070903	Second Branch-Licking Creek
020802070904	Franks Branch-Swift Creek
020802071001	Oldtown Creek-Appomattox River
020802071002	Ashton Creek-Appomattox River
020802080101	Speights Run-Lake Kilby
020802080102	Cohoon Creek
020802080103	Lake Prince
020802080104	Western Branch Reservoir
020802080105	Cedar Lake-Nansemond River
020802080106	Bennett Creek-Nansemond River
020802080201	New Mill Creek-Southern Branch Elizabeth River
020802080202	Big Entry Ditch-Dismal Swamp
020802080203	Deep Creek-Southern Branch Elizabeth River

HUC_12	SUBWATERSHED_DESCRIPTION
020802080204	Eastern Branch Elizabeth River
020802080205	Western Branch Elizabeth River
020802080206	Elizabeth River
020802080301	Streeter Creek-Hampton Roads
020802080302	Willoughby Bay
020802080303	Hampton River-Hampton Roads
020802080304	Hampton Roads Channel
030102020102	Blackwater Swamp
050200040501	Horseshoe Run
050200060101	Maple Run-Youghiogeny River
050200060102	Cherry Creek
050200060104	Little Youghiogheny River
050200060201	Upper Deep Creek Lake
050200060202	Lower Deep Creek Lake
050200060401	Headwaters Casselman River
050200060403	Red Run-Piney Creek
050200060701	Flaugherty Creek
20402070101	Duck Creek

TAB_IBI_PARAMETER

Field Name	Description	Data Type	Length
IBI_PARAMETER (PK)	IBI PARAMETER CODE- IBI Parameter code.	varchar	15
IBI_PARAMETER_DESCRIPTION (NN)	IBI PARAMETER DESCRIPTION- IBI Parameter code description	varchar	75

1) General: This table stores information related exclusively to the IBI_PARAMETER codes in codes in the IBI_METRICS_TABLE. This table contains information to parameter names. Currently accepted PARAMETER and DESCRIPTION designations are as follows:

IBI_PARAMETER	IBI_PARAMETER_DESCRIPTION
ASPT_MOD	Average tolerance Score Per Taxon modified to family level
ASPT_MOD_R	Average tolerance Score Per Taxon modified to family level calculated on Rarefaction Data
BECK	Beck's Index-classical calculation
BECK_R	Beck's Index-classical calculation-calculated on Rarefaction Data
DIPTERA_TAXA_CNT	Number of Diptera Taxa
DIPTERA_TAXA_CNT_R	Number of Diptera Taxa -calculated on Rarefaction Data
EPHEMEROPTERA_TAXA_CNT	Number of Ephemeroptera Taxa
EPHEMEROPTERA_TAXA_CNT_R	Number of Ephemeroptera Taxa-calculated on Rarefaction Data
EPT_TAXA_ABUND	Total abundance of Ephemeroptera, Plecoptera & Trichoptera
EPT_TAXA_ABUND_R	Total abundance of Ephemeroptera, Plecoptera & Trichoptera-calculated on Rarefaction Data
EPT_TAXA_COUNT	Number of Ephemeroptera, Plecoptera & Trichoptera taxa
EPT_TAXA_COUNT_NO_TOL	Number of Ephemeroptera, Plecoptera & Trichoptera taxa excluding tolerant taxa
EPT_TAXA_COUNT_NO_TOL_R	Number of Ephemeroptera, Plecoptera & Trichoptera taxa excluding tolerant taxa-calculated on Rarefaction Data
EPT_TAXA_COUNT_R	Number of Ephemeroptera, Plecoptera & Trichoptera taxa-calculated on Rarefaction Data
FBI	Family level Hilsenhoff Biotic Index
FBI_R	Family level Hilsenhoff Biotic Index-calculated on Rarefaction Data
GOLD	1- relative percentage abundance of Gastropods, Oligochaeta and Diptera
GOLD_R	1- relative percentage abundance of Gastropods, Oligochaeta and Diptera-calculated on Rarefaction Data
MARGALEFS	Margalef Index of community diversity
MARGALEFS_R	Margalef Index of community diversity-calculated on Rarefaction Data -calculated on Rarefaction Data
NCO_TAXA_CNT	Total Taxa Count omitting chironomidae and oligochaeta
NCO_TAXA_CNT_R	Total Taxa Count omitting chironomidae and oligochaeta-calculated on Rarefaction Data
NON_INSECT_TAXA_CNT	Non-Insect Taxa Count
NON_INSECT_TAXA_CNT_R	Non-Insect Taxa Count-calculated on Rarefaction Data
PCT_BURROWER	Percent Burrower Abundance
PCT_BURROWER_R	Percent Burrower Abundance-calculated on Rarefaction Data
PCT_CHIRONOMIDAE	Percent Chironomidae Abundance
PCT_CHIRONOMIDAE_R	Percent Chironomidae Abundance-calculated on Rarefaction Data
PCT_CLIMB	Percent Climbers Abundance
PCT_CLIMB_R	Percent Climbers Abundance-calculated on Rarefaction Data
PCT_CLING	Percent Clinger Abundance
PCT_CLING_R	Percent Clinger Abundance-calculated on Rarefaction Data
PCT_CLINGER_TAXA	Percent Clinger taxa
PCT_CLINGER_TAXA_R	Percent Clinger taxa-calculated on Rarefaction Data
PCT_COLLECT	Percent Collector Abundance
PCT_COLLECT_R	Percent Collector Abundance-calculated on Rarefaction Data
PCT_DIPTERA	Percent Diptera Abundance

IBI_PARAMETER	IBI_PARAMETER_DESCRIPTION
PCT_DIPTERA_R	Percent Diptera Abundance-calculated on Rarefaction Data
PCT_DOM1	Percent Dominant taxa
PCT_DOM1_R	Percent Dominant taxa-calculated on Rarefaction Data
PCT_DOM2	Percent Two Dominant taxa
PCT_DOM2_R	Percent Two Dominant taxa-calculated on Rarefaction Data
PCT_DOM3	Percent Three Dominant taxa
PCT_DOM3_R	Percent Three Dominant taxa-calculated on Rarefaction Data
PCT_EPHEMEROPTERA	Percent Ephemeroptera Abundance
PCT_EPHEMEROPTERA_R	Percent Ephemeroptera Abundance-calculated on Rarefaction Data
PCT_EPT	Percent Ephemeroptera, Plecoptera & Trichoptera Abundance
PCT_EPT_R	Percent Ephemeroptera, Plecoptera & Trichoptera Abundance-calculated on Rarefaction Data
PCT_EPT_TAXA_RICH	Percent Ephemeroptera, Plecoptera & Trichoptera Taxa
PCT_EPT_TAXA_RICH_R	Percent Ephemeroptera, Plecoptera & Trichoptera Taxa-calculated on Rarefaction Data
PCT_FILTERERS	Percent Filterer Abundance
PCT_FILTERERS_R	Percent Filterer Abundance-calculated on Rarefaction Data
PCT_GATHER	Percent Gatherer Abundance
PCT_GATHER_R	Percent Gatherer Abundance-calculated on Rarefaction Data
PCT_LIMESTONE	Percent Limestone Taxa (Isopod + Amphidoda + Ephemerella) Abundance
PCT_LIMESTONE_R	Percent Limestone Taxa (Isopod + Amphidoda + Ephemerella) Abundance-calculated on Rarefaction Data
PCT_NET_CADDISFLY	Percent Net Caddisflies Abundance
PCT_NET_CADDISFLY_R	Percent Net Caddisflies Abundance-calculated on Rarefaction Data
PCT_NON_INSECT	Percent Non-Insects Abundance
PCT_NON_INSECT_R	Percent Non-Insects Abundance-calculated on Rarefaction Data
PCT_PLECOPTERA	Percent Plecoptera Abundance
PCT_PLECOPTERA_R	Percent Plecoptera Abundance-calculated on Rarefaction Data
PCT_PREDATOR	Percent Predator Abundance
PCT_PREDATOR_R	Percent Predator Abundance-calculated on Rarefaction Data
PCT_SCRAPER	Percent Scrapers Abundance
PCT_SCRAPER_R	Percent Scrapers Abundance-calculated on Rarefaction Data
PCT_SENSITIVE	Percent Sensitive (TV<=3) Abundance
PCT_SENSITIVE_R	Percent Sensitive (TV<=3) Abundance-calculated on Rarefaction Data
PCT_SHREDDER	Percent Shredder Abundance
PCT_SHREDDER_R	Percent Shredder Abundance-calculated on Rarefaction Data
PCT_SIMULIIDAE	Percent Simulidate (Blackfly) Abundance
PCT_SIMULIIDAE_R	Percent Simulidate (Blackfly) Abundance-calculated on Rarefaction Data
PCT_SWIMMER	Percent Swimmers Abundance
PCT_SWIMMER_R	Percent Swimmers Abundance-calculated on Rarefaction Data
PCT_TOLERANT	Percent Tolerant (TV>=7) Abundance
PCT_TOLERANT_R	Percent Tolerant (TV>=7) Abundance-calculated on Rarefaction Data
PCT_TRICHOPTERA	Percent Trichoptera Abundance
PCT_TRICHOPTERA_NO_TOL	Percent Trichoptera Abundance excluding Hydropsychidae
PCT_TRICHOPTERA_NO_TOL_R	Percent Trichoptera Abundance excluding Hydropsychidae-calculated on Rarefaction Data
PCT_TRICHOPTERA_R	Percent Trichoptera Abundance-calculated on Rarefaction Data
PLECOPTERA_TAXA_CNT	Plecoptera Taxa count
PLECOPTERA_TAXA_CNT_R	Plecoptera Taxa count-calculated on Rarefaction Data
RATIO_SC_TO_CF	Ratio of Scrapers to Collector Filterers
RATIO_SC_TO_CF_R	Ratio of Scrapers to Collector Filterers-calculated on Rarefaction Data
RATIO_SC_TO_SH	Ratio of Scrapers to Shredders
RATIO_SC_TO_SH_R	Ratio of Scrapers to Shredders-calculated on Rarefaction Data

IBI_PARAMETER	IBI_PARAMETER_DESCRIPTION
RATIO_SH_TO_CG	Ratio of Shredders to Collector Gatherers
RATIO_SH_TO_CG_R	Ratio of Shredders to Collector Gatherers-calculated on Rarefaction Data
SCRAPER_TAXA_CNT	Scraper Taxa Count
SCRAPER_TAXA_CNT_R	Scraper Taxa Count-calculated on Rarefaction Data
SENSITIVE_TAXA_COUNT	Sensitve Taxa Count (TV<=3)
SENSITIVE_TAXA_COUNT_R	Sensitve Taxa Count (TV<=3)-calculated on Rarefaction Data
SIMPSON_DIVERSITY	Simpson's Diversity Index
SIMPSON_DIVERSITY_R	Simpson's Diversity Index-calculated on Rarefaction Data
SW	Shannon Weiner Diversity Index
SW_R	Shannon Weiner Diversity Index-calculated on Rarefaction Data
TAXA_RICH	Total Taxa Count
TAXA_RICH_R	Total Taxa Count -calculated on Rarefaction Data
TOLERANT_TAXA_COUNT	Tolerant Taxa Count (TV>=7)
TOLERANT_TAXA_COUNT_R	Tolerant Taxa Count (TV>=7)-calculated on Rarefaction Data
TOTAL_ABUNDANCE	Total Abundance
TOTAL_ABUNDANCE_R	Total Abundance-calculated on Rarefaction Data
TOTAL_SCORE	Total IBI Score for Site
TRICHOPTERA_TAXA_CNT	Trichoptera Taxa Count
TRICHOPTERA_TAXA_CNT_R	Trichoptera Taxa Count-calculated on Rarefaction Data
TRICHOPTERA_TAXA_COUNT_NO_HYDR	Trichoptera Taxa Count excluding Hydropsychidae
TRICHOPTERA_TAXA_COUNT_NO_R	Trichoptera Taxa Count excluding Hydropsychidae-calculated on Rarefaction Data

TAB_KARST

Field Name	Description	Data Type	Length
KARST (PK)	KARST PARAMETER CODE- KARST area designation code.	varchar	10
KARST_ DESCRIPTION (NN)	KARST_DESCRIPTION- KARST area designation code. description	varchar	50

1) General: This table stores information related exclusively to the KARST codes in codes in the TAB_STATIONS TABLE. Information is based on the USGS Preliminary Map of Potentially Karstic Carbonate Rocks in the Central and Southern Appalachian States: complete OF 2008-1154 map plate. (http://pubs.usgs.gov/of/2008/1154/) This table contains information to parameter names. Currently accepted KARST codes and KARST DESCRIPTION designations are as follows:

KARST	KARST_DECSRIPTION
CPU	COASTAL PLAIN UNCONSOLIDATED SEDIMENTS
FFC	FOLDED FAULTED CARBONATED ROCK
FFCG	FOLDED FAULTED CARBONATED ROCK WITH GLACIAL COVER
GC	GENTLY FOLDED CARBONATED ROCK
GCG	GENTLY FOLDED CARBONATED ROCK WITH GLACIAL COVER
M	MARBLE AND METALIMESTONE
NONE	NON-CARST AREA
TJB	TRIASSIC-JURASSIC AGED BASIN

TAB_LABATORY

Field Name	Description	Data Type	Length
LAB (PK)	LAB-CIMS laboratory code.	varchar	10
LAB_ DESCRIPTION (NN)	LAB_DESCRIPTION-The name of the laboratory.	varchar	150
DIRECTOR	DIRECTOR-The name of the laboratory's director.	varchar	50
LAB_ADDRESS	LAB_ADDRESS-Street (mailing) address of the laboratory.	varchar	100
CITY	CITY-City in the laboratory's mailing address.	varchar	50
STATE	STATEState abbreviation of the laboratory's mailing address.	char	2
ZIP	ZIP-Zip Code of the laboratory's mailing address.	varchar	10
PHONE	PHONE-Laboratory phone number.	varchar	25

1) GENERAL: TAB_LABORATORY contains CIMS laboratory codes and descriptive fields. The lab code identifies the laboratory that analyzed the water samples. This table stores information related exclusively to the LAB codes in codes in the TAB_WQ_DATA table. This table contains information to parameter names. Currently accepted LAB and LAB_DESCRIPTION designations are as follows:

LAB	LAB_DESCRIPTION
AMRL	OLD DOMINION UNIVERSITY APPLIED MARINE RESEARCH LABORATORY (THIS LAB BECAME THE ODU LAB IN MAY, 2000)
BPFL	BLUE PLAINS FIELD LABORATORY
CBL	UNIVERSITY OF MARYLAND CHESAPEAKE BIOLOGICAL LABORATORY
CRL	USEPA-CENTRAL REGIONAL LABORATORY (MOVED TO FT MEADE 1999)
DENREC	DELAWARE DEPARTMENT OF NATURAL RESOURCES AND ENVIRONMENTAL CONTROL
ELB	DISTRICT OF COLUMBIA DEPT OF HEALTH ENVIRONMENTAL LABORATORY BRANCH AT EPA CRL
EPA-UNSP	EPA LABORATORY-UNSPECIFIED
FIELD	FIELD PARAMETER-NO LAB ASSOCIATED WITH SAMPLE
MDHMH	MARYLAND DEPARTMENT OF HEALTH AND MENTAL HYGIENE
MDHMH-WM	MARYLAND DEPARTMENT OF HEALTH AND MENTAL HYGIENE-WESTERN MARYLAND LAB
NYDEC	COLUMBIA ANALYTICAL SERVICES
ODU	OLD DOMINION UNIVERSITY LABORATORY
OWML	OCCOQUAN WATERSHED MONITORING LABORATORY
PADEP	PENNSYLVANIA DEPARTMENT OF ENVIRONMENTAL PROTECTION LAB
SMRP	ST MARY'S RIVER PROJECT
SRBC	SUSQUEHANNA RIVER BASIN COMMISSION LAB
UMCES-AL	UNIVERSITY OF MARYLAND APPALACHIAN LABORATORY
UNKNOWN	UNKNOWN OR LAB NOT SPECIFIED
USGS-KDSL	UNITED STATES GEOLOGICAL SURVEY KENTUCKY DISTRICT SEDIMENT LABORATORY
USGS-NWQL	UNITED STATES GEOLOGICAL SURVEY NATIONAL WATER QUALITY LABORATORY
USGS-SED	USGS SEDIMENT LABORATORY IN KENTUCKY
VADCLS	VIRGINIA DIVISION OF CONSOLIDATED LABORATORY SERVICES
VCU	VIRGINIA COMMONWEALTH UNIVERSITY
VIMS	VIRGINIA INSTITUTE OF MARINE SCIENCE

TAB_LIFE_STAGE

Field Name	Description	Data Type	Length
LIFE_STAGE (PK,FK)	LIFE STAGE CODE- Chesapeake Bay Program Life Stage Code	char	3
LIFE_STATE_ DESCRIPTION (NN)	DESCRIPTION-Detailed Life Stage code Description	varchar	50

1) General: This table stores information in relating to the identification of species life stages in the TAB_TAXONOMIC_COUNT table. The currently accepted LIFE_STAGE values and DESCRIPTIONS are as follows:

LIFE_ STAGE _CODE	LIFE_STAGE_DESCRIPTION
00	EGG
01	YOLK SAC
02	FIN FOLD
03	POST FIN FOLD
04	YEAR CLASS O
05	YEAR CLASS 1 OR OLDER
06	JUVENILES AND ADULTS
07	LARVAE AND JUVENILES AND ADULTS
08	LARVAE AND JUVENILES
09	NAUPLII AND PERITRICHS
10	NAUPLII OR COPEPODITE
11	NAUPLII
12	COPEPODITE
13	ORTHONAUPLII STAGE 1-3
14	METANAUPLII STAGE 4-6
15	COPEPODITE STAGE 1-3
16	COPEPODITE STAGE 4-6
17	CYPRIS LARVAE
18	RESERVED FOR FUTURE USE
19	COPEPOD EGG
20	NYMPH
21	PUPAE
22	PHARATE
23	INSTAR
24	NAIAD
25	HATCHERTY MARKED ORGANISM
26	YEAR CLASS 2 OR OLDER
27	AGE 0 MDDNR HATCHERTY MARKED ORGANISM
28	AGE 1 MDDNR HATCHERTY MARKED ORGANISM
29	AGE 2 OR GREATER MDDNR HATCHERTY MARKED ORGANISM
30	PREZOEA
31	ZOEA
32	METAZOEA
33	MEGALOPS
34	MALE,UNSPECIFIED AGE
35	FEMALE, ADULT
36	FEMALE,JUVENILE
37	MDDNR HATCHERTY MARKED ORGANISM
38	MALE, AGE CLASS 0
39	MALE, AGE CLASS 1
40	NAUPLII STAGE 1
41	NAUPLII STAGE 2

LIFE_ STAGE _CODE	LIFE_STAGE_DESCRIPTION
42	NAUPLII STAGE 3
43	NAUPLII STAGE 4
44	NAUPLII STAGE 5
45	NAUPLII STAGE 6
46	COPEPODITE STAGE 1
47	COPEPODITE STAGE 2
48	COPEPODITE STAGE 3
49	COPEPODITE STAGE 4
50	COPEPODITE STAGE 5
51	COPEPODITE STAGE 6
52	SPECIES A
53	SPECIES B
54	SPECIES C
55	SPECIES D
56	SPECIES E
57	SPECIES F
58	SPECIES A-FULL
59	SPECIES A-EMPTY
60	SPECIES B-FULL
61	SPECIES B-EMPTY
62	SPECIES C-FULL
63	SPECIES C-EMPTY
64	EMBRYO
65	NEONITES
66	MALE, AGE CLASS 2
67	FEMALE, IMMATURE AGE CLASS 0
68	FEMALE, IMMATURE AGE CLASS 1
69	FEMALE, MATURE AGE CLASS 1
70	FEMALE, MATURE AGE CLASS 2
71	FEMALE, MATURE AGE CLASS 0
72	FEMALE, IMMATURE AGE CLASS 2
73	SALPS
74	MALE, ADULT
75	FEMALE,UNSPECIFIED AGE
76	GROUP
77	WITH CAP. SETAE
78	WITHOUT CAP. SETAE
79	SPP.
80	MOLTED
81	UNMOLTED
82	LARGE
83	LARGE-FULL
84	LARGE-EMPTY

LIFE_STAGE_CODE	LIFE_STAGE_DESCRIPTION
85	FULL
86	EMPTY
87	MEDIUM
88	SMALL
89	NOT SPECIFIED
90	EGG- NOT VIABLE
91	SUBADULT
92	POST LARVAL
93	JUVENILE
94	TAXON WITH COUNT STORED AS VOLUME IN MILLILITERS
95	MATURE
96	IMMATURE
97	LARVAE
98	ADULT
99	NOT APPLICABLE
100	20:49UM LENGTH <20UM WIDTH
101	20:49UM LENGTH
102	20:49UM LENGTH 50:99UM WIDTH
103	20:49UM LENGTH 20:49UM WIDTH CUP
104	20:49UM LENGTH 20:49UM WIDTH CONE
105	20:49UM LENGTH 20:49UM WIDTH
106	20:49UM LENGTH >20UM WIDTH
107	>200UM LENGTH
108	20:49UM LENGTH <20UM WIDTH CONE
109	50:99UM LENGTH <20UM WIDTH
110	100:199UM LENGTH >20UM WIDTH
111	>20UM WIDTH
112	<20UM LENGTH
113	<20UM LENGTH <20UM WIDTH CUP
114	<20UM LENGTH <20UM WIDTH CONE
115	<20UM LENGTH <20UM WIDTH
116	<20UM LENGTH CONE
117	20:49UM LENGTH <20UM WIDTH CUP
118	50:99UM LENGTH EMPTY
119	SPECIES C 100:199UM LENGTH 100:199UM WIDTH
120	SPECIES B 50:99UM LENGTH 50:99UM WIDTH
121	SPECIES B 50:99UM LENGTH 20:49UM WIDTH
122	PARVULA GRP FULL
123	PARVULA GRP
124	20:49UM LENGTH FULL
125	BEROIDEA GRP
126	SPECIES C 100:199UM LENGTH 20:49UM WIDTH
127	50:99UM LENGTH
128	50:99UM LENGTH 50:99UM WIDTH CUP
129	50:99UM LENGTH 50:99UM WIDTH CONE
130	50:99UM LENGTH 20:49UM WIDTH CUP
131	50:99UM LENGTH 20:49UM WIDTH CONE
132	50:99UM LENGTH >20UM WIDTH
133	<20UM LENGTH 100:199UM WIDTH
134	<20UM LENGTH 100:199UM WIDTH CONE
135	<20UM LENGTH 100:199UM WIDTH CUP
136	<20UM LENGTH 20:49UM WIDTH
137	<20UM LENGTH 20:49UM WIDTH CONE
138	<20UM LENGTH 20:49UM WIDTH CUP
139	<20UM LENGTH 50:99UM WIDTH
140	<20UM LENGTH 50:99UM WIDTH CONE
141	<20UM LENGTH 50:99UM WIDTH CUP
142	<20UM LENGTH CUP
143	<20UM LENGTH EMPTY

LIFE_STAGE_CODE	LIFE_STAGE_DESCRIPTION
144	<20UM LENGTH FULL
145	<20UM WIDTH
146	<20UM WIDTH EMPTY
147	<20UM WIDTH FULL
148	>200UM
149	>200UM EMPTY
150	>200UM FULL
151	>200UM LENGTH <20UM WIDTH
152	>200UM LENGTH <20UM WIDTH EMPTY
153	>200UM LENGTH <20UM WIDTH FULL
154	>200UM LENGTH >20UM WIDTH
155	>200UM LENGTH >20UM WIDTH EMPTY
156	>200UM LENGTH >20UM WIDTH FULL
157	>200UM LENGTH EMPTY
158	>200UM LENGTH FULL
159	>20UM WIDTH EMPTY
160	>20UM WIDTH FULL
161	100:199UM LENGTH <20UM WIDTH
162	100:199UM LENGTH <20UM WIDTH CONE
163	100:199UM LENGTH <20UM WIDTH CUP
164	100:199UM LENGTH <20UM WIDTH EMPTY
165	100:199UM LENGTH <20UM WIDTH FULL
166	100:199UM LENGTH >20UM WIDTH EMPTY
167	100:199UM LENGTH >20UM WIDTH FULL
168	100:199UM LENGTH 100:199UM WIDTH CONE
169	100:199UM LENGTH 100:199UM WIDTH CUP
170	100:199UM LENGTH 20:49UM WIDTH CONE
171	100:199UM LENGTH 20:49UM WIDTH CUP
172	100:199UM LENGTH 50:99UM WIDTH CONE
173	100:199UM LENGTH 50:99UM WIDTH CUP
174	100:199UM LENGTH CONE
175	100:199UM LENGTH CUP
176	100:199UM LENGTH
177	100:199UM LENGTH EMPTY
178	100:199UM LENGTH FULL
179	20:49UM LENGTH <20UM WIDTH EMPTY
180	20:49UM LENGTH <20UM WIDTH FULL
181	20:49UM LENGTH >20UM WIDTH EMPTY
182	20:49UM LENGTH >20UM WIDTH FULL
183	20:49UM LENGTH 100:199UM WIDTH
184	20:49UM LENGTH 100:199UM WIDTH CONE
185	20:49UM LENGTH 100:199UM WIDTH CUP
186	20:49UM LENGTH 50:99UM WIDTH CONE
187	20:49UM LENGTH 50:99UM WITDH CUP
188	20:49UM LENGTH CONE
189	20:49UM LENGTH CUP
190	20:49UM LENGTH EMPTY
191	RESERVED FOR FUTURE USE
192	50:99UM LENGTH <20UM WIDTH CONE
193	50:99UM LENGTH <20UM WIDTH CUP
194	50:99UM LENGTH <20UM WIDTH EMPTY
195	50:99UM LENGTH <20UM WIDTH FULL
196	50:99UM LENGTH >20UM WIDTH EMPTY
197	50:99UM LENGTH >20UM WIDTH FULL
198	50:99UM LENGTH 100:199UM WIDTH CONE
199	50:99UM LENGTH 100:199UM WIDTH CUP
200	50:99UM LENGTH CONE
201	50:99UM LENGTH CUP
202	RESERVED FOR FUTURE USE
203	RESERVED FOR FUTURE USE

LIFE_ STAGE _CODE	LIFE_STAGE_DESCRIPTION
204	50:99UM LENGTH FULL
205	BEROIDEA GRP EMPTY
206	BEROIDEA GRP FULL
207	LARVAE 20:49UM LENGTH
208	PARVULA GRP EMPTY
209	SMALL EMPTY
210	SMALL FULL
211	SPECIES B 50:99UM LENGTH
212	SPECIES B 50:99UM LENGTH <20UM WIDTH
213	SPECIES B 50:99UM LENGTH 100:199UM WIDTH
214	SPECIES C 100:199UM LENGTH
215	SPECIES C 100:199UM LENGTH <20UM WIDTH
216	SPECIES C 100:199UM LENGTH 50:99UM WIDTH
217	SPECIES V
218	SPECIES W
219	SPECIES X
220	SPECIES Y
221	SPECIES Z
222	SPECIES 1
223	SPECIES 2
224	SPECIES 3
225	COMPLEX
226	SPECIES G
227	SPECIES H

LIFE_ STAGE _CODE	LIFE_STAGE_DESCRIPTION
228	SPECIES I
229	SPECIES J
230	SPECIES K
231	SPECIES L
232	SPECIES M
233	SPECIES N
234	SPECIES O
235	SPECIES P
236	SPECIES Q
237	SPECIES R
238	SPECIES S
239	SPECIES T
240	SPECIES U
241	SPECIES 4
242	SPECIES 5
243	SPECIES 6
244	POLYPS
245	TYPE
246	VARIETY
247	IMMATURE WITH CAP. CHAETE
248	IMMATURE WITHOUT CAP. CHAETE
249	FRAGMEMTS
250	EPHYRA
251	POST EPHYRA

TAB_LL_DATUMS

Field	Description	Type	Length
LL_DATUM (PK)	GEOGRAPHIC DATUM CODE- Latitude/longitude datum code	char	5
LL_DATUM_ DESCRIPTION (NN)	Description-definition of GEOGRAPHIC DATUM	varchar	50

1) GENERAL- This table stored geographic datum descriptions for codes in the TAB_EVENT and TAB_STATIONS tables. The LL_DATUM code defines the datum under which the latitude and longitude measurements for a particular station were calculated. The currently accepted LL_DATUM and DESCRIPTIONS are as follows:

LL_DATUM	LL_DATUM_DESCRIPTION
NAD27	1927 NORTH AMERICAN DATUM
NAD83	1983 NORTH AMERICAN DATUM
WGS84	WORLD GEODETIC SYSTEM 1984
UNID	UNKNOWN DATUM

TAB_IBI_METHOD

Field Name	Description	Data Type	Length
IBI_METHOD (PK,FK)	IBI ANALYTICAL METHOD CODE- Method Description code	char	8
IBI_METHOD_ TITLE	IBI_METHOD_TITLE-IBI method title	varchar	50
IBI_METHOD_ DESCRIPTION (NN)	IBI_METHOD_DESCRIPTION-Basic description of IBI caclucation meton	varchar	max
IBI_METHOD DETAILS	IBI_METHOD_DETAILS- additional details for method	varchar	max

1) General: This table stores information related exclusively to IBI_METHOD codes in the TAB_IBI_METRIC table. This table contains descriptions of the field and laboratory methods for parameter determination. The IBI_METHOD code is used to define the field or lab procedure used to obtain the parameter value. IBI_METHODS designations are as follows:

IBI_METHOD	IBI_METHOD_TITLE
CBP_2010	Chesapeake Bay Program Chessie NT BIBI 2010
CBP_MACS	Cheapeake Bay Program Family level Mid Atlantic Coastal IBI

TAB_WQ_METHOD

Field Name	Description	Data Type	Length
WQ_METHOD_ ID	WQ_METHOD_ID-Autogenerated Number	int	
REPORTING_ PARAMETER (PK,NN)	REPORTING_PARAMETER-CBP Reporting Parameter Name	varchar	15
WQ_METHOD (PK,NN)	WQ_METHOD-CBP Method Code Assignment	char	4
EPA_METHOD	EPA_METHOD-EPA Storet Method code	varchar	50
WQ_TITLE	WQ_TITLE-Analytical Method Title	varchar	100
WQ_ DESCRIPTION	WQ_DESCRIPTION-Analytical Method Title	varchar	max
REFERENCE1	REFERENCE1-Reference for Method	varchar	900
REFERENCE2	REFERENCE2-Reference for Method	varchar	900
REFERENCE3	REFERENCE3-Reference for Method	varchar	900
REFERENCE4	REFERENCE4-Reference for Method	varchar	900
WQ_DETAILS	WQ_DETAILS-additional details	varchar	50
INSTRUMENTS	INSTRUMENTS-Analytical Instrumentation details	varchar	500

1) General: This table stores information related exclusively to WQ_METHOD codes in the TAB_WQ_DATA table. This table contains descriptions of the field and laboratory methods for parameter determination. The METHOD code is used to define the field or lab procedure used to obtain the parameter value. For Currently accepted WQ_METHODS designations PLEASE SEE DATABASE FOR DETAILS.

REPORTING_PARAMETER	WQ_METHOD
ACIDITY	L01
AL	L01
ANC	L01
ANC	L02
AS	L01
BATT	NA
BIOSI	L01
BOAT_SPEED	NA
BOD20F	L01
BOD20W	L01
BOD5F	L01
BOD5W	L01
CA	L01
CD	L01
CDOM_440	L01
CDOM_SLOPE	L01
CHL_A	L01
CHL_B	L01
CHL_C	L01
CHLA	F01
CHLA	L01
CHLA	L02
CHLA	L03
CLF	L01
CLW	L01
CLW	L02
CLW	L03
COD	L01
COD	L02
COD	L03
COLOR	L01
CR	L01
CU	L01
DCU	L01
DIC	L01
DIN	D01
DIN	D01A
DIN	D01B
DIN	D01D
DIN	D02
DIN	D02A
DIN	D02B
DIN	D02D
DO	F01
DO	F02
DO	F03
DO	F04
DO_SAT_M	D01

REPORTING_PARAMETER	WQ_METHOD
DO_SAT_P	F01
DOC	L01
DOC	L02
DOC	L03
DON	D01
DON	D01A
DON	D01B
DON	D01D
DON	D02
DON	D02A
DON	D02B
DON	D02D
DON	D03
DON	D03A
DON	D03B
DON	D03D
DOP	D01
DOP	D01A
DOP	D01B
DOP	D01D
DZN	L01
EPAR_S	F01
EPARD_Z	F01
EPARU_Z	F01
FCOLI_C	L02
FCOLI_M	L01
FCOLI_M	L03
FE_M	L01
FE_M	L02
FE_U	L02
FLOW_AVG	F01
FLOW_INS	F01
FLUOR	NA
FLUORESCENCE	NA
FS	L01
FSS	L01
HARDNESS	F01
HARDNESS	L01
HARDNESS	L02
HARDNESS	L03
HG	L01
IBOD5F	L01
IBOD5W	L01
K	L01
KD	D01
KD	F01
MEASURED_DEPTH	NA
MGF	L01

REPORTING_PARAMETER	WQ_METHOD
MN	L01
NAF	L01
NH4F	L01
NH4F	L02
NH4W	L01
NI	L01
NO23F	C01A
NO23F	D01
NO23F	D01A
NO23F	D01B
NO23F	D01D
NO23F	L01
NO23F	L02
NO23F	L03
NO23W	D01
NO23W	D01A
NO23W	D01B
NO23W	D01D
NO23W	L01
NO23W	L02
NO2F	L01
NO2F	L02
NO2F	L03
NO2W	L01
NO2W	L02
NO2W	L03
NO3F	C01
NO3F	D01
NO3F	D01A
NO3F	D01B
NO3F	D01D
NO3F	L01
NO3W	D01
NO3W	D01A
NO3W	D01B
NO3W	D01D
NO3W	L01
ORP	F01
PB	L01
PC	L01
PH	F01
PH	F02
PHEO	L01
PHEO	L02
PHEO	L03
PIC	L01
PIP	L01
PN	L01

REPORTING_PARAMETER	WQ_METHOD
PO4F	L01
PO4F	L02
PO4F	L03
PO4W	L01
POC	D01
POC	D01A
POC	D01B
POC	D01D
PON	D01
PON	D01A
PON	D01B
PON	D01D
PP	D01
PP	D01A
PP	D01B
PP	D01D
PP	L01
SALINITY	F01
SALINITY	F02
SALINITY	F03
SALINITY	F04
SE	L01
SECCHI	F01
SECCHI	F02
SIF	L01
SIF	L02
SIF	L03
SIGMA_T	D01
SIW	L01
SIW	L02
SIW	L03
SO4F	L01
SO4F	L02
SO4F	L03
SO4F	L04
SO4W	L01
SPCOND	F01
SPCOND	F02
SSC_%FINE	D01
SSC_%SAND	D01
SSC_FINE	L01
SSC_FINE	L02
SSC_SAND	L01
SSC_SAND	L02
SSC_TOTAL	D01
SSC_TOTAL	L01
SSC_TOTAL	L02
TALK	L01

REPORTING_PARAMETER	WQ_METHOD
TCHL_PRE_CAL	F01
TCOLI_C	L02
TCOLI_M	L01
TDN	D01
TDN	D01A
TDN	D01B
TDN	D01D
TDN	D02
TDN	D02A
TDN	D02B
TDN	D02D
TDN	L01
TDN	L02
TDP	L01
TDP	L02
TDP	L03
TDP	L04
TDP	L05
TDS	L01
TKNF	L01
TKNF	L02
TKNF	L03
TKNW	L01
TKNW	L02
TKNW	L03
TN	D01
TN	D01A
TN	D01B
TN	D01D
TN	D02
TN	D02A
TN	D02B
TN	D02D
TN	D03
TN	D03A
TN	D03B
TN	D03D
TN	D04
TN	D04A
TN	D04B
TN	D04D
TN	D05
TN	D05A
TN	D05B
TN	DO5D
TN	L01
TOC	D01
TOC	D01A

REPORTING_PARAMETER	WQ_METHOD
TOC	D01B
TOC	D01D
TOC	L01
TOC	L02
TOC	L03
TON	D01
TON	D01A
TON	D01B
TON	D01D
TON	D02
TON	D02A
TON	D02B
TON	D02D
TON	D03
TON	D03A
TON	D03B
TON	D03D
TOTAL_DEPTH	F01
TOTAL_DEPTH	F02
TOTAL_DEPTH	F03
TOTAL_DEPTH	NA
TP	D01
TP	D01A
TP	D01B
TP	D01D
TP	L01
TP	L02
TP	L03
TP	L04
TP	L05
TS	L01
TSS	L01
TURB_FTU	L01
TURB_JTU	L01
TURB_NTU	F01
TURB_NTU	F02
TURB_NTU	L01
TURB_NTU	UNK
VELOCITY	F01
VELOCITY	F02
VSS	L01
WIDTH	F01
WTEMP	F01
WTEMP	F02
ZN	L01
ZNF	L02

TAB_METHODS_BIO

Field Name	Description	Data Type	Length
BIO_METHOD (PK,FK)	BIO_METHOD_CODE- Method Description code	char	6
BIO_METHOD_ TITLE (NN)	BIO_METHOD_TITLE-Bio enumeration method title	varchar	100
BIO_METHOD_ DESCRIPTION (NN)	BIO_METHOD_DESCRIPTION-Basic description of IBI caclucation meton	varchar	max
BIO_METHOD DETAILS	BIO_METHOD_DETAILS- additional details for method	varchar	max
SUB_COUNT	SUB_COUNT-Subsample size	real	
HABITAT_TYPE	HABITAT_TYPE-Type of stream habitat targeted	varchar	15
TAXON_LEVEL	TAXON_LEVEL- Level of taxonomic identification	varchar	50

1) General: This table stores information related exclusively to BIO_METHOD codes in the TAB_TAXONOMIC_COUNT and TAB_TAXONOMIC_QUANTIFICATION tables. This table contains descriptions of the field and laboratory methods for parameter determination. The BIO_METHOD code is used to define the field or lab procedure used to obtain the parameter value. Currently accepted BIO_METHODS designations are as follows:

BIO_METHOD	BIO_METHOD_TITLE	SUB_COUNT	HABITAT_TYPE	TAXON_LEVEL
BE101A	PADEP RAPID BIO-ASSESSMENT PROTOCOL	200	RIFFLE	FAMILY
BE101B	PADEP RAPID BIO-ASSESSMENT PROTOCOL WITH REPLICATION	200	RIFFLE	FAMILY
BE102	THE NYSDEC STREAM BIOMONITORING- KICK SAMPLE PROTOCOL	100	RIFFLE	GENUS_SPECIES
BE103	MARYLAND BIOLOGICAL STREAM SURVEY BIOASSESSMENT PROTOCOL	100	MULTI	FAMILY_GENUS
BE104	MARYLAND CORE TREND BIOASSESSMENT PROTOCOL		OTHER	FAMILY
BE105	VADEQ RAPID BIO-ASSESSMENT PROTOCOL: SINGLE-HABITAT METHOD	110	RIFFLE	FAMILY
BE106	VADEQ RAPID BIO-ASSESSMENT PROTOCOL: MULTI- HABITAT METHOD	110	MULTI	FAMILY
BE107	WVDEP MODFIED EPA RBP II, SINGLE HABITAT	100	RIFFLE	GENUS
BE107B	WVDEP MODFIED EPA RBP II, SINGLE HABITAT-200 COUNT	200	RIFFLE	GENUS
BE108	MID-ATLANTIC COASTAL STREAMS METHOD	100	MULTI	GENUS
BE109	WVDEP EPA RBP II, MULTI HABITAT	100	MULTI	GENUS_SPECIES
BE110	SRBC RAPID BIOASSESSMENT PROTOCOL (RBP) III	200	RIFFLE	GENUS
BE111	MONTGONERY COUNTY-UNSPECIFIED PROTOCOL	100	MULTI	GENUS
BE112	FAIRFAX COUNTY-RBP FOR PIEDMONT/TRIASSIC PHYSIOGRAPHIC PROVINCES	200	RIFFLE	FAMILY
BE113	FAIRFAX COUNTY-RBP FOR COASTAL PLAIN PHYSIOGRAPHIC PROVINCES	200	MULTI	FAMILY
BE114	ICPRB PROTOCOL-QUANITIATIVE	200	MULTI	FAMILY
BE115	ICPRB PROTOCOL-QUALITIATIVE	200	RIFFLE	FAMILY
BE116	EPA WADEABLE STREAMS ASSESSMENT PROTOCOL	500	MULTI	GENUS
BE117	UNASSIGNED			
BE118	PRINCE GEORGES COUNTY	100	MULTI	GENUS
BE119	USFS	200	RIFFLE	FAMILY
BE120	THE NYSDEC STREAM BIOMONITORING- JAB SAMPLE PROTOCOL	100	MULTI	GENUS_SPECIES
BE121	LOUDEN COUNTY SINGLE HABITAT (RIFFLE)	100	RIFFLE	FAMILY
BE122	UNASSIGNED			

BIO_METHOD	BIO_METHOD_TITLE	SUB_COUNT	HABITAT_TYPE	TAXON_LEVEL
BE123	VCU ASSESSMENT	200	MULTI	GENUS_SPECIES
BE124O	EPA EMAP-OTHER	300	OTHER	GENUS_SPECIES
BE124P	EPA EMAP-POOL	300	POOL	GENUS_SPECIES
BE124R	EPA EMAP-RIFFLE	300	RIFFLE	GENUS_SPECIES
BE125A	USGS-RTH-NOSNAG	500	MULTI	GENUS_SPECIES
BE125B	USGS-RTH-SNAG	500	MULTI	GENUS_SPECIES

TAB_METHODS_HAB

Field Name	Description	Data Type	Length
HAB_METHOD (PK,FK)	HAB_METHOD_CODE- Method Description code	char	6
HAB_METHOD_ TITLE (NN)	HAB_METHOD_TITLE-Bio enumeration method title	varchar	100
HAB_METHOD_DESCRIPTION (NN)	HAB_METHOD_DESCRIPTION-Basic description of IBI caclucation meton	varchar	max
HAB_METHOD DETAILS	HAB_METHOD_DETAILS- additional details for method	varchar	max

1) General: This table stores information related exclusively to HAB_METHOD codes in the TAB_HABITAT_ASSESSMENT tables. This table contains descriptions of the field and laboratory methods for parameter determination. The HAB_METHOD code is used to define the field or lab procedure used to obtain the parameter value. Currently accepted HAB_METHODS designations are as follows:

HAB_METHOD	HAB_METHOD_TITLE
HAB101	EPA RAPID HABITAT ASSESSMENT PROTOCOL
HAB102	MONTGONERY COUNTY-UNSPECIFIED PROTOCOL
HAB103	FAIRFAX COUNTY-RSAT AND MODIFIED RBP FOR PIEDMONT/TRIASSIC PHYSIOGRAPHIC PROVINCES
HAB104	FAIRFAX COUNTY-RSAT AND MODIFIED RBP FOR COASTAL PLAIN PHYSIOGRAPHIC PROVINCES
HAB105	MACS WORKGROUP MODIFIED EPA RAPID HABITAT ASSESSMENT PROTOCOL
HAB106	VCU FAST HABITAT ASSESSMENT
HAB107	VCU SLOW HABITAT ASSESSMENT

TAB_PARAMETER_WQ

Field Name	Description	Data Type	Length
PARAMETER_ID	PARAMETER_ID-An autogenrated number	int	
REPORTING_ PARAMETER (PK,NN)	REPORTING_PARAMETER -Reporting Parameter Code	varchar	15
REPORTING_UNITS (PK,NN)	REPORTING_UNITS- Units of measure for parameter	varchar	10
STORET_CODE	STORET_CODE-EPA STORET Parameter code	varchar	5
USGS_CODE	USGS_CODE-United States Geological Survey Parameter code	varchar	50
REPORTING_ PARAMETER_	Reporting Parameter Description-Parameter description/definition	varchar	100

Field Name	Description	Data Type	Length
DECRIPTION (NN)			
ALT_NAME	ALT_NAME-Alterative parameter name	varchar	75

1) General: This table stores information related exclusively to REPORTING_PARAMETERS AND REPORTING_UNITS codes in codes in the TAB_WQ_DATA table. This table contains information to parameter names and standard detection limits. The following list of parameters represents those parameters that are either directly measured in the field or analyzed in the laboratory. Additional codes may be added as needed. Currently accepted PARAMETER and DESCRIPTION designations are as follows:

REPORTING_PARAMETER	REPORTING_UNITS
AG	UG/L
AS	UG/L
BIOSI	MG/L
BOD20F	MG/L
BOD20W	MG/L
BOD5F	MG/L
BOD5W	MG/L
CAF	MG/L
CD	UG/L
CHL_B	MG/L
CHL_C	MG/L
CHLA	UG/L
CLW	MG/L
COD	MG/L
SPCOND	UMHOS/CM
CR	UG/L
CU	UG/L
DIN	MG/L
DO	MG/L
DO_SAT_M	MG/L
DO_SAT_P	PCT
DOC	MG/L
DON	MG/L
DOP	MG/L
EPAR_S	UM/M**2/S
EPARD_Z	UM/M**2/S
EPARU_Z	UM/M**2/S
FCOLI_C	COL/100 ML
FCOLI_M	MPN/100 ML
FE_M	MG/L
FE_U	UG/L
FLOW_AVG	CFS
FLOW_INS	CFS
FS	MG/L
FSS	MG/L
GAGE_HEIGHT	FT
HARDNESS	MG/L
HG	UG/L

REPORTING_PARAMETER	REPORTING_UNITS
IBOD5F	MG/L
IBOD5W	MG/L
KD	1/M
KF	MG/L
MGF	MG/L
MN	UG/L
NAF	MG/L
NH4F	MG/L
NH4W	MG/L
NI	UG/L
NO23F	MG/L
NO23W	MG/L
NO2F	MG/L
NO2W	MG/L
NO3F	MG/L
NO3W	MG/L
ORP	MV
PB	UG/L
PC	MG/L
PERIPHY	G/M**2
PH	SU
PHEO	UG/L
PIC	MG/L
PIP	MG/L
PN	MG/L
PO4F	MG/L
PO4W	MG/L
PON	MG/L
PP	MG/L
SALINITY	PPT
SE	UG/L
SECCHI	M
SI	UG/L
SIF	MG/L
SIGMA_T	NONE
SIW	MG/L
SN	UG/L
SO3	MG/L

REPORTING_PARAMETER	REPORTING_UNITS
SO4F	MG/L
SO4W	MG/L
TALK	MG/L
TCOLI_C	COL/100 ML
TCOLI_M	MPN/100 ML
TDN	MG/L
TDP	MG/L
TDS	MG/L
TKNF	MG/L
TKNW	MG/L
TN	MG/L
TOC	MG/L
TON	MG/L
TOP	MG/L
TP	MG/L
TS	MG/L
TSS	MG/L
TURB_FTU	FTU
TURB_JTU	JTU
TURB_NTU	NTU
VSS	MG/L
WTEMP	DEG C
ZN	UG/L
TCHL_PRE_CAL	UG/L
FLUORESCENCE	%FS
BATT	VOLTS
BOAT_SPEED	KNOTS

REPORTING_PARAMETER	REPORTING_UNITS
CDOM_440	1/M
CDOM_SLOPE	1/NM
SSC_TOTAL	MG/L
SSC_SAND	MG/L
SSC_FINE	MG/L
SSC_%SAND	PCT
SSC_%FINE	PCT
POC	MG/L
TDZN	UG/L
VELOCITY	M/S
ANC	MEQ/L
CLF	MG/L
ACIDITY	MG/L
AL	MG/L
DZN	UG/L
DCU	UG/L
TOTAL_DEPTH	M
WIDTH	M
CHL_A	UG/ML
CA	MG/L
COLOR	SU
MEASURED_DEPTH	M
DIC	MG/L
K	MG/L
ZNF	UG/L

TAB_PROBLEM_CODE

Field Name	Description	Data Type	Length
PROBLEM_CODE (PK,FK)	PROBLEM_CODE-Code denoting laboratory problem status	VARCHAR	2
PROBLEM_CODE_ DEFINITION	PROBLEM_CODE_CODE DEFINITON- Description of lab problem code.	VARCHAR	100

1) General: This table stores information relating to the laboratory noted analytical problems with a water quality parameter. This code is used primary in the TAB_WQ_DATA table. Problem codes are consistant with those used in other CBP databases. Additional codes may be added as needed. Currently accepted PROBLEM_CODE designations are as follows:

PROBLEM_CODE	PROBLEM_CODE_DESCRIPTION
A	LABORATORY ACCIDENT
AA	FIELD ACCIDENT
B	CHEMICAL MATRIX INTERFERENCE
BB	TORN FILTER PAD
C	INSTRUMENT FAILURE
D	INSUFFICIENT SAMPLE

PROBLEM_CODE	PROBLEM_CODE_DESCRIPTION
DD	SAMPLE SIZE NOT REPORTED (ASSUMED)
E	SAMPLE RECEIVED AFTER HOLDING TIME
F	Post-calibration failure likely due to equipment damage after sampling; data appear normal
FF	MEAN REPORTED DUE TO POOR REPLICATION BETWEEN PADS
GG	SAMPLE ANALYZED AFTER HOLDING TIME

PROBLEM_CODE	PROBLEM_CODE_DESCRIPTION
I	SUSPECT VALUE HAS BEEN VERIFIED CORRECT
J	INCORRECT SAMPLE FRACTION FOR ANALYSIS
JJ	VOLUME FILTERED NOT RECORDED (ASSUMED)
L	LICOR CALIBRATION OFF BY >= 10% PER YEAR. USE WITH CALC KD WHERE PROB OF LU, LS, LB EXIST IN RAW
LB	LICOR CALIBRATION OFF BY >= 10% PER YEAR FOR BOTH AIR AND UPWARD FACING SENSORS
LS	LICOR CALIBRATION OFF BY >= 10% PER YEAR FOR AIR SENSOR
LU	LICOR CALIBRATION OFF BY >= 10% PER YEAR FOR UPWARD FACING SENSOR
MM	OVER 20% OF SAMPLE ADHERED TO POUCH AND OUTSIDE OF PAD
N	NONE
NN	PARTICULATES FOUND IN FILTERED SAMPLE
P	PROVISIONAL DATA
Q	ANALYTE PRESENT; REPORTED VALUE IS ESTIMATED; CONC IS BELOW THE RANGE FOR ACCURATE

PROBLEM_CODE	PROBLEM_CODE_DESCRIPTION
	QUANTITATION
QQ	PART EXCEEDS WHOLE VALUE YET DIFFERENCE IS WITHIN ANALYTICAL PRECISION
R	SAMPLE CONTAMINATED
RR	NO SAMPLE RECEIVED by Lab from Field office
SS	SAMPLE REJECTED DUE TO HIGH SUSPENDED SEDIMENT CONCENTRATION
U	MATRIX PROBLEM RESULTING FROM THE INTERRELATIONSHIP BETWEEN VARIABLES SUCH AS PH AND AMMONIA
UN	For DCDOH data, these values are issues or are nulls with no assigned problem codes. 8/27/2008
V	SAMPLE RESULTS REJECTED DUE TO QC CRITERIA
WW	HIGH OPTICAL DENSITY (750 NM); ACTUAL VALUE RECORDED
X	SAMPLE NOT PRESERVED PROPERLY

TAB_PROGRAM

Name	Description	Data Type	Length
AGENCY_CODE (PK,NN)	AGENCY_CODE-Data Generating Agency Code from Agency Table	char	7
PROGRAM_ CODE (PK,NN)	PROGRAM_CODE-Program Code for Monitoring Program	char	6
PROGRAM_ NAME	PROGRAM_NAME-Name of Monitoring Program	varchar	50
PRINCIPAL_ INVESTIGATOR	PRINCIPAL_INVESTIGATOR-Principle Investigator of Monitoring program	varchar	50
PI_PHONE	PI_PHONE-Contact Phone number for Principle Investigator	varchar	25
PI_EMAIL	PI_EMAIL-Contact Email for Principle Investigator	varchar	100
DATA_ MANAGER	DATA_MANAGER-Primary Data Manager for Monitoring Program	varchar	50
DATA_ MANAGER_ EMAIL	DATA_MANAGER_EMAIL-Contact Email for Primary Data Manager	varchar	100
DATA_ MANAGER_ PHONE	DATA_MANAGER_PHONE- Contact Phone Number for Primary Data Manager	varchar	25
WEB_SITE	WEB_SITE-Monitoring Program Website	varchar	150

1)GENERAL: This table stores information related exclusively to AGENCY_CODE and PROGRAM_CODE in the TAB_PROJECT table. The agency code identifies the organization responsible for processing and storing collected data. Program_Code refers to the group within an agency responsible for data collecton. These codes, taken directly from the currently in the water quality, and living resources databases. Currently accepted AGENCY_CODE and PROGRAM_CODE designations are as follows:

AGENCY_ CODE	PROGRAM_ CODE	AGENCY_PROGRAM_NAME
DNREC	DEBM	Delaware Biological Monitoring Program
FC-DPW	FCWMP	Frederick County Watershed Management Program
FC-SPS	FCSQAP	Fairfax County Stream Quality Assessment Program
HC_DPW	HCBMSA	Howard Co Bio-Monitoring and Assessment Program
LC-DBD	LCSAP	Loudoun County Stream Quality Assessment Program
MC-SPS	MCSMP	Montgomery Co Dept. of Environmental Protection
MDDNR	MBSS	Maryland Biological Stream Survey
MDDNR	MDCT	Maryland Core/Trend Monitoring Network
NYDEC	RSMP	New York Routine Statewide Monitoring Program
PADEP	PAOWQA	Pensylvania Other Water Quality Assessments
PADEP	PASWM	Pensylvania Surface Water Monitoring Program
PADEP	PAUSGS	Pensylvania USGS
PADEP	PAUW	Pensylvania Unassessed Watersheds

AGENCY_ CODE	PROGRAM_ CODE	AGENCY_PROGRAM_NAME
PGC-DER	PGCSS	Prince Georges Co Programs and Planning Division
SRBC	TMDL	SRBC-Watershed Assessment and Protection-TMDL
SRBC	WA	SRBC-Watershed Assessment Program
USEPA	EMAP	EPA-EMAP WADEABLE STREAMS ASSESSMENT
USEPA	MAHA	EPA-MID-ATLANTIC HIGHLANDS ASSESSMENT
USEPA	WSA	EPA-Wadeable Stream Assessment Program
USFS	SA	National Forest Service Stream Assessment
USGS	NAWQA	USGS-National Water Quality Assessment Program
VADEQ	SA	Virginia DEQ Benthic Monitoring Program
VCU	INSTAR	INteractive STream Assessment Resource
WVDEP	SA	West Virginia Div. of Water and Waste Management

TAB_PROJECT

Name	Description	Data Type	Length
PROJECT_ID (AK,NN)	PROJECT_ID-(AGNECY_CODE+PROGRAM_CODE+PROGRAM) AN AUTO GENERATED NUMBER	int	
AGENCY_CODE (CI,U,NNI)	AGENCY_CODE-Data Generating Agency Code from Agency Table	char	7
PROGRAM_ CODE (CI,NN,)	PROGRAM_CODE-Program Code for Monitoring Program	char	6
PROJECT_ NAME (CI,NN,)	PROJECT_NAME-Individual Monitoring Project Name	varchar(100)	100
STARTDATE	STARTDATE-Start Date of Monitoring Project	smalldatetime	
ENDDATE	ENDDATE- End Date of Monitoring Project	smalldatetime	
P_ INVESTIGATOR	P_INVESTIGATOR-Principle Investigator of Monitoring Project	varchar	50
STUDY_ OBJECTIVES	STUDY_OBJECTIVES-Monitoring Project Objectives	varchar	250
COMMENTS	COMMENTS-Additional comment	varchar	MAX

1) TABLE CONSTRAINT: Every event in TAB_EVENT for which there were sample taken of any kind must be associated with a PROJECT RECORD in this table. Project records must be loaded into the database first and all unique records are assigned a PROJECT_ID number. The PROJECT_ID must then be merged onto all event data based on the key fields before data may be loaded into any other primary data tables.

2) PROJECT_ID: The actual primary key for this table is a composite key base on the following fields: PROJECT_ID, STATION_ID, SAMPLE_DATE_TIME. An Auto-Key number is generated for each unique combination of these fields.

3)GENERAL: This table stores information related exclusively to PROJECT_ID in the TAB_EVENTS table. The Project_id identifies the indvidual monitoring Project under which data was collected for. Currently accepted PROJECT_NAME designations are as follows:

PROJECT_ID	AGENCY_CODE	PROGRAM_CODE	PROJECT_NAME
1	DNREC	DEBM	DELAWARE BIOLOGICAL MONITORING PROGRAM
2	FC-SPS	FCSQAP	FAIRFAX COUNTY STREAM PROTECTION STRATEGY
3	MC-SPS	MCSMP	MONTGOMERY CO. COUNTYWIDE STREAM PROTECTION STRATEGY
4	MDDNR	MBSS	MARYLAND BIOLOGICAL STREAM SURVEY 2000-2001
5	MDDNR	MBSS	MARYLAND BIOLOGICAL STREAM SURVEY 2000-2004
6	MDDNR	MBSS	MARYLAND BIOLOGICAL STREAM SURVEY 2005-2009
7	MDDNR	MDCT	MARYLAND CORE/TREND MONITORING NETWORK
8	PADEP	PASWM	ICE SURVEY
9	PADEP	PAUSGS	WQN - MACROINVERTEBRATES
10	PADEP	PAUW	UNASSESSED WATERSHEDS
11	PGC-DER	PGCSS	BIOLOGICAL ASSESSMENT AND MONITORING PROGRAM
12	SRBC	TMDL	AMD SAMPLING
13	SRBC	TMDL	AMD SAMPLING FOR TMDL DEVELOPMENT-2000

PROJECT_ID	AGENCY_CODE	PROGRAM_CODE	PROJECT_NAME
14	SRBC	TMDL	AMD SAMPLING FOR TMDL DEVELOPMENT-2000B
15	SRBC	TMDL	AMD SAMPLING FOR TMDL DEVELOPMENT-2001
16	SRBC	TMDL	AMD SAMPLING FOR TMDL DEVELOPMENT-2002
17	SRBC	TMDL	AMD SAMPLING FOR TMDL DEVELOPMENT-2003
18	SRBC	TMDL	BEAVERDAM BRANCH AND MILL RUN TMDL
19	SRBC	TMDL	CONESTOGA RIVER WATERSHED TMDL
20	SRBC	TMDL	FRANKSTOWN BRANCH JUNIATA RIVER ICE/TMDL
21	SRBC	TMDL	OCTORARO CREEK TMDL
22	SRBC	TMDL	WATER QUALITY SAMPLING FOR TMDL DEVELOPMENT ALIAS: AMD SAMPLING
23	SRBC	WA	CHEMUNG SUBBASIN SURVEY YEAR 1 - 1997
24	SRBC	WA	CHEMUNG SUBBASIN SURVEY YEAR 1 - 2006
25	SRBC	WA	CHEMUNG SUBBASIN SURVEY, YEAR 2 - COHOCTON RIVER WATERSHED
26	SRBC	WA	ICE SAMPLING IN THE YELLOW BREECHES
27	SRBC	WA	INTERSTATE STREAM WATER QUALITY NETWORK
28	SRBC	WA	JUNIATA SUBBASIN SURVEY YEAR 1 - 1995
29	SRBC	WA	JUNIATA SUBBASIN SURVEY YEAR 1 - 2004
30	SRBC	WA	JUNIATA SUBBASIN SURVEY, YEAR 2 - MORRISON COVE
31	SRBC	WA	LARGE RIVER ASSESSMENT 2005
32	SRBC	WA	LARGE RIVER ASSESSMENT 2007
33	SRBC	WA	LARGE RIVER ASSESSMENT PILOT PROJECT
34	SRBC	WA	LOWER SUSQUEHANNA SUBBASIN SURVEY YEAR 1 - 1996
35	SRBC	WA	LOWER SUSQUEHANNA SUBBASIN SURVEY, YEAR 1 - 2005
36	SRBC	WA	LOWER SUSQUEHANNA SUBBASIN SURVEY, YEAR 2 - YELLOW BREECHES WATERSHED
37	SRBC	WA	MIDDLE SUSQUEHANNA SUBBASIN SURVEY - YEAR 2 (WYALUSING CREEK)
38	SRBC	WA	MIDDLE SUSQUEHANNA SUBBASIN SURVEY YEAR 1 - 2001
39	SRBC	WA	NEW YORK EWS
40	SRBC	WA	SCIENCE IN MOTION 2006
41	SRBC	WA	TARGETED WATERSHED GRANT - PAXTON CREEK
42	SRBC	WA	UPPER SUSQUEHANNA SUBBASIN SURVEY YEAR 1 - 1998
43	SRBC	WA	UPPER SUSQUEHANNA SUBBASIN SURVEY YEAR 2 - 2000
44	SRBC	WA	UPPER SUSQUEHANNA SUBBASIN SURVEY YEAR 2 - WHITNEY POINT
45	SRBC	WA	UPPER SUSQUEHANNA SUBBASIN SURVEY, YEAR 1
46	SRBC	WA	WEST BRANCH SUBBASIN SURVEY, YEAR 2 - MORGAN RUN
47	SRBC	WA	WEST BRANCH SUSQUEHANNA SUBBASIN SURVEY YEAR 1 - 2002
48	SRBC	WA	WEST BRANCH SUSQUEHANNA SUBBASIN SURVEY, 1994
49	SRBC	WA	WHITNEY POINT LAKE AND WATERSHED ADAPTIVE MANAGEMENT AND MONITORING PLAN
50	USFS	SA	JEFFERSON/GEORGE WASHINGTON NATIONAL FOREST STREAM ASSESSMENT
51	VADEQ	SA	PROBMON
52	VADEQ	SA	SPECIAL STUDY &/OR MONITORING 2000-2007
53	VADEQ	SA	SPECIAL STUDY &/OR MONITORING 2003-2007
54	VADEQ	SA	TMDL
55	WVDEP	SA	UNASSESSED WATERSHEDS
56	WVDEP	SA	WATERSHED ASSESSMENT PROTOCOL (WAP)
57	PADEP	PASWM	ANTIDEGRADATON SURVEY
58	PADEP	PASWM	CAUSE EFFECT
59	PADEP	PASWM	INTENSIVE UNASSESSED FOLLOW-UP
60	PADEP	PASWM	USE ATTAINABILITY
61	PADEP	PASWM	PROBABALISTIC SURVEY
62	PADEP	PAOWQA	EFFLUENT DOMINATED STUDY

PROJECT_ID	AGENCY_CODE	PROGRAM_CODE	PROJECT_NAME
63	PADEP	PAOWQA	319 FUNDED PROJECT
65	NYDEC	RSMP	STREAM BIOMONITORING PROGRAM
67	WVDEP	SA	RANDOM SURVEYS
68	WVDEP	SA	AMBIENT MONITORING NETWORK
69	WVDEP	SA	LONG TERM MONITORING STATIONS
70	WVDEP	SA	TMDL
71	USEPA	WSA	WADEABLE STREAM ASSESSMENT PROGRAM
73	MDDNR	MBSS	MARYLAND STREAM WADERS PROGRAM
74	PADEP	PASWM	POINT OF FIRST USE
75	PADEP	PASWM	BASIN SURVEY
77	LC-DBD	LCSAP	LOUDOUN COUNTY STREAM ASSESSMENT PROGRAM
78	VCU	INSTAR	VIRGINIA COMMOMWELTH UNIVERSITY INSTAR PROGRAM
79	HC_DPW	HCBMSA	HOWARD COUNTY BIOLOGICAL MONITORING AND ASSESSMENT PROGRAM
86	USEPA	EMAP	EMAP WADEABLE STREAMS ASSESSMENT
87	USEPA	MAHA	MID-ATLANTIC HIGHLANDS ASSESSMENT
93	USGS	NAWQA	NATIONAL WATER-QUALITY ASSESMENT PROGRAM
95	FC-DPW	FCWMP	FREDERICK COUNTY STREAM ASSESSMENT PROGRAM
96	FC-DPW	FCWMP	FREDERICK COUNTY STREAM RESTORATION MONITORING PROGRAM

TAB_QUALIFIERS

Name	Description	Data Type	Length
QUALIFIERS (PK,FK)	QUALIFIER CODE Parameter value qualifier code	char	2
QUALIFIERS _ DESCRIPTION (NN)	DESCRIPTION definition of QUALIFIER	varchar	120

1) General: This table stores information related exclusively to the Qualifiers codes in the TAB_WQ_DATA table. The QUALIFIER code is used to describe the parameter value as less than or greater than the method detection limit or as a calculated value which has been calculated using a method detection limit. Currently accepted QUALIFIERS and DESCRIPTION designations are as follows:

QUALIFIER	QUALIFIER_DESCRIPTION
#	Trace (less than an unknown detectable value)
<	Less than the detection limit of the method
>	Greater than detection limit of method
A	Within Range
J	Estimated value
N	Not detected
NA	Not recorded/parameter value not acceptable

TAB_REPORTING_PREVELANCE

Field Name	Description	Data Type	Length
REPORTING_ PARAMETER_ PREVALANCE (PK,NN)	REPORTING_PARAMETER_PREVALENCE-Prevalance class	char(2)	2
REPORTING_ PARAMETER_ PREVALANCE_ NAME (NN)	REPORTING_PARAMETER_PREVALENCE_NAME-- Prevalance class description.	varchar(50)	50
REPORTING_ PARAMETER_ PREVALANCE_ RANGE	REPORTING_PARAMETER_PREVALENCE_RANGE- Range of organism count of in reporting prevalence class	varchar(50)	50
REPORTING_ PARAMETER_ PREVALANCE_ MAX	REPORTING_PARAMETER_PREVALENCE_MAX- Maximum organism count of reporting range	tinyint	
REPORTING_ PARAMETER_ PREVALANCE_MIN	REPORTING_PARAMETER_PREVALENCE_MIN-Minimun organism count of reporting range	tinyint	
REPORTING_ PARAMETER_ PREVALANCE_ MEAN	REPORTING_PARAMETER_ REVALENCE_MEAN-Mean organism count of reporting range	real	

1) GENERAL: This table stores information related exclusively to REPORTING _PARAMETER_PREVALENCE codes in the TAB_TAXONOMIC_QUANTIFICATION TABLE. In some cases abundace of benthic organism were reported in abundance categories instead of actual organism counts. Prevalence classes are as follows:

REPORTING_ PARAMETER_ PREVALENCE	REPORTING_PARAMETER _PREVALENCE_NAME	REPORTING_PARAMETER _PREVALENCE__RANGE
A	Abundant	25-100
C	Common	10-24
P	Present	3-9
R	Rare	<3
VA	Very Abundant	>100

TAB_ SAMPLE_ REPLICATE_TYPES

Name	Description	Data Type	Length
SAMPLE_ REPLICATE_ TYPE (PK,NN)	SAMPLE_REPLICATE_TYPE-Sample Replicate type code	char	3
SAMPLE_ REPLICATE_ TYPE_ DESCRIPTION	SAMPLE_REPLICATE_TYPE_DESCRIPTION- Sample Replicate type code description	varchar	25

1) GENERAL: This table stores information relating to the type of field replicate samples taken at given site. These codes are used in the TAB_WQ_DATA table. Additional codes may be added as needed. Currently accepted SAMPLE_REPLICATE_TYPE designations are as follows:

SAMPLE_ REPLICATE_TYPE	SAMPLE_REPLICATE_TYPE _DESCRIPTION
CTRL	CONTROL SAMPLE
FL	FIELD AND LAB REPLICATES
FLD	FIELD REPLICATE
FS_AVG	AVERAGE OF TWO FIELD SPLIT SUBSAMPLE VALUES
FS1	FIELD SPLIT SUBSAMPLE 1
FS1/LAV	LABORATORY AVERAGE FOR FIELD SPLIT 1
FS1/LS1	FIELD SPLIT SUBSAMPLE 1/LAB SPLIT SUBSAMPLE 1
FS1/LS2	FIELD SPLIT SUBSAMPLE 1/LAB SPLIT SUBSAMPLE 2
FS1_AVG	AVERAGE OF LAB SPLITS GENERATED FROM FIELD SPLIT SUBSAMPLE 1
FS2	FIELD SPLIT SUBSAMPLE 2
FS2/LAV	LABORATORY AVERAGE FOR FIELD SPLIT TWO
FS2/LS1	FIELD SPLIT SUBSAMPLE2 /LAB SPLIT SUBSAMPLE 1
FS2/LS2	FIELD SPLIT SUBSAMPLE 2/LAB SPLIT SUBSAMPLE 1LAB SPLIT SUBSAMPLE 2
FS2_AVG	AVERAGE OF LAB SPLITS GENERATED FROM FIELD SPLIT SUBSAMPLE 2
FS3	FIELD SPLIT SUBSAMPLE 3
FS4	FIELD SPLIT SUBSAMPLE 4
LAB	LAB REPLICATE

SAMPLE_ REPLICATE_TYPE	SAMPLE_REPLICATE_TYPE _DESCRIPTION
LS1	LAB SPLIT SUBSAMPLE 1
LS2	LAB SPLIT SUBSAMPLE 2
LS3	LAB SPLIT SUBSAMPLE 3
M1	FIELD MEASUREMENT 1
M2	FIELD MEASUREMENT 2
M3	FIELD MEASUREMENT 3
M4	FIELD MEASUREMENT 4
METH	METHOD COMPARISON
S1	SAMPLE 1
S1/LS1	SAMPLE 1/LAB SPLIT SUBSAMPLE 1
S1/LS2	SAMPLE 1/LAB SPLIT SUBSAMPLE 2
S1/LS3	SAMPLE 1/LAB SPLIT SUBSAMPLE 3
S2	SAMPLE 2
S2/LS1	SAMPLE 2/LAB SPLIT SUBSAMPLE 1
S2/LS2	SAMPLE 2/LAB SPLIT SUBSAMPLE 2
S2/LS3	SAMPLE 2/LAB SPLIT SUBSAMPLE 3
S3	SAMPLE 3
SPK	SPIKE SAMPLE
SPK1	SPIKE SAMPLE SUBSAMPLE 1
SPK2	SPIKE SAMPLE SUBSAMPLE 2
SPLT	FIELD SPLIT

TAB_SAMPLE_TYPES

Name	Description	Data Type	Length
SAMPLE_TYPE (PK)	SAMPLE COLLECTION TYPE CODE	varchar	5
DESCRIPTION (NN)	CODE DEFINITION	varchar	130

1) GENERAL: This table stores information relating to the type of field samples taken at given site. These codes are used in the TAB_TAXONOMIC_COUNT, TAB_TAXONOMIC_QUANTIFICATION and TAB_WQ_DATA tables. Additional codes may be added as needed. Currently accepted SAMPLE_TYPE designations are as follows:

SAMPLE_TYPE	SAMPLE_TYPE_DESCRIPTION
C	COMPOSITE SAMPLE, MADE UP OF SUBSAMPLES FROM MULTIPLE DEPTHS
CS	COMPOSITE SAMPLE COLLECTED SPATIALLY
CT	COMPOSITE SAMPLE COLLECTED TEMPORALLY
D	DISCRETE SAMPLE, SAMPLE TAKEN FROM SINGLE DEPTH
HVIC	HORIZONTAL AND VERTICALLY INTEGRATED COMPOSITE SAMPLE
ISM	IN-SITU MEASUREMENT AT DEPTH, NO SAMPLE COLLECTED
ISM_H	IN-SITU MEASUREMENT, COLLECTED AS PART OF A HORIZONTAL TRANSECT
ISM_V	IN-SITU MEASUREMENT, COLLECTED AS PART OF A VERTICAL PROFILE

TAB_SITE_TYPE

Field Name	Description	Data Type	Length
SITE_TYPE (PK,FK)	SITE_TYPE_CODE-Sampling site type	char	2
SITE_TYPE_NAME	SITE_TYPE_NAME-Name of Site type	varchar	25
SITE_TYPE_ DESCRIPTION	SITE_TYPE_DESCRIPTION-Detailed description of site type.	varchar	255

1) General: This table stores information relating to the method of sampling site selection.
This code is used primary in the TAB_EVENT table. Additional codes may be added as needed. Currently accepted SITE_TYPE_CODE designations are as follows:

SITE_TYPE_CODE	SITE_TYPE_NAME
QC	QUALITY CONTROL
R	RANDOM
RR	RANDOM-REFERENCE
TB	TARGET-BASELINE
TD	TARGET-DOWNSTREAM
TR	TARGET-REFERENCE
TS	TARGET-SYSTEMATIC
U	UNSPECIFIED

TAB_UNITS

Field Name	Description	Data	Length
ID	ID-AUTOGENERATED ID	INT	
REPORTING _ UNITS	REPORTING _UNITS-Reporting units for biological Parmameter	varchar	10
REPORTING_UNITS_ DESCRIPTION	REPORTING_UNITS_DESCRIPTION- Detailed description of reporting units for biological parmameter	varchar	50

1) General: This table stores information relating to the type of measurement a parameter is. This code is used primary in the TAB_TAXONOMIC_COUNT table. Additional codes may be added as needed. Currently accepted REPORTING_UNIT designations are as follows:

REPORTING_UNITS	REPORTING_UNITS_DESCRIPTION
1/M	UNITS PER METER
1/NM	UNITS PER NANOMETER
ABS	OPTICAL DENSITY
CFS	CUBIC FEET PER SECOND
COL/100 ML	NUMBER OF COLONIES PER 100 MILLILITERS
DEG C	DEGREES CELSIUS
FT	FEET
FTU	FORMAZIN UNITS
G/M**2	GRAMS PER SQUARE METER
JTU	JACKSON TURBIDITY UNITS
KG/DAY	KILOGRAMS PER DAY
KG/MONTH	KILOGRAMS PER MONTH
KG/YEAR	KILOGRAMS PER YEAR
KNOTS	SPEED
LBS/DAY	POUNDS PER DAY
LBS/MONTH	POUNDS PER DAY
LBS/YEAR	POUNDS PER YEAR
M	METERS
M/S	METERS PER SECOND
MEQ/L	MICRO EQUIVELENTS PER LITER
MG/L	MILLIGRAMS PER LITER

REPORTING_UNITS	REPORTING_UNITS_DESCRIPTION
MPN/100 ML	MOST PROBABLE NUMBER PER 100 MILLILITERS
MV	MILLIVOLTS
NONE	PARAMETER HAS NO ASSOCIATED UNITS
NTU	NEPHELOMETRIC UNITS
PCT	PERCENT
PCT_FS	PERCENT FULL SCALE
PER SAMPLE	NUMBER OF INDIVIDUALS PER SAMPLE
PPT	PARTS PER THOUSAND
SU	SPECIFIC UNITS
TONS/DAY	TONS PER DAY
TONS/MONTH	TONS PER MONTH
TONS/YEAR	TONS PER YEAR
UE/M**2/S	MICROEINSTEINS PER METER SQUARED PER SECOND
UG/L	MICROGRAMS PER LITER
UM/M**2/S	MICROMOLES PER METER SQUARED PER SECOND
UMHOS/CM	MICROHMS PER CENTIMETER
VOLTS	VOLTAGE

STORED INDICATOR SCRIPTS AND TEMPORARY DATA TABLES

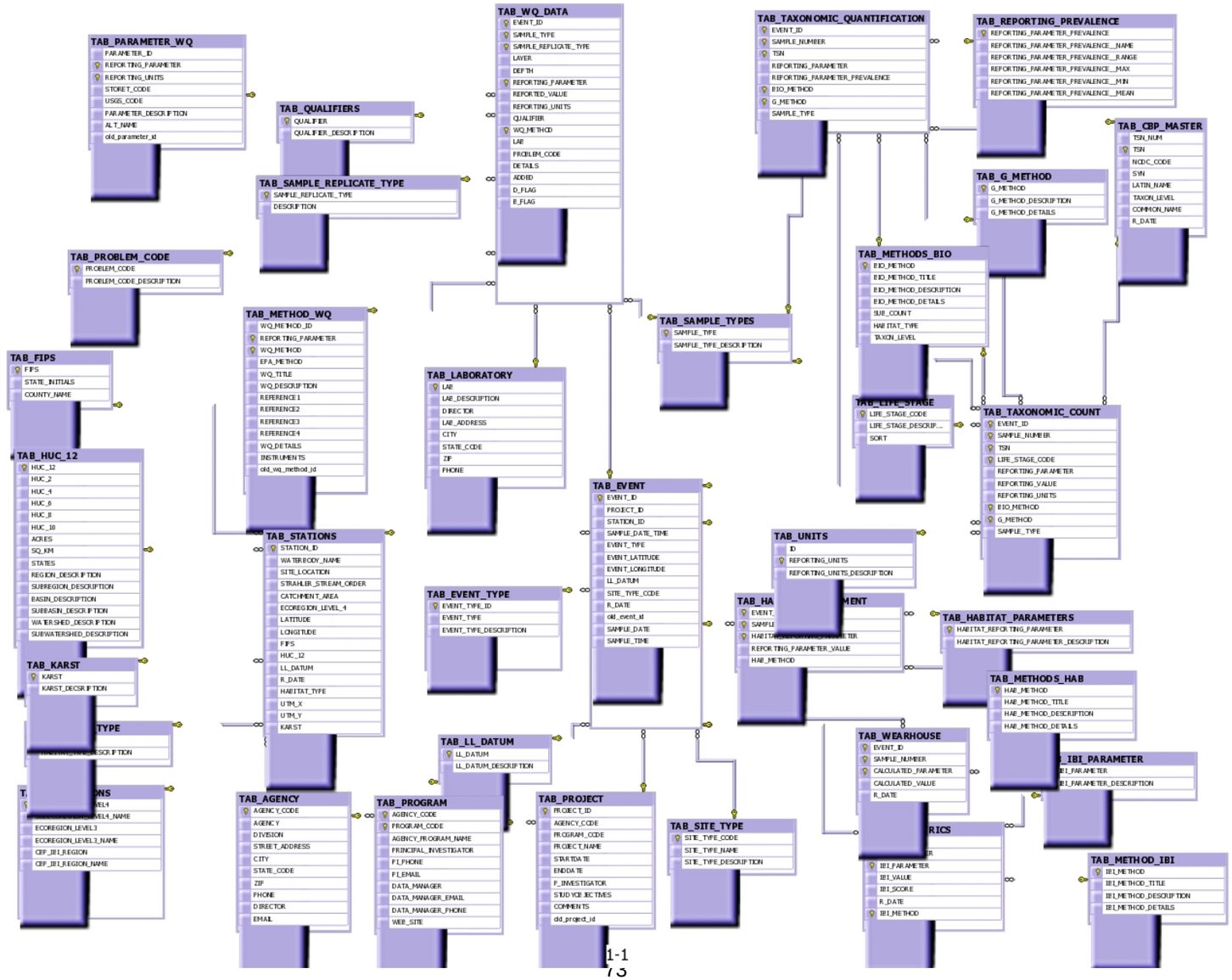
Beginning in January 2012, a series of 53 commonly calculated benthic metrics have been routinely calculated as part of data processing and stored in the TAB_WAREHOUSE table. These metrics are described in detail in the document-Development of a Basin-wide Benthic Index of Biotic Integrity for Non-Tidal Streams and Wadeable Rivers in the Chesapeake Bay Watershed: Final Report to the Chesapeake Bay Program Non-Tidal Water Quality Workgroup. ICPRB Report 11-1. The common metrics are generated by the execution of two stored SQL procedure within the Non-Tidal benthic data base. The procedure processes data for one calendar year at time, it then removes existing data and appends all newly calculated values to the TAB_WAREHOUSE table. The stored procedure usp_MainModuleIndicator draws data from the TAB_TAXONOMIC_COUNT table and calculates metrics on un-rarefacted data. The stored procedure usp_ RarifiedMetrics draws data from the TAB_RAREFACTION_COUNTS table and calculates metrics on rarefacted data. The following two sample lines of SQL script will draw all the data for 2010 and calculate all relevant metrics:

```
execute usp_MainModuleIndicator @year=2010 WAITFOR DELAY '00:00:120'
execute usp_ RarifiedMetrics @year=2010
```

The following temporary tables are created as part of the running of these scripts
Tab_Temp_Event
Tab_Temp_Taxa_Data
Tab_Temp_Family
Tab_Temp_Indicator
Tab_Temp_Ept_Part_1
Temp_PCT_DOM_MULTI
Tab_Temp_Family_2
Temp_DOM_3
Temp_DOM_2
Temp_SW
Tab_Temp_Indicator_Raw
Tab_Temp_Ept_Part_1

For detailed description of the stored procedures see Appendix B & C.

APPENDIX A. DATABASE ENTITY RELATIONSHIP DIAGRAM


TAB_PARAMETER_WQ
PARAMETER_ID
REPORTING_PARAMETER
REPORTING_UNITS
STORET_CODE
USGS_CODE
PARAMETER_DESCRIPTION
ALT_NAME
old_parameter_id
TAB_QUALIFIERS
QUALIFIER
QUALIFIER_DESCRIPTION
TAB_SAMPLE_REPLICATE_TYPE
SAMPLE_REPLICATE_TYPE
DESCRIPTION
TAB_WQ_DATA
EVENT_ID
SAMPLE_TYPE
SAMPLE_REPLICATE_TYPE
LAYER
DEPTH
REPORTING_PARAMETER
REPORTED_VALUE
REPORTING_UNITS
QUALIFIER
WQ_METHOD
LAB
PROBLEM_CODE
DETAILS
ADDED
D_FLAG
B_FLAG
TAB_TAXONOMIC_QUANTIFICATION
EVENT_ID
SAMPLE_NUMBER
TSN
REPORTING_PARAMETER
REPORTING_PARAMETER_PREVALENCE
BIO_METHOD
G_METHOD
SAMPLE_TYPE
TAB_REPORTING_PREVALENCE
REPORTING_PARAMETER_PREVALENCE
REPORTING_PARAMETER_PREVALENCE_NAME
REPORTING_PARAMETER_PREVALENCE_RANGE
REPORTING_PARAMETER_PREVALENCE_MAX
REPORTING_PARAMETER_PREVALENCE_MIN
REPORTING_PARAMETER_PREVALENCE_MEAN
TAB_CBP_MASTER
TSN_NUM
TSN
NODC_CODE
SYN
LATIN_NAME
TAXON_LEVEL
COMMON_NAME
R_DATE
TAB_G_METHOD
G_METHOD
G_METHOD_DESCRIPTION
G_METHOD_DETAILS
TAB_PROBLEM_CODE
PROBLEM_CODE
PROBLEM_CODE_DESCRIPTION
TAB_METHOD_WQ
WQ_METHOD_ID
REPORTING_PARAMETER
WQ_METHOD
EPA_METHOD
WQ_TITLE
WQ_DESCRIPTION
REFERENCE1
REFERENCE2
REFERENCE3
REFERENCE4
WQ_DETAILS
INSTRUMENTS
old_wq_method_id
TAB_SAMPLE_TYPES
SAMPLE_TYPE
SAMPLE_TYPE_DESCRIPTION
TAB_METHODS_BIO
BIO_METHOD
BIO_METHOD_TITLE
BIO_METHOD_DESCRIPTION
BIO_METHOD_DETAILS
SUE_COUNT
HABITAT_TYPE
TAXON_LEVEL
TAB_FIPS
FIPS
STATE_INITIALS
COUNTY_NAME
TAB_LABORATORY
LAB
LAB_DESCRIPTION
DIRECTOR
LAB_ADDRESS
CITY
STATE_CODE
ZIP
PHONE
TAB_LIFE_STAGE
LIFE_STAGE_CODE
LIFE_STAGE_DESCRIP...
SORT
TAB_TAXONOMIC_COUNT
EVENT_ID
SAMPLE_NUMBER
TSN
LIFE_STAGE_CODE
REPORTING_PARAMETER
REPORTING_VALUE
REPORTING_UNITS
BIO_METHOD
G_METHOD
SAMPLE_TYPE
TAB_HUC_12
HUC_12
HUC_2
HUC_4
HUC_6
HUC_8
HUC_10
ACRES
SQ_KM
STATES
REGION_DESCRIPTION
SUBREGION_DESCRIPTION
BASIN_DESCRIPTION
SUBBASIN_DESCRIPTION
WATERSHED_DESCRIPTION
SUBWATERSHED_DESCRIPTION
TAB_STATIONS
STATION_ID
WATERBODY_NAME
SITE_LOCATION
STRAHLER_STREAM_ORDER
CATCHMENT_AREA
ECOREGION_LEVEL_4
LATITUDE
LONGITUDE
FIPS
HUC_12
LL_DATUM
R_DATE
HABITAT_TYPE
UTM_X
UTM_Y
KARST
TAB_EVENT
EVENT_ID
PROJECT_ID
STATION_ID
SAMPLE_DATE_TIME
EVENT_TYPE
EVENT_LATITUDE
EVENT_LONGITUDE
LL_DATUM
SITE_TYPE_CODE
R_DATE
old_event_id
SAMPLE_DATE
SAMPLE_TIME
TAB_UNITS
ID
REPORTING_UNITS
REPORTING_UNITS_DESCRIPTION
TAB_EVENT_TYPE
EVENT_TYPE_ID
EVENT_TYPE
EVENT_TYPE_DESCRIPTION
TAB_KARST
KARST
KARST_DECSRIPTION
TAB_HABITAT_PARAMETERS
HABITAT_REPORTING_PARAMETER
HABITAT_REPORTING_PARAMETER_DESCRIPTION
TAB_METHODS_HAB
HAB_METHOD
HAB_METHOD_TITLE
HAB_METHOD_DESCRIPTION
HAB_METHOD_DETAILS
TAB_LL_DATUM
LL_DATUM
LL_DATUM_DESCRIPTION
TAB_WEARHOUSE
EVENT_ID
SAMPLE_NUMBER
CALCULATED_PARAMETER
CALCULATED_VALUE
R_DATE
TAB_AGENCY
AGENCY_CODE
AGENCY
DIVISION
STREET_ADDRESS
CITY
STATE_CODE
ZIP
PHONE
DIRECTOR
EMAIL
TAB_PROGRAM
AGENCY_CODE
PROGRAM_CODE
AGENCY_PROGRAM_NAME
PRINCIPAL_INVESTIGATOR
PI_PHONE
PI_EMAIL
DATA_MANAGER
DATA_MANAGER_EMAIL
DATA_MANAGER_PHONE
WEB_SITE
TAB_PROJECT
PROJECT_ID
AGENCY_CODE
PROGRAM_CODE
PROJECT_NAME
STARTDATE
ENDDATE
P_INVESTIGATOR
STUDYOBJECTIVES
COMMENTS
old_project_id
TAB_SITE_TYPE
SITE_TYPE_CODE
SITE_TYPE_NAME
SITE_TYPE_DESCRIPTION
TAB_METHOD_IBI
IBI_METHOD
IBI_METHOD_TITLE
IBI_METHOD_DESCRIPTION
IBI_METHOD_DETAILS

APPENDIX B. THE MAINMODULEINDICATOR STORED PROCEDURE

```
USE [Non_Tidal_Benthic]
GO

/****** Object:  StoredProcedure [dbo].[usp_MainModuleIndicator]    Script Date: 04/22/2013 11:58:48 ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

-- =============================================
-- Author:                  Andre Stratton - Vistronix
-- Create date:             May 5, 2011
-- Description:             Calculating all Non-Tidal Metrics
-- Modified Date:     Januaray 23, 2011
-- =============================================
CREATE PROCEDURE [dbo].[usp_MainModuleIndicator]
        -- Add the parameters for the stored procedure here
        @year VARCHAR(8) = NULL
AS

        IF @year IS NULL
        BEGIN
                PRINT 'ERROR: You must specifiy a year in this format: usp_MainIndicator @year = %1994% with %1994%
the encapsulated in single quotes.'
                RETURN
        END

BEGIN TRY
        BEGIN TRANSACTION
-- SET NOCOUNT ON added to prevent extra result sets from
-- interfering with SELECT statements.
        SET NOCOUNT ON;

        DROP TABLE dbo.Tab_Temp_Ept_Part_1
        DROP TABLE dbo.Tab_Temp_Event
        DROP TABLE dbo.Tab_Temp_Family
        DROP TABLE dbo.Tab_Temp_Family_2
        DROP TABLE dbo.Tab_Temp_Indicator
        DROP TABLE dbo.Tab_Temp_Indicator_Raw
        DROP TABLE dbo.Tab_Temp_Taxa_Data
        DROP TABLE dbo.Temp_DOM_2
        DROP TABLE dbo.Temp_DOM_3
        DROP TABLE dbo.Temp_PCT_DOM_MULTI
        DROP TABLE dbo.Temp_SW

        SET ANSI_Warnings ON

        PRINT 'ALL Temp Tables Were Dropped'
        COMMIT TRANSACTION
END TRY
BEGIN CATCH
        PRINT 'Transaction Failed ON DROP TABLE Statement - Will Rollback'
        IF @@TRANCOUNT > 0
        ROLLBACK

        EXECUTE usp_GetErrorInfo
END CATCH

--Step 1: Create Qry_Draw_Event_Data
BEGIN TRY
        BEGIN TRANSACTION
        PRINT 'Step 1: Create Qry_Draw_Event_Data - Begin of Calculating Metrics'
                SELECT dbo.TAB_EVENT.EVENT_ID, dbo.TAB_EVENT.PROJECT_ID,
TAB_ECO_REGIONS.CBP_IBI_REGION, dbo.TAB_EVENT.STATION_ID,DATENAME(yyyy, dbo.TAB_EVENT.SAMPLE_DATE)
AS Yr,
                dbo.TAB_EVENT.SAMPLE_DATE, dbo.TAB_EVENT.SAMPLE_TIME, dbo.TAB_EVENT.R_DATE,
dbo.TAB_EVENT.EVENT_TYPE, TAB_ECO_REGIONS.CBP_IBI_REGION_NAME,
                dbo.TAB_EVENT.EVENT_LATITUDE, dbo.TAB_EVENT.EVENT_LONGITUDE, dbo.TAB_STATIONS.UTM_X,
dbo.TAB_STATIONS.UTM_Y, dbo.TAB_EVENT.SITE_TYPE_CODE
                INTO dbo.Tab_Temp_Event
                FROM dbo.TAB_ECO_REGIONS INNER JOIN dbo.TAB_STATIONS
                        ON dbo.TAB_ECO_REGIONS.SUBECOREGION_LEVEL4 =
dbo.TAB_STATIONS.ECOREGION_LEVEL_4
                                INNER JOIN (dbo.TAB_EVENT INNER JOIN dbo.TAB_EVENT_TYPE
                                                ON dbo.TAB_EVENT.EVENT_TYPE =
dbo.TAB_EVENT_TYPE.EVENT_TYPE_ID)
                        ON dbo.TAB_STATIONS.STATION_ID = dbo.TAB_EVENT.STATION_ID
                WHERE dbo.TAB_EVENT.SAMPLE_DATE LIKE '%' +  @year + '%'
                ORDER BY dbo.TAB_EVENT.EVENT_ID
                PRINT 'Step 1: Create Qry_Draw_Event_Data - Completed Successfully'
                COMMIT TRANSACTION
END TRY
BEGIN CATCH
        PRINT 'Transaction Failed for Step 1: Create Qry_Draw_Event_Data - Will Rollback'
        IF @@TRANCOUNT > 0
        ROLLBACK

        EXECUTE usp_GetErrorInfo
END CATCH
```

```
--Step 2: Create Qry_Draw_Taxa_Data
BEGIN TRY
	BEGIN TRANSACTION
	PRINT 'Step 2: Create Qry_Draw_Taxa_Data'
		SELECT dbo.Tab_Temp_Event.EVENT_ID, dbo.Tab_Temp_Event.CBP_IBI_REGION_NAME,
dbo.Tab_Temp_Event.STATION_ID, dbo.Tab_Temp_Event.SAMPLE_DATE,
		dbo.TAB_TAXONOMIC_COUNT.SAMPLE_NUMBER,dbo.TAB_TAXONOMIC_COUNT.TSN,
dbo.TAB_TAXONOMIC_COUNT.LIFE_STAGE_CODE, dbo.TAB_TAXONOMIC_COUNT.REPORTING_PARAMETER,
		dbo.TAB_TAXONOMIC_COUNT.REPORTING_VALUE,
dbo.TAB_TAXONOMIC_COUNT.REPORTING_UNITS, dbo.TAB_TAXONOMIC_COUNT.BIO_METHOD,
dbo.TAB_TAXONOMIC_COUNT.G_METHOD,
		dbo.TAB_TAXONOMIC_COUNT.SAMPLE_TYPE
		INTO dbo.Tab_Temp_Taxa_Data
		FROM dbo.Tab_Temp_Event
			INNER JOIN dbo.TAB_TAXONOMIC_COUNT
			ON dbo.Tab_Temp_Event.EVENT_ID = dbo.TAB_TAXONOMIC_COUNT.EVENT_ID;
	PRINT 'Step 2: Create Qry_Draw_Taxa_Data - Completed Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 2: Create Qry_Draw_Taxa_Data - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

--Step 3: Create Qry_Delete_Bad_Method_1
BEGIN TRY
	BEGIN TRANSACTION
	PRINT 'Step 3: Create Qry_Delete_Bad_Method_1'
		UPDATE dbo.Tab_Temp_Taxa_Data
		SET dbo.Tab_Temp_Taxa_Data.CBP_IBI_REGION_NAME = 'delete'
		FROM dbo.Tab_Temp_Taxa_Data INNER JOIN TAB_EXCLUSION_METHODS
			ON dbo.Tab_Temp_Taxa_Data.Bio_Method = TAB_EXCLUSION_METHODS.Bio_Method
	PRINT 'Step 3: Create Qry_Delete_Bad_Method_1 - Completed Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 3: Create Qry_Delete_Bad_Method_1 - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

--Step 4: Create Qry_Delete_Bad_Method_2
BEGIN TRY
	BEGIN TRANSACTION
	PRINT 'Step 4: Create Qry_Delete_Bad_Method_2'
		DELETE
		FROM dbo.Tab_Temp_Taxa_Data
		WHERE dbo.Tab_Temp_Taxa_Data.CBP_IBI_REGION_NAME = 'delete'
		PRINT 'Step 4: Create Qry_Delete_Bad_Method_2 - Completed Successfully'
		COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 4: Create Qry_Delete_Bad_Method_2 - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

--Step 5: Create Qry_Sum_To_Family
BEGIN TRY
	BEGIN TRANSACTION
	PRINT 'Step 5: Create Qry_Sum_To_Family'
		SELECT
dbo.Tab_Temp_Taxa_Data.EVENT_ID,dbo.Tab_Temp_Taxa_Data.SAMPLE_NUMBER,dbo.TAB_CONVERT_TAXA.FAMILY_TSN
, dbo.TAB_CONVERT_TAXA.FAMILY,
		SUM(dbo.Tab_Temp_Taxa_Data.REPORTING_VALUE) AS SUM_OF_REPORTING_VALUE,
dbo.TAB_FAMILY.EXCLUSION_TAXA
		INTO dbo.Tab_Temp_Family
		FROM dbo.TAB_CONVERT_TAXA INNER JOIN dbo.Tab_Temp_Taxa_Data
			ON dbo.TAB_CONVERT_TAXA.TSN = dbo.Tab_Temp_Taxa_Data.TSN INNER JOIN
dbo.TAB_FAMILY
			ON dbo.TAB_CONVERT_TAXA.FAMILY_TSN = dbo.TAB_FAMILY.TSN
		GROUP BY dbo.Tab_Temp_Taxa_Data.EVENT_ID, dbo.Tab_Temp_Taxa_Data.SAMPLE_NUMBER,
dbo.TAB_CONVERT_TAXA.FAMILY_TSN, dbo.TAB_CONVERT_TAXA.FAMILY,
		dbo.TAB_FAMILY.EXCLUSION_TAXA;
	PRINT 'Step 5: Create Qry_Sum_To_Family - Completed Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 5: Create Qry_Sum_To_Family  - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK
```

```
	EXECUTE usp_GetErrorInfo
END CATCH

--Step 6: Create Qry_Delete_Exclusion_Taxa
BEGIN TRY
	BEGIN TRANSACTION
	PRINT 'Step 6: Create Qry_Delete_Exclusion_Taxa'
		DELETE
		FROM dbo.Tab_Temp_Family
		WHERE dbo.Tab_Temp_Family.EXCLUSION_TAXA IS NOT NULL
	PRINT 'Step 6: Create Qry_Delete_Exclusion_Taxa - Completed Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 6: Create Qry_Delete_Exclusion_Taxa - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

--Step 7: Create Qry_Make_Indicator_Table and Qry_Compute_Taxa_Rich
BEGIN TRY
	BEGIN TRANSACTION
	PRINT 'Step 7: Create Qry_Make_Indicator_Table and Qry_Compute_Taxa_Rich'
		CREATE TABLE dbo.Tab_Temp_Indicator (EVENT_ID INT NULL, SAMPLE_NUMBER INT NULL,
PARAMETER Varchar(150) NULL, REPORTING_VALUE REAL NULL)

		INSERT INTO dbo.Tab_Temp_Indicator(EVENT_ID, SAMPLE_NUMBER, PARAMETER,
REPORTING_VALUE)
		SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER, 'TAXA_RICH' AS
PARAMETER, COUNT(dbo.Tab_Temp_Family.FAMILY_TSN) AS REPORTING_VALUE
		FROM dbo.Tab_Temp_Family
		GROUP BY dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER
	PRINT 'Step 7: Create Qry_Make_Indicator_Table and Qry_Compute_Taxa_Rich - Completed Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 7: Create Qry_Make_Indicator_Table and Qry_Compute_Taxa_Rich - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

--Step 8: Create Qry_Total_Abundance
BEGIN TRY
	BEGIN TRANSACTION
	PRINT 'Step 8: Create Qry_Total_Abundance - End of Calculating Metrics'
		INSERT INTO dbo.Tab_Temp_Indicator(EVENT_ID, SAMPLE_NUMBER, PARAMETER,
REPORTING_VALUE)
		SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER,
'TOTAL_ABUNDANCE' AS PARAMETER, SUM(dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE)
		AS SUM_OF_REPORTING_VALUE
		FROM dbo.Tab_Temp_Family
		GROUP BY dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER
	PRINT 'Step 8: Create Qry_Total_Abundance - End of Calculating Metrics - Completed Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 8: Create Qry_Total_Abundance - End of Calculating Metrics - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

--Step 9: Create Qry_EPT_TAXA_ONLY_CNT
BEGIN TRY
	BEGIN TRANSACTION
	PRINT 'Step 9: Create Qry_EPT_TAXA_ONLY_CNT - Begin of Calculating Dominance Metrics'
		INSERT INTO dbo.Tab_Temp_Indicator(EVENT_ID, SAMPLE_NUMBER, PARAMETER,
REPORTING_VALUE)
		SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER,
'EPT_TAXA_COUNT' AS PARAMETER, COUNT(dbo.Tab_Temp_Family.FAMILY_TSN) AS COUNT_OF_FAMILY_TSN
		FROM dbo.TAB_EPTD_LOOK_UP INNER JOIN dbo.TAB_FAMILY
		ON dbo.TAB_EPTD_LOOK_UP.EPTD = dbo.TAB_FAMILY.EPTD
			INNER JOIN dbo.Tab_Temp_Family
			ON dbo.TAB_FAMILY.TSN = dbo.Tab_Temp_Family.FAMILY_TSN
		WHERE dbo.TAB_EPTD_LOOK_UP.EPTD <> 'D'
		GROUP BY dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER
	PRINT 'Step 9: Create Qry_EPT_TAXA_ONLY_CNT - Completed Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 9: Qry_EPT_TAXA_ONLY_CNT - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
```

```
END CATCH

--Step 10: Create Qry_EPTD_TAXA_CNT
BEGIN TRY
	BEGIN TRANSACTION
	PRINT 'Step 10: Create Qry_EPTD_TAXA_CNT Part 1'
		SELECT dbo.TAB_FAMILY.TSN, dbo.TAB_FAMILY.Family, dbo.TAB_FAMILY.TOLARANCE_VALUE,
dbo.TAB_FAMILY.GUILD, dbo.TAB_FAMILY.EPTD,
		dbo.TAB_EPTD_LOOK_UP.EPTD_PARAMETER + '_TAXA_CNT' AS PARAMETER
		INTO dbo.Tab_Temp_Ept_Part_1
		FROM dbo.TAB_FAMILY INNER JOIN dbo.TAB_EPTD_LOOK_UP
		ON dbo.TAB_FAMILY.EPTD = dbo.TAB_EPTD_LOOK_UP.EPTD
		WHERE dbo.TAB_FAMILY.EPTD IS NOT NULL
	PRINT 'Step 10: Create Qry_EPTD_TAXA_CNT Part 1 - Completed Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 10: Create Qry_EPTD_TAXA_CNT Part 1 - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

BEGIN TRY
	BEGIN TRANSACTION
	PRINT 'Step 10: Create Qry_EPTD_TAXA_CNT Part 2'
		INSERT INTO dbo.Tab_Temp_Indicator(EVENT_ID, SAMPLE_NUMBER, PARAMETER,
REPORTING_VALUE)
		SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER,
dbo.Tab_Temp_Ept_Part_1.PARAMETER, COUNT(dbo.Tab_Temp_Family.family_tsn) AS COUNT_OF_FAMILY_TSN
		FROM dbo.Tab_Temp_Family LEFT JOIN dbo.Tab_Temp_Ept_Part_1
		ON dbo.Tab_Temp_Family.family_tsn = dbo.Tab_Temp_Ept_Part_1.TSN
		GROUP BY dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER, PARAMETER
	PRINT 'Step 10: Create Qry_EPTD_TAXA_CNT Part 2 - Completed Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 10: Create Qry_EPTD_TAXA_CNT Part 2 - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

--Step 11: Create Qry_TAXA_ONLY_ABUND
BEGIN TRY
	BEGIN TRANSACTION
	PRINT 'Step 11: Create Qry_TAXA_ONLY_ABUND'
		INSERT INTO dbo.Tab_Temp_Indicator ( EVENT_ID, SAMPLE_NUMBER, PARAMETER,
REPORTING_VALUE)
		SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER,
'EPT_TAXA_ABUND' AS PARAMETER,
		SUM(dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE) AS SUM_OF_REPORTING_VALUE
		FROM dbo.TAB_EPTD_LOOK_UP INNER JOIN dbo.TAB_FAMILY
		ON dbo.TAB_EPTD_LOOK_UP.EPTD = dbo.TAB_FAMILY.EPTD
			INNER JOIN dbo.Tab_Temp_Family
			ON dbo.TAB_FAMILY.TSN = dbo.Tab_Temp_Family.FAMIIY_TSN
		WHERE dbo.TAB_EPTD_LOOK_UP.EPTD <> 'D'
		GROUP BY dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER
	PRINT 'Step 11: Create Qry_TAXA_ONLY_ABUND - Completed Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 11: Create Qry_TAXA_ONLY_ABUND - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

--Step 12: Create Qry_CALC_PCT_EPT
BEGIN TRY
	BEGIN TRANSACTION
	PRINT 'Step 12: Create Qry_CALC_PCT_EPT'
		INSERT INTO dbo.Tab_Temp_Indicator(EVENT_ID, SAMPLE_NUMBER, PARAMETER,
REPORTING_VALUE )
		SELECT EA.EVENT_ID, EA.SAMPLE_NUMBER, 'PCT_EPT' AS PARAMETER,
ROUND(((EA.REPORTING_VALUE/TA.REPORTING_VALUE)*100), 2) AS [VALUE]
		FROM (SELECT EVENT_ID, SAMPLE_NUMBER, PARAMETER, REPORTING_VALUE
				FROM dbo.Tab_Temp_Indicator
				WHERE PARAMETER = 'EPT_TAXA_ABUND') AS EA
		INNER JOIN
			(SELECT EVENT_ID, SAMPLE_NUMBER, PARAMETER, REPORTING_VALUE
				FROM dbo.Tab_Temp_Indicator
				WHERE PARAMETER = 'TOTAL_ABUNDANCE') AS TA
		ON (EA.EVENT_ID = TA.EVENT_ID) AND (EA.SAMPLE_NUMBER = TA.SAMPLE_NUMBER)
	PRINT 'Step 12: Create Qry_CALC_PCT_EPT - Completed Successfully'
	COMMIT TRANSACTION
END TRY
```

```
BEGIN CATCH
	PRINT 'Transaction Failed for Step 12: Create Qry_CALC_PCT_EPT - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

--Step 13: Create Qry_EPTD_TAXA_PCT_2
BEGIN TRY
	BEGIN TRANSACTION
	PRINT 'Step 13: Create Qry_EPTD_TAXA_PCT_2'
		INSERT INTO dbo.Tab_Temp_Indicator(EVENT_ID, SAMPLE_NUMBER, PARAMETER,
REPORTING_VALUE)
		SELECT TA.EVENT_ID, ET.SAMPLE_NUMBER, 'PCT_' +  ET.EPTD_PARAMETER AS P,
ROUND(((ET.REPORTING_VALUE/TA.REPORTING_VALUE)*100),2) AS RESULT
		FROM (SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER,
dbo.TAB_EPTD_LOOK_UP.EPTD_PARAMETER,
		SUM(dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE) AS REPORTING_VALUE
				FROM dbo.TAB_EPTD_LOOK_UP INNER JOIN dbo.TAB_FAMILY
				ON dbo.TAB_EPTD_LOOK_UP.EPTD = dbo.TAB_FAMILY.EPTD
					INNER JOIN dbo.Tab_Temp_Family
					ON dbo.TAB_FAMILY.TSN = dbo.Tab_Temp_Family.FAMILY_TSN
					WHERE dbo.TAB_EPTD_LOOK_UP.EPTD IS NOT NULL
				GROUP BY dbo.Tab_Temp_Family.EVENT_ID,
dbo.Tab_Temp_Family.SAMPLE_NUMBER, dbo.TAB_EPTD_LOOK_UP.EPTD_PARAMETER) AS ET
		INNER JOIN
				(SELECT dbo.Tab_Temp_Indicator.EVENT_ID,
dbo.Tab_Temp_Indicator.SAMPLE_NUMBER, dbo.Tab_Temp_Indicator.PARAMETER,
dbo.Tab_Temp_Indicator.REPORTING_VALUE
					FROM dbo.Tab_Temp_Indicator
					WHERE dbo.Tab_Temp_Indicator.PARAMETER = 'TOTAL_ABUNDANCE') AS
TA
		 ON (ET.SAMPLE_NUMBER = TA.SAMPLE_NUMBER) AND (ET.EVENT_ID = TA.EVENT_ID)
	PRINT 'Step 13: Create Qry_EPTD_TAXA_PCT_2 - Completed Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 13: Create Qry_EPTD_TAXA_PCT_2 - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

--Step 14: Create Qry_PCT_EPT_RICH
BEGIN TRY
	BEGIN TRANSACTION
	PRINT 'Step 14: Create Qry_PCT_EPT_RICH'
		INSERT INTO dbo.Tab_Temp_Indicator(EVENT_ID, SAMPLE_NUMBER, PARAMETER,
REPORTING_VALUE)
		SELECT TI.EVENT_ID, TI.SAMPLE_NUMBER, 'PCT_EPT_TAXA_RICH' AS P,
ROUND(((TI.REPORTING_VALUE/TR.REPORTING_VALUE)*100),2) AS RESULT
		FROM (SELECT dbo.Tab_Temp_Indicator.EVENT_ID, dbo.Tab_Temp_Indicator.SAMPLE_NUMBER,
dbo.Tab_Temp_Indicator.PARAMETER, dbo.Tab_Temp_Indicator.REPORTING_VALUE
				FROM dbo.Tab_Temp_Indicator
				WHERE dbo.Tab_Temp_Indicator.PARAMETER = 'EPT_TAXA_COUNT')AS TI
		INNER JOIN
			(SELECT dbo.Tab_Temp_Indicator.EVENT_ID, dbo.Tab_Temp_Indicator.SAMPLE_NUMBER,
dbo.Tab_Temp_Indicator.PARAMETER, dbo.Tab_Temp_Indicator.REPORTING_VALUE
				FROM dbo.Tab_Temp_Indicator
				WHERE dbo.Tab_Temp_Indicator.PARAMETER = 'TAXA_RICH') AS TR
		ON (TI.EVENT_ID = TR.EVENT_ID) AND (TI.SAMPLE_NUMBER = TR.SAMPLE_NUMBER)
	PRINT 'Step 14: Create Qry_PCT_EPT_RICH - Completed Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 14: Create Qry_PCT_EPT_RICH - Will Rollback '
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

--Step 15: Create Qry_EPT_TAXA_CNT_NO_TOL
BEGIN TRY
	BEGIN TRANSACTION
	PRINT 'Step 15: Create Qry_EPT_TAXA_CNT_NO_TOL - End of Calculating Dominance Metrics'
		INSERT INTO dbo.Tab_Temp_Indicator(EVENT_ID, SAMPLE_NUMBER, PARAMETER,
REPORTING_VALUE)
		SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER,
'EPT_TAXA_COUNT_NO_TOL' AS PARAMETER,
		COUNT(dbo.Tab_Temp_Family.FAMILY_TSN) AS CountOfFamily_TSN
		FROM dbo.TAB_EPTD_LOOK_UP INNER JOIN dbo.TAB_FAMILY
		ON dbo.TAB_EPTD_LOOK_UP.EPTD = dbo.TAB_FAMILY.EPTD INNER JOIN dbo.Tab_Temp_Family
			ON dbo.TAB_FAMILY.TSN = dbo.Tab_Temp_Family.FAMILY_TSN
		WHERE dbo.TAB_EPTD_LOOK_UP.EPTD <> 'D' AND dbo.TAB_FAMILY.TOLARANCE_VALUE < 7
		GROUP BY dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER
	PRINT 'Step 15: Create Qry_EPT_TAXA_CNT_NO_TOL - End of Calculating Dominance Metrics - Completed
Successfully'
```

```
		COMMIT TRANSACTION
END TRY
BEGIN CATCH
		PRINT 'Transaction Failed for Step 15: Create Qry_EPT_TAXA_CNT_NO_TOL - End of Calculating Dominance Metrics -
Will Rollback'
		IF @@TRANCOUNT > 0
		ROLLBACK

		EXECUTE usp_GetErrorInfo
END CATCH

--Step 16: Create Qry_DOM_2
BEGIN TRY
		BEGIN TRANSACTION
		PRINT 'Step 16: Create Qry_DOM_2 - Begin of Calculating Diversity Indexes'
			INSERT INTO dbo.Tab_Temp_Indicator (EVENT_ID, SAMPLE_NUMBER, PARAMETER,
REPORTING_VALUE)
			SELECT TA.EVENT_ID, TA.SAMPLE_NUMBER, 'PCT_DOM1' AS PARAMETER,
ROUND(((QD1.MaxSumOfReporting_Value/TA.REPORTING_VALUE)*100),2) AS CALC
			FROM (SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER,
MAX(dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE) AS MaxSumOfReporting_Value
					FROM dbo.Tab_Temp_Family
					GROUP BY dbo.Tab_Temp_Family.EVENT_ID,
dbo.Tab_Temp_Family.SAMPLE_NUMBER) AS QD1
			INNER JOIN
				(SELECT dbo.Tab_Temp_Indicator.EVENT_ID, dbo.Tab_Temp_Indicator.SAMPLE_NUMBER,
dbo.Tab_Temp_Indicator.PARAMETER, dbo.Tab_Temp_Indicator.REPORTING_VALUE
					FROM dbo.Tab_Temp_Indicator
					WHERE dbo.Tab_Temp_Indicator.PARAMETER = 'TOTAL_ABUNDANCE') AS TA
			ON (QD1.EVENT_ID = TA.EVENT_ID) AND (QD1.SAMPLE_NUMBER = TA.SAMPLE_NUMBER)
		PRINT 'Step 16: Create Qry_DOM_2 - Completed Successfully'
		COMMIT TRANSACTION
END TRY
BEGIN CATCH
		PRINT 'Transaction Failed for Step 16: Create Qry_DOM_2 - Will Rollback'
		IF @@TRANCOUNT > 0
		ROLLBACK

		EXECUTE usp_GetErrorInfo
END CATCH

--Step 17: Create Qry_PCT_DOM_MULTI
BEGIN TRY
		BEGIN TRANSACTION
		PRINT 'Step 17: Create Qry_PCT_DOM_MULTI'
			SELECT DISTINCT CAST(dbo.Tab_Temp_Family.EVENT_ID AS VARCHAR(10)) + '_' +
CAST(dbo.Tab_Temp_Family.SAMPLE_NUMBER AS VARCHAR(10)) AS CATEGORY,
			dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER
			INTO dbo.Temp_PCT_DOM_MULTI
			FROM dbo.Tab_Temp_Family
		PRINT 'Step 17: Create Qry_PCT_DOM_MULTI - Completed Successfully'
		COMMIT TRANSACTION
END TRY
BEGIN CATCH
		PRINT 'Transaction Failed for Step 17: Create Qry_PCT_DOM_MULTI - Will Rollback'
		IF @@TRANCOUNT > 0
		ROLLBACK

		EXECUTE usp_GetErrorInfo
END CATCH

--Step 18: Create Qry_CATEGORY_DOM
BEGIN TRY
		BEGIN TRANSACTION
		PRINT 'Step 18: Create Qry_CATEGORY_DOM'
			SELECT dbo.Temp_PCT_DOM_MULTI.CATEGORY, dbo.Tab_Temp_Family.EVENT_ID,
dbo.Tab_Temp_Family.SAMPLE_NUMBER, dbo.Tab_Temp_Family.FAMILY_TSN,
			dbo.Tab_Temp_Family.FAMILY, dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE,
dbo.Tab_Temp_Family.EXCLUSION_TAXA
			INTO dbo.Tab_Temp_Family_2
			FROM dbo.Temp_PCT_DOM_MULTI INNER JOIN dbo.Tab_Temp_Family
			ON dbo.Temp_PCT_DOM_MULTI.EVENT_ID = dbo.Tab_Temp_Family.EVENT_ID AND
dbo.Temp_PCT_DOM_MULTI.SAMPLE_NUMBER = dbo.Tab_Temp_Family.SAMPLE_NUMBER
		PRINT 'Step 18: Create Qry_CATEGORY_DOM - Completed Successfully'
		COMMIT TRANSACTION
END TRY
BEGIN CATCH
		PRINT 'Transaction Failed for Step 18: Create Qry_CATEGORY_DOM - Will Rollback'
		IF @@TRANCOUNT > 0
		ROLLBACK

		EXECUTE usp_GetErrorInfo
END CATCH

--Step 19: Create Qry_DOM_3_1 - PART ONE CREATE of DOM3
BEGIN TRY
		BEGIN TRANSACTION
		PRINT 'Step 19: Create Qry_DOM_3_1 - PART ONE CREATE of DOM3'
			SELECT dbo.Temp_PCT_DOM_MULTI.CATEGORY, dbo.Tab_Temp_Family_2.FAMILY,
dbo.Tab_Temp_Family_2.SUM_OF_REPORTING_VALUE
```

```
		INTO dbo.Temp_DOM_3
		FROM dbo.Temp_PCT_DOM_MULTI INNER JOIN dbo.Tab_Temp_Family_2
		ON dbo.Temp_PCT_DOM_MULTI.CATEGORY = dbo.Tab_Temp_Family_2.CATEGORY
		WHERE dbo.Tab_Temp_Family_2.SUM_OF_REPORTING_VALUE IN (SELECT TOP 3
dbo.Tab_Temp_Family_2.SUM_OF_REPORTING_VALUE

			FROM dbo.Tab_Temp_Family_2

			WHERE dbo.Tab_Temp_Family_2.CATEGORY = dbo.Temp_PCT_DOM_MULTI.CATEGORY

			ORDER BY dbo.Tab_Temp_Family_2.SUM_OF_REPORTING_VALUE DESC)
		ORDER BY dbo.Temp_PCT_DOM_MULTI.CATEGORY,
dbo.Tab_Temp_Family_2.SUM_OF_REPORTING_VALUE DESC
	PRINT 'Step 19: Create Qry_DOM_3_1 - PART ONE CREATE of DOM3 - Completed Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 19: Create Qry_DOM_3_1 - PART ONE CREATE of DOM3 - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

--Step 20: Create Qry_DOM3_3 - PART TWO Create of DOM3
BEGIN TRY
	BEGIN TRANSACTION
	PRINT 'Step 20: Create Qry_DOM3_3 - PART TWO Create of DOM3'
		INSERT INTO dbo.Tab_Temp_Indicator(EVENT_ID, SAMPLE_NUMBER, PARAMETER,
REPORTING_VALUE)
		SELECT TA.EVENT_ID, TA.SAMPLE_NUMBER, 'PCT_DOM3' AS METRIC,
ROUND(((D32.SumOfREPORTING_VALUE/TA.REPORTING_VALUE)*100),2) AS [VALUE]
		FROM (SELECT dbo.Tab_Temp_Indicator.EVENT_ID, dbo.Tab_Temp_Indicator.SAMPLE_NUMBER,
dbo.Tab_Temp_Indicator.PARAMETER, dbo.Tab_Temp_Indicator.REPORTING_VALUE
				FROM dbo.Tab_Temp_Indicator
				WHERE dbo.Tab_Temp_Indicator.PARAMETER = 'TOTAL_ABUNDANCE') AS TA
		INNER JOIN
			(SELECT dbo.Temp_PCT_DOM_MULTI.EVENT_ID,
dbo.Temp_PCT_DOM_MULTI.SAMPLE_NUMBER, SUM(dbo.Temp_DOM_3.SUM_OF_REPORTING_VALUE) AS
SUMOfREPORTING_VALUE
				FROM dbo.Temp_DOM_3 INNER JOIN dbo.Temp_PCT_DOM_MULTI
				ON dbo.Temp_DOM_3.CATEGORY = dbo.Temp_PCT_DOM_MULTI.CATEGORY
				GROUP BY dbo.Temp_PCT_DOM_MULTI.EVENT_ID,
dbo.Temp_PCT_DOM_MULTI.SAMPLE_NUMBER) AS D32
		ON (D32.EVENT_ID = TA.EVENT_ID) AND (D32.SAMPLE_NUMBER = TA.SAMPLE_NUMBER)
	PRINT 'Step 20: Create Qry_DOM3_3 - PART TWO Create of DOM3 - Completed Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 20: Create Qry_DOM3_3 - PART TWO Create of DOM3 - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

--Step 21: Create Qry_DOM_2_1 - PART ONE Creation of DOM2
BEGIN TRY
	BEGIN TRANSACTION
	PRINT 'Step 21: Create Qry_DOM_2_1 - PART ONE Creation of DOM2'
		SELECT dbo.Temp_PCT_DOM_MULTI.CATEGORY, dbo.Temp_DOM_3.FAMILY,
dbo.Temp_DOM_3.SUM_OF_REPORTING_VALUE
		INTO dbo.Temp_DOM_2
		FROM dbo.Temp_PCT_DOM_MULTI INNER JOIN dbo.Temp_DOM_3
		ON dbo.Temp_PCT_DOM_MULTI.CATEGORY = dbo.Temp_DOM_3.CATEGORY
		WHERE dbo.Temp_DOM_3.SUM_OF_REPORTING_VALUE IN (SELECT TOP 2
dbo.Temp_DOM_3.SUM_OF_REPORTING_VALUE

	FROM dbo.Temp_DOM_3

	WHERE dbo.Temp_DOM_3.CATEGORY = dbo.Temp_PCT_DOM_MULTI.CATEGORY

	ORDER BY  dbo.Temp_DOM_3.SUM_OF_REPORTING_VALUE DESC)
		ORDER BY dbo.Temp_PCT_DOM_MULTI.CATEGORY, dbo.Temp_DOM_3.SUM_OF_REPORTING_VALUE
DESC
	PRINT 'Step 21: Create Qry_DOM_2_1 - PART ONE Creation of DOM2 - Completed Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 21: Create Qry_DOM_2_1 - PART ONE Creation of DOM2 - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

--Step 22: Create Qry_DOM2_3 PART TWO Creation of DOM 2
BEGIN TRY
	BEGIN TRANSACTION
	PRINT 'Step 22: Create Qry_DOM2_3 PART TWO Creation of DOM 2 - End of Calculating Diversity Indexes'
```

```
		INSERT INTO dbo.Tab_Temp_Indicator(EVENT_ID, SAMPLE_NUMBER, PARAMETER,
REPORTING_VALUE)
		SELECT TA.EVENT_ID, TA.SAMPLE_NUMBER, 'PCT_DOM2' AS P,
ROUND(((D2.SumOfREPORTING_VALUE/TA.REPORTING_VALUE)*100),2) AS RESULT
		FROM (SELECT dbo.Tab_Temp_Indicator.EVENT_ID, dbo.Tab_Temp_Indicator.SAMPLE_NUMBER,
dbo.Tab_Temp_Indicator.PARAMETER, dbo.Tab_Temp_Indicator.REPORTING_VALUE
			FROM dbo.Tab_Temp_Indicator
			WHERE dbo.Tab_Temp_Indicator.PARAMETER = 'TOTAL_ABUNDANCE') AS TA
		INNER JOIN
			(SELECT dbo.Temp_PCT_DOM_MULTI.EVENT_ID,
dbo.Temp_PCT_DOM_MULTI.SAMPLE_NUMBER, SUM(dbo.Temp_DOM_2.SUM_OF_REPORTING_VALUE) AS
SUMOfREPORTING_VALUE
			FROM dbo.Temp_DOM_2 INNER JOIN dbo.Temp_PCT_DOM_MULTI
			ON dbo.Temp_DOM_2.CATEGORY = dbo.Temp_PCT_DOM_MULTI.CATEGORY
			GROUP BY dbo.Temp_PCT_DOM_MULTI.EVENT_ID,
dbo.Temp_PCT_DOM_MULTI.SAMPLE_NUMBER) AS D2
		ON (TA.SAMPLE_NUMBER = D2.SAMPLE_NUMBER) AND (TA.EVENT_ID = D2.EVENT_ID)
	PRINT 'Step 22: Create Qry_DOM2_3 PART TWO Creation of DOM 2 End of Calculating Diversity Indexes - Completed
Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 22: Create Qry_DOM2_3 PART TWO Creation of DOM 2 - End of Calculating
Diversity Indexes - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

--Step 23 Part 1: Create Shannon Weiner Index Step 1: QRY_SW_2
BEGIN TRY
	BEGIN TRANSACTION
	PRINT 'Step 23: Create Shannon Weiner Index - Part 1: QRY_SW_2 - Begin Calculating Tolarance Metrics'
		SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER,
dbo.Tab_Temp_Family.FAMILY_TSN, dbo.Tab_Temp_Family.FAMILY,
		dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE,
dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE/TA.REPORTING_VALUE AS P,
		LOG(dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE/TA.REPORTING_VALUE) AS LN_P,
		(dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE/TA.REPORTING_VALUE) *
LOG(dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE/TA.REPORTING_VALUE) AS P_LN_P,
		POWER(dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE/TA.REPORTING_VALUE, 2) AS
P2
		INTO dbo.Temp_SW
		FROM (SELECT dbo.Tab_Temp_Indicator.EVENT_ID,
dbo.Tab_Temp_Indicator.SAMPLE_NUMBER, dbo.Tab_Temp_Indicator.PARAMETER,
dbo.Tab_Temp_Indicator.REPORTING_VALUE
			FROM dbo.Tab_Temp_Indicator
			WHERE dbo.Tab_Temp_Indicator.PARAMETER = 'TOTAL_ABUNDANCE') AS
TA INNER JOIN dbo.Tab_Temp_Family
		ON (TA.SAMPLE_NUMBER = dbo.Tab_Temp_Family.SAMPLE_NUMBER) AND (TA.EVENT_ID =
dbo.Tab_Temp_Family.EVENT_ID)
	PRINT 'Step 23: Create Shannon Weiner Index - Part 1: QRY_SW_2 - Begin Calculating Tolarance Metrics - Completed
Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 23: Create Shannon Weiner Index - Part 1: QRY_SW_2 - Begin Calculating Tolarance
Metrics - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

--Step 23 Part 2:QRY_SW_4
BEGIN TRY
	BEGIN TRANSACTION
	PRINT 'Step 23: Create Shannon Weiner Index - Part 2: QRY_SW_4'
		INSERT INTO dbo.Tab_Temp_Indicator (EVENT_ID, SAMPLE_NUMBER, PARAMETER,
REPORTING_VALUE)
		SELECT TS.EVENT_ID, TS.SAMPLE_NUMBER, TS.PARAMETER, ROUND(-1 * TS.SumOfP_LN_P, 4) AS
REPORTING_VALUE
		FROM (SELECT dbo.Temp_SW.EVENT_ID, dbo.Temp_SW.SAMPLE_NUMBER, 'FSW' AS PARAMETER,
SUM(dbo.Temp_SW.P_LN_P) AS SumOfP_LN_P
			FROM dbo.Temp_SW
			GROUP BY dbo.Temp_SW.EVENT_ID, dbo.Temp_SW.SAMPLE_NUMBER) AS TS
	PRINT 'Step 23: Create Shannon Weiner Index - Part 2: QRY_SW_4 - Completed Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 23: Create Shannon Weiner Index - Part 2: QRY_SW_4 - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

--Step 24: Create Qry_SIMPSON_2 - SIMPSONS INDEX
BEGIN TRY
```

```
	BEGIN TRANSACTION
	PRINT 'Step 24: Create Qry_SIMPSON_2 - SIMPSONS INDEX'
		INSERT INTO dbo.Tab_Temp_Indicator(EVENT_ID, SAMPLE_NUMBER, PARAMETER,
REPORTING_VALUE)
		SELECT S1.EVENT_ID, S1.SAMPLE_NUMBER, 'SIMPSON_DIVERSITY' AS PARAMETER, ROUND(1 -
S1.SumOf_P2,4) AS REPORTING_VALUE
		FROM (SELECT dbo.Temp_SW.EVENT_ID, dbo.Temp_SW.SAMPLE_NUMBER, SUM(dbo.Temp_SW.P2) AS
SumOf_P2
			FROM dbo.Temp_SW
			GROUP BY dbo.Temp_SW.EVENT_ID, dbo.Temp_SW.SAMPLE_NUMBER) AS S1
	PRINT 'Step 24: Create Qry_SIMPSON_2 - SIMPSONS INDEX - Completed Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 24: Create Qry_SIMPSON_2 - SIMPSONS INDEX - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

--Step 25: Create Qry_GOLD_3 - Gold Metric
BEGIN TRY
	BEGIN TRANSACTION
	PRINT 'Step 25: Create Qry_GOLD_3 - Gold Metric'
		INSERT INTO dbo.Tab_Temp_Indicator(EVENT_ID, SAMPLE_NUMBER, PARAMETER,
REPORTING_VALUE)
		SELECT TA.EVENT_ID, TA.SAMPLE_NUMBER, 'GOLD' AS PARAMETER, ROUND(1 -
G2.SumOfREPORTING_VALUE/TA.REPORTING_VALUE,4) AS GOLD
		FROM (SELECT dbo.Tab_Temp_Indicator.EVENT_ID, dbo.Tab_Temp_Indicator.SAMPLE_NUMBER,
dbo.Tab_Temp_Indicator.PARAMETER, dbo.Tab_Temp_Indicator.REPORTING_VALUE
			FROM dbo.Tab_Temp_Indicator
			WHERE dbo.Tab_Temp_Indicator.PARAMETER = 'TOTAL_ABUNDANCE') AS TA
		INNER JOIN
			(SELECT G1.EVENT_ID, G1.SAMPLE_NUMBER, SUM(G1.SUM_OF_REPORTING_VALUE) AS
SumOfREPORTING_VALUE
			 FROM (SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER,
dbo.Tab_Temp_Family.FAMILY_TSN, dbo.Tab_Temp_Family.FAMILY,
			 dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE, dbo.TAB_FAMILY.GOLD_TAXA
				FROM dbo.Tab_Temp_Family INNER JOIN dbo.TAB_FAMILY
				ON dbo.Tab_Temp_Family.FAMILY_TSN = dbo.TAB_FAMILY.TSN
				WHERE dbo.TAB_FAMILY.GOLD_TAXA IS NOT NULL) AS G1
			GROUP BY G1.EVENT_ID, G1.SAMPLE_NUMBER) AS G2
		ON (TA.EVENT_ID = G2.EVENT_ID) AND (TA.SAMPLE_NUMBER = G2.SAMPLE_NUMBER)
	PRINT 'Step 25: Create Qry_GOLD_3 - Gold Metric - Completed Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 25: Create Qry_GOLD_3 - Gold Metric - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

--Step 26: Create Qry_SEL_EPTD_3
BEGIN TRY
	BEGIN TRANSACTION
	PRINT 'Step 26: Create Qry_SEL_EPTD_3'
		INSERT INTO dbo.Tab_Temp_Indicator(EVENT_ID, SAMPLE_NUMBER, PARAMETER,
REPORTING_VALUE)
		SELECT SE2.EVENT_ID, SE2.SAMPLE_NUMBER, 'LOG10_SEL_EPTD' AS PARAMETER,
ROUND(LOG10(SE2.SumOfREPORTING_VALUE + 1), 2) AS REPORTING_VALUE
		FROM (SELECT SE1.EVENT_ID, SE1.SAMPLE_NUMBER, SUM(SE1.SUM_OF_REPORTING_VALUE) AS
SumOfREPORTING_VALUE
			FROM(SELECT dbo.Tab_Temp_Family.EVENT_ID,
dbo.Tab_Temp_Family.SAMPLE_NUMBER, dbo.Tab_Temp_Family.Family,
dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE,
			dbo.TAB_FAMILY.SEL_EPTD
				FROM dbo.Tab_Temp_Family INNER JOIN dbo.TAB_FAMILY
				ON dbo.Tab_Temp_Family.FAMILY_TSN = dbo.TAB_FAMILY.TSN
				WHERE dbo.TAB_FAMILY.SEL_EPTD IS NOT NULL) AS SE1
			GROUP BY SE1.EVENT_ID, SE1.SAMPLE_NUMBER) AS SE2
	PRINT 'Step 26: Create Qry_SEL_EPTD_3 - Completed Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 26: Create Qry_SEL_EPTD_3 - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

--Step 27: Create Qry_MARGALEF_1 - MARGALEF INDEX
BEGIN TRY
	BEGIN TRANSACTION
	PRINT 'Step 27: Create Qry_MARGALEF_1 - MARGALEF INDEX - End of Calculating Tolarance Metrics'
		INSERT INTO dbo.Tab_Temp_Indicator(EVENT_ID, SAMPLE_NUMBER, PARAMETER,
REPORTING_VALUE)
```

```
		SELECT TR.EVENT_ID, TR.SAMPLE_NUMBER, 'MARGALEFS' AS PARAMETER, 'VALUE' =
		CASE
			WHEN TR.REPORTING_VALUE = 1 THEN 0 --ROUND((TR.REPORTING_VALUE -1
)/LOG(TA.REPORTING_VALUE), 2)
			--WHEN TR.REPORTING_VALUE = 0 THEN ROUND((TR.REPORTING_VALUE -1
)/LOG(TA.REPORTING_VALUE), 2)
		ELSE
			ROUND((TR.REPORTING_VALUE -1)/LOG(TA.REPORTING_VALUE), 4)

		END
		FROM (SELECT dbo.Tab_Temp_Indicator.EVENT_ID, dbo.Tab_Temp_Indicator.SAMPLE_NUMBER,
dbo.Tab_Temp_Indicator.PARAMETER, dbo.Tab_Temp_Indicator.REPORTING_VALUE
			FROM dbo.Tab_Temp_Indicator
			WHERE dbo.Tab_Temp_Indicator.PARAMETER = 'TAXA_RICH') AS TR
		INNER JOIN
			(SELECT dbo.Tab_Temp_Indicator.EVENT_ID, dbo.Tab_Temp_Indicator.SAMPLE_NUMBER,
dbo.Tab_Temp_Indicator.PARAMETER, dbo.Tab_Temp_Indicator.REPORTING_VALUE
			FROM dbo.Tab_Temp_Indicator
			WHERE dbo.Tab_Temp_Indicator.PARAMETER = 'TOTAL_ABUNDANCE') AS TA
		ON (TR.SAMPLE_NUMBER = TA.SAMPLE_NUMBER) AND (TR.EVENT_ID = TA.EVENT_ID)
	PRINT 'Step 27: Create Qry_MARGALEF_1 - MARGALEF INDEX - End of Calculating Tolarance Metrics - Completed
Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 27: Create Qry_MARGALEF_1 - MARGALEF INDEX - End of Calculating Tolarance
Metrics - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

--Step 28: Create Qry_PCT_TOLERANT_2
BEGIN TRY
	BEGIN TRANSACTION
	PRINT 'Step 28: Create Qry_PCT_TOLERANT_2 - Begin of Calculating Habitat and Guild Metrics'
		INSERT INTO dbo.Tab_Temp_Indicator(EVENT_ID, SAMPLE_NUMBER, PARAMETER,
REPORTING_VALUE)
		SELECT PT1.EVENT_ID, PT1.SAMPLE_NUMBER, 'PCT_TOLERANT' AS P,
ROUND(((PT1.SumOfREPORTING_VALUE/TA.REPORTING_VALUE)*100), 2) AS RESULT
		FROM (SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER,
SUM(dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE) AS SumOfREPORTING_VALUE
			FROM dbo.TAB_FAMILY INNER JOIN dbo.Tab_Temp_Family
			ON dbo.TAB_FAMILY.TSN = dbo.Tab_Temp_Family.family_tsn
			WHERE dbo.TAB_FAMILY.TOLARANCE_VALUE >= 7
			GROUP BY dbo.Tab_Temp_Family.EVENT_ID,
dbo.Tab_Temp_Family.SAMPLE_NUMBER) AS PT1
		INNER JOIN
			(SELECT dbo.Tab_Temp_Indicator.EVENT_ID, dbo.Tab_Temp_Indicator.SAMPLE_NUMBER,
dbo.Tab_Temp_Indicator.PARAMETER, dbo.Tab_Temp_Indicator.REPORTING_VALUE
			FROM dbo.Tab_Temp_Indicator
			WHERE dbo.Tab_Temp_Indicator.PARAMETER = 'TOTAL_ABUNDANCE') AS TA
		ON (PT1.EVENT_ID = TA.EVENT_ID) AND (PT1.SAMPLE_NUMBER = TA.SAMPLE_NUMBER)
	PRINT 'Step 28: Create Qry_PCT_TOLERANT_2 - Completed Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 28: Create Qry_PCT_TOLERANT_2 - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

--Step 29: Create Qry_PCT_SENSITIVE_2
BEGIN TRY
	BEGIN TRANSACTION
	PRINT 'Step 29: Create Qry_PCT_SENSITIVE_2'
		INSERT INTO dbo.Tab_Temp_Indicator(EVENT_ID, SAMPLE_NUMBER, PARAMETER,
REPORTING_VALUE )
		SELECT PS1.EVENT_ID, PS1.SAMPLE_NUMBER, 'PCT_SENSITIVE' AS P,
ROUND(((PS1.SumOfREPORTING_VALUE/TA.REPORTING_VALUE)*100), 2) AS REPORT
		FROM (SELECT dbo.Tab_Temp_Indicator.EVENT_ID, dbo.Tab_Temp_Indicator.SAMPLE_NUMBER,
dbo.Tab_Temp_Indicator.PARAMETER, dbo.Tab_Temp_Indicator.REPORTING_VALUE
			FROM dbo.Tab_Temp_Indicator
			WHERE dbo.Tab_Temp_Indicator.PARAMETER = 'TOTAL_ABUNDANCE') AS TA
		INNER JOIN
			(SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER,
SUM(dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE) AS SumOfREPORTING_VALUE
			FROM dbo.TAB_FAMILY INNER JOIN dbo.Tab_Temp_Family
			ON dbo.TAB_FAMILY.TSN = dbo.Tab_Temp_Family.FAMILY_TSN
			WHERE dbo.TAB_FAMILY.TOLARANCE_VALUE <=3
			GROUP BY dbo.Tab_Temp_Family.EVENT_ID,
dbo.Tab_Temp_Family.SAMPLE_NUMBER) AS PS1
		ON (TA.EVENT_id = PS1.EVENT_ID) AND (TA.SAMPLE_NUMBER = PS1.SAMPLE_NUMBER)
	PRINT 'Step 29: Create Qry_PCT_SENSITIVE_2 - Completed Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
```

```
	PRINT 'Transaction Failed for Step 29: Create Qry_PCT_SENSITIVE_2 - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

--Step 30: Create Qry_ASTP_2
BEGIN TRY
	BEGIN TRANSACTION
	PRINT 'Step 30: Create Qry_ASTP_2'
		INSERT INTO dbo.Tab_Temp_Indicator(EVENT_ID, SAMPLE_NUMBER, PARAMETER,
REPORTING_VALUE)
		SELECT QA1.EVENT_ID, QA1.SAMPLE_NUMBER, 'ASPT_MOD' AS PARAMETER,
ROUND((QA1.SumOfTOLARANCE_VALUE/QA1.CountOfFamily), 2) AS [VALUE]
		FROM (SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER,
SUM(dbo.TAB_FAMILY.TOLARANCE_VALUE) AS SumOfTOLARANCE_VALUE,
		COUNT(dbo.TAB_FAMILY.FAMILY) AS CountOfFamily
			FROM dbo.Tab_Temp_Family INNER JOIN dbo.TAB_FAMILY
			ON dbo.Tab_Temp_Family.FAMILY_TSN = dbo.TAB_FAMILY.TSN
			GROUP BY dbo.Tab_Temp_Family.EVENT_ID,
dbo.Tab_Temp_Family.SAMPLE_NUMBER
			HAVING SUM(dbo.TAB_FAMILY.TOLARANCE_VALUE) IS NOT NULL) AS QA1
	PRINT 'Step 30: Create Qry_ASTP_2 - Completed Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 30: Create Qry_ASTP_2 - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

--Step 31: Create Qry_FBI_Calc_4
BEGIN TRY
	BEGIN TRANSACTION
	PRINT 'Step 31: Create Qry_FBI_Calc_4'
		INSERT INTO dbo.Tab_Temp_Indicator(EVENT_ID, SAMPLE_NUMBER, PARAMETER,
REPORTING_VALUE)
		SELECT QFC3.EVENT_ID, QFC3.SAMPLE_NUMBER, QFC3.PARAMETER, ROUND(QFC3.SumOfCACL2, 4)
AS VAV
		FROM (SELECT QFC2.EVENT_ID, QFC2.SAMPLE_NUMBER, 'FBI' AS PARAMETER,
SUM(ISNULL(QFC2.CACL2,0)) AS SumOfCACL2
			FROM (SELECT TA.EVENT_ID, TA.SAMPLE_NUMBER,
(QFC1.CACL1/TA.REPORTING_VALUE) AS CACL2
				FROM (SELECT dbo.Tab_Temp_Family.EVENT_ID,
dbo.Tab_Temp_Family.SAMPLE_NUMBER, dbo.Tab_Temp_Family.FAMILY_TSN, dbo.Tab_Temp_Family.FAMILY,
				(dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE *
dbo.TAB_FAMILY.TOLARANCE_VALUE) AS CACL1
					FROM dbo.TAB_FAMILY INNER JOIN
dbo.Tab_Temp_Family
					ON dbo.TAB_FAMILY.TSN =
dbo.Tab_Temp_Family.FAMILY_TSN) AS QFC1
				INNER JOIN
					(SELECT dbo.Tab_Temp_Indicator.EVENT_ID,
dbo.Tab_Temp_Indicator.SAMPLE_NUMBER, dbo.Tab_Temp_Indicator.PARAMETER,
					dbo.Tab_Temp_Indicator.REPORTING_VALUE
						FROM dbo.Tab_Temp_Indicator
						WHERE dbo.Tab_Temp_Indicator.PARAMETER
= 'TOTAL_ABUNDANCE') AS TA
				ON (QFC1.EVENT_id = TA.EVENT_ID) AND (QFC1.SAMPLE_NUMBER =
TA.SAMPLE_NUMBER)) AS QFC2
		GROUP BY QFC2.EVENT_ID, QFC2.SAMPLE_NUMBER) AS QFC3
	PRINT 'Step 31: Create Qry_FBI_Calc_4 - Completed Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 31: Create Qry_FBI_Calc_4 - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

--Step 32: Create Qry_Calc_Urban_2
BEGIN TRY
	BEGIN TRANSACTION
	PRINT 'Step 32: Create Qry_Calc_Urban_2'
		INSERT INTO dbo.Tab_Temp_Indicator (EVENT_ID, SAMPLE_NUMBER, PARAMETER,
REPORTING_VALUE)
		SELECT TR.EVENT_ID, TR.SAMPLE_NUMBER, 'PCT_URBAN_INTOL' AS PARAMETER, 'VAL' =
		CASE
			WHEN QCU1.CountOfFamily IS NULL THEN 0
			ELSE
				ROUND(((QCU1.CountOfFamily/TR.REPORTING_VALUE)*100), 2)
		END
		FROM (SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER,
COUNT(dbo.Tab_Temp_Family.FAMILY) AS CountOfFamily, dbo.TAB_FAMILY.INTOERANT_URBAN
			FROM dbo.Tab_Temp_Family INNER JOIN dbo.TAB_FAMILY
```

```
                                        ON dbo.Tab_Temp_Family.FAMILY_TSN = dbo.TAB_FAMILY.TSN
                                        GROUP BY dbo.Tab_Temp_Family.EVENT_ID,
dbo.Tab_Temp_Family.SAMPLE_NUMBER, dbo.TAB_FAMILY.INTOERANT_URBAN
                                        HAVING dbo.TAB_FAMILY.INTOERANT_URBAN = 'X') AS QCU1
                RIGHT JOIN
                        (SELECT dbo.Tab_Temp_Indicator.EVENT_ID, dbo.Tab_Temp_Indicator.SAMPLE_NUMBER,
dbo.Tab_Temp_Indicator.PARAMETER, dbo.Tab_Temp_Indicator.REPORTING_VALUE
                                        FROM dbo.Tab_Temp_Indicator
                                        WHERE dbo.Tab_Temp_Indicator.PARAMETER = 'TAXA_RICH') AS TR
                ON (QCU1.EVENT_ID = TR.EVENT_ID) AND (QCU1.SAMPLE_NUMBER = TR.SAMPLE_NUMBER)
        PRINT 'Step 32: Create Qry_Calc_Urban_2 - Completed Successfully'
        COMMIT TRANSACTION
END TRY
BEGIN CATCH
        PRINT 'Transaction Failed for Step 32: Create Qry_Calc_Urban_2 - Will Rollback'
        IF @@TRANCOUNT > 0
        ROLLBACK

        EXECUTE usp_GetErrorInfo
END CATCH

--Step 33: Create Qry_BECK_4
BEGIN TRY
        BEGIN TRANSACTION
        PRINT 'Step 33: Create Qry_BECK_4'
                INSERT INTO dbo.Tab_Temp_Indicator(EVENT_ID, SAMPLE_NUMBER, PARAMETER,
REPORTING_VALUE)
                SELECT QB3.EVENT_ID, QB3.SAMPLE_NUMBER, 'BECK' AS PARAMETER, ((2 * QB3.BECK_1) +
QB3.BECK_2) AS BI
                FROM(SELECT TA.EVENT_ID, TA.SAMPLE_NUMBER, ISNULL(QB1.CountSumOfREPORTING_VALUE, 0)
AS BECK_1, ISNULL(QB2.CountSumOfREPORTING_VALUE, 0) AS BECK_2
                                        FROM (SELECT dbo.Tab_Temp_Family.EVENT_ID,
dbo.Tab_Temp_Family.SAMPLE_NUMBER, COUNT(dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE) AS
CountSumOfREPORTING_VALUE
                                                        FROM dbo.TAB_FAMILY INNER JOIN dbo.Tab_Temp_Family
                                                        ON dbo.TAB_FAMILY.TSN = dbo.Tab_Temp_Family.FAMILY_TSN
                                                        WHERE dbo.TAB_FAMILY.BECK_CLASS = 2
                                                        GROUP BY dbo.Tab_Temp_Family.EVENT_ID,
dbo.Tab_Temp_Family.SAMPLE_NUMBER) AS QB2
                                        RIGHT JOIN
                                                (SELECT dbo.Tab_Temp_Family.EVENT_ID,
dbo.Tab_Temp_Family.SAMPLE_NUMBER, COUNT(dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE) AS
CountSumOfREPORTING_VALUE
                                                        FROM dbo.TAB_FAMILY INNER JOIN dbo.Tab_Temp_Family
                                                        ON dbo.TAB_FAMILY.TSN = dbo.Tab_Temp_Family.FAMILY_TSN
                                                        WHERE dbo.TAB_FAMILY.BECK_CLASS = 1
                                                        GROUP BY dbo.Tab_Temp_Family.EVENT_ID,
dbo.Tab_Temp_Family.SAMPLE_NUMBER) AS QB1
                                        RIGHT JOIN
                                                (SELECT dbo.Tab_Temp_Indicator.EVENT_ID,
dbo.Tab_Temp_Indicator.SAMPLE_NUMBER, dbo.Tab_Temp_Indicator.PARAMETER,
dbo.Tab_Temp_Indicator.REPORTING_VALUE
                                                        FROM dbo.Tab_Temp_Indicator
                                                        WHERE dbo.Tab_Temp_Indicator.PARAMETER =
'TOTAL_ABUNDANCE') AS TA
                                        ON (QB1.EVENT_ID = TA.EVENT_id) AND (QB1.SAMPLE_NUMBER =
TA.SAMPLE_NUMBER)
                ON (QB2.EVENT_ID = TA.EVENT_id) AND (QB2.SAMPLE_NUMBER = TA.SAMPLE_NUMBER)) AS QB3
        PRINT 'Step 33: Create Qry_BECK_4 - Completed Successfully'
        COMMIT TRANSACTION
END TRY
BEGIN CATCH
        PRINT 'Transaction Failed for Step 33: Create Qry_BECK_4 - Will Rollback'
        IF @@TRANCOUNT > 0
        ROLLBACK

        EXECUTE usp_GetErrorInfo
END CATCH

--Step 34: Create Qry_Count_Sensitive
BEGIN TRY
        BEGIN TRANSACTION
        PRINT 'Step 34: Create Qry_Count_Sensitive'
                INSERT INTO dbo.Tab_Temp_Indicator(EVENT_ID, SAMPLE_NUMBER, PARAMETER,
REPORTING_VALUE)
                SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER,
'SENSITIVE_TAXA_COUNT' AS PARAMETER, COUNT(dbo.TAB_FAMILY.Family) AS CountOfFamily
                FROM dbo.TAB_FAMILY INNER JOIN dbo.Tab_Temp_Family
                ON dbo.TAB_FAMILY.TSN = dbo.Tab_Temp_Family.FAMILY_TSN
                WHERE dbo.TAB_FAMILY.TOLARANCE_VALUE <= 3
                GROUP BY dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER
        PRINT 'Step 34: Create Qry_Count_Sensitive - Completed Successfully'
        COMMIT TRANSACTION
END TRY
BEGIN CATCH
        PRINT 'Transaction Failed for Step 34: Create Qry_Count_Sensitive - Will Rollback'
        IF @@TRANCOUNT > 0
        ROLLBACK

        EXECUTE usp_GetErrorInfo
```

```
END CATCH

--Step 35: Create Qry_Count_Tolerant
BEGIN TRY
	BEGIN TRANSACTION
	PRINT 'Step 35: Create Qry_Count_Tolerant - End of Calculating Habitat and Guild Metrics'
		INSERT INTO dbo.Tab_Temp_Indicator(EVENT_ID, SAMPLE_NUMBER, PARAMETER,
REPORTING_VALUE)
		SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER,
'TOLERANT_TAXA_COUNT' AS PARAMETER, COUNT(dbo.TAB_FAMILY.Family) AS CountOfFamily
		FROM dbo.TAB_FAMILY INNER JOIN dbo.Tab_Temp_Family ON dbo.TAB_FAMILY.TSN =
dbo.Tab_Temp_Family.FAMILY_TSN
		WHERE dbo.TAB_FAMILY.TOLARANCE_VALUE >= 7
		GROUP BY dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER
	PRINT 'Step 35: Create Qry_Count_Tolerant - Completed Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 35: Create Qry_Count_Tolerant - End of Calculating Habitat and Guild Metrics - Will
Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

--Step 36: Create Qry_Calc_Collect_2
BEGIN TRY
	BEGIN TRANSACTION
	PRINT 'Step 36: Create Qry_Calc_Collect_2 - Begin of Calculating Other General Metrics'
		INSERT INTO dbo.Tab_Temp_Indicator(EVENT_ID, SAMPLE_NUMBER, PARAMETER,
REPORTING_VALUE)
		SELECT QCC1.EVENT_ID, QCC1.SAMPLE_NUMBER, 'PCT_COLLECT' AS PARAMETER,
ROUND(((QCC1.SumOfReporting_Value/TA.REPORTING_VALUE)*100),2) AS CALC2
		FROM (SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER,
SUM(dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE) AS SumOfReporting_Value
			FROM dbo.TAB_FAMILY INNER JOIN dbo.Tab_Temp_Family
			ON dbo.TAB_FAMILY.TSN = dbo.Tab_Temp_Family.FAMILY_TSN
			WHERE dbo.TAB_FAMILY.GUILD = 'CG' OR dbo.TAB_FAMILY.GUILD = 'CF'
			GROUP BY dbo.Tab_Temp_Family.EVENT_ID,
dbo.Tab_Temp_Family.SAMPLE_NUMBER) AS QCC1
		INNER JOIN
			(SELECT dbo.Tab_Temp_Indicator.EVENT_ID, dbo.Tab_Temp_Indicator.SAMPLE_NUMBER,
dbo.Tab_Temp_Indicator.PARAMETER, dbo.Tab_Temp_Indicator.REPORTING_VALUE
			FROM dbo.Tab_Temp_Indicator
			WHERE dbo.Tab_Temp_Indicator.PARAMETER = 'TOTAL_ABUNDANCE') AS TA
		ON (QCC1.SAMPLE_NUMBER = TA.SAMPLE_NUMBER) AND (QCC1.EVENT_ID = TA.EVENT_ID)
	PRINT 'Step 36: Create Qry_Calc_Collect_2 - Completed Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 36: Create Qry_Calc_Collect_2 - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

--Step 37: Create Qry_Calc_Cling_2
BEGIN TRY
	BEGIN TRANSACTION
	PRINT 'Step 37: Create Qry_Calc_Cling_2'
		INSERT INTO dbo.Tab_Temp_Indicator(EVENT_ID, SAMPLE_NUMBER, PARAMETER,
REPORTING_VALUE)
		SELECT QCC1.EVENT_ID, QCC1.SAMPLE_NUMBER, 'PCT_CLING' AS PARAMETER,
ROUND(((QCC1.SumOfREPORTING_VALUE/TA.REPORTING_VALUE)*100),2) AS CALC2
		FROM (SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER,
Sum(dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE) AS SumOfREPORTING_VALUE
			FROM dbo.TAB_FAMILY INNER JOIN dbo.Tab_Temp_Family ON dbo.TAB_FAMILY.TSN
= dbo.Tab_Temp_Family.FAMILY_TSN
			WHERE dbo.TAB_FAMILY.HABITAT LIKE '%CN%'
			GROUP BY dbo.Tab_Temp_Family.EVENT_ID,
dbo.Tab_Temp_Family.SAMPLE_NUMBER) AS QCC1
		INNER JOIN
			(SELECT dbo.Tab_Temp_Indicator.EVENT_ID, dbo.Tab_Temp_Indicator.SAMPLE_NUMBER,
dbo.Tab_Temp_Indicator.PARAMETER, dbo.Tab_Temp_Indicator.REPORTING_VALUE
			FROM dbo.Tab_Temp_Indicator
			WHERE dbo.Tab_Temp_Indicator.PARAMETER = 'TOTAL_ABUNDANCE') AS TA
		ON (QCC1.SAMPLE_NUMBER = TA.SAMPLE_NUMBER) AND (QCC1.EVENT_ID = TA.EVENT_ID)
	PRINT 'Step 37: Create Qry_Calc_Cling_2 - Completed Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 37: Create Qry_Calc_Cling_2 - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH
```

```
--Step 38: Create Qry_Calc_Scrape_2
BEGIN TRY
	BEGIN TRANSACTION
	PRINT 'Step 38: Create Qry_Calc_Scrape_2'
		INSERT INTO dbo.Tab_Temp_Indicator(EVENT_ID, SAMPLE_NUMBER, PARAMETER,
REPORTING_VALUE)
		SELECT TA.EVENT_id, TA.SAMPLE_NUMBER, 'SCRAPER_TAXA_CNT' AS PARAMETER,
IsNull(QCS1.CountOfFamily,0) AS [VALUE]
		FROM (SELECT dbo.Tab_Temp_Indicator.EVENT_ID, dbo.Tab_Temp_Indicator.SAMPLE_NUMBER,
dbo.Tab_Temp_Indicator.PARAMETER, dbo.Tab_Temp_Indicator.REPORTING_VALUE
				FROM dbo.Tab_Temp_Indicator
				WHERE dbo.Tab_Temp_Indicator.PARAMETER = 'TOTAL_ABUNDANCE') AS TA
		LEFT JOIN
			(SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER,
COUNT(dbo.Tab_Temp_Family.Family) AS CountOfFamily
				FROM dbo.TAB_FAMILY INNER JOIN dbo.Tab_Temp_Family
				ON dbo.TAB_FAMILY.TSN = dbo.Tab_Temp_Family.family_tsn
				WHERE dbo.TAB_FAMILY.GUILD = 'SC'
				GROUP BY dbo.Tab_Temp_Family.EVENT_ID,
dbo.Tab_Temp_Family.SAMPLE_NUMBER) AS QCS1
		ON (TA.SAMPLE_NUMBER = QCS1.SAMPLE_NUMBER) AND (TA.EVENT_id = QCS1.EVENT_ID)
	PRINT 'Step 38: Create Qry_Calc_Scrape_2 - Completed Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 38: Create Qry_Calc_Scrape_2 - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

-- Step 39: Create Qry_Calc_Climb_2
BEGIN TRY
	BEGIN TRANSACTION
	PRINT 'Step 39: Create Qry_Calc_Climb_2'
		INSERT INTO dbo.Tab_Temp_Indicator(EVENT_ID, SAMPLE_NUMBER, PARAMETER,
REPORTING_VALUE)
		SELECT TA.EVENT_ID, TA.SAMPLE_NUMBER, 'PCT_CLIMB' AS PARAMETER, 'VALUE' =
		CASE
			WHEN QCC1.SumOfREPORTING_VALUE IS NULL THEN 0
			ELSE
				ROUND(((QCC1.SumOfREPORTING_VALUE/TA.REPORTING_VALUE)*100),2)
		END
		FROM (SELECT dbo.Tab_Temp_Indicator.EVENT_ID, dbo.Tab_Temp_Indicator.SAMPLE_NUMBER,
dbo.Tab_Temp_Indicator.PARAMETER, dbo.Tab_Temp_Indicator.REPORTING_VALUE
				FROM dbo.Tab_Temp_Indicator
				WHERE dbo.Tab_Temp_Indicator.PARAMETER = 'TOTAL_ABUNDANCE') AS TA
		LEFT JOIN
			(SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER,
SUM(dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE) AS SumOfREPORTING_VALUE
				FROM dbo.TAB_FAMILY INNER JOIN dbo.Tab_Temp_Family
				ON dbo.TAB_FAMILY.TSN = dbo.Tab_Temp_Family.FAMILY_TSN
				WHERE dbo.TAB_FAMILY.HABITAT LIKE '%CB%'
				GROUP BY dbo.Tab_Temp_Family.EVENT_ID,
dbo.Tab_Temp_Family.SAMPLE_NUMBER) AS QCC1
		ON (TA.EVENT_ID = QCC1.EVENT_ID) AND (TA.SAMPLE_NUMBER = QCC1.SAMPLE_NUMBER)
	PRINT 'Step 39: Create Qry_Calc_Climb_2 - Completed Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 39: Create Qry_Calc_Climb_2 - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

-- Step 40: Create Qry_Calc_Filterers_2
BEGIN TRY
	BEGIN TRANSACTION
	PRINT 'Step 40: Create Qry_Calc_Filterers_2'
		INSERT INTO dbo.Tab_Temp_Indicator(EVENT_ID, SAMPLE_NUMBER, PARAMETER,
REPORTING_VALUE)
		SELECT TA.EVENT_ID, TA.SAMPLE_NUMBER, 'PCT_FILTERERS' AS REPORT,
ROUND(((QCF1.SumOfREPORTING_VALUE/TA.REPORTING_VALUE)*100), 2) AS PCT_FILTERERS
		FROM (SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER,
SUM(dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE) AS SumOfREPORTING_VALUE
				FROM dbo.TAB_FAMILY INNER JOIN dbo.Tab_Temp_Family
				ON dbo.TAB_FAMILY.TSN = dbo.Tab_Temp_Family.FAMILY_TSN
				WHERE dbo.TAB_FAMILY.GUILD = 'CF'
				GROUP BY dbo.Tab_Temp_Family.EVENT_ID,
dbo.Tab_Temp_Family.SAMPLE_NUMBER) AS QCF1
		INNER JOIN
			(SELECT dbo.Tab_Temp_Indicator.EVENT_ID, dbo.Tab_Temp_Indicator.SAMPLE_NUMBER,
dbo.Tab_Temp_Indicator.PARAMETER, dbo.Tab_Temp_Indicator.REPORTING_VALUE
				FROM dbo.Tab_Temp_Indicator
				WHERE dbo.Tab_Temp_Indicator.PARAMETER = 'TOTAL_ABUNDANCE') AS TA

		ON (QCF1.EVENT_ID = TA.EVENT_ID) AND (QCF1.SAMPLE_NUMBER = TA.SAMPLE_NUMBER)
```

```
	PRINT 'Step 40: Create Qry_Calc_Filterers_2 - Completed Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 40: Create Qry_Calc_Filterers_2 - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

--Step 41: Create Qry_Calc_Gather_2
BEGIN TRY
	BEGIN TRANSACTION
	PRINT 'Step 41: Create Qry_Calc_Gather_2'
		INSERT INTO dbo.Tab_Temp_Indicator(EVENT_ID, SAMPLE_NUMBER, PARAMETER,
REPORTING_VALUE)
		SELECT QCG1.EVENT_ID, QCG1.SAMPLE_NUMBER, 'PCT_GATHER' AS PAR,
ROUND(((QCG1.SumOfREPORTING_VALUE/TA.REPORTING_VALUE)*100),2) AS [VALUE]
		FROM (SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER,
SUM(dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE) AS SumOfREPORTING_VALUE
			FROM dbo.TAB_FAMILY
			INNER JOIN dbo.Tab_Temp_Family
			ON dbo.TAB_FAMILY.TSN = dbo.Tab_Temp_Family.FAMILY_TSN
			WHERE dbo.TAB_FAMILY.GUILD = 'CG'
			GROUP BY dbo.Tab_Temp_Family.EVENT_ID,
dbo.Tab_Temp_Family.SAMPLE_NUMBER) AS QCG1
		INNER JOIN
			(SELECT dbo.Tab_Temp_Indicator.EVENT_ID, dbo.Tab_Temp_Indicator.SAMPLE_NUMBER,
dbo.Tab_Temp_Indicator.PARAMETER, dbo.Tab_Temp_Indicator.REPORTING_VALUE
			FROM dbo.Tab_Temp_Indicator
			WHERE dbo.Tab_Temp_Indicator.PARAMETER = 'TOTAL_ABUNDANCE') AS TA
		ON (QCG1.EVENT_ID = TA.EVENT_id) AND (QCG1.SAMPLE_NUMBER = TA.SAMPLE_NUMBER)
	PRINT 'Step 41: Create Qry_Calc_Gather_2 - Completed Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 41: Create Qry_Calc_Gather_2 - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

--Step 42: Create Qry_Scraper_PCT_2
BEGIN TRY
	BEGIN TRANSACTION
	PRINT 'Step 42: Create Qry_Scraper_PCT_2'
	INSERT INTO dbo.Tab_Temp_Indicator (EVENT_ID, SAMPLE_NUMBER, PARAMETER, REPORTING_VALUE)
		SELECT TA.EVENT_ID, TA.SAMPLE_NUMBER, 'PCT_SCRAPER' AS [NAME],
ROUND(((QSP1.SumOfREPORTING_VALUE/TA.REPORTING_VALUE)*100),2) AS REPORT
		FROM (SELECT dbo.Tab_Temp_Indicator.EVENT_ID, dbo.Tab_Temp_Indicator.SAMPLE_NUMBER,
dbo.Tab_Temp_Indicator.PARAMETER, dbo.Tab_Temp_Indicator.REPORTING_VALUE
			FROM dbo.Tab_Temp_Indicator
			WHERE dbo.Tab_Temp_Indicator.PARAMETER = 'TOTAL_ABUNDANCE') AS TA
		INNER JOIN
			(SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER,
SUM(dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE) AS SumOfREPORTING_VALUE
			FROM dbo.Tab_Temp_Family
			INNER JOIN dbo.TAB_FAMILY
			ON dbo.Tab_Temp_Family.FAMILY_TSN = dbo.TAB_FAMILY.TSN
			WHERE dbo.TAB_FAMILY.GUILD = 'SC'
			GROUP BY dbo.Tab_Temp_Family.EVENT_ID,
dbo.Tab_Temp_Family.SAMPLE_NUMBER) AS QSP1
		ON (TA.EVENT_id = QSP1.EVENT_ID) AND (TA.SAMPLE_NUMBER = QSP1.SAMPLE_NUMBER)
	PRINT 'Step 42: Create Qry_Scraper_PCT_2 - Completed Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 42: Create Qry_Scraper_PCT_2 - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

--Step 43: Create Qry_Shred_2
BEGIN TRY
	BEGIN TRANSACTION
	PRINT 'Step 43: Create Qry_Shred_2'
		INSERT INTO dbo.Tab_Temp_Indicator(EVENT_ID, SAMPLE_NUMBER, PARAMETER,
REPORTING_VALUE)
		SELECT QSP1.EVENT_ID, QSP1.SAMPLE_NUMBER, 'PCT_SHREDDER' AS P,
ROUND(((QSP1.SumOfREPORTING_VALUE/TA.REPORTING_VALUE)*100),2) AS RESULT
		FROM (SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER,
SUM(dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE) AS SumOfREPORTING_VALUE
			FROM dbo.Tab_Temp_Family
			INNER JOIN dbo.TAB_FAMILY
			ON dbo.Tab_Temp_Family.FAMILY_TSN = dbo.TAB_FAMILY.TSN
			WHERE dbo.TAB_FAMILY.GUILD = 'SH'
```

```
                                        GROUP BY dbo.Tab_Temp_Family.EVENT_ID,
dbo.Tab_Temp_Family.SAMPLE_NUMBER) AS QSP1
                INNER JOIN
                                (SELECT dbo.Tab_Temp_Indicator.EVENT_ID, dbo.Tab_Temp_Indicator.SAMPLE_NUMBER,
dbo.Tab_Temp_Indicator.PARAMETER, dbo.Tab_Temp_Indicator.REPORTING_VALUE
                                        FROM dbo.Tab_Temp_Indicator
                                        WHERE dbo.Tab_Temp_Indicator.PARAMETER = 'TOTAL_ABUNDANCE') AS TA
                ON (QSP1.EVENT_ID = TA.EVENT_id) AND (QSP1.SAMPLE_NUMBER = TA.SAMPLE_NUMBER)
        PRINT 'Step 43: Create Qry_Shred_2 - Completed Successfully'
        COMMIT TRANSACTION
END TRY
BEGIN CATCH
        PRINT 'Transaction Failed for Step 43: Create Qry_Shred_2 - Will Rollback'
        IF @@TRANCOUNT > 0
        ROLLBACK

        EXECUTE usp_GetErrorInfo
END CATCH

--Step 44: Create Qry_Calc_Swim_2
BEGIN TRY
        BEGIN TRANSACTION
        PRINT 'Step 44: Create Qry_Calc_Swim_2'
                INSERT INTO dbo.Tab_Temp_Indicator (EVENT_ID, SAMPLE_NUMBER, PARAMETER,
REPORTING_VALUE)
                SELECT QCS1.EVENT_ID, QCS1.SAMPLE_NUMBER, 'PCT_SWIMMER' AS P,
ROUND(((QCS1.SumOfREPORTING_VALUE/TA.REPORTING_VALUE)*100),2) AS RESULT
                FROM (SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER,
SUM(dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE) AS SumOfREPORTING_VALUE
                        FROM dbo.TAB_FAMILY INNER JOIN dbo.Tab_Temp_Family
                        ON dbo.TAB_FAMILY.TSN = dbo.Tab_Temp_Family.FAMILY_TSN
                        WHERE dbo.TAB_FAMILY.HABITAT LIKE '%SW%'
                        GROUP BY dbo.Tab_Temp_Family.EVENT_ID,
dbo.Tab_Temp_Family.SAMPLE_NUMBER) AS QCS1
                INNER JOIN
                (SELECT dbo.Tab_Temp_Indicator.EVENT_ID, dbo.Tab_Temp_Indicator.SAMPLE_NUMBER,
dbo.Tab_Temp_Indicator.PARAMETER, dbo.Tab_Temp_Indicator.REPORTING_VALUE
                        FROM dbo.Tab_Temp_Indicator
                        WHERE dbo.Tab_Temp_Indicator.PARAMETER = 'TOTAL_ABUNDANCE') AS
TA
                ON (QCS1.EVENT_ID = TA.EVENT_id) AND (QCS1.SAMPLE_NUMBER = TA.SAMPLE_NUMBER)
        PRINT 'Step 44: Create Qry_Calc_Swim_2 - Completed Successfully'
        COMMIT TRANSACTION
END TRY
BEGIN CATCH
        PRINT 'Transaction Failed for Step 44: Create Qry_Calc_Swim_2 - Will Rollback'
        IF @@TRANCOUNT > 0
        ROLLBACK

        EXECUTE usp_GetErrorInfo
END CATCH

--Step 45: Create Qry_Calc_Cling_Cnt_2
BEGIN TRY
        BEGIN TRANSACTION
        PRINT 'Step 45: Create Qry_Calc_Cling_Cnt_2 - End of Calculating Other General Metrics'
                INSERT INTO dbo.Tab_Temp_Indicator(EVENT_ID, SAMPLE_NUMBER, PARAMETER,
REPORTING_VALUE)
                SELECT QCCC1.EVENT_ID, QCCC1.SAMPLE_NUMBER, 'PCT_CLINGER_TAXA' AS P,
ROUND(((QCCC1.CountSumOfREPORTING_VALUE/TR.REPORTING_VALUE)*100),2) AS RESULT
                FROM (SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER,
COUNT(dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE) AS CountSumOfREPORTING_VALUE
                        FROM dbo.TAB_FAMILY INNER JOIN dbo.Tab_Temp_Family
                        ON dbo.TAB_FAMILY.TSN = dbo.Tab_Temp_Family.FAMILy_TSN
                        WHERE dbo.TAB_FAMILY.HABITAT LIKE '%CN%'
                        GROUP BY dbo.Tab_Temp_Family.EVENT_ID,
dbo.Tab_Temp_Family.SAMPLE_NUMBER) AS QCCC1
                INNER JOIN
                                (SELECT dbo.Tab_Temp_Indicator.EVENT_ID, dbo.Tab_Temp_Indicator.SAMPLE_NUMBER,
dbo.Tab_Temp_Indicator.PARAMETER, dbo.Tab_Temp_Indicator.REPORTING_VALUE
                                        FROM dbo.Tab_Temp_Indicator
                                        WHERE dbo.Tab_Temp_Indicator.PARAMETER = 'TAXA_RICH') AS TR
                ON (QCCC1.EVENT_ID = TR.EVENT_ID) AND (QCCC1.SAMPLE_NUMBER = TR.SAMPLE_NUMBER)
        PRINT 'Step 45: Create Qry_Calc_Cling_Cnt_2 - Completed Successfully'
        COMMIT TRANSACTION
END TRY
BEGIN CATCH
        PRINT 'Transaction Failed for Step 45: Create Qry_Calc_Cling_Cnt_2 - Will Rollback'
        IF @@TRANCOUNT > 0
        ROLLBACK

        EXECUTE usp_GetErrorInfo
END CATCH

--Step 46: Create Qry_Chroninmidae_Pct
BEGIN TRY
        BEGIN TRANSACTION
        PRINT 'Step 46: Create Qry_Chroninmidae_Pct - Begin Calculating Other ...'
                INSERT INTO dbo.Tab_Temp_Indicator(EVENT_ID, SAMPLE_NUMBER, PARAMETER,
REPORTING_VALUE)
```

```
		SELECT QCP1.EVENT_ID, QCP1.SAMPLE_NUMBER, 'PCT_CHIRONOMIDAE' AS PARAMETER,
ROUND(((QCP1.SumOfREPORTING_VALUE/TA.REPORTING_VALUE)*100),2) AS [VALUE]
		FROM (SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER,
SUM(dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE) AS SumOfREPORTING_VALUE
			FROM dbo.Tab_Temp_Family INNER JOIN dbo.TAB_FAMILY ON
dbo.Tab_Temp_Family.FAMILY_TSN = dbo.TAB_FAMILY.TSN
			GROUP BY dbo.Tab_Temp_Family.EVENT_ID,
dbo.Tab_Temp_Family.SAMPLE_NUMBER, dbo.TAB_FAMILY.CHRONONMIDAE
			HAVING dbo.TAB_FAMILY.CHRONONMIDAE IS NOT NULL) AS QCP1
		INNER JOIN
			(SELECT dbo.Tab_Temp_Indicator.EVENT_ID, dbo.Tab_Temp_Indicator.SAMPLE_NUMBER,
dbo.Tab_Temp_Indicator.PARAMETER, dbo.Tab_Temp_Indicator.REPORTING_VALUE
			FROM dbo.Tab_Temp_Indicator
			WHERE dbo.Tab_Temp_Indicator.PARAMETER = 'TOTAL_ABUNDANCE') AS TA
		ON (QCP1.EVENT_ID = TA.EVENT_ID) AND (QCP1.SAMPLE_NUMBER = TA.SAMPLE_NUMBER)
		PRINT 'Step 46: Create Qry_Chroninmidae_Pct - Completed Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 46: Create Qry_Chroninmidae_Pct - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

--Step: 47 Create Qry_Pct_Corbicula
BEGIN TRY
	BEGIN TRANSACTION
	PRINT 'Step 47: Create Qry_Pct_Corbicula'
		INSERT INTO dbo.Tab_Temp_Indicator (EVENT_ID, SAMPLE_NUMBER, PARAMETER,
REPORTING_VALUE)
		SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER, 'PCT_CORBICULA'
AS P, ROUND(((dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE/TA.REPORTING_VALUE)*100),2) AS RESULT
		FROM dbo.Tab_Temp_Family
		INNER JOIN
		(SELECT dbo.Tab_Temp_Indicator.EVENT_ID, dbo.Tab_Temp_Indicator.SAMPLE_NUMBER,
dbo.Tab_Temp_Indicator.PARAMETER, dbo.Tab_Temp_Indicator.REPORTING_VALUE
			FROM dbo.Tab_Temp_Indicator
			WHERE dbo.Tab_Temp_Indicator.PARAMETER = 'TOTAL_ABUNDANCE') AS TA
		ON (dbo.Tab_Temp_Family.EVENT_ID = TA.EVENT_ID) AND (dbo.Tab_Temp_Family.SAMPLE_NUMBER =
TA.SAMPLE_NUMBER)
		WHERE dbo.Tab_Temp_Family.FAMILY_TSN = '0081381'
	PRINT 'Step 47: Create Qry_Pct_Corbicula - Completed Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 47: Create Qry_Pct_Corbicula - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

--Step 48: Create Qry_Net_Caddisfly_2
BEGIN TRY
	BEGIN TRANSACTION
	PRINT 'Step 48: Create Qry_Net_Caddisfly_2'
		INSERT INTO dbo.Tab_Temp_Indicator (EVENT_ID, SAMPLE_NUMBER, PARAMETER,
REPORTING_VALUE)
		SELECT TA.EVENT_ID, QC1.SAMPLE_NUMBER, 'PCT_NET_CADDISFLY' AS P,
ROUND(((QC1.SumOfREPORTING_VALUE/TA.REPORTING_VALUE)*100),2) AS RESULT
		FROM (SELECT dbo.Tab_Temp_Indicator.EVENT_ID, dbo.Tab_Temp_Indicator.SAMPLE_NUMBER,
dbo.Tab_Temp_Indicator.PARAMETER, dbo.Tab_Temp_Indicator.REPORTING_VALUE
			FROM dbo.Tab_Temp_Indicator
			WHERE dbo.Tab_Temp_Indicator.PARAMETER = 'TOTAL_ABUNDANCE') AS TA
		INNER JOIN
			(SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER,
SUM(dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE) AS SumOfREPORTING_VALUE
			FROM dbo.Tab_Temp_Family
			WHERE dbo.Tab_Temp_Family.FAMILY_TSN = '0115257' OR
dbo.Tab_Temp_Family.FAMILY_TSN = '0115398' OR dbo.Tab_Temp_Family.FAMILY_TSN = '0117043'
			GROUP BY dbo.Tab_Temp_Family.EVENT_ID,
dbo.Tab_Temp_Family.SAMPLE_NUMBER) AS QC1
		ON (TA.EVENT_id = QC1.EVENT_ID) AND (TA.SAMPLE_NUMBER = QC1.SAMPLE_NUMBER)
	PRINT 'Step 48: Create Qry_Net_Caddisfly_2 - Completed Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 48: Create Qry_Net_Caddisfly_2 - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

--Step 49: Create Qry_Pct_Caddis_No_Tolerant_2
BEGIN TRY
	BEGIN TRANSACTION
	PRINT 'Step 49: Create Qry_Pct_Caddis_No_Tolerant_2'
```

```
		INSERT INTO dbo.Tab_Temp_Indicator (EVENT_ID, SAMPLE_NUMBER, PARAMETER,
REPORTING_VALUE)
		SELECT QPCNT1.EVENT_ID, QPCNT1.SAMPLE_NUMBER, 'PCT_TRICHOPTERA_NO_TOL' AS P,
ROUND(((QPCNT1.SumOfREPORTING_VALUE/TA.REPORTING_VALUE)*100),2) AS RESULT
		FROM (SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER,
SUM(dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE) AS SumOfREPORTING_VALUE
			FROM dbo.Tab_Temp_Family
			INNER JOIN
			(SELECT dbo.Tab_Temp_Indicator.EVENT_ID,
dbo.Tab_Temp_Indicator.SAMPLE_NUMBER, dbo.Tab_Temp_Indicator.PARAMETER,
dbo.Tab_Temp_Indicator.REPORTING_VALUE
				FROM dbo.Tab_Temp_Indicator
				WHERE dbo.Tab_Temp_Indicator.PARAMETER = 'TOTAL_ABUNDANCE') AS
TA
			ON (dbo.Tab_Temp_Family.SAMPLE_NUMBER = TA.SAMPLE_NUMBER) AND
(dbo.Tab_Temp_Family.EVENT_ID = TA.EVENT_ID)
			INNER JOIN dbo.TAB_FAMILY
			ON dbo.Tab_Temp_Family.FAMILY_TSN = dbo.TAB_FAMILY.TSN
			WHERE dbo.TAB_FAMILY.EPTD = 'T' AND dbo.Tab_Temp_Family.FAMILY_TSN <>
'0115398'
			GROUP BY dbo.Tab_Temp_Family.EVENT_ID,
dbo.Tab_Temp_Family.SAMPLE_NUMBER) AS QPCNT1
		INNER JOIN
			(SELECT dbo.Tab_Temp_Indicator.EVENT_ID, dbo.Tab_Temp_Indicator.SAMPLE_NUMBER,
dbo.Tab_Temp_Indicator.PARAMETER, dbo.Tab_Temp_Indicator.REPORTING_VALUE
			FROM dbo.Tab_Temp_Indicator
			WHERE dbo.Tab_Temp_Indicator.PARAMETER = 'TOTAL_ABUNDANCE') AS TA
		ON (QPCNT1.SAMPLE_NUMBER = TA.SAMPLE_NUMBER) AND (QPCNT1.EVENT_ID = TA.EVENT_ID)
	PRINT 'Step 49: Create Qry_Pct_Caddis_No_Tolerant_2 - Completed Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 49: Create Qry_Pct_Caddis_No_Tolerant_2 - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

--Step 50: Create Qry_Pct_Amphipod_2
BEGIN TRY
	BEGIN TRANSACTION
	PRINT 'Step 50: Create Qry_Pct_Amphipod_2'
		INSERT INTO dbo.Tab_Temp_Indicator (EVENT_ID, SAMPLE_NUMBER, PARAMETER,
REPORTING_VALUE)
		SELECT TA.EVENT_ID, TA.SAMPLE_NUMBER, 'PCT_AMPHIPOD' AS P,
ROUND(((QPA1.SumOfREPORTING_VALUE/TA.REPORTING_VALUE)*100),2) AS RESULT
		FROM (SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER,
SUM(dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE) AS SumOfREPORTING_VALUE
			FROM dbo.Tab_Temp_Family INNER JOIN dbo.TAB_FAMILY
			ON dbo.Tab_Temp_Family.FAMILY_TSN = dbo.TAB_FAMILY.TSN
			WHERE dbo.TAB_FAMILY.[AMPHOPOD/ISOPOD] = 'A'
			GROUP BY dbo.Tab_Temp_Family.EVENT_ID,
dbo.Tab_Temp_Family.SAMPLE_NUMBER) AS QPA1
		INNER JOIN
			(SELECT dbo.Tab_Temp_Indicator.EVENT_ID, dbo.Tab_Temp_Indicator.SAMPLE_NUMBER,
dbo.Tab_Temp_Indicator.PARAMETER, dbo.Tab_Temp_Indicator.REPORTING_VALUE
			FROM dbo.Tab_Temp_Indicator
			WHERE dbo.Tab_Temp_Indicator.PARAMETER = 'TOTAL_ABUNDANCE') AS TA
		ON (QPA1.EVENT_ID = TA.EVENT_ID) AND (QPA1.SAMPLE_NUMBER = TA.SAMPLE_NUMBER)
	PRINT 'Step 50: Create Qry_Pct_Amphipod_2 - Completed Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 50: Create Qry_Pct_Amphipod_2 - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

--Step 51: Create Qry_Pct_Limestone_2
BEGIN TRY
	BEGIN TRANSACTION
	PRINT 'Step 51: Create Qry_Pct_Limestone_2'
		INSERT INTO dbo.Tab_Temp_Indicator (EVENT_ID, SAMPLE_NUMBER, PARAMETER,
REPORTING_VALUE)
		SELECT QPL1.EVENT_ID, QPL1.SAMPLE_NUMBER, 'PCT_LIMESTONE' AS P,
ROUND(((QPL1.SumOfREPORTING_VALUE/TA.REPORTING_VALUE)*100),2) AS RESULT
		FROM (SELECT dbo.Tab_Temp_Indicator.EVENT_ID, dbo.Tab_Temp_Indicator.SAMPLE_NUMBER,
dbo.Tab_Temp_Indicator.PARAMETER, dbo.Tab_Temp_Indicator.REPORTING_VALUE
			FROM dbo.Tab_Temp_Indicator
			WHERE dbo.Tab_Temp_Indicator.PARAMETER = 'TOTAL_ABUNDANCE') AS TA
		INNER JOIN
			(SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER,
SUM(dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE) AS SumOfREPORTING_VALUE
			FROM dbo.Tab_Temp_Family
			INNER JOIN dbo.TAB_FAMILY
			ON dbo.Tab_Temp_Family.FAMILY_TSN = dbo.TAB_FAMILY.TSN
			WHERE dbo.TAB_FAMILY.[AMPHOPOD/ISOPOD]IS NOT NULL
```

```
                                        GROUP BY dbo.Tab_Temp_Family.EVENT_ID,
dbo.Tab_Temp_Family.SAMPLE_NUMBER) AS QPL1
                ON (TA.EVENT_ID = QPL1.EVENT_ID) AND (TA.SAMPLE_NUMBER = QPL1.SAMPLE_NUMBER)
        PRINT 'Step 51: Create Qry_Pct_Limestone_2 - Completed Successfully'
        COMMIT TRANSACTION
END TRY
BEGIN CATCH
        PRINT 'Transaction Failed for Step 51: Create Qry_Pct_Limestone_2 - Will Rollback'
        IF @@TRANCOUNT > 0
        ROLLBACK

        EXECUTE usp_GetErrorInfo
END CATCH

--Step 52: Create Qry_Non_Insect_2
BEGIN TRY
        BEGIN TRANSACTION
        PRINT 'Step 52: Create Qry_Non_Insect_2'
                INSERT INTO dbo.Tab_Temp_Indicator (EVENT_ID, SAMPLE_NUMBER, PARAMETER,
REPORTING_VALUE)
                SELECT TA.EVENT_ID, TA.SAMPLE_NUMBER, 'PCT_NON_INSECT' AS P,
ROUND(((QNI1.SumOfREPORTING_VALUE/TA.REPORTING_VALUE)*100),2) AS RESULT
                FROM (SELECT dbo.Tab_Temp_Indicator.EVENT_ID, dbo.Tab_Temp_Indicator.SAMPLE_NUMBER,
dbo.Tab_Temp_Indicator.PARAMETER, dbo.Tab_Temp_Indicator.REPORTING_VALUE
                                        FROM dbo.Tab_Temp_Indicator
                                        WHERE dbo.Tab_Temp_Indicator.PARAMETER = 'TOTAL_ABUNDANCE') AS TA
                INNER JOIN
                        (SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER,
SUM(dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE) AS SumOfREPORTING_VALUE
                                        FROM dbo.Tab_Temp_Family INNER JOIN dbo.TAB_FAMILY ON
dbo.Tab_Temp_Family.FAMILY_TSN = dbo.TAB_FAMILY.TSN
                                        WHERE dbo.TAB_FAMILY.NON_INSECT = 'X'
                                        GROUP BY dbo.Tab_Temp_Family.EVENT_ID,
dbo.Tab_Temp_Family.SAMPLE_NUMBER) AS QNI1
                ON (TA.EVENT_ID = QNI1.EVENT_ID) AND (TA.SAMPLE_NUMBER = QNI1.SAMPLE_NUMBER)
        PRINT 'Step 52: Create Qry_Non_Insect_2 - Completed Successfully'
        COMMIT TRANSACTION
END TRY
BEGIN CATCH
        PRINT 'Transaction Failed for Step 52: Create Qry_Non_Insect_2 - Will Rollback'
        IF @@TRANCOUNT > 0
        ROLLBACK

        EXECUTE usp_GetErrorInfo
END CATCH

--Step 53: Create Qry_Oligo_2
BEGIN TRY
        BEGIN TRANSACTION
        PRINT 'Step 53: Create Qry_Oligo_2'
                INSERT INTO dbo.Tab_Temp_Indicator (EVENT_ID, SAMPLE_NUMBER, PARAMETER,
REPORTING_VALUE)
                SELECT QO1.EVENT_ID, QO1.SAMPLE_NUMBER, 'PCT_OLIGOCHAETA' AS P,
ROUND(((QO1.SumOfREPORTING_VALUE/TA.REPORTING_VALUE)*100),2) AS RESULT
                FROM (SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER,
SUM(dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE) AS SumOfREPORTING_VALUE
                                        FROM dbo.Tab_Temp_Family
                                        INNER JOIN dbo.TAB_FAMILY
                                        ON dbo.Tab_Temp_Family.FAMILY_TSN = dbo.TAB_FAMILY.TSN
                                        WHERE dbo.TAB_FAMILY.GOLD_TAXA = 'O'
                                        GROUP BY dbo.Tab_Temp_Family.EVENT_ID,
dbo.Tab_Temp_Family.SAMPLE_NUMBER) AS QO1
                INNER JOIN
                        (SELECT dbo.Tab_Temp_Indicator.EVENT_ID, dbo.Tab_Temp_Indicator.SAMPLE_NUMBER,
dbo.Tab_Temp_Indicator.PARAMETER, dbo.Tab_Temp_Indicator.REPORTING_VALUE
                                        FROM dbo.Tab_Temp_Indicator
                                        WHERE dbo.Tab_Temp_Indicator.PARAMETER = 'TOTAL_ABUNDANCE') AS TA
                ON (QO1.EVENT_ID = TA.EVENT_ID) AND (QO1.SAMPLE_NUMBER = TA.SAMPLE_NUMBER)
        PRINT 'Step 53: Create Qry_Oligo_2 - Completed Successfully'
        COMMIT TRANSACTION
END TRY
BEGIN CATCH
        PRINT 'Transaction Failed for Step 53: Create Qry_Oligo_2 - Will Rollback'
        IF @@TRANCOUNT > 0
        ROLLBACK

        EXECUTE usp_GetErrorInfo
END CATCH

--Step 54: Create Qry_Nco_Taxa_Cnt
BEGIN TRY
        BEGIN TRANSACTION
        PRINT 'Step 54: Create Qry_Nco_Taxa_Cnt'
                INSERT INTO dbo.Tab_Temp_Indicator (dbo.Tab_Temp_Indicator.EVENT_ID,
dbo.Tab_Temp_Indicator.SAMPLE_NUMBER, dbo.Tab_Temp_Indicator.PARAMETER,
                dbo.Tab_Temp_Indicator.REPORTING_VALUE)
                SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER, 'NCO_TAXA_CNT'
AS P, COUNT(dbo.Tab_Temp_Family.FAMILY_TSN) AS CountOfFamily_TSN
                FROM dbo.Tab_Temp_Family INNER JOIN dbo.TAB_FAMILY
                ON dbo.Tab_Temp_Family.FAMILY_TSN = dbo.TAB_FAMILY.TSN
```

```
		WHERE dbo.TAB_FAMILY.NCO IS NULL
		GROUP BY dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER
	PRINT 'Step 54: Create Qry_Nco_Taxa_Cnt - Completed Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 54: Create Qry_Nco_Taxa_Cnt - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

--Step 55: Create Qry_Non_Insect_Taxa_Cnt
BEGIN TRY
	BEGIN TRANSACTION
	PRINT 'Step 55: Create Qry_Non_Insect_Taxa_Cnt'
		INSERT INTO dbo.Tab_Temp_Indicator (EVENT_ID, SAMPLE_NUMBER, PARAMETER,
REPORTING_VALUE)
		SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER,
'NON_INSECT_TAXA_CNT' AS P, COUNT(dbo.Tab_Temp_Family.FAMILY_TSN) AS CountOfFamily_TSN
		FROM dbo.Tab_Temp_Family
		INNER JOIN dbo.TAB_FAMILY
		ON dbo.Tab_Temp_Family.FAMILY_TSN = dbo.TAB_FAMILY.TSN
		GROUP BY dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER,
dbo.TAB_FAMILY.NON_INSECT
		HAVING dbo.TAB_FAMILY.NON_INSECT = 'X'
	PRINT 'Step 55: Create Qry_Non_Insect_Taxa_Cnt - Completed Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 55: Create Qry_Non_Insect_Taxa_Cnt - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

--Step 56: Create Qry_Trichoptera_Taxa_Cnt_no_Hydro
BEGIN TRY
	BEGIN TRANSACTION
	PRINT 'Step 56: Create Qry_Trichoptera_Taxa_Cnt_no_Hydro'
		INSERT INTO dbo.Tab_Temp_Indicator (EVENT_ID, SAMPLE_NUMBER, PARAMETER,
REPORTING_VALUE)
		SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER,
'TRICHOPTERA_TAXA_COUNT_NO_HYDR' AS PARAMETER,
		COUNT(dbo.Tab_Temp_Family.FAMILY_TSN) AS CountOfFamily_TSN
		FROM dbo.TAB_EPTD_LOOK_UP INNER JOIN dbo.TAB_FAMILY
		ON dbo.TAB_EPTD_LOOK_UP.EPTD = dbo.TAB_FAMILY.EPTD
			INNER JOIN dbo.Tab_Temp_Family
			ON dbo.Tab_Temp_Family.FAMILY_TSN = dbo.TAB_FAMILY.TSN
		WHERE (dbo.TAB_EPTD_LOOK_UP.EPTD = 't') AND (dbo.Tab_Temp_Family.FAMILY_TSN <> '0115398')
		GROUP BY dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER
	PRINT 'Step 56: Create Qry_Trichoptera_Taxa_Cnt_no_Hydro - Completed Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 56: Create Qry_Trichoptera_Taxa_Cnt_no_Hydro - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

--Step 57: Create Qry_Burrower_Pct_2
BEGIN TRY
	BEGIN TRANSACTION
	PRINT 'Step 57: Create Qry_Burrower_Pct_2'
		INSERT INTO dbo.Tab_Temp_Indicator (EVENT_ID, SAMPLE_NUMBER, PARAMETER,
REPORTING_VALUE)
		SELECT TA.EVENT_id, TA.SAMPLE_NUMBER, 'PCT_BURROWER' AS PARAMETER, 'METRIC_VALUE' =
		CASE
			WHEN QBP1.SumOfREPORTING_VALUE < 1 THEN 0
		ELSE
			ROUND(((QBP1.SumOfREPORTING_VALUE/TA.REPORTING_VALUE)*100),2)
		END
		FROM (SELECT dbo.Tab_Temp_Indicator.EVENT_ID, dbo.Tab_Temp_Indicator.SAMPLE_NUMBER,
dbo.Tab_Temp_Indicator.PARAMETER, dbo.Tab_Temp_Indicator.REPORTING_VALUE
				FROM dbo.Tab_Temp_Indicator
				WHERE dbo.Tab_Temp_Indicator.PARAMETER = 'TOTAL_ABUNDANCE') AS TA
		INNER JOIN
			(SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER,
SUM(dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE) AS SumOfREPORTING_VALUE
				FROM dbo.Tab_Temp_Family INNER JOIN dbo.TAB_FAMILY
				ON dbo.Tab_Temp_Family.FAMILY_TSN = dbo.TAB_FAMILY.TSN
				WHERE dbo.TAB_FAMILY.HABITAT LIKE '%BU%'
				GROUP BY dbo.Tab_Temp_Family.EVENT_ID,
dbo.Tab_Temp_Family.SAMPLE_NUMBER) AS QBP1
		ON (TA.SAMPLE_NUMBER = QBP1.SAMPLE_NUMBER) AND (TA.EVENT_ID = QBP1.EVENT_ID)
	PRINT 'Step 57: Create Qry_Burrower_Pct_2 - Completed Successfully'
```

```
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 57: Create Qry_Burrower_Pct_2 - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

--Step 58: Create Qry_Pct_Predator_2
BEGIN TRY
	BEGIN TRANSACTION
	PRINT 'Step 58: Create Qry_Pct_Predator_2'
		INSERT INTO dbo.Tab_Temp_Indicator (EVENT_ID, SAMPLE_NUMBER, PARAMETER,
REPORTING_VALUE)
		SELECT TA.EVENT_ID, TA.SAMPLE_NUMBER, 'PCT_PREDATOR' AS PARAMETER, 'METRIC' =
		CASE
			WHEN QPP1.SumOfREPORTING_VALUE < 1 THEN 0
		ELSE
			ROUND(((QPP1.SumOfREPORTING_VALUE/TA.REPORTING_VALUE)*100),2)
		END
		FROM (SELECT dbo.Tab_Temp_Indicator.EVENT_ID, dbo.Tab_Temp_Indicator.SAMPLE_NUMBER,
dbo.Tab_Temp_Indicator.PARAMETER, dbo.Tab_Temp_Indicator.REPORTING_VALUE
				FROM dbo.Tab_Temp_Indicator
				WHERE dbo.Tab_Temp_Indicator.PARAMETER = 'TOTAL_ABUNDANCE') AS TA
		INNER JOIN
			(SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER,
SUM(dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE) AS SumOfREPORTING_VALUE
				FROM dbo.Tab_Temp_Family INNER JOIN dbo.TAB_FAMILY
				ON dbo.Tab_Temp_Family.FAMILY_TSN = dbo.TAB_FAMILY.TSN
				WHERE dbo.TAB_FAMILY.GUILD LIKE '%PR%'
				GROUP BY dbo.Tab_Temp_Family.EVENT_ID,
dbo.Tab_Temp_Family.SAMPLE_NUMBER) AS QPP1
		ON (TA.EVENT_ID = QPP1.EVENT_ID) AND (TA.SAMPLE_NUMBER = QPP1.SAMPLE_NUMBER)
	PRINT 'Step 58: Create Qry_Pct_Predator_2 - Completed Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 58: Create Qry_Pct_Predator_2 - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

--Step 59: Create Qry_Pct_Black_Fly
BEGIN TRY
	BEGIN TRANSACTION
	PRINT 'Step 59: Create Qry_Pct_Black_Fly'
		INSERT INTO dbo.Tab_Temp_Indicator (EVENT_ID, SAMPLE_NUMBER, PARAMETER,
REPORTING_VALUE)
		SELECT TA.EVENT_ID, TA.SAMPLE_NUMBER, 'PCT_SIMULIIDAE' AS PARAMETER, 'VAL' =
		CASE
			WHEN dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE < 1 THEN 0
		ELSE

	ROUND(((dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE/TA.REPORTING_VALUE)*100),2)
		END
		FROM dbo.Tab_Temp_Family
		INNER JOIN
		(SELECT dbo.Tab_Temp_Indicator.EVENT_ID, dbo.Tab_Temp_Indicator.SAMPLE_NUMBER,
dbo.Tab_Temp_Indicator.PARAMETER, dbo.Tab_Temp_Indicator.REPORTING_VALUE
				FROM dbo.Tab_Temp_Indicator
				WHERE dbo.Tab_Temp_Indicator.PARAMETER = 'TOTAL_ABUNDANCE') AS TA
		ON (dbo.Tab_Temp_Family.EVENT_ID = TA.EVENT_id) AND (dbo.Tab_Temp_Family.SAMPLE_NUMBER =
TA.SAMPLE_NUMBER)
		WHERE dbo.Tab_Temp_Family.FAMILY_TSN = '0126640'
	PRINT 'Step 59: Create Qry_Pct_Black_Fly - Completed Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 59: Create Qry_Pct_Black_Fly - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

--Step 60: Create Qry_Scraper_To_Collect_Filter
BEGIN TRY
	BEGIN TRANSACTION
	PRINT 'Step 60: Create Qry_Scraper_To_Collect_Filter'
	INSERT INTO dbo.Tab_Temp_Indicator (EVENT_ID, SAMPLE_NUMBER, PARAMETER, REPORTING_VALUE)
	SELECT QSP1.EVENT_ID, QSP1.SAMPLE_NUMBER, 'RATIO_SC_TO_CF' AS PARAMETER,
ROUND(QSP1.SumOfREPORTING_VALUE/QCF1.SumOfREPORTING_VALUE, 3) AS METRIC
	FROM (SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER,
SUM(dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE) AS SumOfREPORTING_VALUE
			FROM dbo.Tab_Temp_Family INNER JOIN dbo.TAB_FAMILY ON
dbo.Tab_Temp_Family.family_tsn=dbo.TAB_FAMILY.TSN
```

```
                        WHERE dbo.TAB_FAMILY.GUILD = 'SC'
                        GROUP BY dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER) AS
QSP1
        INNER JOIN
                (SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER,
SUM(dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE) AS SumOfREPORTING_VALUE
                        FROM dbo.TAB_FAMILY INNER JOIN dbo.Tab_Temp_Family ON
dbo.TAB_FAMILY.TSN=dbo.Tab_Temp_Family.family_tsn
                        WHERE dbo.TAB_FAMILY.GUILD = 'CF'
                        GROUP BY dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER) AS
QCF1
        ON (QSP1.EVENT_ID = QCF1.EVENT_ID) AND (QSP1.SAMPLE_NUMBER = QCF1.SAMPLE_NUMBER)
        PRINT 'Step 60: Create Qry_Scraper_To_Collect_Filter - Completed Successfully'
        COMMIT TRANSACTION
END TRY
BEGIN CATCH
        PRINT 'Transaction Failed for Step 60: Create Qry_Scraper_To_Collect_Filter - Will Rollback'
        IF @@TRANCOUNT > 0
        ROLLBACK

        EXECUTE usp_GetErrorInfo
END CATCH

--Step 61: Create Qry_Ratio_Shredder_To_Gatherer
BEGIN TRY
        BEGIN TRANSACTION
        PRINT 'Step 61: Create Qry_Ratio_Shredder_To_Gatherer'
                INSERT INTO dbo.Tab_Temp_Indicator (EVENT_ID, SAMPLE_NUMBER, PARAMETER,
REPORTING_VALUE)
                SELECT QSP1.EVENT_ID, QSP1.SAMPLE_NUMBER, 'RATIO_SH_TO_CG' AS PAREMTER,
ROUND(QSP1.SumOfREPORTING_VALUE/QCG1.SumOfREPORTING_VALUE,4) AS M
                FROM (SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER,
SUM(dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE) AS SumOfREPORTING_VALUE
                        FROM dbo.Tab_Temp_Family INNER JOIN dbo.TAB_FAMILY
                        ON dbo.Tab_Temp_Family.FAMILY_TSN = dbo.TAB_FAMILY.TSN
                        WHERE dbo.TAB_FAMILY.GUILD = 'SH'
                        GROUP BY dbo.Tab_Temp_Family.EVENT_ID,
dbo.Tab_Temp_Family.SAMPLE_NUMBER) AS QSP1
                INNER JOIN
                        (SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER,
SUM(dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE) AS SumOfREPORTING_VALUE
                        FROM dbo.TAB_FAMILY INNER JOIN dbo.Tab_Temp_Family
                        ON dbo.TAB_FAMILY.TSN = dbo.Tab_Temp_Family.FAMILY_TSN
                        WHERE dbo.TAB_FAMILY.GUILD = 'CG'
                        GROUP BY dbo.Tab_Temp_Family.EVENT_ID,
dbo.Tab_Temp_Family.SAMPLE_NUMBER) AS QCG1
                ON (QSP1.EVENT_ID = QCG1.EVENT_ID) AND (QSP1.SAMPLE_NUMBER = QCG1.SAMPLE_NUMBER)
        PRINT 'Step 61: Create Qry_Ratio_Shredder_To_Gatherer - Completed Successfully'
        COMMIT TRANSACTION
END TRY
BEGIN CATCH
        PRINT 'Transaction Failed for Step 61: Create Qry_Ratio_Shredder_To_Gatherer - Will Rollback'
        IF @@TRANCOUNT > 0
        ROLLBACK

        EXECUTE usp_GetErrorInfo
END CATCH

--Step 62: Create Qry_Ratio_Scrape_To_Shred
BEGIN TRY
        BEGIN TRANSACTION
        PRINT 'Step 62: Create Qry_Ratio_Scrape_To_Shred'
                INSERT INTO dbo.Tab_Temp_Indicator (EVENT_ID, SAMPLE_NUMBER, PARAMETER,
REPORTING_VALUE)
                SELECT QSP1.EVENT_ID, QSP1.SAMPLE_NUMBER, 'RATIO_SC_TO_SH' AS P,
ROUND(QSP1.EVENT_ID/QSHP1.SumOfREPORTING_VALUE,4) AS C
                FROM (SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER,
SUM(dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE) AS SumOfREPORTING_VALUE
                        FROM dbo.Tab_Temp_Family INNER JOIN dbo.TAB_FAMILY
                        ON dbo.Tab_Temp_Family.family_tsn=dbo.TAB_FAMILY.TSN
                        WHERE dbo.TAB_FAMILY.GUILD = 'SC'
                        GROUP BY dbo.Tab_Temp_Family.EVENT_ID,
dbo.Tab_Temp_Family.SAMPLE_NUMBER) AS QSP1
                INNER JOIN
                        (SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER,
SUM(dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE) AS SumOfREPORTING_VALUE
                        FROM dbo.Tab_Temp_Family INNER JOIN dbo.TAB_FAMILY
                        ON dbo.Tab_Temp_Family.FAMILY_TSN = dbo.TAB_FAMILY.TSN
                        WHERE dbo.TAB_FAMILY.GUILD = 'SH'
                        GROUP BY dbo.Tab_Temp_Family.EVENT_ID,
dbo.Tab_Temp_Family.SAMPLE_NUMBER) AS QSHP1
                ON (QSP1.EVENT_ID=QSHP1.EVENT_ID) AND (QSP1.SAMPLE_NUMBER=QSHP1.SAMPLE_NUMBER)
        PRINT 'Step 62: Create Qry_Ratio_Scrape_To_Shred - Completed Successfully'
        COMMIT TRANSACTION
END TRY
BEGIN CATCH
        PRINT 'Transaction Failed for Step 62: Create Qry_Ratio_Scrape_To_Shred - Will Rollback'
        IF @@TRANCOUNT > 0
        ROLLBACK
```

```
	EXECUTE usp_GetErrorInfo
END CATCH

----Step 63: REPLICATE "temp_indicator_raw" including "Decimal Fix"
BEGIN TRY
	BEGIN TRANSACTION
	PRINT 'Step 63: REPLICATE "Tab_Temp_Indicator_Raw" including "Decimal Fix"'
		CREATE TABLE dbo.Tab_Temp_Indicator_Raw (EVENT_ID INT NULL, IBI_PARAMETER Varchar(150)
NULL, CALCULATED_VALUE REAL NULL, CALCULATED_UNITS Varchar(50))

		INSERT INTO  dbo.Tab_Temp_Indicator_Raw(EVENT_ID, IBI_PARAMETER, CALCULATED_VALUE,
CALCULATED_UNITS)
		SELECT dbo.Tab_Temp_Indicator.EVENT_ID, dbo.Tab_Temp_Indicator.PARAMETER,
AVG(dbo.Tab_Temp_Indicator.REPORTING_VALUE), TAB_WORKING_FIX_AVG.UNITS
		FROM dbo.Tab_Temp_Indicator INNER JOIN TAB_WORKING_FIX_AVG
		ON dbo.Tab_Temp_Indicator.PARAMETER = TAB_WORKING_FIX_AVG.PARAMETER
		GROUP BY dbo.Tab_Temp_Indicator.EVENT_ID, dbo.Tab_Temp_Indicator.PARAMETER,
TAB_WORKING_FIX_AVG.UNITS

		UPDATE dbo.Tab_Temp_Indicator_Raw
		SET dbo.Tab_Temp_Indicator_Raw.CALCULATED_VALUE =
		(
			CASE
				WHEN dbo.Tab_Temp_Indicator_Raw.CALCULATED_UNITS = 'COUNT' THEN
ROUND(dbo.Tab_Temp_Indicator_Raw.CALCULATED_VALUE, 1)
				WHEN dbo.Tab_Temp_Indicator_Raw.CALCULATED_UNITS = 'PERCENT' THEN
ROUND(dbo.Tab_Temp_Indicator_Raw.CALCULATED_VALUE, 2)
				WHEN dbo.Tab_Temp_Indicator_Raw.CALCULATED_UNITS = 'UNITLESS' AND
dbo.Tab_Temp_Indicator_Raw.IBI_PARAMETER = 'FBI' THEN
				ROUND(dbo.Tab_Temp_Indicator_Raw.CALCULATED_VALUE, 2)
				WHEN dbo.Tab_Temp_Indicator_Raw.CALCULATED_UNITS = 'UNITLESS' THEN
ROUND(dbo.Tab_Temp_Indicator_Raw.CALCULATED_VALUE, 4)

			ELSE
				ROUND(dbo.Tab_Temp_Indicator_Raw.CALCULATED_VALUE, 0)
			END
		)
	PRINT 'Step 63: REPLICATE "Tab_Temp_Indicator_Raw" including "Decimal Fix" - Completed Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 63: REPLICATE "Tab_Temp_Indicator_Raw" including "Decimal Fix" - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

--Step 64: Delete from TAB_WEARHOUSE where the Tab_Temp_Indicator_Raw Event_Id Exist in TAB_WEARHOUSE
BEGIN TRY
	BEGIN TRANSACTION

	PRINT 'Step 64: Delete from TAB_WEARHOUSE where the Tab_Temp_Indicator_Raw Event_Id Exist in
TAB_WEARHOUSE'

	DELETE TAB_WEARHOUSE
	WHERE EXISTS(SELECT EVENT_ID
				FROM dbo.Tab_Temp_Indicator_Raw
				WHERE TAB_WEARHOUSE.EVENT_ID = dbo.Tab_Temp_Indicator_Raw.EVENT_ID
AND TAB_WEARHOUSE.IBI_PARAMETER = dbo.Tab_Temp_Indicator_Raw.IBI_PARAMETER)

	PRINT 'Step 64: Delete from TAB_WEARHOUSE where the Tab_Temp_Indicator_Raw Event_Id Exist in
TAB_WEARHOUSE - Completed Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 64: Delete from TAB_WEARHOUSE where the Tab_Temp_Indicator_Raw Event_Id
Exist in TAB_WEARHOUSE - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

--Step 65: Insert/Append Into TAB_WEARHOUSE From Tab_Temp_Indicator_Raw
BEGIN TRY
	BEGIN TRANSACTION

	PRINT 'Step 65: Insert Into TAB_WEARHOUSE From Tab_Temp_Indicator_Raw'

	INSERT INTO TAB_WEARHOUSE (EVENT_ID, IBI_PARAMETER, CALCULATED_VALUE, CALCULATED_UNITS)
	SELECT EVENT_ID, IBI_PARAMETER, CALCULATED_VALUE, CALCULATED_UNITS
	FROM dbo.Tab_Temp_Indicator_Raw

	PRINT 'Step 65: Insert/Append Into TAB_WEARHOUSE From Tab_Temp_Indicator_Raw - Completed Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 65: Insert/Append Into TAB_WEARHOUSE From Tab_Temp_Indicator_Raw - Will
Rollback'
```

```
        IF @@TRANCOUNT > 0
        ROLLBACK

        EXECUTE usp_GetErrorInfo
END CATCH

GO
```

APPENDIX C. THE RARIFIEDMETRICS STORED PROCEDURE

```
USE [Non_Tidal_Benthic]
GO

/****** Object:  StoredProcedure [dbo].[usp_RarifiedMetrics]    Script Date: 04/22/2013 12:02:40 ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

-- =============================================
-- Author:                  Andre Stratton - Vistronix
-- Create date:             May 6, 2011
-- Description:             Stored Procedure that is run to verify Rareified Data
-- Modified Date:     January 23, 2012
-- =============================================
CREATE PROCEDURE [dbo].[usp_RarifiedMetrics]
        -- Add the parameters for the stored procedure here
        @year VARCHAR(8) = NULL
AS

        IF @year IS NULL
        BEGIN
                PRINT 'ERROR: You must specifiy a year in this format: usp_MainIndicator @year = %1994% with %1994%
the encapsulated in single quotes.'
                RETURN
        END

BEGIN TRY
        BEGIN TRANSACTION
-- SET NOCOUNT ON added to prevent extra result sets from
-- interfering with SELECT statements.
        SET NOCOUNT ON;

        DROP TABLE dbo.Tab_Temp_Ept_Part_1
        DROP TABLE dbo.Tab_Temp_Event
        DROP TABLE dbo.Tab_Temp_Family
        DROP TABLE dbo.Tab_Temp_Family_2
        DROP TABLE dbo.Tab_Temp_Indicator
        DROP TABLE dbo.Tab_Temp_Indicator_Raw
        DROP TABLE dbo.Tab_Temp_Taxa_Data
        DROP TABLE dbo.Temp_DOM_2
        DROP TABLE dbo.Temp_DOM_3
        DROP TABLE dbo.Temp_PCT_DOM_MULTI
        DROP TABLE dbo.Temp_SW

        SET ANSI_Warnings ON

        PRINT 'ALL Temp Tables Were Dropped'
        COMMIT TRANSACTION
END TRY
BEGIN CATCH
        PRINT 'Transaction Failed ON DROP TABLE Statement - Will Rollback '
        IF @@TRANCOUNT > 0
        ROLLBACK

        EXECUTE usp_GetErrorInfo
END CATCH

--Step 1: Create Qry_Draw_Event_Data
BEGIN TRY
        BEGIN TRANSACTION

        PRINT 'Step 1: Create Qry_Draw_Event_Data - Begin of Calculating Metrics'

        SELECT dbo.TAB_EVENT.EVENT_ID, dbo.TAB_EVENT.PROJECT_ID, TAB_ECO_REGIONS.CBP_IBI_REGION,
dbo.TAB_EVENT.STATION_ID,DATENAME(yyyy, dbo.TAB_EVENT.SAMPLE_DATE) AS Yr,
        dbo.TAB_EVENT.SAMPLE_DATE, dbo.TAB_EVENT.SAMPLE_TIME, dbo.TAB_EVENT.R_DATE,
dbo.TAB_EVENT.EVENT_TYPE, TAB_ECO_REGIONS.CBP_IBI_REGION_NAME,dbo.TAB_EVENT.EVENT_LATITUDE,
        dbo.TAB_EVENT.EVENT_LONGITUDE, dbo.TAB_STATIONS.UTM_X, dbo.TAB_STATIONS.UTM_Y,
dbo.TAB_EVENT.SITE_TYPE_CODE
        INTO dbo.Tab_Temp_Event
        FROM dbo.TAB_ECO_REGIONS INNER JOIN dbo.TAB_STATIONS
                ON dbo.TAB_ECO_REGIONS.SUBECOREGION_LEVEL4 = dbo.TAB_STATIONS.ECOREGION_LEVEL_4
                        INNER JOIN (dbo.TAB_EVENT INNER JOIN dbo.TAB_EVENT_TYPE
                                                ON dbo.TAB_EVENT.EVENT_TYPE =
dbo.TAB_EVENT_TYPE.EVENT_TYPE_ID)
                ON dbo.TAB_STATIONS.STATION_ID = dbo.TAB_EVENT.STATION_ID
        WHERE dbo.TAB_EVENT.SAMPLE_DATE LIKE '%' +  @year + '%'
        ORDER BY dbo.TAB_EVENT.EVENT_ID

        PRINT 'Step 1: Create Qry_Draw_Event_Data - Completed Successfully'
        COMMIT TRANSACTION
END TRY
BEGIN CATCH
        PRINT 'Transaction Failed for Step 1: Create Qry_Draw_Event_Data - Will Rollback'
        IF @@TRANCOUNT > 0
        ROLLBACK

        EXECUTE usp_GetErrorInfo
```

```
END CATCH

--Step 2: Create Qry_Draw_Taxa_Data
BEGIN TRY
	BEGIN TRANSACTION

	PRINT 'Step 2: Create Qry_Draw_Taxa_Data'

	SELECT dbo.Tab_Temp_Event.EVENT_ID, dbo.Tab_Temp_Event.CBP_IBI_REGION_NAME,
dbo.Tab_Temp_Event.STATION_ID, dbo.Tab_Temp_Event.SAMPLE_DATE,
	dbo.TAB_RAREFACTION_COUNTS.SAMPLE_NUMBER, dbo.TAB_RAREFACTION_COUNTS.TSN,
dbo.TAB_RAREFACTION_COUNTS.LIFE_STAGE_CODE,
	dbo.TAB_RAREFACTION_COUNTS.REPORTING_PARAMETER,dbo.TAB_RAREFACTION_COUNTS.REPORTING_V
ALUE, dbo.TAB_RAREFACTION_COUNTS.REPORTING_UNITS,
	dbo.TAB_RAREFACTION_COUNTS.BIO_METHOD, dbo.TAB_RAREFACTION_COUNTS.G_METHOD,
dbo.TAB_RAREFACTION_COUNTS.SAMPLE_TYPE
	INTO dbo.Tab_Temp_Taxa_Data
	FROM dbo.Tab_Temp_Event
		INNER JOIN dbo.TAB_RAREFACTION_COUNTS
		ON dbo.Tab_Temp_Event.EVENT_ID = dbo.TAB_RAREFACTION_COUNTS.EVENT_ID;

	PRINT 'Step 2: Create Qry_Draw_Taxa_Data - Completed Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 2: Create Qry_Draw_Taxa_Data - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

--Step 3: Create Qry_Delete_Bad_Method_1
BEGIN TRY
	BEGIN TRANSACTION

	PRINT 'Step 3: Create Qry_Delete_Bad_Method_1'

	UPDATE dbo.Tab_Temp_Taxa_Data
	SET dbo.Tab_Temp_Taxa_Data.CBP_IBI_REGION_NAME = 'delete'
	FROM dbo.Tab_Temp_Taxa_Data
		INNER JOIN TAB_EXCLUSION_METHODS
		ON dbo.Tab_Temp_Taxa_Data.Bio_Method = TAB_EXCLUSION_METHODS.Bio_Method

	PRINT 'Step 3: Create Qry_Delete_Bad_Method_1 - Completed Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 3: Create Qry_Delete_Bad_Method_1 - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

--Step 4: Create Qry_Delete_Bad_Method_2
BEGIN TRY
	BEGIN TRANSACTION

	PRINT 'Step 4: Create Qry_Delete_Bad_Method_2'

	DELETE
	FROM dbo.Tab_Temp_Taxa_Data
	WHERE dbo.Tab_Temp_Taxa_Data.CBP_IBI_REGION_NAME = 'delete'

	PRINT 'Step 4: Create Qry_Delete_Bad_Method_2 - Completed Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 4: Create Qry_Delete_Bad_Method_2 - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

--Step 5: Create Qry_Sum_To_Family
BEGIN TRY
	BEGIN TRANSACTION

	PRINT 'Step 5: Create Qry_Sum_To_Family'

	SELECT
dbo.Tab_Temp_Taxa_Data.EVENT_ID,dbo.Tab_Temp_Taxa_Data.SAMPLE_NUMBER,dbo.TAB_CONVERT_TAXA.FAMILY_TSN
, dbo.TAB_CONVERT_TAXA.FAMILY,
	SUM(dbo.Tab_Temp_Taxa_Data.REPORTING_VALUE) AS SUM_OF_REPORTING_VALUE,
dbo.TAB_FAMILY.EXCLUSION_TAXA
	INTO dbo.Tab_Temp_Family
	FROM dbo.TAB_CONVERT_TAXA INNER JOIN dbo.Tab_Temp_Taxa_Data
		ON dbo.TAB_CONVERT_TAXA.TSN = dbo.Tab_Temp_Taxa_Data.TSN INNER JOIN dbo.TAB_FAMILY
```

```
                ON dbo.TAB_CONVERT_TAXA.FAMILY_TSN = dbo.TAB_FAMILY.TSN
        GROUP BY dbo.Tab_Temp_Taxa_Data.EVENT_ID, dbo.Tab_Temp_Taxa_Data.SAMPLE_NUMBER,
dbo.TAB_CONVERT_TAXA.FAMILY_TSN, dbo.TAB_CONVERT_TAXA.FAMILY,
        dbo.TAB_FAMILY.EXCLUSION_TAXA;

        PRINT 'Step 5: Create Qry_Sum_To_Family - Completed Successfully'
        COMMIT TRANSACTION
END TRY
BEGIN CATCH
        PRINT 'Transaction Failed for Step 5: Create Qry_Sum_To_Family - Will Rollback'
        IF @@TRANCOUNT > 0
        ROLLBACK

        EXECUTE usp_GetErrorInfo
END CATCH

--Step 6: Create Qry_Delete_Exclusion_Taxa
BEGIN TRY
        BEGIN TRANSACTION

        PRINT 'Step 6: Create Qry_Delete_Exclusion_Taxa'

        DELETE
        FROM dbo.Tab_Temp_Family
        WHERE dbo.Tab_Temp_Family.EXCLUSION_TAXA IS NOT NULL

        PRINT 'Step 6: Create Qry_Delete_Exclusion_Taxa - Completed Successfully'
        COMMIT TRANSACTION
END TRY
BEGIN CATCH
        PRINT 'Transaction Failed for Step 6: Create Qry_Delete_Exclusion_Taxa - Will Rollback'
        IF @@TRANCOUNT > 0
        ROLLBACK

        EXECUTE usp_GetErrorInfo
END CATCH

--Step 7: Create Qry_Make_Indicator_Table and Qry_Compute_Taxa_Rich
BEGIN TRY
        BEGIN TRANSACTION

        PRINT 'Step 7: Create Qry_Make_Indicator_Table and Qry_Compute_Taxa_Rich'

        CREATE TABLE dbo.Tab_Temp_Indicator (EVENT_ID INT NULL, SAMPLE_NUMBER INT NULL, PARAMETER
Varchar(150) NULL, REPORTING_VALUE REAL NULL)

        INSERT INTO dbo.Tab_Temp_Indicator(EVENT_ID, SAMPLE_NUMBER, PARAMETER, REPORTING_VALUE)
        SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER, 'TAXA_RICH' AS
PARAMETER, COUNT(dbo.Tab_Temp_Family.FAMILY_TSN) AS REPORTING_VALUE
        FROM dbo.Tab_Temp_Family
        GROUP BY dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER

        PRINT 'Step 7: Create Qry_Make_Indicator_Table and Qry_Compute_Taxa_Rich - Completed Successfully'
        COMMIT TRANSACTION
END TRY
BEGIN CATCH
        PRINT 'Transaction Failed for Step 7: Create Qry_Make_Indicator_Table and Qry_Compute_Taxa_Rich - Will Rollback'
        IF @@TRANCOUNT > 0
        ROLLBACK

        EXECUTE usp_GetErrorInfo
END CATCH

--Step 8: Create Qry_Total_Abundance
BEGIN TRY
        BEGIN TRANSACTION

        PRINT 'Step 8: Create Qry_Total_Abundance'

        INSERT INTO dbo.Tab_Temp_Indicator(EVENT_ID, SAMPLE_NUMBER, PARAMETER, REPORTING_VALUE)
        SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER, 'TOTAL_ABUNDANCE' AS
PARAMETER,
        SUM(dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE) AS SUM_OF_REPORTING_VALUE
        FROM dbo.Tab_Temp_Family
        GROUP BY dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER

        PRINT 'Step 8: Create Qry_Total_Abundance - Completed Successfully'
        COMMIT TRANSACTION
END TRY
BEGIN CATCH
        PRINT 'Transaction Failed for Step 8: Create Qry_Total_Abundance - Will Rollback'
        IF @@TRANCOUNT > 0
        ROLLBACK

        EXECUTE usp_GetErrorInfo
END CATCH

--Step 9: Create Qry_EPT_TAXA_ONLY_CNT
BEGIN TRY
        BEGIN TRANSACTION
```

```
	PRINT 'Step 9: Create Qry_EPT_TAXA_ONLY_CNT'

	INSERT INTO dbo.Tab_Temp_Indicator(EVENT_ID, SAMPLE_NUMBER, PARAMETER, REPORTING_VALUE)
	SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER, 'EPT_TAXA_COUNT' AS
PARAMETER, COUNT(dbo.Tab_Temp_Family.FAMILY_TSN) AS COUNT_OF_FAMILY_TSN
	FROM dbo.TAB_EPTD_LOOK_UP INNER JOIN dbo.TAB_FAMILY
	ON dbo.TAB_EPTD_LOOK_UP.EPTD = dbo.TAB_FAMILY.EPTD
		INNER JOIN dbo.Tab_Temp_Family
		ON dbo.TAB_FAMILY.TSN = dbo.Tab_Temp_Family.FAMILY_TSN
	WHERE dbo.TAB_EPTD_LOOK_UP.EPTD <> 'D'
	GROUP BY dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER

	PRINT 'Step 9: Create Qry_EPT_TAXA_ONLY_CNT - Completed Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 9: Create Qry_EPT_TAXA_ONLY_CNT - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

--Step 10: Create Qry_EPTD_TAXA_CNT
BEGIN TRY
	BEGIN TRANSACTION

	PRINT 'Step 10: Create Qry_EPTD_TAXA_CNT Part 1'

	SELECT dbo.TAB_FAMILY.TSN, dbo.TAB_FAMILY.Family, dbo.TAB_FAMILY.TOLARANCE_VALUE,
dbo.TAB_FAMILY.GUILD, dbo.TAB_FAMILY.EPTD,
	dbo.TAB_EPTD_LOOK_UP.EPTD_PARAMETER + '_TAXA_CNT' AS PARAMETER
	INTO dbo.Tab_Temp_Ept_Part_1
	FROM dbo.TAB_FAMILY INNER JOIN dbo.TAB_EPTD_LOOK_UP
	ON dbo.TAB_FAMILY.EPTD = dbo.TAB_EPTD_LOOK_UP.EPTD
	WHERE dbo.TAB_FAMILY.EPTD IS NOT NULL

	PRINT 'Step 10: Create Qry_EPTD_TAXA_CNT Part 1 - Completed Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 10: Create Qry_EPTD_TAXA_CNT Part 1 - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH
BEGIN TRY
	BEGIN TRANSACTION

	PRINT 'Step 10: Create Qry_EPTD_TAXA_CNT Part 2'

	INSERT INTO dbo.Tab_Temp_Indicator(EVENT_ID, SAMPLE_NUMBER, PARAMETER, REPORTING_VALUE)
	SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER,
dbo.Tab_Temp_Ept_Part_1.PARAMETER, COUNT(dbo.Tab_Temp_Family.family_tsn) AS COUNT_OF_FAMILY_TSN
	FROM dbo.Tab_Temp_Family LEFT JOIN dbo.Tab_Temp_Ept_Part_1
	ON dbo.Tab_Temp_Family.family_tsn = dbo.Tab_Temp_Ept_Part_1.TSN
	GROUP BY dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER, PARAMETER

	PRINT 'Step 10: Create Qry_EPTD_TAXA_CNT Part 2 - Completed Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 10: Create Qry_EPTD_TAXA_CNT Part 2 - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

--Step 11: Create Qry_TAXA_ONLY_ABUND
BEGIN TRY
	BEGIN TRANSACTION

	PRINT 'Step 11: Create Qry_TAXA_ONLY_ABUND'

	INSERT INTO dbo.Tab_Temp_Indicator ( EVENT_ID, SAMPLE_NUMBER, PARAMETER, REPORTING_VALUE)
	SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER, 'EPT_TAXA_ABUND' AS
PARAMETER,
	SUM(dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE) AS SUM_OF_REPORTING_VALUE
	FROM dbo.TAB_EPTD_LOOK_UP INNER JOIN dbo.TAB_FAMILY
	ON dbo.TAB_EPTD_LOOK_UP.EPTD = dbo.TAB_FAMILY.EPTD
		INNER JOIN dbo.Tab_Temp_Family
		ON dbo.TAB_FAMILY.TSN = dbo.Tab_Temp_Family.FAMIlY_TSN
	WHERE dbo.TAB_EPTD_LOOK_UP.EPTD <> 'D'
	GROUP BY dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER

	PRINT 'Step 11: Create Qry_TAXA_ONLY_ABUND - Completed Successfully'
	COMMIT TRANSACTION
```

```
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 11: Create Qry_TAXA_ONLY_ABUND - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

--Step 12: Create Qry_CALC_PCT_EPT
BEGIN TRY
	BEGIN TRANSACTION

	PRINT 'Step 12: Create Qry_CALC_PCT_EPT'

	INSERT INTO dbo.Tab_Temp_Indicator(EVENT_ID, SAMPLE_NUMBER, PARAMETER, REPORTING_VALUE )
	SELECT EA.EVENT_ID, EA.SAMPLE_NUMBER, 'PCT_EPT' AS PARAMETER,
ROUND(((EA.REPORTING_VALUE/TA.REPORTING_VALUE)*100), 2) AS [VALUE]
	FROM (SELECT EVENT_ID, SAMPLE_NUMBER, PARAMETER, REPORTING_VALUE
			FROM dbo.Tab_Temp_Indicator
			WHERE PARAMETER = 'EPT_TAXA_ABUND') AS EA
	INNER JOIN
		(SELECT EVENT_ID, SAMPLE_NUMBER, PARAMETER, REPORTING_VALUE
			FROM dbo.Tab_Temp_Indicator
			WHERE PARAMETER = 'TOTAL_ABUNDANCE') AS TA
	ON (EA.EVENT_ID = TA.EVENT_ID) AND (EA.SAMPLE_NUMBER = TA.SAMPLE_NUMBER)

	PRINT 'Step 12: Create Qry_CALC_PCT_EPT - Completed Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 12: Create Qry_CALC_PCT_EPT - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

--Step 13: Create Qry_EPTD_TAXA_PCT_2
BEGIN TRY
	BEGIN TRANSACTION

	PRINT 'Step 13: Create Qry_EPTD_TAXA_PCT_2'

	INSERT INTO dbo.Tab_Temp_Indicator(EVENT_ID, SAMPLE_NUMBER, PARAMETER, REPORTING_VALUE)
	SELECT TA.EVENT_ID, ET.SAMPLE_NUMBER, 'PCT_' +  ET.EPTD_PARAMETER AS P,
ROUND(((ET.REPORTING_VALUE/TA.REPORTING_VALUE)*100),2) AS RESULT
	FROM (SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER,
dbo.TAB_EPTD_LOOK_UP.EPTD_PARAMETER,
	SUM(dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE) AS REPORTING_VALUE
			FROM dbo.TAB_EPTD_LOOK_UP INNER JOIN dbo.TAB_FAMILY
			ON dbo.TAB_EPTD_LOOK_UP.EPTD = dbo.TAB_FAMILY.EPTD
				INNER JOIN dbo.Tab_Temp_Family
				ON dbo.TAB_FAMILY.TSN = dbo.Tab_Temp_Family.FAMILY_TSN
				WHERE dbo.TAB_EPTD_LOOK_UP.EPTD IS NOT NULL
			GROUP BY dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER,
dbo.TAB_EPTD_LOOK_UP.EPTD_PARAMETER) AS ET
	INNER JOIN
			(SELECT dbo.Tab_Temp_Indicator.EVENT_ID, dbo.Tab_Temp_Indicator.SAMPLE_NUMBER,
dbo.Tab_Temp_Indicator.PARAMETER, dbo.Tab_Temp_Indicator.REPORTING_VALUE
				FROM dbo.Tab_Temp_Indicator
				WHERE dbo.Tab_Temp_Indicator.PARAMETER = 'TOTAL_ABUNDANCE') AS TA
	 ON (ET.SAMPLE_NUMBER = TA.SAMPLE_NUMBER) AND (ET.EVENT_ID = TA.EVENT_ID)

	PRINT 'Step 13: Create Qry_EPTD_TAXA_PCT_2 - Completed Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 13: Create Qry_EPTD_TAXA_PCT_2 - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

--Step 14: Create Qry_PCT_EPT_RICH
BEGIN TRY
	BEGIN TRANSACTION

	PRINT 'Step 14: Create Qry_PCT_EPT_RICH'

	INSERT INTO dbo.Tab_Temp_Indicator(EVENT_ID, SAMPLE_NUMBER, PARAMETER, REPORTING_VALUE)
	SELECT TI.EVENT_ID, TI.SAMPLE_NUMBER, 'PCT_EPT_TAXA_RICH' AS P,
ROUND(((TI.REPORTING_VALUE/TR.REPORTING_VALUE)*100),2) AS RESULT
	FROM (SELECT dbo.Tab_Temp_Indicator.EVENT_ID, dbo.Tab_Temp_Indicator.SAMPLE_NUMBER,
dbo.Tab_Temp_Indicator.PARAMETER, dbo.Tab_Temp_Indicator.REPORTING_VALUE
			FROM dbo.Tab_Temp_Indicator
			WHERE dbo.Tab_Temp_Indicator.PARAMETER = 'EPT_TAXA_COUNT')AS TI
	INNER JOIN
```

```
                (SELECT dbo.Tab_Temp_Indicator.EVENT_ID, dbo.Tab_Temp_Indicator.SAMPLE_NUMBER,
dbo.Tab_Temp_Indicator.PARAMETER, dbo.Tab_Temp_Indicator.REPORTING_VALUE
                        FROM dbo.Tab_Temp_Indicator
                        WHERE dbo.Tab_Temp_Indicator.PARAMETER = 'TAXA_RICH') AS TR
        ON (TI.EVENT_ID = TR.EVENT_ID) AND (TI.SAMPLE_NUMBER = TR.SAMPLE_NUMBER)

        PRINT 'Step 14: Create Qry_PCT_EPT_RICH - Completed Successfully'
        COMMIT TRANSACTION
END TRY
BEGIN CATCH
        PRINT 'Transaction Failed for Step 14: Create Qry_PCT_EPT_RICH - Will Rollback'
        IF @@TRANCOUNT > 0
        ROLLBACK

        EXECUTE usp_GetErrorInfo
END CATCH

--Step 15: Create Qry_EPT_TAXA_CNT_NO_TOL
BEGIN TRY
        BEGIN TRANSACTION

        PRINT 'Step 15: Create Qry_EPT_TAXA_CNT_NO_TOL'

        INSERT INTO dbo.Tab_Temp_Indicator(EVENT_ID, SAMPLE_NUMBER, PARAMETER, REPORTING_VALUE)
        SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER,
'EPT_TAXA_COUNT_NO_TOL' AS PARAMETER, COUNT(dbo.Tab_Temp_Family.FAMILY_TSN) AS CountOfFamily_TSN
        FROM dbo.TAB_EPTD_LOOK_UP INNER JOIN dbo.TAB_FAMILY
        ON dbo.TAB_EPTD_LOOK_UP.EPTD = dbo.TAB_FAMILY.EPTD INNER JOIN dbo.Tab_Temp_Family
                ON dbo.TAB_FAMILY.TSN = dbo.Tab_Temp_Family.FAMILY_TSN
        WHERE dbo.TAB_EPTD_LOOK_UP.EPTD <> 'D' AND dbo.TAB_FAMILY.TOLARANCE_VALUE < 7
        GROUP BY dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER

        PRINT 'Step 15: Create Qry_EPT_TAXA_CNT_NO_TOL - Completed Successfully'
        COMMIT TRANSACTION
END TRY
BEGIN CATCH
        PRINT 'Transaction Failed for Step 15: Create Qry_EPT_TAXA_CNT_NO_TOL - Will Rollback'
        IF @@TRANCOUNT > 0
        ROLLBACK

        EXECUTE usp_GetErrorInfo
END CATCH

--Step 16: Create Qry_DOM_2
BEGIN TRY
        BEGIN TRANSACTION

        PRINT 'Step 16: Create Qry_DOM_2'

        INSERT INTO dbo.Tab_Temp_Indicator (EVENT_ID, SAMPLE_NUMBER, PARAMETER, REPORTING_VALUE)
        SELECT TA.EVENT_ID, TA.SAMPLE_NUMBER, 'PCT_DOM1' AS PARAMETER,
ROUND(((QD1.MaxSumOfReporting_Value/TA.REPORTING_VALUE)*100),2) AS CALC
        FROM (SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER,
MAX(dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE) AS MaxSumOfReporting_Value
                FROM dbo.Tab_Temp_Family
                GROUP BY dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER) AS
QD1
        INNER JOIN
                (SELECT dbo.Tab_Temp_Indicator.EVENT_ID, dbo.Tab_Temp_Indicator.SAMPLE_NUMBER,
dbo.Tab_Temp_Indicator.PARAMETER, dbo.Tab_Temp_Indicator.REPORTING_VALUE
                FROM dbo.Tab_Temp_Indicator
                WHERE dbo.Tab_Temp_Indicator.PARAMETER = 'TOTAL_ABUNDANCE') AS TA
        ON (QD1.EVENT_ID = TA.EVENT_ID) AND (QD1.SAMPLE_NUMBER = TA.SAMPLE_NUMBER)

        PRINT 'Step 16: Create Qry_DOM_2 - Completed Successfully'
        COMMIT TRANSACTION
END TRY
BEGIN CATCH
        PRINT 'Transaction Failed for Step 16: Create Qry_DOM_2 - Will Rollback'
        IF @@TRANCOUNT > 0
        ROLLBACK

        EXECUTE usp_GetErrorInfo
END CATCH

--Step 17: Create Qry_PCT_DOM_MULTI
BEGIN TRY
        BEGIN TRANSACTION

        PRINT 'Step 17: Create Qry_PCT_DOM_MULTI'

        SELECT DISTINCT CAST(dbo.Tab_Temp_Family.EVENT_ID AS VARCHAR(10)) + '_' +
CAST(dbo.Tab_Temp_Family.SAMPLE_NUMBER AS VARCHAR(10)) AS CATEGORY,
        dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER
        INTO dbo.Temp_PCT_DOM_MULTI
        FROM dbo.Tab_Temp_Family

        PRINT 'Step 17: Create Qry_PCT_DOM_MULTI - Completed Successfully'
        COMMIT TRANSACTION
END TRY
```

```
BEGIN CATCH
	PRINT 'Transaction Failed for Step 17: Create Qry_PCT_DOM_MULTI - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

--Step 18: Create Qry_CATEGORY_DOM
BEGIN TRY
	BEGIN TRANSACTION

	PRINT 'Step 18: Create Qry_CATEGORY_DOM'

	SELECT dbo.Temp_PCT_DOM_MULTI.CATEGORY, dbo.Tab_Temp_Family.EVENT_ID,
dbo.Tab_Temp_Family.SAMPLE_NUMBER, dbo.Tab_Temp_Family.FAMILY_TSN,
	dbo.Tab_Temp_Family.FAMILY, dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE,
dbo.Tab_Temp_Family.EXCLUSION_TAXA
	INTO dbo.Tab_Temp_Family_2
	FROM dbo.Temp_PCT_DOM_MULTI INNER JOIN dbo.Tab_Temp_Family
	ON dbo.Temp_PCT_DOM_MULTI.EVENT_ID = dbo.Tab_Temp_Family.EVENT_ID AND
dbo.Temp_PCT_DOM_MULTI.SAMPLE_NUMBER = dbo.Tab_Temp_Family.SAMPLE_NUMBER

	PRINT 'Step 18: Create Qry_CATEGORY_DOM - Completed Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 18: Create Qry_CATEGORY_DOM - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

--Step 19: Create Qry_DOM_3_1 - PART ONE CREATE of DOM3
BEGIN TRY
	BEGIN TRANSACTION

	PRINT 'Step 19: Create Qry_DOM_3_1 - PART ONE CREATE of DOM3'

	SELECT dbo.Temp_PCT_DOM_MULTI.CATEGORY, dbo.Tab_Temp_Family_2.FAMILY,
dbo.Tab_Temp_Family_2.SUM_OF_REPORTING_VALUE
	INTO dbo.Temp_DOM_3
	FROM dbo.Temp_PCT_DOM_MULTI INNER JOIN dbo.Tab_Temp_Family_2
	ON dbo.Temp_PCT_DOM_MULTI.CATEGORY = dbo.Tab_Temp_Family_2.CATEGORY
	WHERE dbo.Tab_Temp_Family_2.SUM_OF_REPORTING_VALUE IN (SELECT TOP 3
dbo.Tab_Temp_Family_2.SUM_OF_REPORTING_VALUE

	FROM dbo.Tab_Temp_Family_2

	WHERE dbo.Tab_Temp_Family_2.CATEGORY = dbo.Temp_PCT_DOM_MULTI.CATEGORY

	ORDER BY dbo.Tab_Temp_Family_2.SUM_OF_REPORTING_VALUE DESC)
	ORDER BY dbo.Temp_PCT_DOM_MULTI.CATEGORY, dbo.Tab_Temp_Family_2.SUM_OF_REPORTING_VALUE
DESC

	PRINT 'Step 19: Create Qry_DOM_3_1 - PART ONE CREATE of DOM3 - Completed Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 19: Create Qry_DOM_3_1 - PART ONE CREATE of DOM3 - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

--Step 20: Create Qry_DOM3_3 - PART TWO Create of DOM3
BEGIN TRY
	BEGIN TRANSACTION

	PRINT 'Step 20: Create Qry_DOM3_3 - PART TWO Create of DOM3'

	INSERT INTO dbo.Tab_Temp_Indicator(EVENT_ID, SAMPLE_NUMBER, PARAMETER, REPORTING_VALUE)
	SELECT TA.EVENT_ID, TA.SAMPLE_NUMBER, 'PCT_DOM3' AS METRIC,
ROUND(((D32.SumOfREPORTING_VALUE/TA.REPORTING_VALUE)*100),2) AS [VALUE]
	FROM (SELECT dbo.Tab_Temp_Indicator.EVENT_ID, dbo.Tab_Temp_Indicator.SAMPLE_NUMBER,
dbo.Tab_Temp_Indicator.PARAMETER, dbo.Tab_Temp_Indicator.REPORTING_VALUE
				FROM dbo.Tab_Temp_Indicator
				WHERE dbo.Tab_Temp_Indicator.PARAMETER = 'TOTAL_ABUNDANCE') AS TA
	INNER JOIN
			(SELECT dbo.Temp_PCT_DOM_MULTI.EVENT_ID, dbo.Temp_PCT_DOM_MULTI.SAMPLE_NUMBER,
SUM(dbo.Temp_DOM_3.SUM_OF_REPORTING_VALUE) AS SUMOfREPORTING_VALUE
				FROM dbo.Temp_DOM_3 INNER JOIN dbo.Temp_PCT_DOM_MULTI
				ON dbo.Temp_DOM_3.CATEGORY = dbo.Temp_PCT_DOM_MULTI.CATEGORY
				GROUP BY dbo.Temp_PCT_DOM_MULTI.EVENT_ID,
dbo.Temp_PCT_DOM_MULTI.SAMPLE_NUMBER) AS D32
	ON (D32.EVENT_ID = TA.EVENT_ID) AND (D32.SAMPLE_NUMBER = TA.SAMPLE_NUMBER)

	PRINT 'Step 20: Create Qry_DOM3_3 - PART TWO Create of DOM3 - Completed Successfully'
```

```
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 20: Create Qry_DOM3_3 - PART TWO Create of DOM3 - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

--Step 21: Create Qry_DOM_2_1 - PART ONE Creation of DOM2
BEGIN TRY
	BEGIN TRANSACTION

	PRINT 'Step 21: Create Qry_DOM_2_1 - PART ONE Creation of DOM2'

	SELECT dbo.Temp_PCT_DOM_MULTI.CATEGORY, dbo.Temp_DOM_3.FAMILY,
dbo.Temp_DOM_3.SUM_OF_REPORTING_VALUE
	INTO dbo.Temp_DOM_2
	FROM dbo.Temp_PCT_DOM_MULTI INNER JOIN dbo.Temp_DOM_3
	ON dbo.Temp_PCT_DOM_MULTI.CATEGORY = dbo.Temp_DOM_3.CATEGORY
	WHERE dbo.Temp_DOM_3.SUM_OF_REPORTING_VALUE IN (SELECT TOP 2
dbo.Temp_DOM_3.SUM_OF_REPORTING_VALUE

	FROM dbo.Temp_DOM_3

	WHERE dbo.Temp_DOM_3.CATEGORY = dbo.Temp_PCT_DOM_MULTI.CATEGORY

	ORDER BY  dbo.Temp_DOM_3.SUM_OF_REPORTING_VALUE DESC)
	ORDER BY dbo.Temp_PCT_DOM_MULTI.CATEGORY, dbo.Temp_DOM_3.SUM_OF_REPORTING_VALUE DESC

	PRINT 'Step 21: Create Qry_DOM_2_1 - PART ONE Creation of DOM2 - Completed Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 21: Create Qry_DOM_2_1 - PART ONE Creation of DOM2 - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

--Step 22: Create Qry_DOM2_3 PART TWO Creation of DOM 2
BEGIN TRY
	BEGIN TRANSACTION

	PRINT 'Step 22: Create Qry_DOM2_3 PART TWO Creation of DOM 2'

	INSERT INTO dbo.Tab_Temp_Indicator(EVENT_ID, SAMPLE_NUMBER, PARAMETER, REPORTING_VALUE)
	SELECT TA.EVENT_ID, TA.SAMPLE_NUMBER, 'PCT_DOM2' AS P,
ROUND(((D2.SumOfREPORTING_VALUE/TA.REPORTING_VALUE)*100),2) AS RESULT
	FROM (SELECT dbo.Tab_Temp_Indicator.EVENT_ID, dbo.Tab_Temp_Indicator.SAMPLE_NUMBER,
dbo.Tab_Temp_Indicator.PARAMETER, dbo.Tab_Temp_Indicator.REPORTING_VALUE
				FROM dbo.Tab_Temp_Indicator
				WHERE dbo.Tab_Temp_Indicator.PARAMETER = 'TOTAL_ABUNDANCE') AS TA
	INNER JOIN
		(SELECT dbo.Temp_PCT_DOM_MULTI.EVENT_ID, dbo.Temp_PCT_DOM_MULTI.SAMPLE_NUMBER,
SUM(dbo.Temp_DOM_2.SUM_OF_REPORTING_VALUE) AS SUMOfREPORTING_VALUE
				FROM dbo.Temp_DOM_2 INNER JOIN dbo.Temp_PCT_DOM_MULTI
				ON dbo.Temp_DOM_2.CATEGORY = dbo.Temp_PCT_DOM_MULTI.CATEGORY
				GROUP BY dbo.Temp_PCT_DOM_MULTI.EVENT_ID,
dbo.Temp_PCT_DOM_MULTI.SAMPLE_NUMBER) AS D2
	ON (TA.SAMPLE_NUMBER = D2.SAMPLE_NUMBER) AND (TA.EVENT_ID = D2.EVENT_ID)

	PRINT 'Step 22: Create Qry_DOM2_3 PART TWO Creation of DOM 2 - Completed Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 22: Create Qry_DOM2_3 PART TWO Creation of DOM 2 - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

--Step 23 Part 1: Create Shannon Weiner Index Step 1: QRY_SW_2
BEGIN TRY
	BEGIN TRANSACTION

	PRINT 'Step 23: Create Shannon Weiner Index - Part 1: QRY_SW_2 - Begin Calculating Tolarance Metrics'

	SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER,
dbo.Tab_Temp_Family.FAMILY_TSN, dbo.Tab_Temp_Family.FAMILY,
	dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE,
	dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE/TA.REPORTING_VALUE AS P,
LOG(dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE/TA.REPORTING_VALUE) AS LN_P,
	(dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE/TA.REPORTING_VALUE) *
LOG(dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE/TA.REPORTING_VALUE) AS P_LN_P,
	POWER(dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE/TA.REPORTING_VALUE, 2) AS P2
	INTO dbo.Temp_SW
```

```
	FROM (SELECT dbo.Tab_Temp_Indicator.EVENT_ID, dbo.Tab_Temp_Indicator.SAMPLE_NUMBER,
dbo.Tab_Temp_Indicator.PARAMETER, dbo.Tab_Temp_Indicator.REPORTING_VALUE
				FROM dbo.Tab_Temp_Indicator
				WHERE dbo.Tab_Temp_Indicator.PARAMETER = 'TOTAL_ABUNDANCE') AS TA INNER JOIN
dbo.Tab_Temp_Family
	ON (TA.SAMPLE_NUMBER = dbo.Tab_Temp_Family.SAMPLE_NUMBER) AND (TA.EVENT_ID =
dbo.Tab_Temp_Family.EVENT_ID)
	PRINT 'Step 23: Create Shannon Weiner Index - Part 1: QRY_SW_2 - Begin Calculating Tolarance Metrics - Completed
Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 23: Create Shannon Weiner Index - Part 1: QRY_SW_2 - Begin Calculating Tolarance
Metrics - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

--Step 23 Part 2:QRY_SW_4
BEGIN TRY
	BEGIN TRANSACTION

	PRINT 'Step 23: Create Shannon Weiner Index - Part 2: QRY_SW_4'

	INSERT INTO dbo.Tab_Temp_Indicator (EVENT_ID, SAMPLE_NUMBER, PARAMETER, REPORTING_VALUE)
	SELECT TS.EVENT_ID, TS.SAMPLE_NUMBER, TS.PARAMETER, ROUND(-1 * TS.SumOfP_LN_P, 4) AS
REPORTING_VALUE
	FROM (SELECT dbo.Temp_SW.EVENT_ID, dbo.Temp_SW.SAMPLE_NUMBER, 'FSW_R' AS PARAMETER,
SUM(dbo.Temp_SW.P_LN_P) AS SumOfP_LN_P
				FROM dbo.Temp_SW
				GROUP BY dbo.Temp_SW.EVENT_ID, dbo.Temp_SW.SAMPLE_NUMBER) AS TS

	PRINT 'Step 23: Create Shannon Weiner Index - Part 2: QRY_SW_4 - Completed Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 23: Create Shannon Weiner Index - Part 2: QRY_SW_4 - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

--Step 24: Create Qry_SIMPSON_2 - SIMPSONS INDEX
BEGIN TRY
	BEGIN TRANSACTION

	PRINT 'Step 24: Create Qry_SIMPSON_2 - SIMPSONS INDEX'

	INSERT INTO dbo.Tab_Temp_Indicator(EVENT_ID, SAMPLE_NUMBER, PARAMETER, REPORTING_VALUE)
	SELECT S1.EVENT_ID, S1.SAMPLE_NUMBER, 'SIMPSON_DIVERSITY' AS PARAMETER, ROUND(1 -
S1.SumOf_P2,4) AS REPORTING_VALUE
	FROM (SELECT dbo.Temp_SW.EVENT_ID, dbo.Temp_SW.SAMPLE_NUMBER, SUM(dbo.Temp_SW.P2) AS
SumOf_P2
				FROM dbo.Temp_SW
				GROUP BY dbo.Temp_SW.EVENT_ID, dbo.Temp_SW.SAMPLE_NUMBER) AS S1
	PRINT 'Step 24: Create Qry_SIMPSON_2 - SIMPSONS INDEX - Completed Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 24: Create Qry_SIMPSON_2 - SIMPSONS INDEX - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

--Step 25: Create Qry_GOLD_3 - Gold Metric
BEGIN TRY
	BEGIN TRANSACTION

	PRINT 'Step 25: Create Qry_GOLD_3 - Gold Metric'

	INSERT INTO dbo.Tab_Temp_Indicator(EVENT_ID, SAMPLE_NUMBER, PARAMETER, REPORTING_VALUE)
	SELECT TA.EVENT_ID, TA.SAMPLE_NUMBER, 'GOLD' AS PARAMETER, ROUND(1 -
G2.SumOfREPORTING_VALUE/TA.REPORTING_VALUE,4) AS GOLD
	FROM (SELECT dbo.Tab_Temp_Indicator.EVENT_ID, dbo.Tab_Temp_Indicator.SAMPLE_NUMBER,
dbo.Tab_Temp_Indicator.PARAMETER, dbo.Tab_Temp_Indicator.REPORTING_VALUE
				FROM dbo.Tab_Temp_Indicator
				WHERE dbo.Tab_Temp_Indicator.PARAMETER = 'TOTAL_ABUNDANCE') AS TA
	INNER JOIN
		(SELECT G1.EVENT_ID, G1.SAMPLE_NUMBER, SUM(G1.SUM_OF_REPORTING_VALUE) AS
SumOfREPORTING_VALUE
		 FROM (SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER,
dbo.Tab_Temp_Family.FAMILY_TSN, dbo.Tab_Temp_Family.FAMILY,
		 dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE, dbo.TAB_FAMILY.GOLD_TAXA
					FROM dbo.Tab_Temp_Family INNER JOIN dbo.TAB_FAMILY
					ON dbo.Tab_Temp_Family.FAMILY_TSN = dbo.TAB_FAMILY.TSN
```

```
                                        WHERE dbo.TAB_FAMILY.GOLD_TAXA IS NOT NULL) AS G1
                        GROUP BY G1.EVENT_ID, G1.SAMPLE_NUMBER) AS G2
        ON (TA.EVENT_ID = G2.EVENT_ID) AND (TA.SAMPLE_NUMBER = G2.SAMPLE_NUMBER)

        PRINT 'Step 25: Create Qry_GOLD_3 - Gold Metric - Completed Successfully'
        COMMIT TRANSACTION
END TRY
BEGIN CATCH
        PRINT 'Transaction Failed for Step 25: Create Qry_GOLD_3 - Gold Metric - Will Rollback'
        IF @@TRANCOUNT > 0
        ROLLBACK

        EXECUTE usp_GetErrorInfo
END CATCH

--Step 26: Create Qry_SEL_EPTD_3
BEGIN TRY
        BEGIN TRANSACTION

        PRINT 'Step 26: Create Qry_SEL_EPTD_3'

        INSERT INTO dbo.Tab_Temp_Indicator(EVENT_ID, SAMPLE_NUMBER, PARAMETER, REPORTING_VALUE)
        SELECT SE2.EVENT_ID, SE2.SAMPLE_NUMBER, 'LOG10_SEL_EPTD' AS PARAMETER,
ROUND(LOG10(SE2.SumOfREPORTING_VALUE + 1), 2) AS REPORTING_VALUE
        FROM (SELECT SE1.EVENT_ID, SE1.SAMPLE_NUMBER, SUM(SE1.SUM_OF_REPORTING_VALUE) AS
SumOfREPORTING_VALUE
                        FROM(SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER,
dbo.Tab_Temp_Family.Family, dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE,
                        dbo.TAB_FAMILY.SEL_EPTD
                                FROM dbo.Tab_Temp_Family INNER JOIN dbo.TAB_FAMILY
                                ON dbo.Tab_Temp_Family.FAMILY_TSN = dbo.TAB_FAMILY.TSN
                                WHERE dbo.TAB_FAMILY.SEL_EPTD IS NOT NULL) AS SE1
                        GROUP BY SE1.EVENT_ID, SE1.SAMPLE_NUMBER) AS SE2
        PRINT 'Step 26: Create Qry_SEL_EPTD_3 - Completed Successfully'
        COMMIT TRANSACTION
END TRY
BEGIN CATCH
        PRINT 'Transaction Failed for Step 26: Create Qry_SEL_EPTD_3 - Will Rollback'
        IF @@TRANCOUNT > 0
        ROLLBACK

        EXECUTE usp_GetErrorInfo
END CATCH

--Step 27: Create Qry_MARGALEF_1 - MARGALEF INDEX
BEGIN TRY
        BEGIN TRANSACTION

        PRINT 'Step 27: Create Qry_MARGALEF_1 - MARGALEF INDEX - End of Calculating Tolarance Metrics'

        INSERT INTO dbo.Tab_Temp_Indicator(EVENT_ID, SAMPLE_NUMBER, PARAMETER, REPORTING_VALUE)
        SELECT TR.EVENT_ID, TR.SAMPLE_NUMBER, 'MARGALEFS' AS PARAMETER, 'VALUE' =
        CASE
                WHEN TR.REPORTING_VALUE = 1 THEN 0 --ROUND((TR.REPORTING_VALUE -1
)/LOG(TA.REPORTING_VALUE), 2)
                --WHEN TR.REPORTING_VALUE = 0 THEN ROUND((TR.REPORTING_VALUE -1
)/LOG(TA.REPORTING_VALUE), 2)
        ELSE
                ROUND((TR.REPORTING_VALUE -1)/LOG(TA.REPORTING_VALUE), 4)
        END
        FROM (SELECT dbo.Tab_Temp_Indicator.EVENT_ID, dbo.Tab_Temp_Indicator.SAMPLE_NUMBER,
dbo.Tab_Temp_Indicator.PARAMETER, dbo.Tab_Temp_Indicator.REPORTING_VALUE
                        FROM dbo.Tab_Temp_Indicator
                        WHERE dbo.Tab_Temp_Indicator.PARAMETER = 'TAXA_RICH') AS TR
        INNER JOIN
                 (SELECT dbo.Tab_Temp_Indicator.EVENT_ID, dbo.Tab_Temp_Indicator.SAMPLE_NUMBER,
dbo.Tab_Temp_Indicator.PARAMETER, dbo.Tab_Temp_Indicator.REPORTING_VALUE
                        FROM dbo.Tab_Temp_Indicator
                        WHERE dbo.Tab_Temp_Indicator.PARAMETER = 'TOTAL_ABUNDANCE') AS TA
        ON (TR.SAMPLE_NUMBER = TA.SAMPLE_NUMBER) AND (TR.EVENT_ID = TA.EVENT_ID)

        PRINT 'Step 27: Create Qry_MARGALEF_1 - MARGALEF INDEX - End of Calculating Tolarance Metrics - Completed
Successfully'
        COMMIT TRANSACTION
END TRY
BEGIN CATCH
        PRINT 'Transaction Failed for Step 27: Create Qry_MARGALEF_1 - MARGALEF INDEX - End of Calculating Tolarance
Metrics - Will Rollback'
        IF @@TRANCOUNT > 0
        ROLLBACK

        EXECUTE usp_GetErrorInfo
END CATCH

--Step 28: Create Qry_PCT_TOLERANT_2
BEGIN TRY
        BEGIN TRANSACTION

        PRINT 'Step 28: Create Qry_PCT_TOLERANT_2 - Begin of Calculating Habitat and Guild Metrics'
```

```
	INSERT INTO dbo.Tab_Temp_Indicator(EVENT_ID, SAMPLE_NUMBER, PARAMETER, REPORTING_VALUE)
	SELECT PT1.EVENT_ID, PT1.SAMPLE_NUMBER, 'PCT_TOLERANT' AS P,
ROUND(((PT1.SumOfREPORTING_VALUE/TA.REPORTING_VALUE)*100), 2) AS RESULT
	FROM (SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER,
SUM(dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE) AS SumOfREPORTING_VALUE
				FROM dbo.TAB_FAMILY INNER JOIN dbo.Tab_Temp_Family
				ON dbo.TAB_FAMILY.TSN = dbo.Tab_Temp_Family.family_tsn
				WHERE dbo.TAB_FAMILY.TOLARANCE_VALUE >= 7
				GROUP BY dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER) AS
PT1
	INNER JOIN
		(SELECT dbo.Tab_Temp_Indicator.EVENT_ID, dbo.Tab_Temp_Indicator.SAMPLE_NUMBER,
dbo.Tab_Temp_Indicator.PARAMETER, dbo.Tab_Temp_Indicator.REPORTING_VALUE
				FROM dbo.Tab_Temp_Indicator
				WHERE dbo.Tab_Temp_Indicator.PARAMETER = 'TOTAL_ABUNDANCE') AS TA
	ON (PT1.EVENT_ID = TA.EVENT_ID) AND (PT1.SAMPLE_NUMBER = TA.SAMPLE_NUMBER)

	PRINT 'Step 28: Create Qry_PCT_TOLERANT_2 - Completed Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 28: Create Qry_PCT_TOLERANT_2 - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

--Step 29: Create Qry_PCT_SENSITIVE_2
BEGIN TRY
	BEGIN TRANSACTION

	PRINT 'Step 29: Create Qry_PCT_SENSITIVE_2'

	INSERT INTO dbo.Tab_Temp_Indicator(EVENT_ID, SAMPLE_NUMBER, PARAMETER, REPORTING_VALUE )
	SELECT PS1.EVENT_ID, PS1.SAMPLE_NUMBER, 'PCT_SENSITIVE' AS P,
ROUND(((PS1.SumOfREPORTING_VALUE/TA.REPORTING_VALUE)*100), 2) AS REPORT
	FROM (SELECT dbo.Tab_Temp_Indicator.EVENT_ID, dbo.Tab_Temp_Indicator.SAMPLE_NUMBER,
dbo.Tab_Temp_Indicator.PARAMETER, dbo.Tab_Temp_Indicator.REPORTING_VALUE
				FROM dbo.Tab_Temp_Indicator
				WHERE dbo.Tab_Temp_Indicator.PARAMETER = 'TOTAL_ABUNDANCE') AS TA
	INNER JOIN
			(SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER,
SUM(dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE) AS SumOfREPORTING_VALUE
				FROM dbo.TAB_FAMILY INNER JOIN dbo.Tab_Temp_Family
				ON dbo.TAB_FAMILY.TSN = dbo.Tab_Temp_Family.FAMILY_TSN
				WHERE dbo.TAB_FAMILY.TOLARANCE_VALUE <=3
				GROUP BY dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER) AS
PS1
	ON (TA.EVENT_id = PS1.EVENT_ID) AND (TA.SAMPLE_NUMBER = PS1.SAMPLE_NUMBER)

	PRINT 'Step 29: Create Qry_PCT_SENSITIVE_2 - Completed Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 29: Create Qry_PCT_SENSITIVE_2 - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

--Step 30: Create Qry_ASTP_2
BEGIN TRY
	BEGIN TRANSACTION

	PRINT 'Step 30: Create Qry_ASTP_2'

	INSERT INTO dbo.Tab_Temp_Indicator(EVENT_ID, SAMPLE_NUMBER, PARAMETER, REPORTING_VALUE)
	SELECT QA1.EVENT_ID, QA1.SAMPLE_NUMBER, 'ASPT_MOD' AS PARAMETER,
ROUND((QA1.SumOfTOLARANCE_VALUE/QA1.CountOfFamily), 2) AS [VALUE]
	FROM (SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER,
SUM(dbo.TAB_FAMILY.TOLARANCE_VALUE) AS SumOfTOLARANCE_VALUE,
	COUNT(dbo.TAB_FAMILY.FAMILY) AS CountOfFamily
				FROM dbo.Tab_Temp_Family INNER JOIN dbo.TAB_FAMILY
				ON dbo.Tab_Temp_Family.FAMILY_TSN = dbo.TAB_FAMILY.TSN
				GROUP BY dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER
				HAVING SUM(dbo.TAB_FAMILY.TOLARANCE_VALUE) Is NOT NULL) AS QA1

	PRINT 'Step 30: Create Qry_ASTP_2 - Completed Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 30: Create Qry_ASTP_2 - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH
```

```
--Step 31: Create Qry_FBI_Calc_4
BEGIN TRY
	BEGIN TRANSACTION

	PRINT 'Step 31: Create Qry_FBI_Calc_4'

	INSERT INTO dbo.Tab_Temp_Indicator(EVENT_ID, SAMPLE_NUMBER, PARAMETER, REPORTING_VALUE)
	SELECT QFC3.EVENT_ID, QFC3.SAMPLE_NUMBER, QFC3.PARAMETER, ROUND(QFC3.SumOfCACL2, 4) AS VAV
	FROM (SELECT QFC2.EVENT_ID, QFC2.SAMPLE_NUMBER, 'FBI' AS PARAMETER, SUM(QFC2.CACL2) AS
SumOfCACL2
		FROM (SELECT TA.EVENT_ID, TA.SAMPLE_NUMBER, (QFC1.CACL1/TA.REPORTING_VALUE)
AS CACL2
			FROM (SELECT dbo.Tab_Temp_Family.EVENT_ID,
dbo.Tab_Temp_Family.SAMPLE_NUMBER, dbo.Tab_Temp_Family.FAMILY_TSN,
			dbo.Tab_Temp_Family.FAMILY,
(dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE * dbo.TAB_FAMILY.TOLARANCE_VALUE) AS CACL1
				FROM dbo.TAB_FAMILY INNER JOIN
dbo.Tab_Temp_Family
				ON dbo.TAB_FAMILY.TSN =
dbo.Tab_Temp_Family.FAMILY_TSN) AS QFC1
			INNER JOIN
				(SELECT dbo.Tab_Temp_Indicator.EVENT_ID,
dbo.Tab_Temp_Indicator.SAMPLE_NUMBER, dbo.Tab_Temp_Indicator.PARAMETER,
				dbo.Tab_Temp_Indicator.REPORTING_VALUE
					FROM dbo.Tab_Temp_Indicator
					WHERE dbo.Tab_Temp_Indicator.PARAMETER =
'TOTAL_ABUNDANCE') AS TA
			ON (QFC1.EVENT_id = TA.EVENT_ID) AND (QFC1.SAMPLE_NUMBER =
TA.SAMPLE_NUMBER)) AS QFC2
	GROUP BY QFC2.EVENT_ID, QFC2.SAMPLE_NUMBER) AS QFC3

	PRINT 'Step 31: Create Qry_FBI_Calc_4 - Completed Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 31: Create Qry_FBI_Calc_4 - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

--Step 32: Create Qry_Calc_Urban_2
BEGIN TRY
	BEGIN TRANSACTION

	PRINT 'Step 32: Create Qry_Calc_Urban_2'

	INSERT INTO dbo.Tab_Temp_Indicator (EVENT_ID, SAMPLE_NUMBER, PARAMETER, REPORTING_VALUE)
	SELECT TR.EVENT_ID, TR.SAMPLE_NUMBER, 'PCT_URBAN_INTOL' AS PARAMETER, 'VAL' =
	CASE
		WHEN QCU1.CountOfFamily IS NULL THEN 0
		ELSE
			ROUND(((QCU1.CountOfFamily/TR.REPORTING_VALUE)*100), 2)
	END
	FROM (SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER,
COUNT(dbo.Tab_Temp_Family.FAMILY) AS CountOfFamily, dbo.TAB_FAMILY.INTOERANT_URBAN
			FROM dbo.Tab_Temp_Family INNER JOIN dbo.TAB_FAMILY
			ON dbo.Tab_Temp_Family.FAMILY_TSN = dbo.TAB_FAMILY.TSN
			GROUP BY dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER,
dbo.TAB_FAMILY.INTOERANT_URBAN
			HAVING dbo.TAB_FAMILY.INTOERANT_URBAN = 'X') AS QCU1
	RIGHT JOIN
		(SELECT dbo.Tab_Temp_Indicator.EVENT_ID, dbo.Tab_Temp_Indicator.SAMPLE_NUMBER,
dbo.Tab_Temp_Indicator.PARAMETER, dbo.Tab_Temp_Indicator.REPORTING_VALUE
			FROM dbo.Tab_Temp_Indicator
			WHERE dbo.Tab_Temp_Indicator.PARAMETER = 'TAXA_RICH') AS TR
	ON (QCU1.EVENT_ID = TR.EVENT_ID) AND (QCU1.SAMPLE_NUMBER = TR.SAMPLE_NUMBER)

	PRINT 'Step 32: Create Qry_Calc_Urban_2 - Completed Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 32: Create Qry_Calc_Urban_2 - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

--Step 33: Create Qry_BECK_4
BEGIN TRY
	BEGIN TRANSACTION

	PRINT 'Step 33: Create Qry_BECK_4'

	INSERT INTO dbo.Tab_Temp_Indicator(EVENT_ID, SAMPLE_NUMBER, PARAMETER, REPORTING_VALUE)
	SELECT QB3.EVENT_ID, QB3.SAMPLE_NUMBER, 'BECK' AS PARAMETER, ((2 * QB3.BECK_1) + QB3.BECK_2) AS
BI
```

```
                FROM(SELECT TA.EVENT_ID, TA.SAMPLE_NUMBER, ISNULL(QB1.CountSumOfREPORTING_VALUE, 0)
AS BECK_1, ISNULL(QB2.CountSumOfREPORTING_VALUE, 0) AS BECK_2
                        FROM (SELECT dbo.Tab_Temp_Family.EVENT_ID,
dbo.Tab_Temp_Family.SAMPLE_NUMBER, COUNT(dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE) AS
CountSumOfREPORTING_VALUE
                                FROM dbo.TAB_FAMILY INNER JOIN dbo.Tab_Temp_Family
                                ON dbo.TAB_FAMILY.TSN = dbo.Tab_Temp_Family.FAMILY_TSN
                                WHERE dbo.TAB_FAMILY.BECK_CLASS = 2
                                GROUP BY dbo.Tab_Temp_Family.EVENT_ID,
dbo.Tab_Temp_Family.SAMPLE_NUMBER) AS QB2
                        RIGHT JOIN
                                (SELECT dbo.Tab_Temp_Family.EVENT_ID,
dbo.Tab_Temp_Family.SAMPLE_NUMBER, COUNT(dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE) AS
CountSumOfREPORTING_VALUE
                                FROM dbo.TAB_FAMILY INNER JOIN dbo.Tab_Temp_Family
                                ON dbo.TAB_FAMILY.TSN = dbo.Tab_Temp_Family.FAMILY_TSN
                                WHERE dbo.TAB_FAMILY.BECK_CLASS = 1
                                GROUP BY dbo.Tab_Temp_Family.EVENT_ID,
dbo.Tab_Temp_Family.SAMPLE_NUMBER) AS QB1
                                RIGHT JOIN
                                (SELECT dbo.Tab_Temp_Indicator.EVENT_ID,
dbo.Tab_Temp_Indicator.SAMPLE_NUMBER, dbo.Tab_Temp_Indicator.PARAMETER,
dbo.Tab_Temp_Indicator.REPORTING_VALUE
                                FROM dbo.Tab_Temp_Indicator
                                WHERE dbo.Tab_Temp_Indicator.PARAMETER =
'TOTAL_ABUNDANCE') AS TA
                                ON (QB1.EVENT_ID = TA.EVENT_id) AND (QB1.SAMPLE_NUMBER =
TA.SAMPLE_NUMBER)
                        ON (QB2.EVENT_ID = TA.EVENT_id) AND (QB2.SAMPLE_NUMBER =
TA.SAMPLE_NUMBER)) AS QB3

                PRINT 'Step 33: Create Qry_BECK_4 - Completed Successfully'
                COMMIT TRANSACTION
        END TRY
BEGIN CATCH
        PRINT 'Transaction Failed for Step 33: Create Qry_BECK_4 - Will Rollback'
        IF @@TRANCOUNT > 0
        ROLLBACK

        EXECUTE usp_GetErrorInfo
END CATCH

--Step 34: Create Qry_Count_Sensitive
BEGIN TRY
        BEGIN TRANSACTION

        PRINT 'Step 34: Create Qry_Count_Sensitive'

        INSERT INTO dbo.Tab_Temp_Indicator(EVENT_ID, SAMPLE_NUMBER, PARAMETER, REPORTING_VALUE)
        SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER, 'SENSITIVE_TAXA_COUNT'
AS PARAMETER, COUNT(dbo.TAB_FAMILY.Family) AS CountOfFamily
        FROM dbo.TAB_FAMILY INNER JOIN dbo.Tab_Temp_Family
        ON dbo.TAB_FAMILY.TSN = dbo.Tab_Temp_Family.FAMILY_TSN
        WHERE dbo.TAB_FAMILY.TOLARANCE_VALUE <= 3
        GROUP BY dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER

        PRINT 'Step 34: Create Qry_Count_Sensitive - Completed Successfully'
        COMMIT TRANSACTION
END TRY
BEGIN CATCH
        PRINT 'Transaction Failed for Step 34: Create Qry_Count_Sensitive - Will Rollback'
        IF @@TRANCOUNT > 0
        ROLLBACK

        EXECUTE usp_GetErrorInfo
END CATCH

--Step 35: Create Qry_Count_Tolerant
BEGIN TRY
        BEGIN TRANSACTION

        PRINT 'Step 35: Create Qry_Count_Tolerant - End of Calculating Habitat and Guild Metrics'

        INSERT INTO dbo.Tab_Temp_Indicator(EVENT_ID, SAMPLE_NUMBER, PARAMETER, REPORTING_VALUE)
        SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER,  'TOLERANT_TAXA_COUNT'
AS PARAMETER, COUNT(dbo.TAB_FAMILY.Family) AS CountOfFamily
        FROM dbo.TAB_FAMILY INNER JOIN dbo.Tab_Temp_Family ON dbo.TAB_FAMILY.TSN =
dbo.Tab_Temp_Family.FAMILY_TSN
        WHERE dbo.TAB_FAMILY.TOLARANCE_VALUE >= 7
        GROUP BY dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER

        PRINT 'Step 35: Create Qry_Count_Tolerant - Completed Successfully'
        COMMIT TRANSACTION
END TRY
BEGIN CATCH
        PRINT 'Transaction Failed for Step 35: Create Qry_Count_Tolerant - End of Calculating Habitat and Guild Metrics - Will
Rollback'
        IF @@TRANCOUNT > 0
        ROLLBACK
```

```
	EXECUTE usp_GetErrorInfo
END CATCH

--Step 36: Create Qry_Calc_Collect_2
BEGIN TRY
	BEGIN TRANSACTION

	PRINT 'Step 36: Create Qry_Calc_Collect_2 - Begin of Calculating Other General Metrics'

	INSERT INTO dbo.Tab_Temp_Indicator(EVENT_ID, SAMPLE_NUMBER, PARAMETER, REPORTING_VALUE)
	SELECT QCC1.EVENT_ID, QCC1.SAMPLE_NUMBER, 'PCT_COLLECT' AS PARAMETER,
ROUND(((QCC1.SumOfReporting_Value/TA.REPORTING_VALUE)*100),2) AS CALC2
	FROM (SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER,
SUM(dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE) AS SumOfReporting_Value
				FROM dbo.TAB_FAMILY INNER JOIN dbo.Tab_Temp_Family
				ON dbo.TAB_FAMILY.TSN = dbo.Tab_Temp_Family.FAMILY_TSN
				WHERE dbo.TAB_FAMILY.GUILD = 'CG' OR dbo.TAB_FAMILY.GUILD = 'CF'
				GROUP BY dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER) AS
QCC1
	INNER JOIN
		(SELECT dbo.Tab_Temp_Indicator.EVENT_ID, dbo.Tab_Temp_Indicator.SAMPLE_NUMBER,
dbo.Tab_Temp_Indicator.PARAMETER, dbo.Tab_Temp_Indicator.REPORTING_VALUE
				FROM dbo.Tab_Temp_Indicator
				WHERE dbo.Tab_Temp_Indicator.PARAMETER = 'TOTAL_ABUNDANCE') AS TA
	ON (QCC1.SAMPLE_NUMBER = TA.SAMPLE_NUMBER) AND (QCC1.EVENT_ID = TA.EVENT_ID)

	PRINT 'Step 36: Create Qry_Calc_Collect_2 - Completed Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 36: Create Qry_Calc_Collect_2 - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

--Step 37: Create Qry_Calc_Cling_2
BEGIN TRY
	BEGIN TRANSACTION

	PRINT 'Step 37: Create Qry_Calc_Cling_2'

	INSERT INTO dbo.Tab_Temp_Indicator(EVENT_ID, SAMPLE_NUMBER, PARAMETER, REPORTING_VALUE)
	SELECT QCC1.EVENT_ID, QCC1.SAMPLE_NUMBER, 'PCT_CLING' AS PARAMETER,
ROUND(((QCC1.SumOfREPORTING_VALUE/TA.REPORTING_VALUE)*100),2) AS CALC2
	FROM (SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER,
Sum(dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE) AS SumOfREPORTING_VALUE
				FROM dbo.TAB_FAMILY INNER JOIN dbo.Tab_Temp_Family ON dbo.TAB_FAMILY.TSN =
dbo.Tab_Temp_Family.FAMILY_TSN
				WHERE dbo.TAB_FAMILY.HABITAT LIKE '%CN%'
				GROUP BY dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER) AS
QCC1
	INNER JOIN
		(SELECT dbo.Tab_Temp_Indicator.EVENT_ID, dbo.Tab_Temp_Indicator.SAMPLE_NUMBER,
dbo.Tab_Temp_Indicator.PARAMETER, dbo.Tab_Temp_Indicator.REPORTING_VALUE
				FROM dbo.Tab_Temp_Indicator
				WHERE dbo.Tab_Temp_Indicator.PARAMETER = 'TOTAL_ABUNDANCE') AS TA
	ON (QCC1.SAMPLE_NUMBER = TA.SAMPLE_NUMBER) AND (QCC1.EVENT_ID = TA.EVENT_ID)

	PRINT 'Step 37: Create Qry_Calc_Cling_2 - Completed Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 37: Create Qry_Calc_Cling_2 - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

--Step 38: Create Qry_Calc_Scrape_2
BEGIN TRY
	BEGIN TRANSACTION

	PRINT 'Step 38: Create Qry_Calc_Scrape_2'

	INSERT INTO dbo.Tab_Temp_Indicator(EVENT_ID, SAMPLE_NUMBER, PARAMETER, REPORTING_VALUE)
	SELECT TA.EVENT_id, TA.SAMPLE_NUMBER, 'SCRAPER_TAXA_CNT' AS PARAMETER,
IsNull(QCS1.CountOfFamily,0) AS [VALUE]
	FROM (SELECT dbo.Tab_Temp_Indicator.EVENT_ID, dbo.Tab_Temp_Indicator.SAMPLE_NUMBER,
dbo.Tab_Temp_Indicator.PARAMETER, dbo.Tab_Temp_Indicator.REPORTING_VALUE
				FROM dbo.Tab_Temp_Indicator
				WHERE dbo.Tab_Temp_Indicator.PARAMETER = 'TOTAL_ABUNDANCE') AS TA
	LEFT JOIN
		(SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER,
COUNT(dbo.Tab_Temp_Family.Family) AS CountOfFamily
				FROM dbo.TAB_FAMILY INNER JOIN dbo.Tab_Temp_Family
				ON dbo.TAB_FAMILY.TSN = dbo.Tab_Temp_Family.family_tsn
				WHERE dbo.TAB_FAMILY.GUILD = 'SC'
```

```
                GROUP BY dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER) AS
QCS1
        ON (TA.SAMPLE_NUMBER = QCS1.SAMPLE_NUMBER) AND (TA.EVENT_id = QCS1.EVENT_ID)

        PRINT 'Step 38: Create Qry_Calc_Scrape_2 - Completed Successfully'
        COMMIT TRANSACTION
END TRY
BEGIN CATCH
        PRINT 'Transaction Failed for Step 38: Create Qry_Calc_Scrape_2 - Will Rollback'
        IF @@TRANCOUNT > 0
        ROLLBACK

        EXECUTE usp_GetErrorInfo
END CATCH

-- Step 39: Create Qry_Calc_Climb_2
BEGIN TRY
        BEGIN TRANSACTION

        PRINT 'Step 39: Create Qry_Calc_Climb_2'

        INSERT INTO dbo.Tab_Temp_Indicator(EVENT_ID, SAMPLE_NUMBER, PARAMETER, REPORTING_VALUE)
        SELECT TA.EVENT_ID, TA.SAMPLE_NUMBER, 'PCT_CLIMB' AS PARAMETER, 'VALUE' =
        CASE
                WHEN QCC1.SumOfREPORTING_VALUE IS NULL THEN 0
                ELSE
                        ROUND(((QCC1.SumOfREPORTING_VALUE/TA.REPORTING_VALUE)*100),2)
        END
        FROM (SELECT dbo.Tab_Temp_Indicator.EVENT_ID, dbo.Tab_Temp_Indicator.SAMPLE_NUMBER,
dbo.Tab_Temp_Indicator.PARAMETER, dbo.Tab_Temp_Indicator.REPORTING_VALUE
                        FROM dbo.Tab_Temp_Indicator
                        WHERE dbo.Tab_Temp_Indicator.PARAMETER = 'TOTAL_ABUNDANCE') AS TA
        LEFT JOIN
                (SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER,
SUM(dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE) AS SumOfREPORTING_VALUE
                        FROM dbo.TAB_FAMILY INNER JOIN dbo.Tab_Temp_Family
                        ON dbo.TAB_FAMILY.TSN = dbo.Tab_Temp_Family.FAMILY_TSN
                        WHERE dbo.TAB_FAMILY.HABITAT LIKE '%CB%'
                        GROUP BY dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER) AS
QCC1
        ON (TA.EVENT_ID = QCC1.EVENT_ID) AND (TA.SAMPLE_NUMBER = QCC1.SAMPLE_NUMBER)

        PRINT 'Step 39: Create Qry_Calc_Climb_2 - Completed Successfully'
        COMMIT TRANSACTION
END TRY
BEGIN CATCH
        PRINT 'Transaction Failed for Step 39: Create Qry_Calc_Climb_2 - Will Rollback'
        IF @@TRANCOUNT > 0
        ROLLBACK

        EXECUTE usp_GetErrorInfo
END CATCH

-- Step 40: Create Qry_Calc_Filterers_2
BEGIN TRY
        BEGIN TRANSACTION

        PRINT 'Step 40: Create Qry_Calc_Filterers_2'

        INSERT INTO dbo.Tab_Temp_Indicator(EVENT_ID, SAMPLE_NUMBER, PARAMETER, REPORTING_VALUE)
        SELECT TA.EVENT_ID, TA.SAMPLE_NUMBER, 'PCT_FILTERERS' AS REPORT,
ROUND(((QCF1.SumOfREPORTING_VALUE/TA.REPORTING_VALUE)*100), 2) AS PCT_FILTERERS
        FROM (SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER,
SUM(dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE) AS SumOfREPORTING_VALUE
                        FROM dbo.TAB_FAMILY INNER JOIN dbo.Tab_Temp_Family
                        ON dbo.TAB_FAMILY.TSN = dbo.Tab_Temp_Family.FAMILY_TSN
                        WHERE dbo.TAB_FAMILY.GUILD = 'CF'
                        GROUP BY dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER) AS
QCF1
        INNER JOIN
                (SELECT dbo.Tab_Temp_Indicator.EVENT_ID, dbo.Tab_Temp_Indicator.SAMPLE_NUMBER,
dbo.Tab_Temp_Indicator.PARAMETER, dbo.Tab_Temp_Indicator.REPORTING_VALUE
                        FROM dbo.Tab_Temp_Indicator
                        WHERE dbo.Tab_Temp_Indicator.PARAMETER = 'TOTAL_ABUNDANCE') AS TA

        ON (QCF1.EVENT_ID = TA.EVENT_ID) AND (QCF1.SAMPLE_NUMBER = TA.SAMPLE_NUMBER)

        PRINT 'Step 40: Create Qry_Calc_Filterers_2 - Completed Successfully'
        COMMIT TRANSACTION
END TRY
BEGIN CATCH
        PRINT 'Transaction Failed for Step 40: Create Qry_Calc_Filterers_2 - Will Rollback'
        IF @@TRANCOUNT > 0
        ROLLBACK

        EXECUTE usp_GetErrorInfo
END CATCH

--Step 41: Create Qry_Calc_Gather_2
BEGIN TRY
```

```
	BEGIN TRANSACTION

	PRINT 'Step 41: Create Qry_Calc_Gather_2'

	INSERT INTO dbo.Tab_Temp_Indicator(EVENT_ID, SAMPLE_NUMBER, PARAMETER, REPORTING_VALUE)
	SELECT QCG1.EVENT_ID, QCG1.SAMPLE_NUMBER, 'PCT_GATHER' AS PAR,
ROUND(((QCG1.SumOfREPORTING_VALUE/TA.REPORTING_VALUE)*100),2) AS [VALUE]
	FROM (SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER,
SUM(dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE) AS SumOfREPORTING_VALUE
				FROM dbo.TAB_FAMILY
				INNER JOIN dbo.Tab_Temp_Family
				ON dbo.TAB_FAMILY.TSN = dbo.Tab_Temp_Family.FAMILY_TSN
				WHERE dbo.TAB_FAMILY.GUILD = 'CG'
				GROUP BY dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER) AS
QCG1
	INNER JOIN
		(SELECT dbo.Tab_Temp_Indicator.EVENT_ID, dbo.Tab_Temp_Indicator.SAMPLE_NUMBER,
dbo.Tab_Temp_Indicator.PARAMETER, dbo.Tab_Temp_Indicator.REPORTING_VALUE
				FROM dbo.Tab_Temp_Indicator
				WHERE dbo.Tab_Temp_Indicator.PARAMETER = 'TOTAL_ABUNDANCE') AS TA
	ON (QCG1.EVENT_ID = TA.EVENT_id) AND (QCG1.SAMPLE_NUMBER = TA.SAMPLE_NUMBER)

	PRINT 'Step 41: Create Qry_Calc_Gather_2 - Completed Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 41: Create Qry_Calc_Gather_2 - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

--Step 42: Create Qry_Scraper_PCT_2
BEGIN TRY
	BEGIN TRANSACTION

	PRINT 'Step 42: Create Qry_Scraper_PCT_2'

	INSERT INTO dbo.Tab_Temp_Indicator (EVENT_ID, SAMPLE_NUMBER, PARAMETER, REPORTING_VALUE)
	SELECT TA.EVENT_ID, TA.SAMPLE_NUMBER, 'PCT_SCRAPER' AS [NAME],
ROUND(((QSP1.SumOfREPORTING_VALUE/TA.REPORTING_VALUE)*100),2) AS REPORT
	FROM (SELECT dbo.Tab_Temp_Indicator.EVENT_ID, dbo.Tab_Temp_Indicator.SAMPLE_NUMBER,
dbo.Tab_Temp_Indicator.PARAMETER, dbo.Tab_Temp_Indicator.REPORTING_VALUE
				FROM dbo.Tab_Temp_Indicator
				WHERE dbo.Tab_Temp_Indicator.PARAMETER = 'TOTAL_ABUNDANCE') AS TA
	INNER JOIN
		(SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER,
SUM(dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE) AS SumOfREPORTING_VALUE
				FROM dbo.Tab_Temp_Family
				INNER JOIN dbo.TAB_FAMILY
				ON dbo.Tab_Temp_Family.FAMILY_TSN = dbo.TAB_FAMILY.TSN
				WHERE dbo.TAB_FAMILY.GUILD = 'SC'
				GROUP BY dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER) AS
QSP1
	ON (TA.EVENT_id = QSP1.EVENT_ID) AND (TA.SAMPLE_NUMBER = QSP1.SAMPLE_NUMBER)

	PRINT 'Step 42: Create Qry_Scraper_PCT_2 - Completed Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 42: Create Qry_Scraper_PCT_2 - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

--Step 43: Create Qry_Shred_2
BEGIN TRY
	BEGIN TRANSACTION

	PRINT 'Step 43: Create Qry_Shred_2'

	INSERT INTO dbo.Tab_Temp_Indicator(EVENT_ID, SAMPLE_NUMBER, PARAMETER, REPORTING_VALUE)
	SELECT QSP1.EVENT_ID, QSP1.SAMPLE_NUMBER, 'PCT_SHREDDER' AS P,
ROUND(((QSP1.SumOfREPORTING_VALUE/TA.REPORTING_VALUE)*100),2) AS RESULT
	FROM (SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER,
SUM(dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE) AS SumOfREPORTING_VALUE
				FROM dbo.Tab_Temp_Family
				INNER JOIN dbo.TAB_FAMILY
				ON dbo.Tab_Temp_Family.FAMILY_TSN = dbo.TAB_FAMILY.TSN
				WHERE dbo.TAB_FAMILY.GUILD = 'SH'
				GROUP BY dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER) AS
QSP1
	INNER JOIN
		(SELECT dbo.Tab_Temp_Indicator.EVENT_ID, dbo.Tab_Temp_Indicator.SAMPLE_NUMBER,
dbo.Tab_Temp_Indicator.PARAMETER, dbo.Tab_Temp_Indicator.REPORTING_VALUE
				FROM dbo.Tab_Temp_Indicator
				WHERE dbo.Tab_Temp_Indicator.PARAMETER = 'TOTAL_ABUNDANCE') AS TA
```

```
	ON (QSP1.EVENT_ID = TA.EVENT_id) AND (QSP1.SAMPLE_NUMBER = TA.SAMPLE_NUMBER)

	PRINT 'Step 43: Create Qry_Shred_2 - Completed Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 43: Create Qry_Shred_2 - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

--Step 44: Create Qry_Calc_Swim_2
BEGIN TRY
	BEGIN TRANSACTION

	PRINT 'Step 44: Create Qry_Calc_Swim_2'

	INSERT INTO dbo.Tab_Temp_Indicator (EVENT_ID, SAMPLE_NUMBER, PARAMETER, REPORTING_VALUE)
	SELECT QCS1.EVENT_ID, QCS1.SAMPLE_NUMBER, 'PCT_SWIMMER' AS P,
ROUND(((QCS1.SumOfREPORTING_VALUE/TA.REPORTING_VALUE)*100),2) AS RESULT
	FROM (SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER,
SUM(dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE) AS SumOfREPORTING_VALUE
				FROM dbo.TAB_FAMILY INNER JOIN dbo.Tab_Temp_Family
				ON dbo.TAB_FAMILY.TSN = dbo.Tab_Temp_Family.FAMILY_TSN
				WHERE dbo.TAB_FAMILY.HABITAT LIKE '%SW%'
				GROUP BY dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER) AS
QCS1
	INNER JOIN
	(SELECT dbo.Tab_Temp_Indicator.EVENT_ID, dbo.Tab_Temp_Indicator.SAMPLE_NUMBER,
dbo.Tab_Temp_Indicator.PARAMETER, dbo.Tab_Temp_Indicator.REPORTING_VALUE
				FROM dbo.Tab_Temp_Indicator
				WHERE dbo.Tab_Temp_Indicator.PARAMETER = 'TOTAL_ABUNDANCE') AS TA
	ON (QCS1.EVENT_ID = TA.EVENT_id) AND (QCS1.SAMPLE_NUMBER = TA.SAMPLE_NUMBER)

	PRINT 'Step 44: Create Qry_Calc_Swim_2 - Completed Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 44: Create Qry_Calc_Swim_2 - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

--Step 45: Create Qry_Calc_Cling_Cnt_2
BEGIN TRY
	BEGIN TRANSACTION

	PRINT 'Step 45: Create Qry_Calc_Cling_Cnt_2 - End of Calculating Other General Metrics'

	INSERT INTO dbo.Tab_Temp_Indicator(EVENT_ID, SAMPLE_NUMBER, PARAMETER, REPORTING_VALUE)
	SELECT QCCC1.EVENT_ID, QCCC1.SAMPLE_NUMBER, 'PCT_CLINGER_TAXA' AS P,
ROUND(((QCCC1.CountSumOfREPORTING_VALUE/TR.REPORTING_VALUE)*100),2) AS RESULT
	FROM (SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER,
COUNT(dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE) AS CountSumOfREPORTING_VALUE
				FROM dbo.TAB_FAMILY INNER JOIN dbo.Tab_Temp_Family
				ON dbo.TAB_FAMILY.TSN = dbo.Tab_Temp_Family.FAMILy_TSN
				WHERE dbo.TAB_FAMILY.HABITAT LIKE '%CN%'
				GROUP BY dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER) AS
QCCC1
	INNER JOIN
		(SELECT dbo.Tab_Temp_Indicator.EVENT_ID, dbo.Tab_Temp_Indicator.SAMPLE_NUMBER,
dbo.Tab_Temp_Indicator.PARAMETER, dbo.Tab_Temp_Indicator.REPORTING_VALUE
				FROM dbo.Tab_Temp_Indicator
				WHERE dbo.Tab_Temp_Indicator.PARAMETER = 'TAXA_RICH') AS TR
	ON (QCCC1.EVENT_ID = TR.EVENT_ID) AND (QCCC1.SAMPLE_NUMBER = TR.SAMPLE_NUMBER)

	PRINT 'Step 45: Create Qry_Calc_Cling_Cnt_2 - Completed Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 45: Create Qry_Calc_Cling_Cnt_2 - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

--Step 46: Create Qry_Chroninmidae_Pct
BEGIN TRY
	BEGIN TRANSACTION

	PRINT 'Step 46: Create Qry_Chroninmidae_Pct - Begin Calculating Other ...'

	INSERT INTO dbo.Tab_Temp_Indicator(EVENT_ID, SAMPLE_NUMBER, PARAMETER, REPORTING_VALUE)
	SELECT QCP1.EVENT_ID, QCP1.SAMPLE_NUMBER, 'PCT_CHIRONOMIDAE' AS PARAMETER,
ROUND(((QCP1.SumOfREPORTING_VALUE/TA.REPORTING_VALUE)*100),2) AS [VALUE]
```

```
	FROM (SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER,
SUM(dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE) AS SumOfREPORTING_VALUE
				FROM dbo.Tab_Temp_Family INNER JOIN dbo.TAB_FAMILY ON
dbo.Tab_Temp_Family.FAMILY_TSN = dbo.TAB_FAMILY.TSN
				GROUP BY dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER,
dbo.TAB_FAMILY.CHRONONMIDAE
				HAVING dbo.TAB_FAMILY.CHRONONMIDAE IS NOT NULL) AS QCP1
	INNER JOIN
		(SELECT dbo.Tab_Temp_Indicator.EVENT_ID, dbo.Tab_Temp_Indicator.SAMPLE_NUMBER,
dbo.Tab_Temp_Indicator.PARAMETER, dbo.Tab_Temp_Indicator.REPORTING_VALUE
				FROM dbo.Tab_Temp_Indicator
				WHERE dbo.Tab_Temp_Indicator.PARAMETER = 'TOTAL_ABUNDANCE') AS TA
	ON (QCP1.EVENT_ID = TA.EVENT_ID) AND (QCP1.SAMPLE_NUMBER = TA.SAMPLE_NUMBER)

	PRINT 'Step 46: Create Qry_Chroninmidae_Pct - Completed Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 46: Create Qry_Chroninmidae_Pct - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

--Step: 47 Create Qry_Pct_Corbicula
BEGIN TRY
	BEGIN TRANSACTION

	PRINT 'Step 47: Create Qry_Pct_Corbicula'

	INSERT INTO dbo.Tab_Temp_Indicator (EVENT_ID, SAMPLE_NUMBER, PARAMETER, REPORTING_VALUE)
	SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER, 'PCT_CORBICULA' AS P,
	ROUND(((dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE/TA.REPORTING_VALUE)*100),2) AS RESULT
	FROM dbo.Tab_Temp_Family
	INNER JOIN
	(SELECT dbo.Tab_Temp_Indicator.EVENT_ID, dbo.Tab_Temp_Indicator.SAMPLE_NUMBER,
dbo.Tab_Temp_Indicator.PARAMETER, dbo.Tab_Temp_Indicator.REPORTING_VALUE
				FROM dbo.Tab_Temp_Indicator
				WHERE dbo.Tab_Temp_Indicator.PARAMETER = 'TOTAL_ABUNDANCE') AS TA
	ON (dbo.Tab_Temp_Family.EVENT_ID = TA.EVENT_ID) AND (dbo.Tab_Temp_Family.SAMPLE_NUMBER =
TA.SAMPLE_NUMBER)
	WHERE dbo.Tab_Temp_Family.FAMILY_TSN = '0081381'

	PRINT 'Step 47: Create Qry_Pct_Corbicula - Completed Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 47: Create Qry_Pct_Corbicula - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

--Step 48: Create Qry_Net_Caddisfly_2
BEGIN TRY
	BEGIN TRANSACTION

	PRINT 'Step 48: Create Qry_Net_Caddisfly_2'

	INSERT INTO dbo.Tab_Temp_Indicator (EVENT_ID, SAMPLE_NUMBER, PARAMETER, REPORTING_VALUE)
	SELECT TA.EVENT_ID, QC1.SAMPLE_NUMBER, 'PCT_NET_CADDISFLY' AS P,
ROUND(((QC1.SumOfREPORTING_VALUE/TA.REPORTING_VALUE)*100),2) AS RESULT
	FROM (SELECT dbo.Tab_Temp_Indicator.EVENT_ID, dbo.Tab_Temp_Indicator.SAMPLE_NUMBER,
dbo.Tab_Temp_Indicator.PARAMETER, dbo.Tab_Temp_Indicator.REPORTING_VALUE
				FROM dbo.Tab_Temp_Indicator
				WHERE dbo.Tab_Temp_Indicator.PARAMETER = 'TOTAL_ABUNDANCE') AS TA
	INNER JOIN
		(SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER,
SUM(dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE) AS SumOfREPORTING_VALUE
				FROM dbo.Tab_Temp_Family
				WHERE dbo.Tab_Temp_Family.FAMILY_TSN = '0115257' OR dbo.Tab_Temp_Family.FAMILY_TSN
= '0115398' OR dbo.Tab_Temp_Family.FAMILY_TSN = '0117043'
				GROUP BY dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER) AS
QC1
	ON (TA.EVENT_id = QC1.EVENT_ID) AND (TA.SAMPLE_NUMBER = QC1.SAMPLE_NUMBER)

	PRINT 'Step 48: Create Qry_Net_Caddisfly_2 - Completed Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 48: Create Qry_Net_Caddisfly_2 - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

--Step 49: Create Qry_Pct_Caddis_No_Tolerant_2
```

```
BEGIN TRY
	BEGIN TRANSACTION

	PRINT 'Step 49: Create Qry_Pct_Caddis_No_Tolerant_2'

	INSERT INTO dbo.Tab_Temp_Indicator (EVENT_ID, SAMPLE_NUMBER, PARAMETER, REPORTING_VALUE)
	SELECT QPCNT1.EVENT_ID, QPCNT1.SAMPLE_NUMBER, 'PCT_TRICHOPTERA_NO_TOL' AS P,
ROUND(((QPCNT1.SumOfREPORTING_VALUE/TA.REPORTING_VALUE)*100),2) AS RESULT
	FROM (SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER,
SUM(dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE) AS SumOfREPORTING_VALUE
				FROM dbo.Tab_Temp_Family
				INNER JOIN
				(SELECT dbo.Tab_Temp_Indicator.EVENT_ID, dbo.Tab_Temp_Indicator.SAMPLE_NUMBER,
dbo.Tab_Temp_Indicator.PARAMETER, dbo.Tab_Temp_Indicator.REPORTING_VALUE
					FROM dbo.Tab_Temp_Indicator
					WHERE dbo.Tab_Temp_Indicator.PARAMETER = 'TOTAL_ABUNDANCE') AS TA
				ON (dbo.Tab_Temp_Family.SAMPLE_NUMBER = TA.SAMPLE_NUMBER) AND
(dbo.Tab_Temp_Family.EVENT_ID = TA.EVENT_ID)
				INNER JOIN dbo.TAB_FAMILY
				ON dbo.Tab_Temp_Family.FAMILY_TSN = dbo.TAB_FAMILY.TSN
				WHERE dbo.TAB_FAMILY.EPTD = 'T' AND dbo.Tab_Temp_Family.FAMILY_TSN <> '0115398'
				GROUP BY dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER) AS
QPCNT1
	INNER JOIN
		(SELECT dbo.Tab_Temp_Indicator.EVENT_ID, dbo.Tab_Temp_Indicator.SAMPLE_NUMBER,
dbo.Tab_Temp_Indicator.PARAMETER, dbo.Tab_Temp_Indicator.REPORTING_VALUE
				FROM dbo.Tab_Temp_Indicator
				WHERE dbo.Tab_Temp_Indicator.PARAMETER = 'TOTAL_ABUNDANCE') AS TA
	ON (QPCNT1.SAMPLE_NUMBER = TA.SAMPLE_NUMBER) AND (QPCNT1.EVENT_ID = TA.EVENT_ID)

	PRINT 'Step 49: Create Qry_Pct_Caddis_No_Tolerant_2 - Completed Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 49: Create Qry_Pct_Caddis_No_Tolerant_2 - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

--Step 50: Create Qry_Pct_Amphipod_2
BEGIN TRY
	BEGIN TRANSACTION

	PRINT 'Step 50: Create Qry_Pct_Amphipod_2'

	INSERT INTO dbo.Tab_Temp_Indicator (EVENT_ID, SAMPLE_NUMBER, PARAMETER, REPORTING_VALUE)
	SELECT TA.EVENT_ID, TA.SAMPLE_NUMBER, 'PCT_AMPHIPOD' AS P,
ROUND(((QPA1.SumOfREPORTING_VALUE/TA.REPORTING_VALUE)*100),2) AS RESULT
	FROM (SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER,
SUM(dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE) AS SumOfREPORTING_VALUE
				FROM dbo.Tab_Temp_Family INNER JOIN dbo.TAB_FAMILY
				ON dbo.Tab_Temp_Family.FAMILY_TSN = dbo.TAB_FAMILY.TSN
				WHERE dbo.TAB_FAMILY.[AMPHOPOD/ISOPOD] = 'A'
				GROUP BY dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER) AS
QPA1
	INNER JOIN
		(SELECT dbo.Tab_Temp_Indicator.EVENT_ID, dbo.Tab_Temp_Indicator.SAMPLE_NUMBER,
dbo.Tab_Temp_Indicator.PARAMETER, dbo.Tab_Temp_Indicator.REPORTING_VALUE
				FROM dbo.Tab_Temp_Indicator
				WHERE dbo.Tab_Temp_Indicator.PARAMETER = 'TOTAL_ABUNDANCE') AS TA
	ON (QPA1.EVENT_ID = TA.EVENT_ID) AND (QPA1.SAMPLE_NUMBER = TA.SAMPLE_NUMBER)

	PRINT 'Step 50: Create Qry_Pct_Amphipod_2 - Completed Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 50: Create Qry_Pct_Amphipod_2 - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

--Step 51: Create Qry_Pct_Limestone_2
BEGIN TRY
	BEGIN TRANSACTION

	PRINT 'Step 51: Create Qry_Pct_Limestone_2'

	INSERT INTO dbo.Tab_Temp_Indicator (EVENT_ID, SAMPLE_NUMBER, PARAMETER, REPORTING_VALUE)
	SELECT QPL1.EVENT_ID, QPL1.SAMPLE_NUMBER, 'PCT_LIMESTONE' AS P,
ROUND(((QPL1.SumOfREPORTING_VALUE/TA.REPORTING_VALUE)*100),2) AS RESULT
	FROM (SELECT dbo.Tab_Temp_Indicator.EVENT_ID, dbo.Tab_Temp_Indicator.SAMPLE_NUMBER,
dbo.Tab_Temp_Indicator.PARAMETER, dbo.Tab_Temp_Indicator.REPORTING_VALUE
				FROM dbo.Tab_Temp_Indicator
				WHERE dbo.Tab_Temp_Indicator.PARAMETER = 'TOTAL_ABUNDANCE') AS TA
	INNER JOIN
```

```
                (SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER,
SUM(dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE) AS SumOfREPORTING_VALUE
                        FROM dbo.Tab_Temp_Family
                        INNER JOIN dbo.TAB_FAMILY
                        ON dbo.Tab_Temp_Family.FAMILY_TSN = dbo.TAB_FAMILY.TSN
                        WHERE dbo.TAB_FAMILY.[AMPHIPOD/ISOPOD]IS NOT NULL
                        GROUP BY dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER) AS
QPL1
        ON (TA.EVENT_ID = QPL1.EVENT_ID) AND (TA.SAMPLE_NUMBER = QPL1.SAMPLE_NUMBER)

        PRINT 'Step 51: Create Qry_Pct_Limestone_2 - Completed Successfully'
        COMMIT TRANSACTION
END TRY
BEGIN CATCH
        PRINT 'Transaction Failed for Step 51: Create Qry_Pct_Limestone_2 - Will Rollback'
        IF @@TRANCOUNT > 0
        ROLLBACK

        EXECUTE usp_GetErrorInfo
END CATCH

--Step 52: Create Qry_Non_Insect_2
BEGIN TRY
        BEGIN TRANSACTION

        PRINT 'Step 52: Create Qry_Non_Insect_2'

        INSERT INTO dbo.Tab_Temp_Indicator (EVENT_ID, SAMPLE_NUMBER, PARAMETER, REPORTING_VALUE)
        SELECT TA.EVENT_ID, TA.SAMPLE_NUMBER, 'PCT_NON_INSECT' AS P,
ROUND(((QNI1.SumOfREPORTING_VALUE/TA.REPORTING_VALUE)*100),2) AS RESULT
        FROM (SELECT dbo.Tab_Temp_Indicator.EVENT_ID, dbo.Tab_Temp_Indicator.SAMPLE_NUMBER,
dbo.Tab_Temp_Indicator.PARAMETER, dbo.Tab_Temp_Indicator.REPORTING_VALUE
                        FROM dbo.Tab_Temp_Indicator
                        WHERE dbo.Tab_Temp_Indicator.PARAMETER = 'TOTAL_ABUNDANCE') AS TA
        INNER JOIN
                (SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER,
SUM(dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE) AS SumOfREPORTING_VALUE
                        FROM dbo.Tab_Temp_Family INNER JOIN dbo.TAB_FAMILY ON
dbo.Tab_Temp_Family.FAMILY_TSN = dbo.TAB_FAMILY.TSN
                        WHERE dbo.TAB_FAMILY.NON_INSECT = 'X'
                        GROUP BY dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER) AS
QNI1
        ON (TA.EVENT_ID = QNI1.EVENT_ID) AND (TA.SAMPLE_NUMBER = QNI1.SAMPLE_NUMBER)

        PRINT 'Step 52: Create Qry_Non_Insect_2 - Completed Successfully'
        COMMIT TRANSACTION
END TRY
BEGIN CATCH
        PRINT 'Transaction Failed for Step 52: Create Qry_Non_Insect_2 - Will Rollback'
        IF @@TRANCOUNT > 0
        ROLLBACK

        EXECUTE usp_GetErrorInfo
END CATCH

--Step 53: Create Qry_Oligo_2
BEGIN TRY
        BEGIN TRANSACTION

        PRINT 'Step 53: Create Qry_Oligo_2'

        INSERT INTO dbo.Tab_Temp_Indicator (EVENT_ID, SAMPLE_NUMBER, PARAMETER, REPORTING_VALUE)
        SELECT QO1.EVENT_ID, QO1.SAMPLE_NUMBER, 'PCT_OLIGOCHAETA' AS P,
ROUND(((QO1.SumOfREPORTING_VALUE/TA.REPORTING_VALUE)*100),2) AS RESULT
        FROM (SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER,
SUM(dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE) AS SumOfREPORTING_VALUE
                        FROM dbo.Tab_Temp_Family
                        INNER JOIN dbo.TAB_FAMILY
                        ON dbo.Tab_Temp_Family.FAMILY_TSN = dbo.TAB_FAMILY.TSN
                        WHERE dbo.TAB_FAMILY.GOLD_TAXA = 'O'
                        GROUP BY dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER) AS
QO1
        INNER JOIN
                (SELECT dbo.Tab_Temp_Indicator.EVENT_ID, dbo.Tab_Temp_Indicator.SAMPLE_NUMBER,
dbo.Tab_Temp_Indicator.PARAMETER, dbo.Tab_Temp_Indicator.REPORTING_VALUE
                        FROM dbo.Tab_Temp_Indicator
                        WHERE dbo.Tab_Temp_Indicator.PARAMETER = 'TOTAL_ABUNDANCE') AS TA
        ON (QO1.EVENT_ID = TA.EVENT_ID) AND (QO1.SAMPLE_NUMBER = TA.SAMPLE_NUMBER)

        PRINT 'Step 53: Create Qry_Oligo_2 - Completed Successfully'
        COMMIT TRANSACTION
END TRY
BEGIN CATCH
        PRINT 'Transaction Failed for Step 53: Create Qry_Oligo_2 - Will Rollback'
        IF @@TRANCOUNT > 0
        ROLLBACK

        EXECUTE usp_GetErrorInfo
END CATCH
```

```
--Step 54: Create Qry_Nco_Taxa_Cnt
BEGIN TRY
	BEGIN TRANSACTION

	PRINT 'Step 54: Create Qry_Nco_Taxa_Cnt'

	INSERT INTO dbo.Tab_Temp_Indicator (dbo.Tab_Temp_Indicator.EVENT_ID,
dbo.Tab_Temp_Indicator.SAMPLE_NUMBER, dbo.Tab_Temp_Indicator.PARAMETER,
	dbo.Tab_Temp_Indicator.REPORTING_VALUE)
	SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER, 'NCO_TAXA_CNT' AS P,
COUNT(dbo.Tab_Temp_Family.FAMILY_TSN) AS CountOfFamily_TSN
	FROM dbo.Tab_Temp_Family INNER JOIN dbo.TAB_FAMILY
	ON dbo.Tab_Temp_Family.FAMILY_TSN = dbo.TAB_FAMILY.TSN
	WHERE dbo.TAB_FAMILY.NCO IS NULL
	GROUP BY dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER

	PRINT 'Step 54: Create Qry_Nco_Taxa_Cnt - Completed Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 54: Create Qry_Nco_Taxa_Cnt - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

--Step 55: Create Qry_Non_Insect_Taxa_Cnt
BEGIN TRY
	BEGIN TRANSACTION

	PRINT 'Step 55: Create Qry_Non_Insect_Taxa_Cnt'

	INSERT INTO dbo.Tab_Temp_Indicator (EVENT_ID, SAMPLE_NUMBER, PARAMETER, REPORTING_VALUE)
	SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER, 'NON_INSECT_TAXA_CNT'
AS P, COUNT(dbo.Tab_Temp_Family.FAMILY_TSN) AS CountOfFamily_TSN
	FROM dbo.Tab_Temp_Family
	INNER JOIN dbo.TAB_FAMILY
	ON dbo.Tab_Temp_Family.FAMILY_TSN = dbo.TAB_FAMILY.TSN
	GROUP BY dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER,
dbo.TAB_FAMILY.NON_INSECT
	HAVING dbo.TAB_FAMILY.NON_INSECT = 'X'

	PRINT 'Step 55: Create Qry_Non_Insect_Taxa_Cnt - Completed Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 55: Create Qry_Non_Insect_Taxa_Cnt - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

--Step 56: Create Qry_Trichoptera_Taxa_Cnt_no_Hydro
BEGIN TRY
	BEGIN TRANSACTION

	PRINT 'Step 56: Create Qry_Trichoptera_Taxa_Cnt_no_Hydro'

	INSERT INTO dbo.Tab_Temp_Indicator (EVENT_ID, SAMPLE_NUMBER, PARAMETER, REPORTING_VALUE)
	SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER,
'TRICHOPTERA_TAXA_COUNT_NO_HYDR' AS PARAMETER,
	COUNT(dbo.Tab_Temp_Family.FAMILY_TSN) AS CountOfFamily_TSN
	FROM dbo.TAB_EPTD_LOOK_UP INNER JOIN dbo.TAB_FAMILY
	ON dbo.TAB_EPTD_LOOK_UP.EPTD = dbo.TAB_FAMILY.EPTD
		INNER JOIN dbo.Tab_Temp_Family
		ON dbo.Tab_Temp_Family.FAMILY_TSN = dbo.TAB_FAMILY.TSN
	WHERE (dbo.TAB_EPTD_LOOK_UP.EPTD = 't') AND (dbo.Tab_Temp_Family.FAMILY_TSN <> '0115398')
	GROUP BY dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER

	PRINT 'Step 56: Create Qry_Trichoptera_Taxa_Cnt_no_Hydro - Completed Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 56: Create Qry_Trichoptera_Taxa_Cnt_no_Hydro - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

--Step 57: Create Qry_Burrower_Pct_2
BEGIN TRY
	BEGIN TRANSACTION

	PRINT 'Step 57: Create Qry_Burrower_Pct_2'

	INSERT INTO dbo.Tab_Temp_Indicator (EVENT_ID, SAMPLE_NUMBER, PARAMETER, REPORTING_VALUE)
	SELECT TA.EVENT_id, TA.SAMPLE_NUMBER, 'PCT_BURROWER' AS PARAMETER, 'METRIC_VALUE' =
```

```
	CASE
		WHEN QBP1.SumOfREPORTING_VALUE < 1 THEN 0
	ELSE
		ROUND(((QBP1.SumOfREPORTING_VALUE/TA.REPORTING_VALUE)*100),2)
	END
	FROM (SELECT dbo.Tab_Temp_Indicator.EVENT_ID, dbo.Tab_Temp_Indicator.SAMPLE_NUMBER,
dbo.Tab_Temp_Indicator.PARAMETER, dbo.Tab_Temp_Indicator.REPORTING_VALUE
				FROM dbo.Tab_Temp_Indicator
				WHERE dbo.Tab_Temp_Indicator.PARAMETER = 'TOTAL_ABUNDANCE') AS TA
	INNER JOIN
		 (SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER,
SUM(dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE) AS SumOfREPORTING_VALUE
				FROM dbo.Tab_Temp_Family INNER JOIN dbo.TAB_FAMILY
				ON dbo.Tab_Temp_Family.FAMILY_TSN = dbo.TAB_FAMILY.TSN
				WHERE dbo.TAB_FAMILY.HABITAT LIKE '%BU%'
				GROUP BY dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER) AS
QBP1
	ON (TA.SAMPLE_NUMBER = QBP1.SAMPLE_NUMBER) AND (TA.EVENT_ID = QBP1.EVENT_ID)

	PRINT 'Step 57: Create Qry_Burrower_Pct_2 - Completed Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 57: Create Qry_Burrower_Pct_2 - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

--Step 58: Create Qry_Pct_Predator_2
BEGIN TRY
	BEGIN TRANSACTION

	PRINT 'Step 58: Create Qry_Pct_Predator_2'

	INSERT INTO dbo.Tab_Temp_Indicator (EVENT_ID, SAMPLE_NUMBER, PARAMETER, REPORTING_VALUE)
	SELECT TA.EVENT_ID, TA.SAMPLE_NUMBER, 'PCT_PREDATOR' AS PARAMETER, 'METRIC' =
	CASE
		WHEN QPP1.SumOfREPORTING_VALUE < 1 THEN 0
	ELSE
		ROUND(((QPP1.SumOfREPORTING_VALUE/TA.REPORTING_VALUE)*100),2)
	END
	FROM (SELECT dbo.Tab_Temp_Indicator.EVENT_ID, dbo.Tab_Temp_Indicator.SAMPLE_NUMBER,
dbo.Tab_Temp_Indicator.PARAMETER, dbo.Tab_Temp_Indicator.REPORTING_VALUE
				FROM dbo.Tab_Temp_Indicator
				WHERE dbo.Tab_Temp_Indicator.PARAMETER = 'TOTAL_ABUNDANCE') AS TA
	INNER JOIN
		 (SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER,
SUM(dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE) AS SumOfREPORTING_VALUE
				FROM dbo.Tab_Temp_Family INNER JOIN dbo.TAB_FAMILY
				ON dbo.Tab_Temp_Family.FAMILY_TSN = dbo.TAB_FAMILY.TSN
				WHERE dbo.TAB_FAMILY.GUILD LIKE '%PR%'
				GROUP BY dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER) AS
QPP1
	ON (TA.EVENT_ID = QPP1.EVENT_ID) AND (TA.SAMPLE_NUMBER = QPP1.SAMPLE_NUMBER)

	PRINT 'Step 58: Create Qry_Pct_Predator_2 - Completed Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 58: Create Qry_Pct_Predator_2 - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

--Step 59: Create Qry_Pct_Black_Fly
BEGIN TRY
	BEGIN TRANSACTION

	PRINT 'Step 59: Create Qry_Pct_Black_Fly'

	INSERT INTO dbo.Tab_Temp_Indicator (EVENT_ID, SAMPLE_NUMBER, PARAMETER, REPORTING_VALUE)
	SELECT TA.EVENT_ID, TA.SAMPLE_NUMBER, 'PCT_SIMULIIDAE' AS PARAMETER, 'VAL' =
	CASE
		WHEN dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE < 1 THEN 0
	ELSE
		ROUND(((dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE/TA.REPORTING_VALUE)*100),2)
	END
	FROM dbo.Tab_Temp_Family
	INNER JOIN
	(SELECT dbo.Tab_Temp_Indicator.EVENT_ID, dbo.Tab_Temp_Indicator.SAMPLE_NUMBER,
dbo.Tab_Temp_Indicator.PARAMETER, dbo.Tab_Temp_Indicator.REPORTING_VALUE
				FROM dbo.Tab_Temp_Indicator
				WHERE dbo.Tab_Temp_Indicator.PARAMETER = 'TOTAL_ABUNDANCE') AS TA
	ON (dbo.Tab_Temp_Family.EVENT_ID = TA.EVENT_id) AND (dbo.Tab_Temp_Family.SAMPLE_NUMBER =
TA.SAMPLE_NUMBER)
	WHERE dbo.Tab_Temp_Family.FAMILY_TSN = '0126640'
```

```
	PRINT 'Step 59: Create Qry_Pct_Black_Fly - Completed Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 59: Create Qry_Pct_Black_Fly - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

--Step 60: Create Qry_Scraper_To_Collect_Filter
BEGIN TRY
	BEGIN TRANSACTION

	PRINT 'Step 60: Create Qry_Scraper_To_Collect_Filter'

	INSERT INTO dbo.Tab_Temp_Indicator (EVENT_ID, SAMPLE_NUMBER, PARAMETER, REPORTING_VALUE)
	SELECT QSP1.EVENT_ID, QSP1.SAMPLE_NUMBER, 'RATIO_SC_TO_CF' AS PARAMETER,
ROUND(QSP1.SumOfREPORTING_VALUE/QCF1.SumOfREPORTING_VALUE, 3) AS METRIC
	FROM (SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER,
SUM(dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE) AS SumOfREPORTING_VALUE
		FROM dbo.Tab_Temp_Family INNER JOIN dbo.TAB_FAMILY ON
dbo.Tab_Temp_Family.family_tsn=dbo.TAB_FAMILY.TSN
		WHERE dbo.TAB_FAMILY.GUILD = 'SC'
		GROUP BY dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER) AS
QSP1
	INNER JOIN
		(SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER,
SUM(dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE) AS SumOfREPORTING_VALUE
		FROM dbo.TAB_FAMILY INNER JOIN dbo.Tab_Temp_Family ON
dbo.TAB_FAMILY.TSN=dbo.Tab_Temp_Family.family_tsn
		WHERE dbo.TAB_FAMILY.GUILD = 'CF'
		GROUP BY dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER) AS
QCF1
	ON (QSP1.EVENT_ID = QCF1.EVENT_ID) AND (QSP1.SAMPLE_NUMBER = QCF1.SAMPLE_NUMBER)

	PRINT 'Step 60: Create Qry_Scraper_To_Collect_Filter - Completed Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 60: Create Qry_Scraper_To_Collect_Filter - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

--Step 61: Create Qry_Ratio_Shredder_To_Gatherer
BEGIN TRY
	BEGIN TRANSACTION

	PRINT 'Step 61: Create Qry_Ratio_Shredder_To_Gatherer'

  INSERT INTO dbo.Tab_Temp_Indicator (EVENT_ID, SAMPLE_NUMBER, PARAMETER, REPORTING_VALUE)
	SELECT QSP1.EVENT_ID, QSP1.SAMPLE_NUMBER, 'RATIO_SH_TO_CG' AS PAREMTER,
ROUND(QSP1.SumOfREPORTING_VALUE/QCG1.SumOfREPORTING_VALUE,4) AS M
	FROM (SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER,
SUM(dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE) AS SumOfREPORTING_VALUE
		FROM dbo.Tab_Temp_Family INNER JOIN dbo.TAB_FAMILY
		ON dbo.Tab_Temp_Family.FAMILY_TSN = dbo.TAB_FAMILY.TSN
		WHERE dbo.TAB_FAMILY.GUILD = 'SH'
		GROUP BY dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER) AS
QSP1
	INNER JOIN
		(SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER,
SUM(dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE) AS SumOfREPORTING_VALUE
		FROM dbo.TAB_FAMILY INNER JOIN dbo.Tab_Temp_Family
		ON dbo.TAB_FAMILY.TSN = dbo.Tab_Temp_Family.FAMILY_TSN
		WHERE dbo.TAB_FAMILY.GUILD = 'CG'
		GROUP BY dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER) AS
QCG1
	ON (QSP1.EVENT_ID = QCG1.EVENT_ID) AND (QSP1.SAMPLE_NUMBER = QCG1.SAMPLE_NUMBER)

	PRINT 'Step 61: Create Qry_Ratio_Shredder_To_Gatherer - Completed Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 61: Create Qry_Ratio_Shredder_To_Gatherer - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

--Step 62: Create Qry_Ratio_Scrape_To_Shred
BEGIN TRY
	BEGIN TRANSACTION
```

```
	PRINT 'Step 62: Create Qry_Ratio_Scrape_To_Shred'

	INSERT INTO dbo.Tab_Temp_Indicator (EVENT_ID, SAMPLE_NUMBER, PARAMETER, REPORTING_VALUE)
	SELECT QSP1.EVENT_ID, QSP1.SAMPLE_NUMBER, 'RATIO_SC_TO_SH' AS P,
ROUND(QSP1.EVENT_ID/QSHP1.SumOfREPORTING_VALUE,4) AS C
	FROM (SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER,
SUM(dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE) AS SumOfREPORTING_VALUE
				FROM dbo.Tab_Temp_Family INNER JOIN dbo.TAB_FAMILY
				ON dbo.Tab_Temp_Family.family_tsn=dbo.TAB_FAMILY.TSN
				WHERE dbo.TAB_FAMILY.GUILD = 'SC'
				GROUP BY dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER) AS
QSP1
	INNER JOIN
		(SELECT dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER,
SUM(dbo.Tab_Temp_Family.SUM_OF_REPORTING_VALUE) AS SumOfREPORTING_VALUE
				FROM dbo.Tab_Temp_Family INNER JOIN dbo.TAB_FAMILY
				ON dbo.Tab_Temp_Family.FAMILY_TSN = dbo.TAB_FAMILY.TSN
				WHERE dbo.TAB_FAMILY.GUILD = 'SH'
				GROUP BY dbo.Tab_Temp_Family.EVENT_ID, dbo.Tab_Temp_Family.SAMPLE_NUMBER) AS
QSHP1
	ON (QSP1.EVENT_ID=QSHP1.EVENT_ID) AND (QSP1.SAMPLE_NUMBER=QSHP1.SAMPLE_NUMBER)

	PRINT 'Step 62: Create Qry_Ratio_Scrape_To_Shred - Completed Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 62: Create Qry_Ratio_Scrape_To_Shred - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

----Step 63: REPLICATE "temp_indicator_raw" including "Decimal Fix"
BEGIN TRY
	BEGIN TRANSACTION

	PRINT 'Step 63: REPLICATE "temp_indicator_raw" including "Decimal Fix"'

	CREATE TABLE dbo.Tab_Temp_Indicator_Raw (EVENT_ID INT NULL, IBI_PARAMETER Varchar(150) NULL,
CALCULATED_VALUE REAL NULL, CALCULATED_UNITS Varchar(50))

	INSERT INTO  dbo.Tab_Temp_Indicator_Raw(EVENT_ID, IBI_PARAMETER, CALCULATED_VALUE,
CALCULATED_UNITS)
	SELECT dbo.Tab_Temp_Indicator.EVENT_ID, dbo.Tab_Temp_Indicator.PARAMETER,
AVG(dbo.Tab_Temp_Indicator.REPORTING_VALUE), TAB_WORKING_FIX_AVG.UNITS
	FROM dbo.Tab_Temp_Indicator INNER JOIN TAB_WORKING_FIX_AVG
	ON dbo.Tab_Temp_Indicator.PARAMETER = TAB_WORKING_FIX_AVG.PARAMETER
	GROUP BY dbo.Tab_Temp_Indicator.EVENT_ID, dbo.Tab_Temp_Indicator.PARAMETER,
TAB_WORKING_FIX_AVG.UNITS

	UPDATE dbo.Tab_Temp_Indicator_Raw
	SET dbo.Tab_Temp_Indicator_Raw.CALCULATED_VALUE =
	(
		CASE
			WHEN dbo.Tab_Temp_Indicator_Raw.CALCULATED_UNITS = 'COUNT' THEN
ROUND(dbo.Tab_Temp_Indicator_Raw.CALCULATED_VALUE, 1)
			WHEN dbo.Tab_Temp_Indicator_Raw.CALCULATED_UNITS = 'PERCENT' THEN
ROUND(dbo.Tab_Temp_Indicator_Raw.CALCULATED_VALUE, 2)
			WHEN dbo.Tab_Temp_Indicator_Raw.CALCULATED_UNITS = 'UNITLESS' AND
dbo.Tab_Temp_Indicator_Raw.IBI_PARAMETER = 'FBI' THEN
			ROUND(dbo.Tab_Temp_Indicator_Raw.CALCULATED_VALUE, 2)
			WHEN dbo.Tab_Temp_Indicator_Raw.CALCULATED_UNITS = 'UNITLESS' THEN
ROUND(dbo.Tab_Temp_Indicator_Raw.CALCULATED_VALUE, 4)

		ELSE
			ROUND(dbo.Tab_Temp_Indicator_Raw.CALCULATED_VALUE, 0)
		END
	)

	PRINT 'Step 63: REPLICATE "Tab_Temp_Indicator_raw" including "Decimal Fix" - Completed Successfully'
	COMMIT TRANSACTION
END TRY
BEGIN CATCH
	PRINT 'Transaction Failed for Step 63: REPLICATE "Tab_Temp_Indicator_raw" including "Decimal Fix" - Will Rollback'
	IF @@TRANCOUNT > 0
	ROLLBACK

	EXECUTE usp_GetErrorInfo
END CATCH

--Step 64: Update Rarieified Parameter Data with _R
BEGIN TRY
	BEGIN TRANSACTION

	PRINT 'Step 64: Update Rarieified Parameter Data with _R'

		UPDATE dbo.Tab_Temp_Indicator_Raw
		SET dbo.Tab_Temp_Indicator_Raw.IBI_PARAMETER = dbo.Tab_Temp_Indicator_Raw.IBI_PARAMETER +
'_R'
```

```
		PRINT 'Step 64: Update Rarieified Parameter Data with _R - Completed Successfully'
		COMMIT TRANSACTION
END TRY
BEGIN CATCH
		PRINT 'Transaction Failed for Step 64: Update Rarieified Parameter Data with _R - Will Rollback'
		IF @@TRANCOUNT > 0
		ROLLBACK

		EXECUTE usp_GetErrorInfo
END CATCH

--Step 65: Delete from TAB_WEARHOUSE where the Tab_Temp_Indicator_Raw Event_Id Exist in TAB_WEARHOUSE
BEGIN TRY
		BEGIN TRANSACTION

		PRINT 'Step 65: Delete from TAB_WEARHOUSE where the Tab_Temp_Indicator_Raw Event_Id Exist in
TAB_WEARHOUSE'

		DELETE TAB_WEARHOUSE
		WHERE EXISTS(SELECT EVENT_ID
					FROM dbo.Tab_Temp_Indicator_Raw
					WHERE TAB_WEARHOUSE.EVENT_ID = dbo.Tab_Temp_Indicator_Raw.EVENT_ID
AND TAB_WEARHOUSE.IBI_PARAMETER = dbo.Tab_Temp_Indicator_Raw.IBI_PARAMETER)

		PRINT 'Step 65: Delete from TAB_WEARHOUSE where the Tab_Temp_Indicator_Raw Event_Id Exist in
TAB_WEARHOUSE - Completed Successfully'
		COMMIT TRANSACTION
END TRY
BEGIN CATCH
		PRINT 'Transaction Failed for Step 65: Delete from TAB_WEARHOUSE where the Tab_Temp_Indicator_Raw Event_Id
Exist in TAB_WEARHOUSE - Will Rollback'
		IF @@TRANCOUNT > 0
		ROLLBACK

		EXECUTE usp_GetErrorInfo
END CATCH

--Step 66: Insert/Append Into TAB_WEARHOUSE From Tab_Temp_Indicator_Raw
BEGIN TRY
		BEGIN TRANSACTION

		PRINT 'Step 66: Insert Into TAB_WEARHOUSE From Tab_Temp_Indicator_Raw'

		INSERT INTO TAB_WEARHOUSE (EVENT_ID, IBI_PARAMETER, CALCULATED_VALUE, CALCULATED_UNITS)
		SELECT EVENT_ID, IBI_PARAMETER, CALCULATED_VALUE, CALCULATED_UNITS
		FROM dbo.Tab_Temp_Indicator_Raw

		PRINT 'Step 66: Insert/Append Into TAB_WEARHOUSE From Tab_Temp_Indicator_Raw - Completed Successfully'
		COMMIT TRANSACTION
END TRY
BEGIN CATCH
		PRINT 'Transaction Failed for Step 66: Insert/Append Into TAB_WEARHOUSE From Tab_Temp_Indicator_Raw - Will
Rollback'
		IF @@TRANCOUNT > 0
		ROLLBACK

		EXECUTE usp_GetErrorInfo
END CATCH

GO
```