# Centralized Cruise Database - Data Integration SOP

## Overview:

The Centralized Cruise Database (CCD) was developed to manage all operational cruise-level information in the PIFSC enterprise database that other scientific data schemas can reference directly. The CCD Data Integration SOP was developed to define the process of integrating the CCD data into any scientific enterprise database schema with cruise-based data collection using Cruise Legs as the integration point. Integrating data across database schemas has some additional considerations that should be handled with a standard approach to ensure maintainability and security. In this SOP the CCD will be referred to as the Source Schema and the scientific data schema that will reference the CCD will be referred to as the Referring Schema. The Centralized CTD Data System was integrated with the CCD and is provided as an example of a Referring Schema in the SOP.

## Resources:

* [Centralized Cruise Database - Technical Documentation](Centralized%20Cruise%20Database%20-%20Technical%20Documentation.docx)
* Git Repository URL: [git@gitlab.pifsc.gov:centralized-data-tools/centralized-cruise-database.git](mailto:git@gitlab.pifsc.gov:centralized-data-tools/centralized-cruise-database.git)

## Procedure:

### Grant the Referring Schema permissions

* + Grant the minimum required permissions to access the appropriate CCD Source Schema objects
    - Open [grant\_external\_schema\_privs.sql](../SQL/queries/grant_external_schema_privs.sql)
    - Replace the [EXTERNAL SCHEMA] placeholders with the Referring Schema name (e.g. CEN\_CTD).
    - Execute the generated permissions script on the CCD Source Schema (CEN\_CRUISE).

### Update the Referring Schema object(s)

* + Update the appropriate Referring Schema object(s) to reference the CCD Cruise Leg (CEN\_CRUISE.CCD\_CRUISE\_LEGS) table.
    - Open [integrate\_external\_data\_set.sql](../SQL/queries/integrate_external_data_set.sql)
    - Replace the defined placeholders based on the comments in the SQL file for as many objects in the Referring Schema that must directly reference the CCD Cruise Leg table.
    - Execute the generated DDL script on the Referring Schema (e.g. CEN\_CTD) to define the foreign key relationship for each object. Referential integrity will be enforced for all foreign key references to CCD Cruise Legs.

### Develop methods to associate the CCD Cruise Leg data

* + Develop methods to associate the Referring Schema data with the appropriate CCD Cruise Leg data by setting the foreign key value (CRUISE\_LEG\_ID) of the Referring Schema data records. For this particular data integration it is recommended to utilize the CCD CCD\_CRUISE\_PKG.LEG\_ALIAS\_TO\_CRUISE\_LEG\_ID\_FN external package function to determine the foreign key value of the given cruise leg/cruise leg alias so that developers are not required to know about the implementation details of the CCD Cruise Leg business rules. This approach allows the foreign key value to be set in an application, script, or query. There are multiple options for associating the Referring Schema data with the corresponding Cruise Leg data, the exact implementation will depend on the use case:
    - Insert/Update queries can be used to associate the Referring Schema data with the CCD data. These queries can be implemented directly in PL/SQL scripts, packages, functions, and procedures, or in applications, or by executing them directly on the Referring Schema if there is a readily available cruise leg alias that can be translated using the CCD CCD\_CRUISE\_PKG.LEG\_ALIAS\_TO\_CRUISE\_LEG\_ID\_FN package function.
    - In simple examples an end-user application can implement the Cruise Leg as a drop down field by directly querying from the appropriate CCD views.
      * For example a web interface can implement the CCD CCD\_CRUISE\_LEGS\_V view from the CCD to populate a drop down list with Cruise Leg information and use the selected Cruise Leg's associated primary key value to create the association. This approach does not require the use of the CCD CCD\_CRUISE\_PKG.LEG\_ALIAS\_TO\_CRUISE\_LEG\_ID\_FN package function.
    - The CCD external package function can be implemented within PL/SQL package(s)/function(s) defined within the Referring Schema so the schema-specific logic can reside there.
      * For example the CTD Data System's CTD\_PKG.PARSE\_CRUISE\_LEG\_FROM\_PATH package function extracts the Cruise Leg alias from a CTD file's directory path using specific business rules and then utilizes the CCD CCD\_CRUISE\_PKG.LEG\_ALIAS\_TO\_CRUISE\_LEG\_ID\_FN external package function to translate the parsed value to the CRUISE\_LEG\_ID so the Referring Schema data objects can be associated with the corresponding CCD Cruise Legs.
    - The Referring Schema package function(s) or the CCD package function can be implemented in PL/SQL package(s)/procedure(s) to associate the Referring Schema data with the corresponding CCD Cruise Legs.
      * For example the CTD Data System's CTD\_PKG.REFRESH\_CAST\_CRUISE\_LEG package procedure utilizes the CTD Data System's CTD\_PKG.PARSE\_CRUISE\_LEG\_FROM\_PATH package function to resolve the CRUISE\_LEG\_ID value and use it to update the existing CTD cast's associated Cruise Leg.
      * Best practice: develop an automated method to detect the missing/mismatched Cruise Leg associations in the Referring Schema objects and refresh the associations with the appropriate CCD Cruise Legs.
        + For example the CTD Data System's CTD\_PKG.REFRESH\_BLANK\_MIS\_CASTS package procedure queries for all CTD casts that have blank/mismatched Cruise Legs and executes the CTD Data System's CTD\_PKG.REFRESH\_CAST\_CRUISE\_LEG package procedure for those casts to update the Cruise Leg information. This package procedure is also available for execution in the CTD web application by authorized users.
    - In more complex examples an application can implement the Referring Schema's functions/procedures to associate the Referring Schema objects with the CCD Cruise Legs
      * For example the CTD import module utilizes the CTD data system's CTD\_PKG.PARSE\_CRUISE\_LEG\_FROM\_PATH package function to determine the CRUISE\_LEG\_ID based on the given CTD data file path and saves that value so the module can use it when inserting the CTD cast record.

### Develop view(s) in the Referring Schema to retrieve the related CCD information

* + Define view(s) in the given Referring Schema that relate the scientific data to the CCD data based on the foreign key relationship defined in [step 2](#step2) by utilizing the shared CCD views (e.g. CEN\_CRUISE.CCD\_CRUISE\_LEGS\_V).
    - For example the CTD data system defines a CTD\_CAST\_CRUISES\_V view that retrieves CTD cast and related Cruise Leg information using the CCD CCD\_CRUISE\_LEGS\_V view based on the CRUISE\_LEG\_ID foreign key relationship. Multiple views are then based on the CTD\_CAST\_CRUISES\_V for various purposes.

### Develop QC view(s) in the Referring Schema

* + Develop QC view(s) in the Referring Schema to identify missing/invalid associations with the CCD Cruise Legs so these associations can be updated accordingly.
    - Define view(s) that return records in the Referring Schema that have missing/invalid CRUISE\_LEG\_ID values so these record association issues can be resolved. Missing/Invalid Cruise Leg associations should be addressed before exporting the data for reporting, analysis, etc.
      * For example the CTD data system has two QC views: CTD\_QC\_CAST\_FILES\_V that identifies missing Cruise Leg information for CTD casts and CTD\_QC\_CRUISE\_CAST\_INFO\_V that identifies mismatched Cruise Leg information for CTD casts.