

The `xxmx_validate_code_comb_cvr_pkg` package appears to be a utility tool used within an Oracle database environment, particularly for validating and managing General Ledger (GL) code combinations in the context of a data migration project. It interacts with a series of transformation tables that seem to contain migrated financial and accounting data.

Here's a summarized functionality of the package based on the technical documentation provided:

- gl_code_comb_cvr Procedure:**
 - Purpose:** Validates GL code combinations by cross-referencing multiple source or staging tables against a central table called `XXMX_GL_ACCOUNT_TRANSFORMS`. This procedure ensures that only new or missing combinations are added to this central table to maintain consistency and integrity of the GL codes.
 - Operation:** Uses a cursor to select unique combinations, employs bulk operations for efficient data handling, and commits the transactions or rolls back if exceptions are encountered.
- tot_count Procedure:**
 - Purpose:** Counts the total number of records present in the `XXMX_GL_ACCOUNT_TRANSFORMS` table that are associated with a given table name.
 - Operation:** Performs a simple SQL count operation and returns the total number of rows that match the criteria.
- acct_combination Procedure:**
 - Purpose:** Retrieves a set of account combinations from the `XXMX_GL_ACCOUNT_TRANSFORMS` table, potentially for further processing or validation.
 - Operation:** Provides a paginated output via a cursor by skipping a defined number of rows, allowing for a manageable batch of data to be processed at a time.

Overall Summary: The `xxmx_validate_code_comb_cvr_pkg` package is a comprehensive tool designed for validating and managing financial account code combinations during the data migration process in a financial or accounting system. It provides functionalities for inserting new valid GL code combinations into a transformation table, counting the total number of these combinations, and retrieving specific subsets of combinations for further analysis or processing. Error handling is built into each procedure to ensure the robustness of the data migration validation processes. This package is likely to be a crucial component of the data migration and integration phase in an enterprise resource planning (ERP) system implementation or during periodic data refreshes.

```
## acct_combination
```

Package xmxm_validate_code_comb_cvr_pkg Documentation

Overview

The `xxmx_validate_code_comb_cvr_pkg` package is designed to validate code combinations against CVR (Code Combination Validation Rules) for different entities during data migration. This package consists of various procedures that manipulate data within tables associated with general ledger code combinations and their transformations.

Structure

The package contains the following three procedures:

- gl_code_comb_cvr
- tot_count
- acct_combination

Procedure: gl_code_comb_cvr

This procedure is used to validate the GL code combinations from various transformation tables against the `XXMX_GL_ACCOUNT_TRANSFORMS` table. The procedure expects a list of transformation tables to be processed as a comma-separated string.

Parameters

- `p_xfm_tbl`: The name(s) of the transformation table(s) as a comma-separated string.
- `x_status`: The status of execution; returns 'SUCCESS' or an error message.

Logic

The procedure performs the following steps:

- Extracts metadata information related to the transformation tables.
- Opens a cursor, `get_code_comb_cur`, that selects data from multiple transformation tables which do not exist in the `XXMX_GL_ACCOUNT_TRANSFORMS` table.
- Bulk collects the selected data and inserts it into the `XXMX_GL_ACCOUNT_TRANSFORMS` table in batches using a `FORALL` statement.
- Commits the transaction if successful or rolls back and provides an error message if an exception occurs.

Exceptions

Catches any exceptions that occur and sets the `x_status` output parameter with the error message.

Procedure: tot_count

This procedure is used to count the number of entries in the `XXMX_GL_ACCOUNT_TRANSFORMS` table based on the provided transformation table(s).

Parameters

- `p_xfm_tbl`: The name(s) of the transformation table(s) as a comma-separated string.
- `x_count`: The output count of entries.
- `x_status`: The status of execution; returns 'SUCCESS' or an error message.

Logic

The procedure simply performs a `SELECT count(1)` query on the `XXMX_GL_ACCOUNT_TRANSFORMS` table with filtering based on the provided transformation table(s).

Exceptions

Catches any exceptions that occur and sets the `x_status` output parameter with the error message.

Procedure: acct_combination

This procedure is used to retrieve a set of GL code combinations from the `XXMX_GL_ACCOUNT_TRANSFORMS` table.

Parameters

- `p_xfm_tbl`: The name(s) of the transformation table(s) as a comma-separated string.
- `p_cur`: The output ref cursor containing the selected data.
- `x_status`: The status of execution; returns 'SUCCESS' or an error message.
- `p_var_skip`: The number of rows to skip in the result set (used for pagination).

Logic

The procedure performs the following steps:

- Opens the output ref cursor `p_cur` with a `SELECT` statement that retrieves GL code combinations from the `XXMX_GL_ACCOUNT_TRANSFORMS` table, applying the specified `p_var_skip` offset.
- Orders the result by `S_NO` and fetches the next 800 rows only (hardcoded pagination).

Exceptions

Catches any exceptions that occur, closes the cursor if open, and sets the `x_status` output parameter with the error message.

Usage Examples

Here are some usage examples for the procedures within this package:

Using gl_code_comb_cvr

```
DECLARE
    x_status varchar2(100);
BEGIN
    xmxm_validate_code_comb_cvr_pkg.gl_code_comb_cvr(p_xfm_tbl => 'XXMX_GL_OPENING_BALANCES_XFM', x_status => x_status);
    DBMS_OUTPUT.put_line('Status: ' || x_status);
END;
```

Using tot_count

```
DECLARE
    x_count number;
    x_status varchar2(100);
BEGIN
    xmxm_validate_code_comb_cvr_pkg.tot_count(p_xfm_tbl => 'XXMX_GL_OPENING_BALANCES_XFM', x_count => x_count, x_status => x_status);
    IF x_status = 'SUCCESS' THEN
        DBMS_OUTPUT.put_line('Total Count: ' || x_count);
    ELSE
        DBMS_OUTPUT.put_line('Error: ' || x_status);
    END IF;
END;
```

Using acct_combination

```
DECLARE
    p_cur sys_refcursor;
    x_status varchar2(100);
BEGIN
    xmxm_validate_code_comb_cvr_pkg.acct_combination(p_xfm_tbl => 'XXMX_GL_OPENING_BALANCES_XFM', p_cur => p_cur, x_status => x_status, p_var_skip => 0);
    -- Process the cursor
    CLOSE p_cur;
END;
```

Additional Information

The package relies on specific naming and type conventions for its variables and parameters, which improves code readability and maintainability. It employs bulk operations for efficiency and uses exception handling to manage errors. Proper usage of this package requires understanding of the underlying database schema and the purpose of each procedure within the context of the application's data migration processes.

```
## tot_count
```

XXMX_VALIDATE_CODE_COMB_CVR_PKG Technical Documentation

Introduction

This documentation provides an in-depth technical description of the `xxmx_validate_code_comb_cvr_pkg` package which is part of a larger software application. The purpose of this package is to validate code combinations against CVR (Chart of Accounts Validation Rules) for all entities during data migration. This package contains three procedures: `gl_code_comb_cvr`, `tot_count`, and `acct_combination`.

Package Specification

The package specification defines the interface to the package. It declares the procedures that can be called externally.

Procedures

gl_code_comb_cvr

This procedure validates GL code combinations and populates them into a transformation table.

Parameters:

- `p_xfm_tbl`: IN parameter of type `VARCHAR2`, representing the transformation table.
- `x_status`: OUT parameter of type `VARCHAR2`, representing the status of execution.

tot_count

This procedure counts the total number of new or invalid code combinations from the transformation table.

Parameters:

- `p_xfm_tbl`: IN parameter of type `VARCHAR2`.
- `x_count`: OUT parameter of type `NUMBER`.
- `x_status`: OUT parameter of type `VARCHAR2`.

acct_combination

This procedure returns a cursor to the caller with the account combinations.

Parameters:

- `p_xfm_tbl`: IN parameter of type `VARCHAR2`.
- `p_cur`: OUT parameter of type `SYS_REFCURSOR`.
- `x_status`: OUT parameter of type `VARCHAR2`.
- `p_var_skip`: IN parameter of type `NUMBER`, representing the offset to skip rows for pagination.

Package Body

The package body contains the implementation logic for the procedures declared in the package specification.

Procedure gl_code_comb_cvr

The `gl_code_comb_cvr` procedure is the core procedure for validating GL code combinations. It uses a cursor `get_code_comb_cur` to select distinct code combinations from various staging tables (transformations tables) such as `xxmx_gl_opening_balances_xfm`, `xxmx_gl_summary_balances_xfm`, `xxmx_gl_historical_rates_xfm`, and others. It checks if the selected combinations exist in the `XXMX_GL_ACCOUNT_TRANSFORMS` table and if not, it inserts them into the `XXMX_GL_ACCOUNT_TRANSFORMS` as new records.

The procedure uses bulk collect and `FORALL` to insert data, optimizing performance for large data sets. It captures the migration metadata before opening the cursor and uses exception handling to manage errors during execution.

Procedure tot_count

The `tot_count` procedure counts the number of records in the `XXMX_GL_ACCOUNT_TRANSFORMS` table for the specified transformation table(s). It outputs the count through the `x_count` parameter and captures any SQL errors during its execution.

Procedure acct_combination

The `acct_combination` procedure opens a cursor that selects account combinations from the `XXMX_GL_ACCOUNT_TRANSFORMS` table where the status is 'NEW' or 'Invalid'. It supports pagination through the `p_var_skip` parameter and handles any errors during cursor operations.

Error Handling

Each procedure within the package includes an `EXCEPTION` block that handles errors using the `OTHERS` handler. It provides a standardized error message using the SQL built-in functions `SQLERRM` and `SQLCODE`.

Execution Flow

- `gl_code_comb_cvr`: Populates new account combinations into the transformation table.
- `tot_count`: Retrieves the count of new or invalid combinations.
- `acct_combination`: Provides account combinations to the caller for further processing.

Notes

- The package makes use of the `xxmx_utilities_pkg.gcn_bulkcollectlimit` constant which is not defined in the given code but is expected to be declared in the `xxmx_utilities_pkg` package.
- Each procedure commits changes to the database upon successful execution and rolls back changes if an error occurs.

Example Usage

The package is designed to be used in the following sequence:

- Call `gl_code_comb_cvr` to validate and populate account combinations.
- Use `tot_count` to get the total count of combinations.
- Execute `acct_combination` to retrieve specific account combinations for further operations.

Conclusion

The `xxmx_validate_code_comb_cvr_pkg` package provides a robust mechanism for validating and managing GL code combinations during data migration. Through efficient cursor operations, bulk data processing, and comprehensive error handling, this package ensures the integrity and consistency of GL code combinations in the system.

```
## gl_code_comb_cvr
```

xxmx_validate_code_comb_cvr_pkg

This package is designed to validate code combinations against CVR for various entities during data migration.

Table of Contents

- Package Specification
 - Procedures
- Package Body
 - Procedures
 - Cursors
 - Types
 - Exceptions
 - Constants
 - Variables
- Usage

Package Specification

Procedures

The package specification `xxmx_validate_code_comb_cvr_pkg` defines three procedures:

1. gl_code_comb_cvr

- Parameters:
 - `p_xfm_tbl` IN `varchar2` - The name of the table to validate.
 - `x_status` OUT `varchar2` - The status of the execution.
- Description: This procedure validates GL code combinations by inserting non-existent records into the `XXMX_GL_ACCOUNT_TRANSFORMS` table.

2. tot_count

- Parameters:
 - `p_xfm_tbl` IN `varchar2` - The name of the table to retrieve the count for.
 - `x_count` OUT `number` - The count of records.
 - `x_status` OUT `varchar2` - The status of the execution.
- Description: This procedure counts the total number of records in the `XXMX_GL_ACCOUNT_TRANSFORMS` table based on the provided table name.

3. acct_combination

- Parameters:
 - `p_xfm_tbl` IN `varchar2` - The name of the table for which to fetch account combinations.
 - `p_cur` OUT `sys_refcursor` - A ref cursor for the result set.
 - `x_status` OUT `varchar2` - The status of the execution.
 - `p_var_skip` IN `number` - The number of rows to skip in the result set.
- Description: This procedure fetches a subset of account combinations from the `XXMX_GL_ACCOUNT_TRANSFORMS` table based on the provided table name and the number of rows to skip.

Package Body

Procedures

gl_code_comb_cvr

This is the main procedure of the package. It opens a cursor `get_code_comb_cur` that selects records from different source tables and checks for their existence in the `XXMX_GL_ACCOUNT_TRANSFORMS` table.

The source tables include:

- `XXMX_GL_OPENING_BALANCES_XFM`
- `XXMX_GL_SUMMARY_BALANCES_XFM`
- `XXMX_GL_HISTORICAL_RATES_XFM`
- `XXMX_SCM_PO_DISTRIBUTIONS_STD_XFM`
- `XXMX_AP_SUPP_SITE_ASSIGNS_XFM`
- `XXMX_FA_MASS_ADDITIONS_XFM`
- `XXMX_PPM_PRJ_LBRCOST_XFM`
- `XXMX_PPM_PRJ_MISCCOST_XFM`
- `XXMX_FA_MASS_ADDITION_DIST_XFM`
- `XXMX_PER_ASSIGNMENTS_M_XFM`

The procedure then performs a `BULK COLLECT` fetch with a limit defined by the `xxmx_utilities_pkg.gcn_bulkcollectlimit` constant and inserts new records into `XXMX_GL_ACCOUNT_TRANSFORMS`. It uses the `FORALL` statement for efficient bulk insert operations.

If an exception occurs, the procedure closes the cursor if it is open, performs a `ROLLBACK`, and sets the `x_status` accordingly.

tot_count

This procedure simply counts the number of records in the `XXMX_GL_ACCOUNT_TRANSFORMS` table that match the provided table name (`p_xfm_tbl`) and returns the count via `x_count`. If an error occurs, `x_status` is set to contain the error details.

acct_combination

This procedure fetches account combinations from the `XXMX_GL_ACCOUNT_TRANSFORMS` table based on the provided table name (`p_xfm_tbl`). It opens a cursor and returns a result set limited to 800 rows starting after a number of rows defined by `p_var_skip`. If an exception occurs, it closes the cursor if open and sets the `x_status`.

Cursors

get_code_comb_cur

A cursor defined to retrieve GL code combination records from various transformation tables. It uses several subqueries with `UNION` to gather records from different source tables.

Types

get_code_comb_t

A table type based on the `get_code_comb_cur%rowtype`.

Exceptions

E_LOADERERROR

An exception raised when there is an error during the loading of migration metadata.

Constants

No constants are explicitly declared in the package body, but comments indicate a section for constant declarations.

Variables

- `v_APPLICATION_SUITE` - Stores the application suite name.
- `v_APPLICATION` - Stores the application name.
- `v_BUSINESS_ENTITY` - Stores the business entity name.
- `v_SUB_ENTITY` - Stores the sub-entity name.

Usage

- To validate GL code combinations, call the procedure `gl_code_comb_cvr` with the appropriate table name.
- To count the records in the `XXMX_GL_ACCOUNT_TRANSFORMS` table, call `tot_count`.
- To fetch a subset of account combinations, call `acct_combination`.

Errors encountered are reported back through the `x_status` output parameter for each procedure.

Summary

The `xxmx_validate_code_comb_cvr_pkg` package provides a set of procedures to support the validation of GL code combinations against CVR during data migration. It features bulk operations for efficient processing and error handling mechanisms.