Data Acquisition Standard Operating Procedures

Big Bend Seagrasses & Nature Coast Aquatic Preserves - Seagrass Monitoring (ID# 560)

Last Updated: 2/17/2024

Program Summary

Seagrass Monitoring

URLs

- Program - https://floridadep.gov/fco/aquatic-preserve/locations/big-bend-seagrasses-aquatic-preserve
- DDI - https://data.florida-seacar.org/programs/details/560

Contacts

Contact Name	Organization	Email	Phone
Timothy Jones	Big Bend Seagrasses Aquatic Preserves Manager	Timothy.W.Jones@dep.state.fl.us	
Trisha Green	Big Bend Seagrasses Aquatic Preserves - Environmental Specialist	Trisha.Green@dep.state.fl.us	
Morgan Edwards	Biologist - Nature Coast Aquatic Preserve	morgan18@ufl.edu	
Jamie Hammond	Biologist - Nature Coast Aquatic Preserve	jmelyn1987@ufl.edu	
Katherine Suchanec	Big Bend Seagrasses Aquatic Preserves - Environmental Specialist I	Katherine.Suchanec@FloridaDEP.gov	352-228-6032

Data Tables

- Data_560A_Final
- Data_560A_Final
- Data_560A_Load
- Data_560A_Load
- Data_560A_Load_temp_oldstyle
- Data_560B_Final
- Data_560B_Load
- Data_560C_Final
- Data_560C_Load

Data Stored Procedures

- usp_Data_560A_Load_insert
- usp_Data_560A_Load_insert
- usp_Data_560B_Load_insert
- usp_Data_560C_Load_insert
- usp_combined_sav_insert_560A
- usp_combined_sav_insert_560A
- usp_combined_wq_wc_nut_insert_560B
- usp_combined_wq_wc_nut_insert_560C

Data Acquisition Standard Operating Procedures: ProgramID *560*

Date Created: *11/05/2020*

Created By: *Claude Kershaw*

Date Modified: *01/16/2024*

Created By: *Jennifer Baker*

Data File Paths:

1. Data Path: "U:\MiscProjects\SEACAR_FDEP\Data\ID_0560_BBSAP_SAV\DataToLoad\ "
2. Data: NCAP_Program560_SAVdata_YYYYPOR_AllSystems.xlsx
3. Data: BBSAP_Program560_WaterQualityData_YYYY_AllSystems.xlsx
4. Data: BBSAP_Program560_SAVdata_YYYY_AllSystems.xlsx
5. Data: GPS Coordinates for BBSAP and NCAP Stations YYYY.xlsx

DDI URL: http://dev.seacar.waterinstitute.usf.edu/datadiscovery/programs/details/560

Procedure Overview:

***Note: This program is "is essentially two programs" if the data *looks* like POR, it is **NOT** truly POR for ALL of the data. **DO NOT** truncate the final table. You can truncate the load and use the where not exists exception in the load proc to filter out duplicated data. (as of 11/29/2021) **Also**, watch for station names. ***

1. Use SQL Server Import Export Wizard to load data file "NCAP_Program560_SAVdata_YYYYPOR_AllSystems.xlsx" and "BBSAP_Program560_SAVdata_YYYY_AllSystems.xlsx" for **SAV Data** into **Data_560A_Load.**
2. Use SQL Server Import Export Wizard to load data file "NCAP_Program560_SAVdata_YYYYPOR_AllSystems.xlsx" for **Water Quality Data** into **Data_560B_Load.**
    a. **NOTE: VERT_SECCHI_m is mapped to Secchi**
3. Use SQL Server Import Export Wizard to load data file "BBSAP_Program560_WaterQualitydata_YYYY_AllSystems.xlsx" for **Water Quality Data** into **Data_560C_Load.**
    a. **NOTE: Depth(m) is mapped to Secchi**
4. Load the locations from "Master Lat Long Spreadsheet.xlsx" into **Locations_560A**.
5. Use SQL Server Import Export Wizard to load the file "**BBSAP_Program560_WaterQualityData_YYYY_AllSystems.xlsx**" into table **Data_560C_Load.**
6. Execute procedure usp_Data_560*_Load_insert to load the data into table **Data_560*_Final.**

7. Add new Monitoring Locations into the **SampleLocation_Point** table from the **Locations_560A table**.
8. Add new Monitoring Locations into the **SampleLocation_Point** table from the **Data_560*_Final tables**.
9. Add new Monitoring Locations into the **SampleLocation** table. This will generate a LocationID for each Monitoring Location.
10. Update the **SampleLocation_Point** table with the LocationID generated in the **SampleLocation** table. Run procedure usp_SampleLocation_Point_update to do this.
11. Update the LocationID column in table **Data_560A_Final** with the LocationID in the **SampleLocation** table. Join on the [STATION] column in **Data_560A_Final** and the ProgramLocationID column in **SampleLocation.**
12. Update the LocationID column in table **Data_560B_Final** with the LocationID in the **SampleLocation** table. Join on the [STATION] column in **Data_560B_Final** and the ProgramLocationID column in **SampleLocation.**

Data Tables

1. Data_560*_Load
2. Data_560*_Final
3. Locations_560A

Data Stored Procedures

1. usp_Data_560*_Load_insert
2. usp_SampleLocation_Point_update

GIS Procedures

1. The Monitoring Location information can be found in the Load table **Data_560*_Load.**
2. Complete steps 8 through 14 in the “Procedure Overview” section of this document.

```
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
CREATE PROC [legacy].[usp_combined_sav_insert_560A]
AS
BEGIN
SET NOCOUNT ON;
SET XACT_ABORT ON;

-- Delete Existing Data
exec usp_delete_combined 16, 'Combined_SAV'

-- Constants - PLEASE SET NOW!!
DECLARE @dataLoadCode varchar(10) = '560A';
DECLARE @combinedTable varchar(50) = 'Combined_SAV'
DECLARE @parameterID int

-- Setup data load
DECLARE @runBy varchar(50) = SYSTEM_USER;
DECLARE @programID int, @dataStreamID int;

SELECT@dataStreamID = DataStreamID,
@programID = ProgramID
FROMDataStreamProcedure
WHEREDataLoadCode = @dataLoadCode;
ÿ
-- Insert data
SET @parameterID = 22

INSERT INTO Combined_SAV (ProgramID, DataStreamID, ParameterID, LocationID,  SampleDate, SpeciesID
SamplingMethod1, SamplingMethod2, ReportingLevel, QuadSize_m2, Grid, ResultValue, Depth_M, DateAdd
QuadIdentifier, SiteIdentifier)
SELECT@programID, @dataStreamID, @parameterID, a.LocationID, DATEFROMPARTS(YEAR, MONTH(a.MONTH +
' 1 2000'), 1), c.SpeciesID,  'Random', 'Natural', 'Quadrat', 1.00, NULL, a.COVERAGE, NULL, GETDAT
 [REPLICATE],  STATION
FROMData_560A_Finala
INNER JOINref_conversion_species b on a.SPECIES = b.OriginalCommonIdentifier AND b.DataStreamID =
@dataStreamID
INNER JOIN  ref_species c on b.SpeciesID = c.SpeciesID
WHEREa.COVERAGE IS NOT NULL
ANDa.locationid is not null
ÿ
--exec usp_combined_data_tracking_insert @parameterID = @parameterID, @ProgramID = @programID,
@dataStreamID = @dataStreamID, @CombinedTableName = @combinedTable, @NumRowsFinal = @@ROWCOUNT,
@LastUpdateBy = @runBy

-- Insert data
SET @parameterID = 22

INSERT INTO Combined_SAV (ProgramID, DataStreamID, ParameterID, LocationID,  SampleDate, SpeciesID
SamplingMethod1, SamplingMethod2, ReportingLevel, QuadSize_m2, Grid, ResultValue, Depth_M,  DateAd
 QuadIdentifier, SiteIdentifier)
SELECT@programID, @dataStreamID, @parameterID, a.LocationID,  DATEFROMPARTS(YEAR, MONTH(a.MONTH
+ ' 1 2000'), 1), b.SpeciesID, 'Random', 'Natural', 'Quadrat', 1.00, NULL, a.TOTAL_GRASS, NULL,
GETDATE(), [REPLICATE], STATION
FROMData_560A_Finala
INNER JOINref_conversion_species b on 'TOTAL_GRASS' = b.OriginalCommonIdentifier AND b.DataStreamI
= @dataStreamID
WHEREa.TOTAL_GRASS IS NOT NULL
ANDa.locationid is not null
ÿ
--exec usp_combined_data_tracking_insert @parameterID = @parameterID, @ProgramID = @programID,
@dataStreamID = @dataStreamID, @CombinedTableName = @combinedTable, @NumRowsFinal = @@ROWCOUNT,
@LastUpdateBy = @runBy

-- Insert data
```

```
SET @parameterID = 22

INSERT INTO Combined_SAV (ProgramID, DataStreamID, ParameterID, LocationID,  SampleDate, SpeciesID
SamplingMethod1, SamplingMethod2, ReportingLevel, QuadSize_m2, Grid, ResultValue, Depth_M, DateAdd
QuadIdentifier, SiteIdentifier)
SELECT@programID, @dataStreamID, @parameterID, a.LocationID,  DATEFROMPARTS(YEAR, MONTH(a.MONTH
+ ' 1 2000'), 1), b.SpeciesID,  'Random', 'Natural', 'Quadrat', 1.00, NULL, a.TOTAL_MACROALGAE, NU
GETDATE(),  [REPLICATE], STATION
FROMData_560A_Finala
INNER JOINref_conversion_species b on 'TOTAL_MACROALGAE' = b.OriginalCommonIdentifier AND
b.DataStreamID = @dataStreamID
WHEREISNUMERIC(a.TOTAL_MACROALGAE) = 1
ANDa.locationid is not null
ÿ
--exec usp_combined_data_tracking_insert @parameterID = @parameterID, @ProgramID = @programID,
@dataStreamID = @dataStreamID, @CombinedTableName = @combinedTable, @NumRowsFinal = @@ROWCOUNT,
@LastUpdateBy = @runBy

/*
SELECT *
FROM Combined_SAV

SELECT Distinct ProgramID, b.IndicatorName, c.ParameterName, NumRowsCombined
FROM Combined_Data_Tracking a
INNER JOIN Indicator b on a.IndicatorID = b.IndicatorID
INNER JOIN Combined_Parameters c on a.ParameterID = c.ParameterID
WHERE b.Habitat = 'Submerged Aquatic Vegetation'

SELECT *
FROM Data_560A_Final

SELECT *
FROM Combined_Parameters a
INNER JOIN Indicator b on a.IndicatorID = b.IndicatorID
where b.Habitat = 'Submerged Aquatic Vegetation'

SELECT *
FROM DataStreamProcedure
WHERE ProgramID = 560

SELECT *
FROM ref_species
where habitat = 'Submerged Aquatic Vegetation'
and scientificname like '%calcer%'

SELECT *
FROM ref_conversion_species
WHERE ProgramID = 560

exec usp_delete_combined 16, 'Combined_SAV'
*/

END




GO
```

```
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
CREATE PROC [legacy].[usp_Data_560A_Load_insert]
AS
BEGIN
SET NOCOUNT ON
SET XACT_ABORT ON

INSERT INTO Data_560A_Final (SYSTEM, DAY, MONTH, YEAR, STATION, REPLICATE, SPECIES, COVERAGE, SCAL
URCHINS, GRASS_ALGAE, TOTAL_SAV, TOTAL_GRASS, TOTAL_MACROALGAE, EPIPHYTE_DENSITY, SEDIMENT_TYPE,
COMMENTS, TEMP_C, SAL_ppt, DO_mg_L, pH_SU, CANOPY_HEIGHT_1, CANOPY_HEIGHT_2, CANOPY_HEIGHT_3, DEPT
Percent_Coverage)
SELECT SYSTEM, DAY, MONTH, YEAR, STATION, REPLICATE, SPECIES, COVERAGE, SCALLOPS, URCHINS,
GRASS_ALGAE, TOTAL_SAV, TOTAL_GRASS, TOTAL_MACROALGAE, EPIPHYTE_DENSITY, SEDIMENT_TYPE, COMMENTS,
TEMP_C, SAL_ppt, DO_mg_L, pH_SU, CANOPY_HEIGHT_1, CANOPY_HEIGHT_2, CANOPY_HEIGHT_3, DEPTH_m,
Percent_Coverage
FROMData_560A_Loada
WHERENOT EXISTS (SELECT *
                  FROM Data_560A_Final b
  WHERE ISNULL(a.[DAY], '') = ISNULL(b.[DAY],'')
  AND ISNULL(a.[MONTH],'') = ISNULL(b.[MONTH],'')
  AND ISNULL(a.[YEAR],'') = ISNULL(b.[YEAR],'')
  AND ISNULL(a.[SYSTEM],'') = ISNULL(b.[SYSTEM],'')
  AND ISNULL(a.STATION,'') = ISNULL(b.STATION,'')
  AND ISNULL(a.SPECIES, '') = ISNULL(b.SPECIES, '')
  )

-- AFTER Matching to SampleLocation to populate LocationID - run these updates to standardize the
Station Name
/*
update Data_560A_Final
set station = replace(station,'ck0','ck')
where LocationID is null
and station like 'ck0%'
ÿ
update Data_560A_Final
set station = replace(station,'DBKB ','DBKB')
where LocationID is null
and station like 'DBKB %'
ÿ
update Data_560A_Final
set station = replace(station,'SMAR ','SMAR')
where LocationID is null
and station like 'SMAR %'
ÿ
update Data_560A_Final
set station = replace(station,'SMAR','SMAR0')
where LocationID is null
and station in (
'SMAR1'
,'SMAR2'
,'SMAR3'
,'SMAR4'
,'SMAR5'
,'SMAR6'
,'SMAR7'
,'SMAR8'
,'SMAR9'
)
*/

END
```

```
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
CREATE PROC [dbo].[usp_combined_sav_insert_560A]
AS
BEGIN
SET NOCOUNT ON;
SET XACT_ABORT ON;

-- Constants - PLEASE SET NOW!!
DECLARE @dataLoadCode varchar(10) = '560A';
DECLARE @combinedTable varchar(50) = 'Combined_SAV'
DECLARE @parameterID int

-- Setup data load
DECLARE @runBy varchar(50) = SYSTEM_USER;
DECLARE @programID int, @dataStreamID int;

SELECT@dataStreamID = DataStreamID,
@programID = ProgramID
FROMDataStreamProcedure
WHEREDataLoadCode = @dataLoadCode;
ÿ
--delete existing data
exec usp_delete_combined @datastreamID, @combinedTable

--Insert Data
INSERT INTO Combined_SAV (ProgramID, DataStreamID, ParameterID, LocationID,  SampleDate, SpeciesID
SamplingMethod1, SamplingMethod2, ReportingLevel, QuadSize_m2, Grid, ResultValue, Depth_M, DateAdd
QuadIdentifier, SiteIdentifier)
SELECT@programID, @dataStreamID, ccp.targetparameterID, a.LocationID, SampleDate, c.SpeciesID,
'Random', 'Natural', 'Quadrat', 1.00, NULL, a.ResultValue, Depth_M, GETDATE(),  [REPLICATE],  STAT
FROMData_560A_Finala
INNER JOINref_conversion_species b on a.SPECIES = b.OriginalCommonIdentifier AND b.DataStreamID =
@dataStreamID and OriginalColumnName = 'SPECIES'
INNER JOIN  ref_species c on b.SpeciesID = c.SpeciesID
INNER JOINCombined_Conversion_Parameters ccp on ccp.OriginalParameter = a.MeasuredParameter and
ccp.DatastreamID = @dataStreamID
WHEREa.ResultValue IS NOT NULL
ANDa.locationid is not null
ANDa.SampleDate is not null
ANDb.SpeciesID is not null
ÿ
ÿ


/*
SELECT *
FROM Combined_SAV

SELECT Distinct ProgramID, b.IndicatorName, c.ParameterName, NumRowsCombined
FROM Combined_Data_Tracking a
INNER JOIN Indicator b on a.IndicatorID = b.IndicatorID
INNER JOIN Combined_Parameters c on a.ParameterID = c.ParameterID
WHERE b.Habitat = 'Submerged Aquatic Vegetation'

SELECT *
FROM Data_560A_Final

SELECT *
FROM Combined_Parameters a
INNER JOIN Indicator b on a.IndicatorID = b.IndicatorID
where b.Habitat = 'Submerged Aquatic Vegetation'

SELECT *
FROM DataStreamProcedure
```

```
WHERE ProgramID = 560

SELECT *
FROM ref_species
where habitat = 'Submerged Aquatic Vegetation'
and scientificname like '%calcer%'

SELECT *
FROM ref_conversion_species
WHERE ProgramID = 560

exec usp_delete_combined 16, 'Combined_SAV'
*/

END









GO
```

```
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON

CREATE PROC [dbo].[usp_combined_wq_wc_nut_insert_560B]
AS
BEGIN
SET NOCOUNT ON;
SET XACT_ABORT ON;

-- Constants - PLEASE SET NOW!!
DECLARE @dataLoadCode varchar(10) = '560B';
DECLARE @combinedTable varchar(50) = 'Combined_wq_wc_nut'
DECLARE @parameterID int

-- Setup data load
DECLARE @runBy varchar(50) = SYSTEM_USER;
DECLARE @programID int, @dataStreamID int;

SELECT@dataStreamID = DataStreamID,
@programID = ProgramID
FROMDataStreamProcedure
WHEREDataLoadCode = @dataLoadCode;
ÿ
--delete existing data
exec usp_delete_combined @datastreamID, @combinedTable

INSERT INTO Combined_WQ_WC_NUT (ProgramID, DataStreamID,ParameterID,LocationID,ActivityType,
SampleDate, ResultValue)
SELECT @programID,@dataStreamID, b.TargetParameterID, LocationID, 'Sample', SampleDate , Result
FROM Data_560B_Final a
LEFT JOIN Combined_Conversion_Parameters b on b.OriginalParameter = a.MeasuredParameter and
b.DatastreamID = @dataStreamID
WHERE   ISNUMERIC(result) = 1
ANDa.LocationID IS NOT NULL
ANDa.SampleDate IS NOT NULL
ANDb.TargetParameterID IS NOT NULL




/*
SELECT distinct Bottom
FROM Data_560B_Final

SELECT distinct RelativeDepth
FROM Combined_WQ_WC_NUT

SELECT distinct MeasuredParameter
FROM data_560b_final

SELECT *
FROM Combined_Parameters a
INNER JOIN Indicator b on a.IndicatorID = b.IndicatorID
where b.Habitat = 'Water Column'
SELECT *
FROM DataStreamProcedure where programid = 560
ÿ
SELECT *
FROM Combined_Conversion_Parameters
WHERE datastreamid = 1444

insert into Combined_Conversion_Parameters (DatastreamID, OriginalParameter, OriginalUnits,
TargetParameterID, TargetParameterUnits, ParameterConversion)
values
(1444,'TEMP_C', 'C', '3','Degrees C',NULL ),
```

```
(1444,'SAL_ppt', 'ppt', '2','ppt',NULL ),
(1444,'pH_SU', 'SU', '4','None',NULL ),
(1444,'DO_mg_L', 'mg_L', '1','mg/L',NULL )

*/

END

GO
```

```
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
CREATE PROC [dbo].[usp_combined_wq_wc_nut_insert_560C]
AS
BEGIN
SET NOCOUNT ON;
SET XACT_ABORT ON;

-- Constants - PLEASE SET NOW!!
DECLARE @dataLoadCode varchar(10) = '560C';
DECLARE @combinedTable varchar(50) = 'Combined_wq_wc_nut'
DECLARE @parameterID int

-- Setup data load
DECLARE @runBy varchar(50) = SYSTEM_USER;
DECLARE @programID int, @dataStreamID int;

SELECT@dataStreamID = DataStreamID,
@programID = ProgramID
FROMDataStreamProcedure
WHEREDataLoadCode = @dataLoadCode;
ÿ
--delete existing data
exec usp_delete_combined @datastreamID, @combinedTable

INSERT INTO Combined_WQ_WC_NUT (ProgramID, DataStreamID,ParameterID,LocationID,ActivityType,
SampleDate, ResultValue)
SELECT @programID,@dataStreamID, b.TargetParameterID, LocationID, 'Sample', SampleDate , Result
FROM Data_560C_Final a
LEFT JOIN Combined_Conversion_Parameters b on b.OriginalParameter = a.MeasuredParameter and
b.DatastreamID = @dataStreamID
WHERE   ISNUMERIC(result) = 1
ANDa.LocationID IS NOT NULL
ANDa.SampleDate IS NOT NULL
ANDb.TargetParameterID IS NOT NULL




/*
SELECT distinct Bottom
FROM Data_560B_Final

SELECT distinct RelativeDepth
FROM Combined_WQ_WC_NUT

SELECT distinct MeasuredParameter
FROM data_560b_final

SELECT *
FROM Combined_Parameters a
INNER JOIN Indicator b on a.IndicatorID = b.IndicatorID
where b.Habitat = 'Water Column'
SELECT *
FROM DataStreamProcedure where programid = 560
ÿ
SELECT *
FROM Combined_Conversion_Parameters

insert into Combined_Conversion_Parameters (DatastreamID, OriginalParameter, OriginalUnits,
TargetParameterID, TargetParameterUnits, ParameterConversion)
values
(1444,'TEMP_C', 'C', '3','Degrees C',NULL ),
(1444,'SAL_ppt', 'ppt', '2','ppt',NULL ),
(1444,'pH_SU', 'SU', '4','None',NULL ),
```

```
(1444,'DO_mg_L', 'mg_L', '1','mg/L',NULL )

*/

END

GO
```

```
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
CREATE PROC [dbo].[usp_Data_560A_Load_insert]
AS
BEGIN
SET NOCOUNT ON
SET XACT_ABORT ON
ÿ

ÿ
--Pivoting load table into the final table for 560A
INSERT INTO Data_560A_Final (SYSTEM, DAY, MONTH, YEAR, STATION, REPLICATE, SPECIES,  SCALLOPS, URC
GRASS_ALGAE, EPIPHYTE_DENSITY, SEDIMENT_TYPE, CANOPY_HEIGHT_1, CANOPY_HEIGHT_2, CANOPY_HEIGHT_3,
COMMENTS, sampledate, ResultValue, MeasuredParameter)
SELECTSYSTEM, DAY, MONTH, YEAR, STATION, REPLICATE, SPECIES, SCALLOPS, URCHINS, GRASS_ALGAE,
EPIPHYTE_DENSITY, SEDIMENT_TYPE, CANOPY_HEIGHT_cm_1, CANOPY_HEIGHT_cm_2, CANOPY_HEIGHT_cm_3, COMME
sampledate, Result, MeasuredParameter
FROM (
SELECTSYSTEM, DAY, MONTH, YEAR, STATION, REPLICATE, SPECIES, PERCENT_COVERAGE, BB_COVERAGE,
PERCENT_DRIFT, SCALLOPS, URCHINS, GRASS_ALGAE, TOTAL_SAV_PERCENT, [TOTAL _SAV_BB] as TOTAL_SAV_BB,
TOTAL_GRASS_PERCENT, TOTAL_GRASS_BB, TOTAL_MACROALGAE_PERCENT, TOTAL_MACROALGAE_BB, EPIPHYTE_DENSI
SEDIMENT_TYPE, CANOPY_HEIGHT_cm_1, CANOPY_HEIGHT_cm_2, CANOPY_HEIGHT_cm_3, COMMENTS, sampledate
FROMData_560A_Loadload
 ) p
UNPIVOT
(Result FOR MeasuredParameter IN (PERCENT_COVERAGE, BB_COVERAGE, PERCENT_DRIFT, TOTAL_SAV_PERCENT,
TOTAL_SAV_BB, TOTAL_GRASS_PERCENT, TOTAL_GRASS_BB, TOTAL_MACROALGAE_PERCENT, TOTAL_MACROALGAE_BB))
WHERE  a.MeasuredParameter IS NOT NULL
  AND isnumeric([result]) = 1
  AND NOT EXISTS (SELECT *
                  FROM Data_560a_Final b
  WHERE ISNULL(a.[DAY], '') = ISNULL(b.[DAY],'')
  AND ISNULL(a.[MONTH],'') = ISNULL(b.[MONTH],'')
  AND ISNULL(a.[YEAR],'') = ISNULL(b.[YEAR],'')
  AND ISNULL(a.[SYSTEM],'') = ISNULL(b.[SYSTEM],'')
  AND ISNULL(a.STATION,'') = ISNULL(b.STATION,'')
  AND ISNULL(a.MeasuredParameter,'') = ISNULL(b.MeasuredParameter,'')
  )




/*
INSERT INTO Data_560A_Final (SYSTEM, DAY, MONTH, YEAR, STATION, REPLICATE, SPECIES, COVERAGE, SCAL
URCHINS, GRASS_ALGAE, TOTAL_SAV, TOTAL_GRASS, TOTAL_MACROALGAE, EPIPHYTE_DENSITY, SEDIMENT_TYPE,
COMMENTS)
SELECT SYSTEM, DAY, MONTH, YEAR, STATION, REPLICATE, SPECIES, PERCENT_COVERAGE, SCALLOPS,
URCHINS, GRASS_ALGAE, TOTAL_SAV_PERCENT, TOTAL_GRASS_PERCENT, TOTAL_MACROALGAE_PERCENT,
EPIPHYTE_DENSITY, SEDIMENT_TYPE, COMMENTS
FROMData_560A_Loada
WHERENOT EXISTS (SELECT *
                  FROM Data_560A_Final b
  WHERE ISNULL(a.[DAY], '') = ISNULL(b.[DAY],'')
  AND ISNULL(a.[MONTH],'') = ISNULL(b.[MONTH],'')
  AND ISNULL(a.[YEAR],'') = ISNULL(b.[YEAR],'')
  AND ISNULL(a.[SYSTEM],'') = ISNULL(b.[SYSTEM],'')
  AND ISNULL(a.STATION,'') = ISNULL(b.STATION,'')
  AND ISNULL(a.SPECIES, '') = ISNULL(b.SPECIES, '')
  )

--Handles the Months that are written out
UPDATE a
SET SampleDate = DATEFROMPARTS(YEAR, MONTH(trim(a.MONTH)), 1)
FROM  Data_560A_Final a
```

```
WHERE SampleDate is null
and ISNUMERIC(a.month) = 0

--Handles Months that are numeric already
UPDATE a
SET SampleDate = DATEFROMPARTS(YEAR, a.MONTH, 1)
FROM  Data_560A_Final a
WHERE SampleDate is null
and ISNUMERIC(a.month) = 1

ÿ
--Updates all LocationIDS
update a
SET a.LocationID  = b.LocationID
FROM Data_560A_Final a
INNER JOIN samplelocation b on a.STATION= b.ProgramLocationID
WHERE b.ProgramID = 560 and a.LocationID is null

*/

-- AFTER Matching to SampleLocation to populate LocationID - run these updates to standardize the
Station Name
/*
update Data_560A_Final
set station = replace(station,'ck0','ck')
where LocationID is null
and station like 'ck0%'
ÿ
update Data_560A_Final
set station = replace(station,'DBKB ','DBKB')
where LocationID is null
and station like 'DBKB %'
ÿ
update Data_560A_Final
set station = replace(station,'SMAR ','SMAR')
where LocationID is null
and station like 'SMAR %'
ÿ
update Data_560A_Final
set station = replace(station,'SMAR','SMAR0')
where LocationID is null
and station in (
'SMAR1'
,'SMAR2'
,'SMAR3'
,'SMAR4'
,'SMAR5'
,'SMAR6'
,'SMAR7'
,'SMAR8'
,'SMAR9'
)
*/

END




GO
```

```
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
CREATE PROC [dbo].[usp_Data_560B_Load_insert]
AS
BEGIN
SET NOCOUNT ON
SET XACT_ABORT ON


--For Vertical Secchi Parameter update where the value is 'B' to the Depth where the depth is not
UPDATE a
SET a.VERT_SECCHI_m = a.DEPTH_m
FROM Data_560B_Load a
WHERE a.VERT_SECCHI_m = 'B' and a.DEPTH_m is not null

--Pivoting load table into the final table for 560B
INSERT INTO Data_560B_Final (SYSTEM, DAY, MONTH, YEAR, STATION, DATA_RECORDER, MeasuredParameter,
Result, NOTES)
SELECT SYSTEM, DAY, MONTH, YEAR, STATION, DATA_RECORDER, MeasuredParameter, Result, NOTES
FROM (
SELECT SYSTEM, DAY, MONTH, YEAR, STATION, DATA_RECORDER, TEMP_C, [DO_mg/L], SAL_ppt, pH_SU,
VERT_SECCHI_m, HORIZ_SECCHI_m, DEPTH_m, NOTES
FROMData_560B_Loadload
 ) p
UNPIVOT
(Result FOR MeasuredParameter IN (TEMP_C, [DO_mg/L], SAL_ppt, pH_SU, VERT_SECCHI_m, HORIZ_SECCHI_m
DEPTH_m)) a
WHERENOT EXISTS (SELECT *
                  FROM Data_560B_Final b
  WHERE ISNULL(a.[DAY], '') = ISNULL(b.[DAY],'')
  AND ISNULL(a.[MONTH],'') = ISNULL(b.[MONTH],'')
  AND ISNULL(a.[YEAR],'') = ISNULL(b.[YEAR],'')
  AND ISNULL(a.[SYSTEM],'') = ISNULL(b.[SYSTEM],'')
  AND ISNULL(a.STATION,'') = ISNULL(b.STATION,'')
  AND ISNULL(a.MeasuredParameter,'') = ISNULL(b.MeasuredParameter,'')
  )
  AND a.MeasuredParameter IS NOT NULL


ÿ
----Handles the Months that are written out
--UPDATE a
--SET SampleDate = DATEFROMPARTS(YEAR, MONTH(trim(a.MONTH)), 1)
--FROM  Data_560B_Final a
--WHERE SampleDate is null
--and ISNUMERIC(a.month) = 0

--Handles Months that are numeric already
UPDATE a
SET SampleDate = DATEFROMPARTS(a.YEAR, a.MONTH, ISNULL(a.Day, 1))
FROM  Data_560B_Final a
WHERE SampleDate is null
and ISNUMERIC(a.month) = 1

--Updates all LocationIDS
update a
SET a.LocationID  = b.LocationID
FROM Data_560B_Final a
INNER JOIN samplelocation b on a.STATION= b.ProgramLocationID
WHERE b.ProgramID = 560 and a.LocationID is null



/*
```

```
SELECT *
FROM Data_560B_load

SELECT *
FROM Data_560B_final

*/

END




GO
```

```
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
CREATE PROC [dbo].[usp_Data_560C_Load_insert]
AS
BEGIN
SET NOCOUNT ON
SET XACT_ABORT ON



INSERT INTO Data_560C_Final (SYSTEM, DAY, MONTH, YEAR, STATION, REPLICATE, MeasuredParameter, Resu
SELECT SYSTEM, DAY, MONTH, YEAR, STATION, REPLICATE, MeasuredParameter, Result
FROM (
SELECTSYSTEM, DAY, MONTH, YEAR, STATION, Replicate, TEMP_C, SAL_ppt, [DO_mg L] as [DO_mg_L], pH_SU
[Depth (m)] as SECCHI_Depth_m
FROMData_560C_Loadload
 ) p
UNPIVOT
(Result FOR MeasuredParameter IN (TEMP_C, SAL_ppt, [DO_mg_L], pH_SU,  SECCHI_Depth_m)) a
WHERE a.MeasuredParameter IS NOT NULL
  and NOT EXISTS
(SELECT *
FROM Data_560C_Final b
WHERE ISNULL(a.[DAY], '') = ISNULL(b.[DAY],'')
AND ISNULL(a.[MONTH],'') = ISNULL(b.[MONTH],'')
AND ISNULL(a.[YEAR],'') = ISNULL(b.[YEAR],'')
AND ISNULL(a.[SYSTEM],'') = ISNULL(b.[SYSTEM],'')
AND ISNULL(a.STATION,'') = ISNULL(b.STATION,'')
AND ISNULL(a.MeasuredParameter,'') = ISNULL(b.MeasuredParameter,'')
)

--Handles the Months that are written out
UPDATE a
SET SampleDate = DATEFROMPARTS(YEAR, MONTH(MONTH( a.MONTH + ' 1 2000') ), 1)
FROM  Data_560C_Final a
WHERE SampleDate is null
and ISNUMERIC(a.month) = 0


--Handles Months that are numeric already
UPDATE a
SET SampleDate = DATEFROMPARTS(YEAR, a.MONTH, 1)
FROM  Data_560C_Final a
WHERE SampleDate is null
and ISNUMERIC(a.month) = 1

--Updates all LocationIDS
update a
SET a.LocationID  = b.LocationID
FROM Data_560C_Final a
INNER JOIN samplelocation b on replace(a.STATION,' ','')= replace(b.ProgramLocationID,' ' ,'')
WHERE b.ProgramID = 560 and a.LocationID is null



ÿ

END
```

GO