

Oracle® Argus Mart

Extensibility Guide

Release 1.0

E39156-01

April 2013

Oracle Argus Mart Extensibility Guide, Release 1.0

E39156-01

Copyright © 2013 Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

This software and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Contents

1 Overview

1.1	Scope	1-1
1.2	Assumptions	1-1

2 Argus Mart ETL

2.1	Introduction to Argus Mart ETL.....	2-1
2.2	ETL Hooks.....	2-1
2.2.1	List of Argus Mart ETL Hooks	2-2
2.2.2	Viewing Argus Safety User Interface.....	2-2
2.2.3	Executing ETL Hooks.....	2-3
2.3	Configuring Flexible Data Re-categorization Code List	2-5
2.4	Customizing Tables	2-6
2.4.1	Customizing RM Tables.....	2-6
2.4.1.1	Populating a Column Added in the Argus Safety table	2-6
2.4.1.2	Populating a Calculated Column in RM	2-7
2.4.2	Customizing SM Tables	2-8
2.4.2.1	Defining a New Column in SM table.....	2-9
2.4.2.2	Populating a Custom UVT table	2-9

Preface

The AM is a data source software product that can be used for analysis and reporting in medical product safety and pharmacovigilance. The primary data for AM are the adverse event cases managed by the Oracle Argus Safety application. The AM product consists of:

- A predefined AM data model containing Signal and Reporting tables.
- Pre-built ODI based interfaces that are linked to Oracle PL/SQL based packages.

The Argus Safety application serves as the primary source of data for AM. The ODI software extracts the data from the Argus Safety database, transforms and loads the data into the AM. Once the ODI tool loads the data into the AM data mart, it is available for the Argus Mart users for querying and reporting activities.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit

<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Finding Information and Patches on My Oracle Support

Your source for the latest information about Oracle Argus Mart is Oracle Support's self-service Web site, My Oracle Support (formerly MetaLink).

Always visit the My Oracle Support Web site for the latest information, including alerts, release notes, documentation, and patches.

Getting the Oracle Argus Mart Standard Configuration Media Pack

The Oracle Argus Mart media pack is available both as physical media and as a disk image from the Oracle E-Delivery Web site. The media pack contains the technology stack products and the Oracle Argus Mart application. To receive the physical media, order it from Oracle Store at <https://oraclestore.oracle.com>.

To download the Oracle Argus Mart media pack from eDelivery, do the following:

1. Navigate to <http://edelivery.oracle.com> and log in.
2. From the **Select a Product Pack drop-down** list, select **Health Sciences**.
3. From the **Platform** drop-down list, select the appropriate operating system.
4. Click **Go**.
5. Select Oracle Argus Mart Media Pack for Operating System and click **Continue**.
6. Download the software.

Creating a My Oracle Support Account

You must register at My Oracle Support to obtain a user name and password account before you can enter the Web site.

To register for My Oracle Support:

1. Open a Web browser to <http://support.oracle.com>.
2. Click the **Register here** link to create a My Oracle Support account. The registration page opens.
3. Follow the instructions on the registration page.

Signing In to My Oracle Support

To sign in to My Oracle Support:

1. Open a Web browser to <http://support.oracle.com>.
2. Click **Sign In**.
3. Enter your user name and password.
4. Click **Go** to open the My Oracle Support home page.

Searching for Knowledge Articles by ID Number or Text String

The fastest way to search for product documentation, release notes, and white papers is by the article ID number.

To search by the article ID number:

1. Sign in to My Oracle Support at <http://support.oracle.com>.
2. Locate the Search box in the upper right corner of the My Oracle Support page.
3. Click the Sources icon to the left of the search box, and then select Article ID from the list.
4. Enter the Article ID number in the text box.
5. Click the magnifying glass icon to the right of the Search box (or press the Enter key) to execute your search.

The Knowledge page displays the results of your search. If the article is found, click the link to view the abstract, text, attachments, and related products.

In addition to searching by article ID, you can use the following My Oracle Support tools to browse and search the knowledge base:

- **Product Focus** — On the Knowledge page, you can drill into a product area through the Browse Knowledge menu on the left side of the page. In the Browse any Product, By Name field, type in part of the product name, and then select the product from the list. Alternatively, you can click the arrow icon to view the

complete list of Oracle products and then select your product. This option lets you focus your browsing and searching on a specific product or set of products.

- **Refine Search** — Once you have results from a search, use the Refine Search options on the right side of the Knowledge page to narrow your search and make the results more relevant.
- **Advanced Search** — You can specify one or more search criteria, such as source, exact phrase, and related product, to find knowledge articles and documentation.

Finding Patches on My Oracle Support

Be sure to check My Oracle Support for the latest patches, if any, for your product. You can search for patches by patch ID or number, or by product or family.

To locate and download a patch:

1. Sign in to My Oracle Support at <http://support.oracle.com>.
2. Click the **Patches & Updates** tab.

The Patches & Updates page opens and displays the Patch Search region. You have the following options:

- In the **Patch ID** or **Number** field, enter the primary bug number of the patch you want. This option is useful if you already know the patch number.
 - To find a patch by product name, release, and platform, click the **Product or Family** link to enter one or more search criteria.
3. Click **Search** to execute your query. The Patch Search Results page opens.
 4. Click the patch ID number. The system displays details about the patch. In addition, you can view the Read Me file before downloading the patch.
 5. Click **Download**. Follow the instructions on the screen to download, save, and install the patch files.

Finding Certification Information

Certifications provide access to product certification information for Oracle and third party products. A product is certified for support on a specific release of an operating system on a particular hardware platform, for example, Oracle Database 10g Release 2 (10.2.0.1.0) on Sun Solaris 10 (SPARC). To find certification information:

1. Sign in to My Oracle Support at <http://support.oracle.com>.
2. Click the **Certifications** tab. The Certifications page opens and displays the Find Certifications region.
3. In **Select Product**, enter Oracle Argus Mart.
4. Click the **Go to Certifications** icon.

The right pane displays the certification information.

5. Select a certification to view the certification details.

Known Installation and Configuration Issues

Oracle maintains a list of installation and configuration issues that you can download from My Oracle Support (MOS). For information about these issues, please see Note ID 1326918.1.

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

This chapter comprises the following topics:

- [Scope](#)
- [Assumptions](#)

1.1 Scope

This document contains information that helps a database expert to extend Argus Mart data structure/logic for custom data.

1.2 Assumptions

This guide assumes the following:

- Argus Mart (AM) is dependent on Argus Safety (AS).
- You should avoid executing any such functionality that may be implemented in a simpler and more general way in Argus Safety.
- It is not necessary to wrap PL/SQL components of the ETL.

Argus Mart ETL

This chapter comprises the following topics:

- [Introduction to Argus Mart ETL](#)
- [ETL Hooks](#)
- [Configuring Flexible Data Re-categorization Code List](#)
- [Customizing Tables](#)

Note: The user interface in the application may vary from the figures displayed in the subsequent sections.

2.1 Introduction to Argus Mart ETL

Argus Mart is a data source for analysis and reporting in medical product safety and pharmacovigilance. The primary data is the adverse event cases managed by the Argus Safety application. Argus Mart consists of one or more data marts and the software to **Extract** the source data from Argus Safety; **Transform** the data, and **Load** it into the Argus Mart data marts. The two sets of tables used in Argus Mart are:

- Reporting Mart (RM)
- Signal Mart (SM)

The data can be customized before or after fetching data from Argus Safety database with the help of ETL Hooks. The next section comprises the details about ETL Hooks.

2.2 ETL Hooks

ETL Hooks are the custom procedures that are used to extend the existing functionality or behavior of the Argus Mart ETL.

ETL Hooks can be executed to perform actions while fetching data from the source database to the destination database, as required by the users.

The data fetched from Argus Safety database can be modified before or after uploading the data into the Argus Mart database. You can populate customized (non-standard/user-defined) columns to the standard RM and SM tables. Besides, you can also populate customized table(s).

For more information on populating customized columns or tables, refer to [Section 2.4, "Customizing Tables"](#).

ETL Hooks perform these actions at different stages of both initial and incremental ETL.

The ODI Load Plan lists all the ETL scenarios, as shown below:

Figure 2–1 ODI Load Plan

#	Steps Hierarchy	Enabled	Scenario/Variable	Restart
0	root_step	<input checked="" type="checkbox"/>		Restart from failure
1	SCN_compile_schema_stage	<input checked="" type="checkbox"/>	SCN_compile_schema_stage Version 001	Restart from failed step
2	SCN_compile_schema_mart	<input checked="" type="checkbox"/>	SCN_compile_schema_mart Version 001	Restart from failed step
3	SCN_compile_schema_rls	<input checked="" type="checkbox"/>	SCN_compile_schema_rls Version 001	Restart from failed step
4	SCN_etl_pre_req_checks_initial	<input checked="" type="checkbox"/>	SCN_etl_pre_req_checks_initial Version 001	Restart from failed step
5	SCN_mark_etl_state_to_started	<input checked="" type="checkbox"/>	SCN_mark_etl_state_to_started Version 001	Restart from failed step
6	SCN_set_etl_status_initial_start	<input checked="" type="checkbox"/>	SCN_set_etl_status_initial_start Version 001	Restart from failed step
7	SCN_maintain_logs_history	<input checked="" type="checkbox"/>	SCN_maintain_logs_history Version 001	Restart from failed step
8	SCN_etl_log_start_time	<input checked="" type="checkbox"/>	SCN_etl_log_start_time Version 001	Restart from failed step
9	SCN_update_etl_jobs_stage_start	<input checked="" type="checkbox"/>	SCN_update_etl_jobs_stage_start Version 001	Restart from failed step
10	SCN_populate_control_table	<input checked="" type="checkbox"/>	SCN_populate_control_table Version 001	Restart from failed step
11	SCN_truncate_dict_tables	<input checked="" type="checkbox"/>	SCN_truncate_dict_tables Version 001	Restart from failed step
12	SCN_delete_stage_schema_stats	<input checked="" type="checkbox"/>	SCN_delete_stage_schema_stats Version 001	Restart from failed step
13	SCN_pop_etl_enterprise_to_proces	<input checked="" type="checkbox"/>	SCN_pop_etl_enterprise_to_proces Version 001	Restart from failed step
14	SCN populate profile switches	<input checked="" type="checkbox"/>	SCN populate profile switches Version 001	Restart from failed step
15	SCN_p_exec_etl_custom_hooks_PRE_STAGE_TABLES_POPULATION	<input checked="" type="checkbox"/>	SCN_p_exec_etl_custom_hooks_PRE_STAGE...	Restart from failed step
16	SCN_change_schema_tables_logging(0,0)	<input checked="" type="checkbox"/>	SCN_change_schema_tables_logging(0,0) Ver...	Restart from failed step
17	SCN_calc_etl_high_water_mark	<input checked="" type="checkbox"/>	SCN_calc_etl_high_water_mark Version 001	Restart from failed step

This section comprises of the following topics:

- [List of Argus Mart ETL Hooks](#)
- [Viewing Argus Safety User Interface](#)
- [Executing ETL Hooks](#)

2.2.1 List of Argus Mart ETL Hooks

Argus Mart comprises the following ETL Hooks:

- CUSTOM ROUTINE BEFORE STAGE TABLES POPULATION: This hook can be executed before populating the Staging tables.
- CUSTOM ROUTINE BEFORE REPORTING TABLES POPULATION: This hook can be executed before populating the Reporting tables (or RM tables).
- CUSTOM ROUTINE AFTER REPORTING TABLES POPULATION: This hook can be executed after populating the Reporting tables (or RM tables).
- CUSTOM ROUTINE BEFORE SIGNAL HELPER TABLES POPULATION: This hook can be executed before populating the Signal Helper tables.
- CUSTOM ROUTINE AFTER SIGNAL HELPER TABLES POPULATION: This hook can be executed after populating the Signal Helper tables.
- CUSTOM ROUTINE AFTER ETL: This hook can be executed after running the initial or incremental ETL (post-ETL commit).

2.2.2 Viewing Argus Safety User Interface

ETL Hooks help in customizing the existing ETL to suit your requirements through Argus Safety user interface.

Follow the steps given below to view the Argus Safety user interface:

1. Log in to the Argus Safety application.
2. From the menu bar, click **Argus Console**.
The menu bar updates for Argus Console.
3. From the new menu bar, click **System Configuration**.

4. Click **System Management (Common Profile Switches)**.

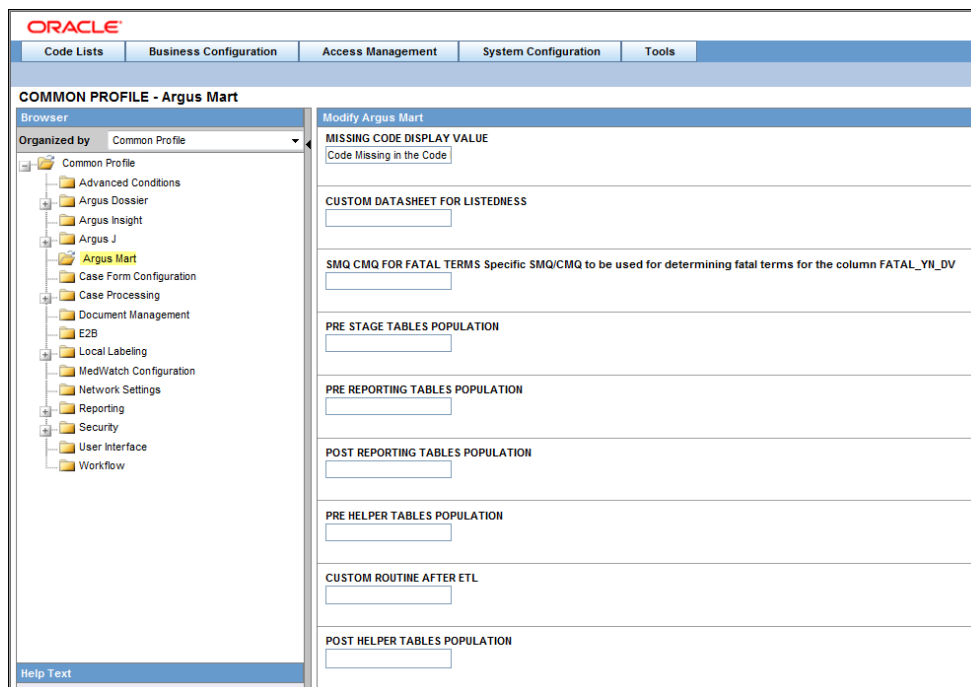
On the left hand side of the screen, a list of all system-managed common profile switches appears.

5. Expand the Common Profile folder and click **Argus Mart**.

The **Modify Argus Mart** page appears.

Note: ETL hooks are the global-level switches, visible on Argus Safety Console when you are logged-in through the default enterprise. These switches are visible only after installing and creating Argus Mart schema.

Figure 2–2 Argus Safety User Interface



2.2.3 Executing ETL Hooks

The ETL Hooks can be executed at different staging levels.

1. Before populating the reporting tables, execute CUSTOM ROUTINE BEFORE STAGE TABLES POPULATION, and/or CUSTOM ROUTINE BEFORE REPORTING TABLES POPULATION.
2. Before populating the signal tables, execute ETL Hooks CUSTOM ROUTINE AFTER REPORTING TABLES POPULATION, CUSTOM ROUTINE BEFORE SIGNAL HELPER TABLES POPULATION, and/or CUSTOM ROUTINE AFTER SIGNAL HELPER TABLES POPULATION.
3. After populating reporting and signal tables and before populating the results, execute CUSTOM ROUTINE AFTER ETL.

Figure 2-3 Sequence of ETL Hook Execution

Follow the steps given below to execute an ETL Hook:

1. On the **Modify Argus Mart** page, enter a procedure name in the text box relevant to the ETL Hook execution stage. The ETL type can be initial or incremental where the customized data will be fetched. The ETL Hook will look for the database object that matches the procedure name.

A procedure name denotes the ETL Hook that comprises the custom procedure, or steps. Based on the stage (initial or incremental) where this procedure is being called, the data is modified before sending it to the destination tables.

2. If the database object (or procedure) is found, the ETL is executed as follows:
 - If the ETL executes without any errors, then move to the next procedure.
 - If the ETL executes with errors, then log the error(s) and exit.

When executing ETL, if there is any error while populating the staging schema tables, error(s) are logged in the table ETL_STAGE_LOG, whereas while populating the mart schema tables, error(s) are logged in the table ETL_MART_LOG.

3. If the database object (or procedure) is not found, then log the error(s) and fail the ETL. In this case, the ETL may be executed if you have explicitly created an exception-handling for such cases to absorb any exceptions and move on to the next procedure.

Optionally, to resolve this issue, create a procedure of that name, provide an existing procedure name, or remove the configuration.

WARNING: ■ You should not modify the existing data or names of the objects. Although additional objects can be created, all changes should first be tested on a test environment, before implementing them in a production environment.

- No data commit should be done within ETL Hooks.
-
-

2.3 Configuring Flexible Data Re-categorization Code List

Flexible Data Re-categorization is an Argus Safety functionality through which users can define code list display values (LM tables) in different languages.

Argus Safety maintains this data in the tables `CODE_LIST_MASTER`, `CODE_LIST_CODE_ATTRIBUTES`, and `CODE_LIST_DETAIL_DISCRETE` for supported languages.

The Argus Mart ETL populates the code list data in the tables `RM_CODE_LIST_MASTER`, `RM_CODE_LIST_CODE_ATTRIBUTES`, and `RM_CODE_LIST_DETAIL_DISCRETE`, which are used for processing the SM table columns.

Based on the applicable language, the SM table fields are divided into the following categories:

- **LI - Language Independent fields:** These fields are applicable for first-human language, or second-human language parameters. The values are the same, irrespective of the language code.
- **FL - First Language fields:** These fields are populated on the basis of the AM Common Profile Switch `FIRST_HUMAN_LANGUAGE`, which is present in Argus Console.

The default value of this switch is **en**. As a result, the default language in which these fields will be populated is English.

If the user configures this switch with any other available language and if its corresponding data is also present in the Flexible Data Re-categorization code list tables, these first language fields will contain data as per the configured language.

- **SL - Second Language fields:** These fields are populated on the basis of the AM Common Profile Switch `SECOND_HUMAN_LANGUAGE`, which is present in Argus Console.

The default value of this switch is `NULL`. This implies that the second language fields will be populated in the SM tables only after this switch is configured.

If the user configures this switch with any other configured language (like German), and:

- If its corresponding data is also present in the Flexible Data Re-categorization code list tables, these second language fields will contain data as per the configured language.
- If its corresponding data is not available in the Flexible Data Re-categorization code list tables, these second language fields will contain data as per the English language.
- **EN - English fields:** These fields are always populated for only the English language.
- **J - Japanese fields:** These fields are applicable only if the second-human language parameter is set to Japanese language.
- **EN_ABBRV:** These fields use the `EN_ABBRV` language for the specified code list.
- **SM:** These fields use the SM language for the specified code list.

The following is an example to configure a code list display value in a new language for an already existing code in Argus Safety:

Example 2–1 Configuring Flexible Data Re-categorization

For a code list GENDER, data in the table CODE_LIST_DETAIL_DISCRETE for code 1 is available in the following four decode contexts (languages):

Figure 2–4 Original Decode Contexts (Languages)

CODE_LIST_ID	DECODE_CONTEXT	CODE	DISPLAY_VALUE	PREFERRED	SORT	LAST_UPDATE_TIME	ENTERPRISE_ID
GENDER	en	1	Male	0	(null)	05-FEB-13	1
GENDER	ja	1	男性	0	(null)	05-FEB-13	1
GENDER	E2B	1	1	0	(null)	05-FEB-13	1
GENDER	SM	1	M	0	(null)	05-FEB-13	1

To configure the same code 1 in the code list GENDER for a new language such as GERMAN (decode context **ge**), populate the table CODE_LIST_DETAIL_DISCRETE with required values in the GERMAN language.

```
INSERT INTO CODE_LIST_DETAIL_DISCRETE (CODE_LIST_ID, DECODE_CONTEXT, CODE,
DISPLAY_VALUE, PREFERRED, SORT, LAST_UPDATE_TIME, ENTERPRISE_ID) VALUES (
'GENDER', 'ge', 1, 'männlich', 0, null, sysdate, 1);
```

Figure 2–5 New Decode Contexts (Languages)

CODE_LIST_ID	DECODE_CONTEXT	CODE	DISPLAY_VALUE	PREFERRED	SORT	LAST_UPDATE_TIME	ENTERPRISE_ID
GENDER	en	1	Male	0	(null)	05-FEB-13	1
GENDER	ja	1	男性	0	(null)	05-FEB-13	1
GENDER	E2B	1	1	0	(null)	05-FEB-13	1
GENDER	SM	1	M	0	(null)	05-FEB-13	1
GENDER	ge	1	männlich	0	(null)	20-FEB-13	1

2.4 Customizing Tables

You can customize RM and SM tables, as explained in the following sections:

- [Customizing RM Tables](#)
- [Customizing SM Tables](#)

2.4.1 Customizing RM Tables

This section comprises information on the following actions:

- [Populating a Column Added in the Argus Safety table](#)
- [Populating a Calculated Column in RM](#)

2.4.1.1 Populating a Column Added in the Argus Safety table

You can populate new column(s) to existing RM tables that are added to the Argus Safety tables.

To populate the new column:

1. Check mapping between the source and the target tables, through the table *ETL_SIGNAL_TABLE_MAPPING*.
2. Based on the mapping between the Argus Safety and RM tables, fetch the desired column in the RM table through ETL Hooks.

There can be different source types to fetch data into tables at initial or incremental stages, such as:

- ATOS - Argus Safety to Staging tables
- DTOS - DLP to Staging tables
- STORM - Staging to RM tables
- STOSM - PRE_SM to SM Tables

Example 2-2 Populating a new column in the RM table

Follow the steps given below to populate a new column in the table RM_CASE_MASTER:

1. Check the mapping between RM_CASE_MASTER, and the source table (SDLP_CASE_MASTER) to fetch columns from the staging table into the RM table. In this case, the source type is STORM.
2. Check the source table for target table (SDLP_CASE_MASTER), where columns are fetched from Argus Safety into the staging table. In this case, the source type can be ATOS/DTOS.

Figure 2-6 Populating New Columns in the RM Table

The screenshot shows a Query Builder window with two SQL queries. The first query is: `SELECT * FROM etl_signal_table_mapping WHERE target_tab_name='SDLP_CASE_MASTER';` The second query is: `select * from etl_signal_table_mapping where source_tab_name='SDLP_CASE_MASTER';` Below the queries, there is a 'Query Result' pane showing a table with 2 rows and 7 columns: TABLE_ID, SOURCE_TAB_NAME, TARGET_TAB_NAME, SOURCE_TYPE, EXECUTE_STATUS, and CATEGORY. The first row has values: 1, 1020, DLP_CASE_MASTER, SDLP_CASE_MASTER, DTOS, 1 CASE. The second row has values: 2, 3140, SDLP_CASE_MASTER, RM_CASE_MASTER, STORM, 1 CASE.

TABLE_ID	SOURCE_TAB_NAME	TARGET_TAB_NAME	SOURCE_TYPE	EXECUTE_STATUS	CATEGORY
1	1020 DLP_CASE_MASTER	SDLP_CASE_MASTER	DTOS		1 CASE
2	3140 SDLP_CASE_MASTER	RM_CASE_MASTER	STORM		1 CASE

3. Once the mapping of the tables has been identified, fetch these column(s) from Argus Safety/DLP into Argus Mart, as required.

2.4.1.2 Populating a Calculated Column in RM

You can populate customized calculated column(s) to the existing RM tables that do not exist in the Argus Safety database. You can also populate column(s) which are calculated on the basis of the existing columns in a table.

To populate such custom columns, create a procedure, and use ETL Hooks to execute them.

Example 2-3 Creating a custom procedure

Creating a procedure called P_UPD_RM_CASE_PAT_INFO, to populate a new column called PAT_FULL_NAME in the table RM_CASE_PAT_INFO. The value of this column is calculated, and populated from the existing columns of RM_CASE_PAT_INFO. Here, we concatenate three columns PAT_INITIALS, PAT_FIRSTNAME, and PAT_LASTNAME to populate this value as PAT_FULL_NAME.

1. Create a procedure called P_UPD_RM_CASE_PAT_INFO.

- In the table RM_CASE_PAT_INFO, populate a new column called PAT_FULL_NAME.
The value of this column is calculated, and populated from the existing columns of RM_CASE_PAT_INFO.
- Concatenate three columns PAT_INITIALS, PAT_FIRSTNAME, and PAT_LASTNAME to populate this value as PAT_FULL_NAME.

Figure 2–7 Creating Custom Procedure

```
CREATE OR REPLACE PROCEDURE p_upd_rm_case_pat_info AS
BEGIN
  UPDATE rm_case_pat_info
    SET pat_full_name = pat_initials || ' ' || pat_firstname || ' ' || pat_lastname
    WHERE pat_full_name IS NULL;
EXCEPTION
  WHEN OTHERS THEN
    pkg_sm_logging.p_etl_mart_log (SUBSTR (SQLERRM, 1, 300),
                                  'p_upd_rm_case_pat_info',
                                  'Error during updation of RM_CASE_PAT_INFO.');
```

- Once this procedure is created, call this procedure using ETL Hook PRE_REPORTING_TABLES_POPULATION.
The ETL Hook will populate this new column PAT_FULL_NAME in SDLP_CASE_PAT_INFO.

2.4.2 Customizing SM Tables

When using ETL Hooks to populate data into SM tables, it is recommended to use RM tables as the source, with join(s) to the chunk table ETL_SM_CASES_TO_PROCESS_CHUNK.

Do not use RM Views such as V_ETL_CASE_MASTER, because each view is joined individually to the chunk table. This can lead to sluggish performance of the application if ETL Hook extracts data from the multiple views.

Example 2–4 Populating data into SM tables using the chunk table

In the example given below, two RM tables RM_CASE_MASTER and RM_CASE_PAT_INFO have been joined with the chunk table ETL_SM_CASES_TO_PROCESS_CHUNK to fetch data into the SM table with the help of the ETL Hooks.

Figure 2–8 Populating Data into SM tables using Chunk Table

```
SELECT *
FROM rm_case_master rcm, rm_case_pat_info rcpi, etl_sm_cases_to_process_chunk chnk
WHERE chnk.enterprise_id = rcm.enterprise_id
AND chnk.case_id = rcm.case_id
AND chnk.validstart >= rcm.effective_start_date
AND chnk.validstart < rcm.effective_end_date
AND chnk.enterprise_id = rcpi.enterprise_id
AND chnk.case_id = rcpi.case_id
AND chnk.validstart >= rcpi.effective_start_date
AND chnk.validstart < rcpi.effective_end_date;
```

This section comprises information on the following actions:

- [Defining a New Column in SM table](#)
- [Populating a Custom UVT table](#)

2.4.2.1 Defining a New Column in SM table

New column(s) can be added before or after fetching data into SM tables. The columns to be added into SM can have simple or complex calculations.

- Simple calculation-based columns: SM Views are created based on Argus Safety tables. When a customized column having simple calculations is to be added in an SM table, these calculations are done on SM Views, such as:
 - `select <column-name> from <table-name> table where <column-name> DateOfBirth = '1990';`
 - The definition of these SM Views is based on RM tables and is not updated automatically. When calculations are done, make sure to update the views, to reflect the changes. Once SM Views are updated, use ETL Hooks to fetch this column(s) into SM table, as required.
- Complex calculation-based columns: *PRE_SM* table is a set of tables that comprises the staging data for SM tables. You can perform complex calculations on these tables and then fetch them into SM tables using ETL Hooks.

The basic procedure to define a new column is the same for both simple and complex calculation-based column(s) - SM View and *PRE_SM* tables.

Simple and complex calculations are segregated to improve performance of execution of the data.

2.4.2.2 Populating a Custom UVT table

The Argus Mart ETL maintains Unique Value Tables (UVT) for first-human language and second-human language, as defined in the system. The UVTs contain the list of distinct values available for categorical data items in the case data.

To populate a custom UVT table, you must first check the source and target table mapping from the table *ETL_UVT_SM_MAPPING*.

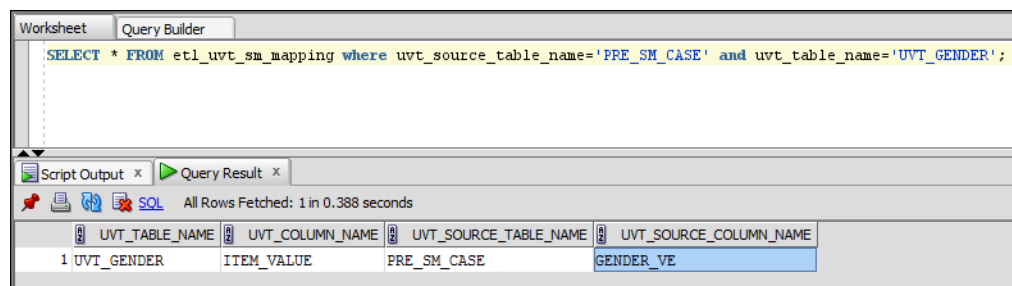
Fetch the data from the source table into the target table as per the mapping done above.

Example 2-5 Populating a custom UVT table

Follow the steps given below to populate a column in the table *UVT_GENDER*:

1. Check the mapping between the source table *PRE_SM_CASE* and the target table *UVT_GENDER* from the table *ETL_UVT_SM_MAPPING*.

Figure 2-9 Populating Custom UVT Table



-
2. After mapping the tables and their related columns, you can populate the column(s), as required.