

Data Acquisition Standard Operating Procedures

Oyster Sentinel (ID# 456)

Last Updated: 2/17/2024

Program Summary

Water quality, oyster health and oyster body size collected in Charlotte Harbor and Apalachicola Bay. Water temperature (T) and salinity (S) are collected above the oyster reef at the time of the collection. (We have used mercury thermometers and YSI T/S meters for the collection of T and refractometers and YSI T/S meters for the determination of S.) We try to collect at least 10 oysters < 75mm and 10 oysters > 75 mm. Length is measured to the nearest mm.

URLs

- Program - <https://www.shellbudget.org/>
- DDI - <https://data.florida-seacar.org/programs/details/456>

Contacts

Contact Name	Organization	Email	Phone
David X Gallegos		dgallego@my.uno.edu	
Thomas M Soniat	University of New Orleans Professor/Research	tsoniat@uno.edu	504.280.1324

Data Tables

- Data_456A_Final
- Data_456A_Load
- Data_456B_Final
- Data_456B_Load

Data Stored Procedures

- usp_combined_wq_wc_nut_insert_456A

Data Acquisition Standard Operating Procedures: ProgramID 456

Date Created: 04/23/2019

Created By: *Claude Kershaw*

Data File Path:

1. "\\forest.usf.edu\data\PDrive\CAS-WI\Misc Projects\SEACAR_FDEP\Data\ID_456\DataToLoad\ID_0456_OysterSentinel_FloridaDermoData.xlsx"

DDI URL: <http://dev.seacar.waterinstitute.usf.edu/datadiscovery/programs/details/456>

Contact Information:

Contact Name: David X Gallegos

Contact Organization: Program - Oyster Sentinel

Contact Email: dgallego@my.uno.edu

Contact Phone:

Procedure Overview:

1. Use SQL Server Import Export Wizard to load the worksheet "Dermo Stations" into table **Locations_456A**.
2. Use SQL Server Import Export Wizard to load the worksheet "Dermo Collections" into table **Data_456A_Load**.
3. Use SQL Server Import Export Wizard to load the file "Dermo Samples" into table **Data_456B_Load**. Add the column [dermo_station_id] from table **Data_456A_Load** by joining on [dermo_collection_id] and [id].
4. Execute procedure usp_Data_456*_Load_insert to load the data into table **Data_456*_Final**.
5. Add the Monitoring Locations from tables **Locations_456A** to the **SampleLocation_Point** table if they do not exist there already.
6. Add new Monitoring Locations into the **SampleLocation** table. This will generate a LocationID for each Monitoring Location.
7. Update the **SampleLocation_Point** and **SampleLocation_Line** table with the LocationID generated in the **SampleLocation** table. Run procedure usp_SampleLocation_Point_update to do this.
8. Update the LocationID column in table **Data_456A_Final** with the LocationID in the **SampleLocation** table. Join on the [dermo_station_id] column in **Data_456A_Final** and the ProgramLocationID column in **SampleLocation**.

Data Tables

1. Data_456*_Load
2. Data_456*_Final

Data Stored Procedures

1. usp_Data_456*_Load_insert
2. usp_SampleLocation_Point_update

GIS Procedures

1. The Monitoring Location information is found in the tables **Locations_456A**.
2. Complete steps 5 through 8 in the "Procedure Overview" section of this document.

```
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
```

```
CREATE PROC [dbo].[usp_combined_wq_wc_nut_insert_456A]
```

```
AS
```

```
BEGIN
```

```
SET NOCOUNT ON;
```

```
SET XACT_ABORT ON;
```

```
-- Constants - PLEASE SET NOW!!
```

```
DECLARE @dataLoadCode varchar(10) = '456A';
```

```
DECLARE @combinedTable varchar(50) = 'Combined_WQ_WC_NUT';
```

```
DECLARE @ParameterID int;
```

```
DECLARE @TotalRows int;
```

```
ÿ
```

```
-- Setup data load
```

```
DECLARE @runBy varchar(50) = SYSTEM_USER;
```

```
DECLARE @programID int, @dataStreamID int;
```

```
SELECT @dataStreamID = DataStreamID,
```

```
@programID = ProgramID
```

```
FROM DataStreamProcedure
```

```
WHERE DataLoadCode = @dataLoadCode;
```

```
ÿ
```

```
PRINT (@dataStreamID)
```

```
ÿ
```

```
-- Insert data
```

```
SET @TotalRows = 0
```

```
Set @ParameterID = 2--Salinity
```

```
INSERT INTO Combined_WQ_WC_NUT (ProgramID, DataStreamID, ParameterID, LocationID,
Activity_Start_Date_Time, TOTAL_DEPTH, Total_Depth_Unit, RELATIVE_DEPTH, Org_Result_Value,
VALUE_QUALIFIER, DateAdded)
```

```
SELECT @programID, @dataStreamID, @ParameterID, a.LocationID, CAST(a.collected_at as datetime),
```

```
dbo.udf_convert_units('ft', 'm', a.depth_feet), 'm', NULL, a.salinity_ppt, NULL, GETDATE()
```

```
FROM Data_456A_Final a
```

```
Where a.salinity_ppt is not null
```

```
and a.LocationID is not null
```

```
SET @TotalRows = @TotalRows + @@ROWCOUNT
```

```
exec usp_combined_data_tracking_insert @parameterID = @ParameterID, @ProgramID = @programID,
@dataStreamID = @dataStreamID, @CombinedTableName = @combinedTable, @NumRowsFinal = @TotalRows,
@LastUpdateBy = @runBy
```

```
-- Insert data
```

```
SET @TotalRows = 0
```

```
Set @ParameterID = 3--Temp
```

```
INSERT INTO Combined_WQ_WC_NUT (ProgramID, DataStreamID, ParameterID, LocationID,
Activity_Start_Date_Time, TOTAL_DEPTH, Total_Depth_Unit, RELATIVE_DEPTH, Org_Result_Value,
VALUE_QUALIFIER, DateAdded)
```

```
SELECT @programID, @dataStreamID, @ParameterID, a.LocationID, CAST(a.collected_at as datetime),
```

```
dbo.udf_convert_units('ft', 'm', a.depth_feet), 'm', NULL, a.temperature_celsius, NULL, GETDATE()
```

```
FROM Data_456A_Final a
```

```
Where a.temperature_celsius is not null
```

```
and a.LocationID is not null
```

```
SET @TotalRows = @TotalRows + @@ROWCOUNT
```

```
exec usp_combined_data_tracking_insert @parameterID = @ParameterID, @ProgramID = @programID,
@dataStreamID = @dataStreamID, @CombinedTableName = @combinedTable, @NumRowsFinal = @TotalRows,
```

@LastUpdateBy = @runBy

```
/*
SELECT *
FROM Combined_WQ_WC_NUT

SELECT *
FROM Data_456A_Final

SELECT *
FROM Combined_Parameters a
join Indicator b on a.IndicatorID = b.IndicatorID
where b.Habitat = 'Water Column'
and b.IndicatorName <> 'Nekton'

SELECT *
FROM DataStreamProcedure
WHERE ProgramID = 456

SELECT *
FROM Combined_Data_tracking
where programid = 456

exec usp_delete_combined 1260, 'Combined_WQ_WC_NUT'
exec [usp_combined_wq_wc_nut_insert_456A]
*/

END
```

GO