

# MimamsuProPlus

## *User's Guide*

# Content

- Overview.....	3
- Quick Summary.....	5
- Example(HR Database).....	6

## DORM Studio:

- Open DORM Studio.....	8
- Create New/Open Map.....	9
- DOBJ: Add/Edit Static Attributes Table .....	11
- DOBJ: Create Static Attributes Table.....	13
- DOBJ: Add/Edit Aperiodic Attributes Table.....	15
- DOBJ: Create Aperiodic Attributes Table.....	16
- DOBJ: Add/Edit Periodic Attributes Table.....	18
- DOBJ: Create Periodic Attributes Table.....	19
- R-DOBJ: Add/Edit Static Attributes Table .....	21
- R-DOBJ: Create Static Attributes Table.....	22
- R-DOBJ: Add/Edit Aperiodic Attributes Table.....	24
- R-DOBJ: Create Aperiodic Attributes Table.....	25
- R-DOBJ: Add/Edit Periodic Attributes Table.....	26
- R-DOBJ: Create Periodic Attributes Table.....	27
- Add/Edit LOOK-UP.....	28
- Create LOOK-UP table.....	29
- Add/Edit RANGE.....	31
- Create RANGE table.....	32
- View Map.....	34
- Auto Map.....	35
- Verify Map.....	36
- Export/Import/Copy.....	37
- Standard names for tables and columns.....	38

## STORE:

- Create New.....	39
- Build RelTree.....	40
- Path Runner.....	41
- Open.....	42
- Update.....	43
- Create New Selection: Select Attribute/s.....	44
- Create New Selection: Select Object/s.....	45
- Run Selection.....	46
- Open Dataset.....	47
- Object Relations Summary.....	47
- Analysis Tool : General Functions.....	48
- Analysis Tool : Dobj Functions.....	50
- Analysis Tool : Aggregate Functions.....	51
- Analysis Tool : Analytic Functions.....	52
- Analysis Tool : String Functions.....	53
- Analysis Tool : Date Functions.....	54
- Dataset Tool .....	55
- Re-Run Selection.....	56

## Metadata:

- Dataset Metadata.....	57
- Statistic Metadata.....	58
- Subset Metadata.....	59

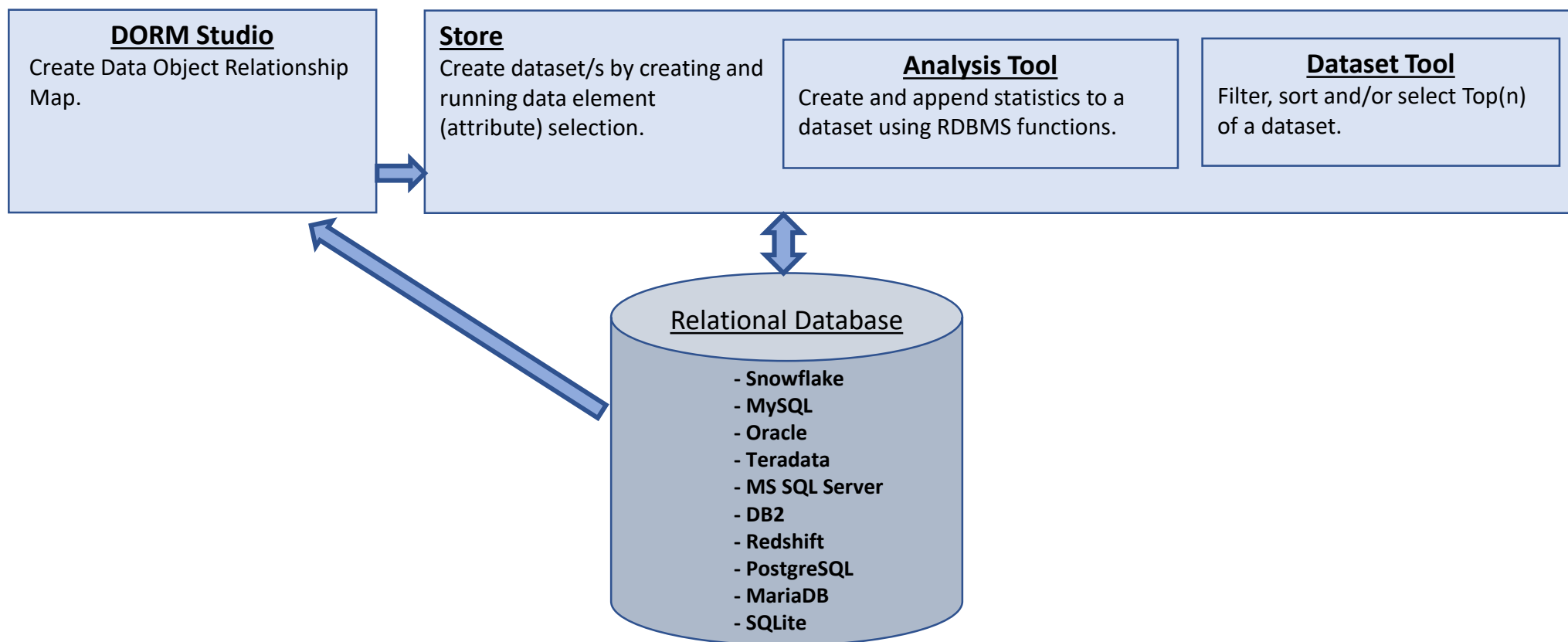
## Appendix:

- Appendix-1:Relational Data Object Model.....	60
- Appendix-2:Build RelTree.....	67
- Appendix-3:PathRunner.....	68
- Appendix-4:Create Dataset.....	69
- Appendix-5:Object Relations Summary.....	70
- Appendix-6:Create and Append Statistic.....	71

# Overview

MimamsuProPlus consists of four components (DORM Studio, Store, Analysis Tool and Dataset Tool). DORM Studio is a facility to create Data Object Relationship Map (DORM) of relational database(with RDOM\*) in terms of objects and attributes. Store is a facility to create dataset(with metadata) of selected attributes, as well as (optionally) run 'Path Runner' and 'Build RelTree'. Analysis Tool is a facility to perform statistical analysis, and append the statistics to the dataset. Dataset Tool is to create subset and/or filter the dataset.

Datasets, statistical analysis and subsets get stored as database table/s. For datasets and subsets a metadata table gets created together with the data table. Datasets, statistical analysis and subsets tables (together with metadata tables) can also be exported as CSV files.



\* see [Appendix-1: Symmetric Relational Data Object Model](#) for details.

**DORM Studio**: includes facility to create Data Object Relationship Map (DORM) as well as copy, import and export map features. In addition to individual interfaces for adding DOBJ, R\_DOBJ, LOOK-UP and RANGE the facility includes interfaces to create tables for all four types of objects as well (i.e. create RDOM database). It also includes 'AutoMap' and 'Verify Map' features. The 'AutoMap' adds (i.e. maps) tables and columns with standard names (see '[Standard Names for Tables and Columns](#)' section) as DOBJ and R-DOBJ components automatically. The 'Verify Map' verifies database/schema content for adherence to Relational Data Object Model.

**Store**: primarily is an interface, created using DORM, to create and run data element(attributes) selection to create dataset (with [Dataset Metadata](#)<sup>4</sup>). Store also includes 'Analysis Tool' and 'Dataset Tool' as part of 'Work with Dataset' feature. Additionally, Store includes 'Build RelTree\*' and 'PathRunner\*\*'.

**Object Relations Summary**: Each dataset gets created with Object Relations Summary. It is an interactive presentation of all combinations of related object counts in the dataset. The summary shows total counts of the objects at the top. The bottom part of the summary is interactive; it shows from left to right all combinations of related object counts, from left to right in descending order of object combination size.

**Analysis Tool**: is an interface consisting of six types of functions, 1) General Functions (i.e. CASE STATEMENT, EXPRESSION), 2) DOBJ Functions (i.e. COUNT, FREQUENCY COUNT, EMBED-FN-CODE), 3) Aggregate Functions, 4) Analytic Functions, 5) String Functions and 6) Date/Time Functions. Metadata about appended statistic gets added to [Dataset Metadata](#)<sup>4</sup>. Also, when a statistic's table is exported as CSV file, two files get created; a file for statistic and a file for [Statistic Metadata](#)<sup>5</sup>. [*Note: All functions are built in functions of the database system in use.*]

**Dataset Tool**: includes features to subset and/or order a dataset. It also includes features to save a subset as database table and export the subset as CSV file. Each Saved subset is created with associated [Subset Metadata](#)<sup>6</sup> table.

\* See '[Appendix-2 Build RelTree](#)' for details.

\*\* See '[Appendix-3 PathRunner](#)' for details.

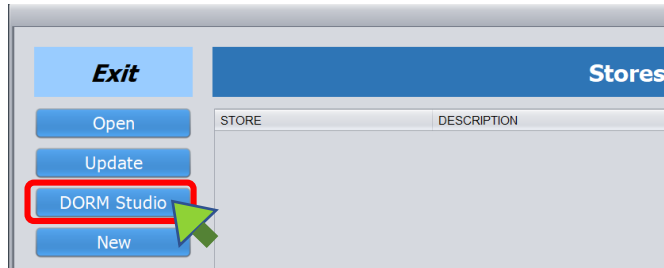
<sup>4</sup> See '[Dataset Metadata](#)' for details.

<sup>5</sup> See '[Statistic Metadata](#)' for details.

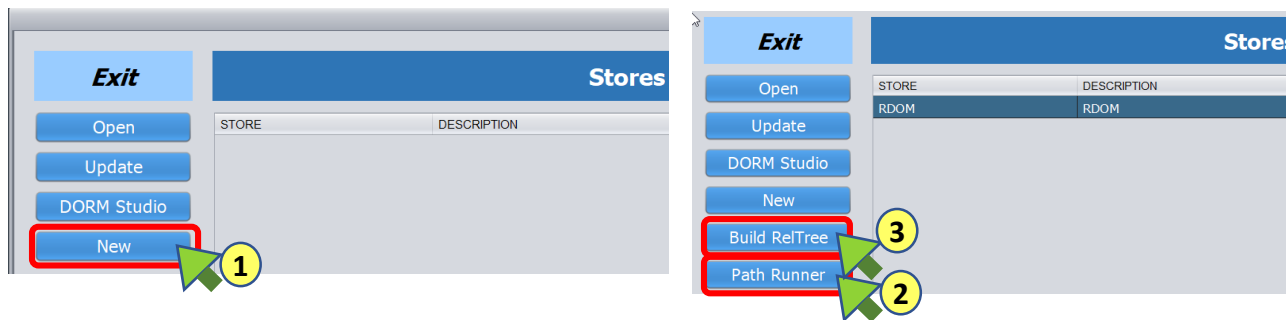
<sup>6</sup> See '[Subset Metadata](#)' for details.

# Quick Summary

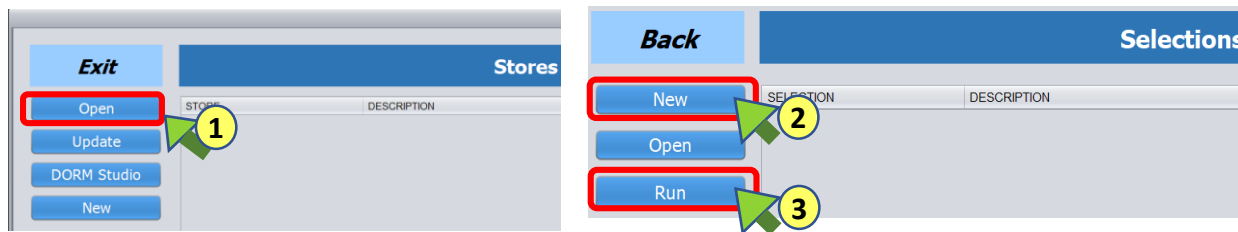
**Step 1:** Create Data Object Relationship Map (DORM) of the database/schemas (with RDOM\*) using 'DORM Studio'.



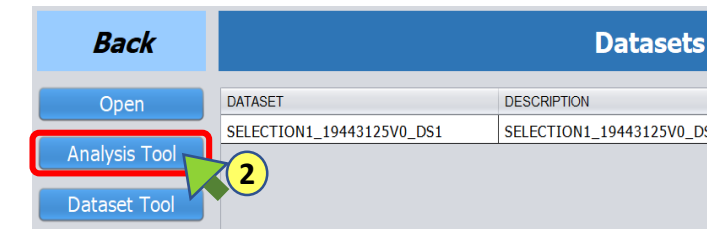
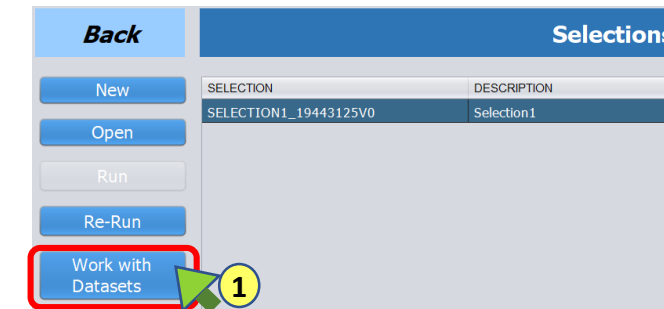
**Step 2:** Create 'Store' using the DORM. (Optionally) run 'PathRunner' and 'Build RelTree'.



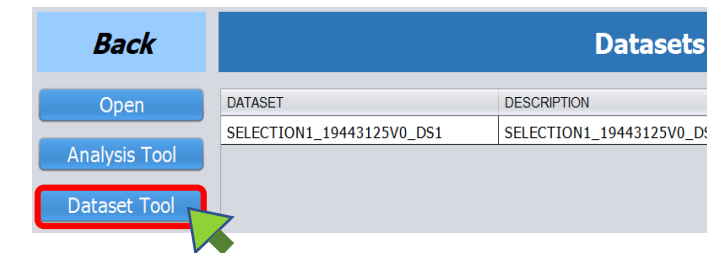
**Step 3:** Create and run data element 'Selection' to create dataset/s.



**Step 4:** Run (and append) statistics on the dataset.



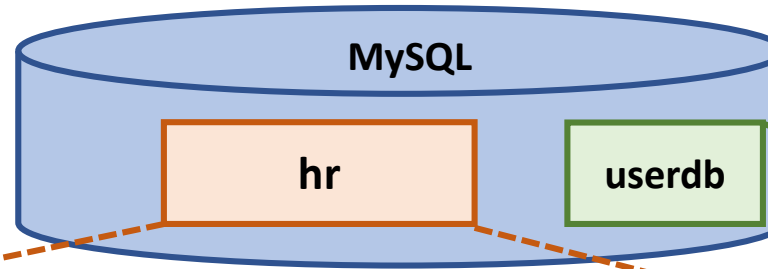
**Step 5:** Subset and/or filter the dataset.



\* see [Appendix-1: Symmetric Relational Data Object Model](#) for details.

## Example (HR Database)

There are two databases (i.e. schemas) 'hr' and 'userdb' on MySQL server. 'hr' database tables and views contain employee and department data and 'userdb' is used to store datasets, statistical analysis and subset tables. For the most part this guide refers to this example.



### EMP

employee	
123	EMPLOYEE_ID
123	MANAGER_ID
ABC	FIRST_NAME
ABC	LAST_NAME
ABC	EMAIL
ABC	PHONE_NUMBER
🕒	HIRE_DATE
123	SALARY
123	COMMISSION_PCT
ABC	MGR_FIRST_NAME
ABC	MGR_LAST_NAME

jobs_history	
1	EMPLOYEE_ID
🕒	START_DATE
🕒	END_DATE
ABC	JOB_ID
123	DEPARTMENT_ID
1	INSTANCE_SEQ

emppay_jan2009	
1	EMPLOYEE_ID
123	GROSS_PAY
123	NET_PAY

emppay_feb2009	
1	EMPLOYEE_ID
123	GROSS_PAY
123	NET_PAY

emppay_mar2009	
1	EMPLOYEE_ID
123	GROSS_PAY
123	NET_PAY

### EMP\_DEPT

emp_x_dept	
123	EMPLOYEE_ID
123	DEPARTMENT_ID

### DEPT

department	
ABC	DPTMGR_FIRST_NAME
ABC	DPTMGR_LAST_NAME
ABC	DPTMGR_EMAIL
ABC	DPTMGR_PHONE_NUMBER
🕒	DPTMGR_HIRE_DATE
ABC	DPTMGR_JOB_ID
123	DPTMGR_SALARY
123	DPTMGR_COMMISSION_PCT
123	DEPARTMENT_ID
ABC	DEPARTMENT_NAME
123	LOCATION_ID

### LOCATION

dept_location	
123	LOCATION_ID
ABC	STREET_ADDRESS
ABC	POSTAL_CODE
ABC	CITY
ABC	STATE_PROVINCE
ABC	COUNTRY_NAME
ABC	REGION_NAME

### JOBS

jobs	
ABC	JOB_ID
ABC	JOB_TITLE
123	MIN_SALARY
123	MAX_SALARY

### SALARY\_RANK

salary_range	
23	lbound
23	ubound
ABC	salary_rank
ABC	range

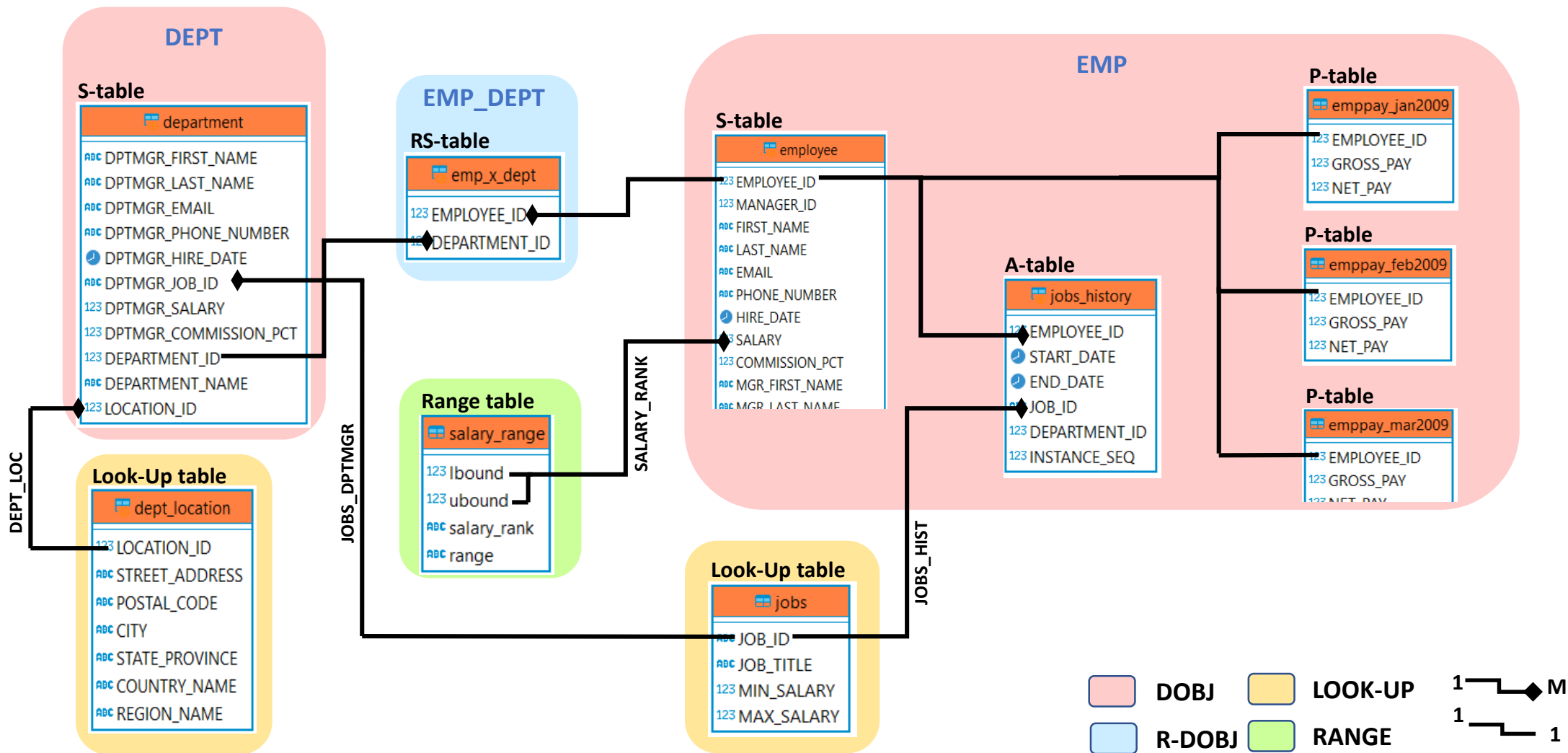
**Datasets**

**Analysis**

**Subsets**

- DOBJ
- R-DOBJ
- LOOK-UP
- RANGE

## Example (HR Database) (RDOM diagram)



In terms of relational database, DEPARTMENT\_ID and EMPLOYEE\_ID are primary-keys (not NULL) for 'department' and 'employee' tables (i.e. S-tables) respectively. For 'jobs\_history' table (i.e. A-table) EMPLOYEE\_ID + INSTANCE\_SEQ is primary-key where EMPLOYEE\_ID is foreign-key to 'employee' table. For 'emppay\_jan2009', 'emppay\_feb2009' and 'emppay\_mar2009' tables (i.e. P-tables) EMPLOYEE\_ID is primary-key and foreign-key to 'employee' table. For 'emp\_x\_dept' table (i.e. RS-table) EMPLOYEE\_ID + DEPARTMENT\_ID is primary-key (not NULL) and are foreign-keys to 'employee' and 'department' tables respectively.

- Launch MimamsuProPlus
- Open DORM Studio



Enter MimamsuProPlus.



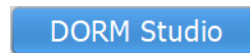
Close MimamsuProPlus.



Open selected store.



Update selected store (after associated map update).



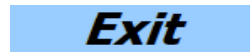
Open DORM Studio.



Create new store.



Delete selected store.

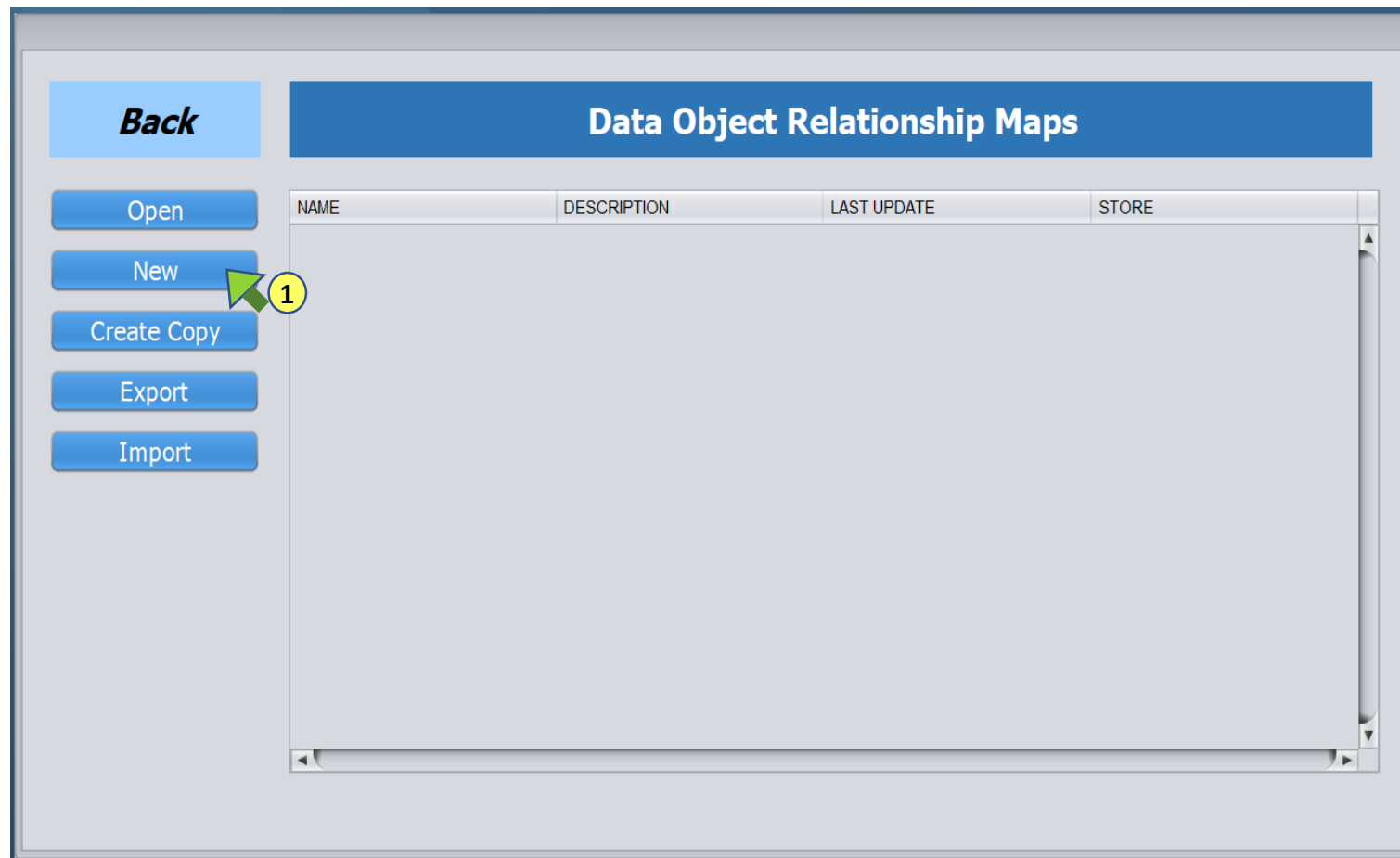


Exit MimamsuProPlus.



## DORM Studio:

### - New map



Open

Open selected map.

New

Create new map.

Create Copy

Create duplicate copy of the selected map.

Export

Export selected map as a file.

Import

Import map from a file.

**Back**

Return to main menu (or application home page).

NAME

Map name.

DESCRIPTION

Map description.

LAST UPDATE

Last update timestamp.

STORE

Name of associated store.

## DORM Studio:

- New map
- New DOBJ

If prompted, copy/paste the link into browser, download the driver file and add using folder button.

RDBMS	Server URL
Snowflake	//<org>-<account>.snowflakecomputing.com/?db=<MyDatabase>
Teradata	//<IP or hostname>/DATABASE=<MyDatabase>,DBS_PORT=1025
Oracle	@<hostname>:1521:<sid>
SQLServer	//<hostname>:1433 (Instance Name: <instance_name>)
Redshift	//<cloudhost>.redshift.amazonaws.com:5439/<MyDatabase>
DB2	//<hostname>:50000/<DatabaseName>
MySQL	//<hostname>:3306/<?parameters>
MariaDB	//<hostname>:3306/<?parameters>
PostgreSQL	//<hostname>:5432/<DatabaseName>
SQLite	<folderPath>/<folder_name>

**Name** Enter name for the new map.

**Description** Enter brief description of the map.

**Database Type** Select database type from menu.

**Server URL** Enter server address of the database.

**Username** Enter your username for the database.

**Password** Enter your password for the database.

**Connect** Connect(logon) to the database.

**MyDatabase(space)** Select a database space for datasets, analysis and subsets.

**>** Add database to selected databases.

**<** Remove database from selected databases.

**Save DB Info** Save database type, database selections, address (URL) and Username.

**Auto Map** Auto-create map entries for DOBJ and R-DOBJ components based on the standard table and column names. (See '[Standard Names for Tables and Columns](#)'.)

**Verify Map** Check for NULL and duplicate values in ID columns; and verify objects relationships.

**DOBJ** Create/edit Data Object/s.

**R-DOBJ** Create/edit Relationship Data Object/s.

**LOOK-UP** Create/edit Look-Up object.

**RANGE** Create/edit Range object.

**Close** Close map. (all add/edits will be saved.)

**X** Close map. (all add/edits will be saved.)

**View Map** View map in a tabular form.

**X** Delete map.

# DORM Studio:

- New DOBJ 'EMP'

- Add Static Table

\* Create new.

Dobj Name Enter name of the Dobj.

Description Enter small description of the Dobj.

➤ Add selected table's info to Dobj.

✕ Delete from the map.

✕ Exclude attribute from the Dobj.

Static Attributes Table Select from left and add Static attributes table.

Dobj-ID Column Select Dobj-ID column from drop down.

Row Date/Time Column Select row timestamp column from drop down.

Aperiodic Attributes Table Select from left and add Aperiodic attributes table

Instance Update Seq. Number Column Select row update seq. column from drop down.

Periodic Attributes Table Select from left and add Periodic attributes table

Period Type Create new Period Type(or select from drop down)

Period Name Enter Period name.

Attribute Description Description of selected (from table below) attribute (editable).

Attribute Information Additional info. about selected (from table below) attribute (editable).

Save Save add/edits.

Cancel Cancel add/edits.

Done Close Dobj

View Map View map in a tabular form.

# DORM Studio:

- Edit 'EMP' DOBJ
- Edit Static Table

MyDatabase(space) EMP \* Dobj Name EMP View Map

userdb Description EMPLOYEES

Database(schema) / Tables

Static Attributes Table x Aperiodic Attributes Table x Periodic Attributes Table x

employee --- SELECT --- --- SELECT ---

Dobj-ID Column Instance Update Seq. Number Column \* Period Type Period Name

EMPLOYEE\_ID --- SELECT --- --- SELECT ---

Row Date/Time Column(Optional) Row Date/Time Column(Optional) Row Date/Time Column(Optional)

Attribute Column Attribute Description Attribute Information

Included Attributes (18)

	NAME	DESCRIPTION	INFORMATION
1	COMMISSION_PCT	COMMISSION_PCT	decimal

Included Attributes (17) Excluded Attributes (1)

	NAME	DESCRIPTION	INFORMATION
1	COMMISSION_PCT	COMMISSION_PCT	decimal

Included Attributes (18)

	NAME	DESCRIPTION	INFORMATION
1	COMMISSION_PCT	COMMISSION_PCT	decimal
2	EMAIL	EMAIL	varchar
3	FIRST_NAME	FIRST_NAME	varchar
4	HIRF_DATE	HIRF_DATE	date

Error:

Hint:

Save Done

- \* Create new.
- Dobj Name Enter name of the Dobj.
- Description Enter small description of the Dobj.
- > Add selected table's info to Dobj.
- x Delete from the map.
- Included Attributes Shows included attributes.
- Excluded Attributes Shows excluded attributes.
- Exclude attribute from DOBJ.
- + Include attribute in DOBJ.
- Static Attributes Table Select from left and add Static attributes table.
- Dobj-ID Column Select Dobj-ID column from drop down.
- Row Date/Time Column Select row timestamp column from drop down.
- Aperiodic Attributes Table Select from left and add Aperiodic attributes table
- Instance Update Seq. Number Column Select row update seq. column from drop down.
- Periodic Attributes Table Select from left and add Periodic attributes table
- Period Type Create new Period Type(or select from drop down)
- Period Name Enter Period name.
- Attribute Description Description of selected (from table below) attribute (editable).
- Attribute Information Additional info. about selected (from table below) attribute (editable).
- Save Save add/edits.
- Cancel Cancel add/edits.
- Done Close Dobj
- View Map View map in a tabular form.

# DORM Studio:

## - Create Static Attributes Table

The screenshot shows the DORM Studio interface for creating a static attributes table. The main form contains the following fields and controls:

- MyDatabase(space)**: userdb
- Database(schema) / Tables**: MySQL (//localhost:3306/?allowPublicKe) > hr
- Obj Name**: OBJECT1 (Callout 1)
- Description**: Object1 (Callout 3)
- Static Attributes Table**: OBJ1STATIC (Callout 4)
- Dobj-ID Column**: OBJ1\_ID (Callout 6)
- Attribute Column**: Attribute1 (Callout 7)
- Attribute Description**: Attribute1\_Desc (Callout 9)
- Save**: (Callout 10)

Additional elements include a table with columns NAME, DESCRIPTION, and INFORMATION, and a 'View Map' button. A legend on the right explains the icons: a yellow exclamation mark for 'Optional' and a red asterisk for 'Repeat (7, 8, 9) for each Attribute'.

! Optional

\* Repeat (7, 8, 9) for each Attribute.

( Continued on next page.)

\* Create new.

**Dobj Name** Enter name of the Data Object.

**Description** Enter small description of the Data Object.

X Delete from the map.

X Exclude attribute from the Look-Up.

**Static Attributes Table** New Static attributes table name

**Dobj-ID Column** New Dobj-ID column name.

**Attribute Column** New attribute column name.

**Attribute Description** Description of attribute (editable).

**Attribute Information** Additional info. about attribute (editable).

**Row Date/Time Column** Column for row update date/time.

**Save** Save add/edits.

**Cancel** Cancel add/edits.

**View Map** View map in a tabular form.

## DORM Studio:

### - Create Static Attributes Table

**New Table**

**Create Table**

Database: userdb

Table: OBJ1STATIC

Columns:

- OBJ1\_ID
- ATTRIBUTE1 \*

Definition:

Column: ATTRIBUTE1

Data Type: VARCHAR \*

Width / Constraint: (10) !

DDL Statement:

```
CREATE TABLE userdb.OBJ1STATIC (  
  OBJ1_ID VARCHAR (10)  
  ,ATTRIBUTE1 VARCHAR (10)  
)
```

Execute DDL

Cancel

- \* For each column repeat ①, ② and ③ (for some data types such as VARCHAR)
- ! After typing Width (and/or Constraint) press Enter or click in DDL Statement box on the right.

**Database** MyDatabase (space)

**Table** Table to be create

**Columns** Columns to be created

**Data Type** Select Data Type for the column

**Width / Constraint** Enter Width and/or constraint for the column

**Execute DDL**

Create table in the Database

**Cancel**

Cancel table creation

## DORM Studio:

- Edit 'EMP' DOBJ
- Add Aperiodic Table

The screenshot shows the DORM Studio interface for editing a DOBJ named 'EMP'. The 'Description' field contains 'EMPLOYEES'. The 'jobs\_history' table is selected in the tree view. The configuration panel shows the 'Instance Update Seq. Number Column' set to 'INSTANCE\_SEQ'. The 'Attribute Column' is 'DEPARTMENT\_ID' and the 'Attribute Description' is 'HIST\_DEPARTMENT\_ID'. The 'Attribute Information' is 'decimal'. A table below lists the attributes:

	NAME	DESCRIPTION	INFORMATION	
1	DEPARTMENT_ID	HIST_DEPARTMENT_ID	decimal	✕
2	END_DATE	HIST_END_DATE	date	✕
3	JOB_ID	HIST_JOB_ID	varchar	✕
4	START_DATE	HIST_START_DATE	date	✕

Numbered callouts (1-7) highlight the following elements:

- Star icon for creating a new DOBJ.
- Tree view showing the selected table.
- Arrow icon for adding a table's info to the DOBJ.
- Drop-down menu for the Instance Update Seq. Number Column.
- Attribute table row for DEPARTMENT\_ID.
- Star icon for editing attribute description/information.
- Save and Cancel buttons.

\* To change attribute description and/or information select (5) and edit (6).

\* Create new.

**Dobj Name** Enter name of the Dobj.

**Description** Enter small description of the Dobj.

➤ Add selected table's info to Dobj.

✕ Delete from the map.

✕ Exclude attribute from the Dobj.

**Static Attributes Table** Select from left and add Static attributes table.

**Dobj-ID Column** Select Dobj-ID column from drop down.

**Row Date/Time Column** Select row timestamp column from drop down.

**Aperiodic Attributes Table** Select from left and add Aperiodic attributes table

**Instance Update Seq. Number Column** Select row update seq. column from drop down.

**Periodic Attributes Table** Select from left and add Periodic attributes table

**Period Type** Create new Period Type(or select from drop down)

**Period Name** Enter Period name.

**Attribute Description** Description of selected (from table below) attribute (editable).

**Attribute Information** Additional info. about selected (from table below) attribute (editable).

**Save** Save add/edits.

**Cancel** Cancel add/edits.

**Done** Close Dobj

**Close** Close DOBJ interface.

**View Map** View map in a tabular form.

## DORM Studio:

### - Create Aperiodic Attributes Table

MyDatabase(space) OBJECT1 \* Dobj Name OBJECT1 View Map

userdb Description Object1

Database(schema) / Tables

- \* Static Attributes Table
- \* Aperiodic Attributes Table
- \* Periodic Attributes Table

OBJ1STATIC OBJ1APERIODIC1 --- SELECT ---

Dobj-ID Column Instance Update Seq. Number Column \* Period Type Period Name

--- SELECT --- SEQ\_NUM

Row Date/Time Column(Optional) Row Date/Time Column(Optional) Row Date/Time Column(Optional)

--- SELECT ---

\* Attribute Column Attribute Description Attribute Information

APRDCATTR1 APRDCATTR1

	NAME	DESCRIPTION	INFORMATION
1	APRDCATTR1	APRDCATTR1	

Error: 0/0 Hint: Save 10 Cancel

! Optional

\* Repeat ( 4 5 6 )  
for each Attribute.

( Continued on next page.)

\* Create new.

Dobj Name Name of the Data Object.

Description Description of the Data Object.

X Delete from the map.

X Exclude attribute from the Look-Up.

Aperiodic Attributes Table New Aperiodic attributes table name

Instance Update Seq. Number Column New sequence number column name.

Attribute Column New attribute column name.

Attribute Description Description of attribute (editable).

Attribute Information Additional info. about attribute (editable).

Row Date/Time Column Column for row update date/time.

Save Save add/edits.

Cancel Cancel add/edits.

View Map View map in a tabular form.



## DORM Studio:

### - Create Aperiodic Attributes Table

**New Table**

**Create Table**

Database: userdb

Table: OBJ1APERIODIC1

Columns:

- OBJ1\_ID
- SEQ\_NUM
- APRDCATTR1

Definition:

Column: APRDCATTR1

Data Type: VARCHAR

Width / Constraint: (15)

DDL Statement:

```
CREATE TABLE userdb.OBJ1APERIODIC1 (  
OBJ1_ID VARCHAR (10)  
,SEQ_NUM INT  
,APRDCATTR1 VARCHAR (15)  
)
```

Execute DDL

Cancel

- \* For each column repeat ①, ② and ③ (for some data types such as VARCHAR)
- ❗ After typing Width (and/or Constraint) press Enter or click in DDL Statement box on the right.

**Database** MyDatabase (space)

**Table** Table to be create

**Columns** Columns to be created

**Data Type** Select Data Type for the column

**Width / Constraint** Enter Width and/or constraint for the column

**Execute DDL**

Create table in the Database

**Cancel**

Cancel table creation

# DORM Studio: -Add Periodic Tables

\* Create New (3) or select (3) Period Type for each Periodic Attributes Table

\* Create new.

**Dobj Name** Enter name of the Dobj.

**Description** Enter small description of the Dobj.

➤ Add selected table's info to Dobj.

✕ Delete from the map.

✕ Exclude attribute from the Dobj.

**Static Attributes Table** Select from left and add Static attributes table.

**Dobj-ID Column** Select Dobj-ID column from drop down.

**Row Date/Time Column** Select row timestamp column from drop down.

**Aperiodic Attributes Table** Select from left and add Aperiodic attributes table

**Instance Update Seq. Number Column** Select row update seq. column from drop down.

**Periodic Attributes Table** Select from left and add Periodic attributes table

**Period Type** Create new Period Type(or select from drop down)

**Period Name** Enter Period name.

**Attribute Description** Description of selected (from table below) attribute (editable).

**Attribute Information** Additional info. about selected (from table below) attribute (editable).

**Save** Save add/edits.

**Cancel** Cancel add/edits.

**Done** Close Dobj

**Close** Close DOBJ interface.

**View Map** View map in a tabular form.

# DORM Studio:

## - Create Periodic Attributes Table

The screenshot shows the DORM Studio interface for creating a Periodic Attributes Table. The main window is titled "MyDatabase(space)" and contains a form with the following fields and controls:

- Dobj Name:** OBJECT1
- Description:** Object1
- Database(schema) / Tables:** MySQL (//localhost:3306/?allowPublicKe) > hr
- Static Attributes Table:** OBJ1STATIC
- Aperiodic Attributes Table:** --- SELECT ---
- Periodic Attributes Table:** FREQ1TABLE1
- Dobj-ID Column:** --- SELECT ---
- Instance Update Seq. Number Column:** --- SELECT ---
- Period Type:** P1
- Period Name:** N1
- Row Date/Time Column(Optional):** --- SELECT ---
- Attribute Column:** F1ATTRIBUTE1
- Attribute Description:** F1ATTRIBUTE1
- Attribute Information:** (empty)

A table at the bottom lists the attributes:

	NAME	DESCRIPTION	INFORMATION
1	F1ATTRIBUTE1	F1ATTRIBUTE1	

Numbered callouts (1-8) and icons (asterisk, exclamation mark, X) highlight specific features and actions:

- 1: Asterisk icon for "Periodic Attributes Table"
- 2: Arrow icon for "Period Type"
- 3: Asterisk icon for "Period Type" dropdown
- 4: Arrow icon for "Period Name" dropdown
- 5: Asterisk icon for "Attribute Column" dropdown
- 6: Asterisk icon for "Attribute Column" text field
- 7: Asterisk icon for "Attribute Description" text field
- 8: Asterisk icon for "Save" button

Legend:

- ! Optional
- \* Repeat (5 6 7) for each Attribute.

\* Create New (3) or select (3) Period Type for each Periodic Attributes Table

! Optional

\* Repeat (5 6 7) for each Attribute.

( Continued on next page.)

\* Create new.

**Dobj Name** Name of the Data Object.

**Description** Description of the Data Object.

X Delete from the map.

X Exclude attribute from the Look-Up.

**Periodic Attributes Table** New Periodic attributes table name

**Period Type** Type of periodic table

**Period Name** Name of this periodic table

**Attribute Column** New attribute column name.

**Attribute Description** Description of attribute (editable).

**Attribute Information** Additional info. about attribute (editable).

**Row Date/Time Column** Column for row update date/time.

**Save** Save add/edits.

**Cancel** Cancel add/edits.

**View Map** View map in a tabular form.

## DORM Studio:

### - Create Periodic Attributes Table

**Create Table**

Database: userdb

Table: FREQ1TABLE1

Columns:

- OBJ1\_ID
- F1ATTRIBUTE1

Definition:

Column: F1ATTRIBUTE1

Data Type: VARCHAR

Width / Constraint: (8)

DDL Statement:

```
CREATE TABLE userdb.FREQ1TABLE1 (  
  OBJ1_ID VARCHAR (10)  
  ,F1ATTRIBUTE1 VARCHAR (8)  
)
```

Execute DDL

Cancel

- \* For each column repeat ①, ② and ③ (for some data types such as VARCHAR)
- ❗ After typing Width (and/or Constraint) press Enter or click in DDL Statement box on the right.

**Database** MyDatabase (space)

**Table** Table to be create

**Columns** Columns to be created

**Data Type** Select Data Type for the column

**Width / Constraint** Enter Width and/or constraint for the column

**Execute DDL**

Create table in the Database

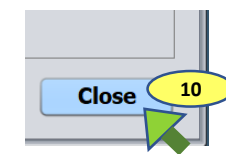
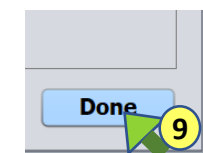
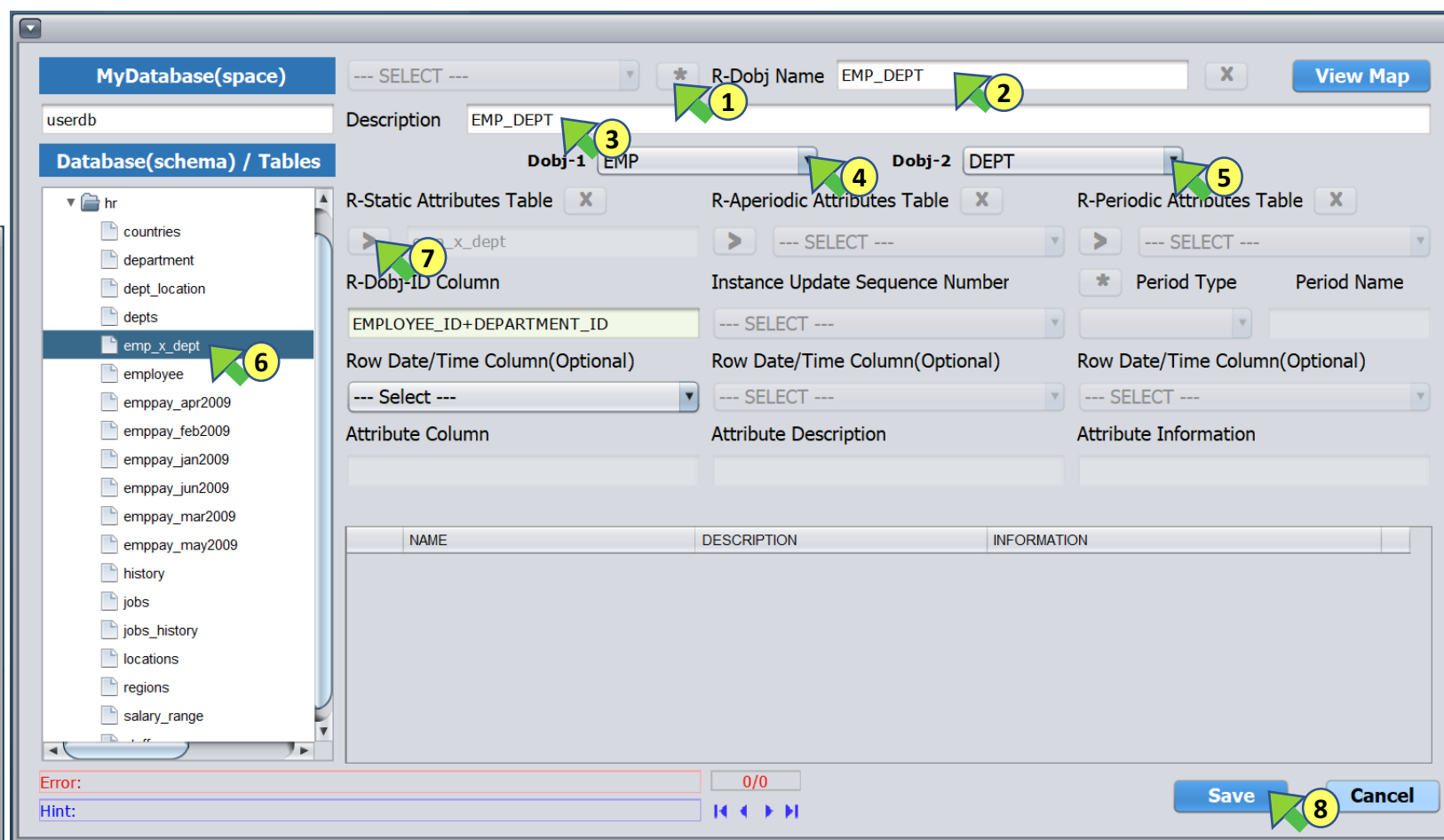
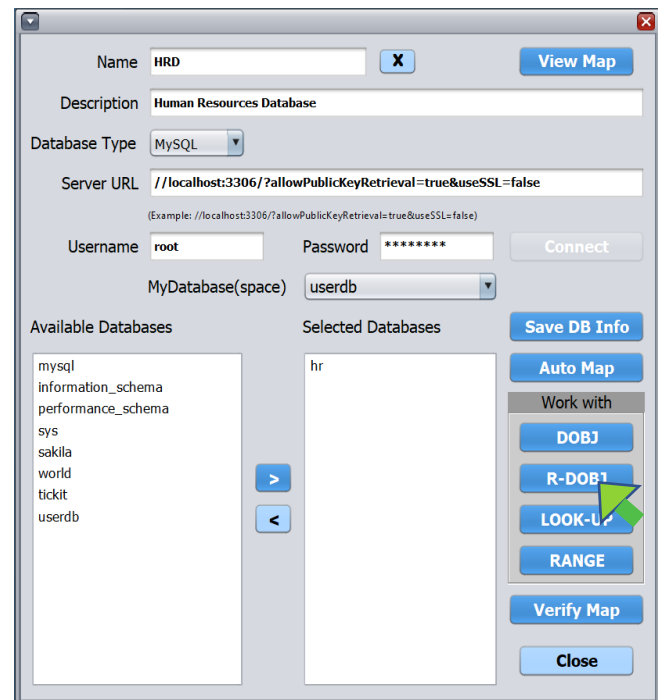
**Cancel**

Cancel table creation

# DORM Studio:

- New R-DOBJ 'EMP\_DEPT'

- Add R-Static Table



\* Create new.

R-Dobj Name Enter name of the R-Dobj.

Description Enter small description of the R-Dobj.

Dobj-1 Select a Dobj.

Dobj-2 Select a Dobj

> Add selected table's info to R-Dobj.

X Delete from the map.

R-Static Attributes Table Select from left and add Static attributes table.

R-Dobj-ID Column Dobj-ID columns of both Dobjs.

Row Date/Time Column Select row timestamp column from drop down.

R-Aperiodic Attributes Table Select from left and add Aperiodic attributes table

Instance Update Seq. Number Column Select row update seq. column from drop down.

R-Periodic Attributes Table Select from left and add Periodic attributes table

Period Type Create new Period Type(or select from drop down)

Period Name Enter Period name.

Attribute Description Description of selected (from table below) attribute (editable).

Attribute Information Additional info. about selected (from table below) attribute (editable).

Save Save add/edits.

Cancel Cancel add/edits.

Done Close R-Dobj

Close Close R-DOBJ interface.

View Map View map in a tabular form.

# DORM Studio:

## - Create R-Static Attributes Table

! Optional

\* Repeat (8, 9, 10) for each Attribute.

( Continued on next page.)

\* Create new.

**Dobj Name** Enter name of the Data Object.

**Description** Enter small description of the Data Object.

X Delete from the map.

X Exclude attribute from the Look-Up.

**Static Attributes Table** New R-Static attributes table name

**Attribute Column** New attribute column name.

**Attribute Description** Description of attribute (editable).

**Attribute Information** Additional info. about attribute (editable).

**Row Date/Time Column** Column for row update date/time (optional).

**Save** Save add/edits.

**Cancel** Cancel add/edits.

**View Map** View map in a tabular form.

## DORM Studio: - Create R-Static Attributes Table

**New Table**

**Create Table**

Database: userdb

Table: EMP\_X\_DEPT

Columns:

- EMPLOYEE\_ID
- DEPARTMENT\_ID
- ATTR1\_COL**

Definition:

Column: ATTR1\_COL

Data Type: VARCHAR

Width / Constraint: (10)

DDL Statement:

```
CREATE TABLE userdb.EMP_X_DEPT (  
EMPLOYEE_ID decimal  
,DEPARTMENT_ID decimal  
,ATTR1_COL VARCHAR (10)  
)
```

Execute DDL

Cancel

- \* For each column repeat ①, ② and ③ (for some data types such as VARCHAR)
- ❗ After typing Width (and/or Constraint) press Enter or click in DDL Statement box on the right.

**Database** MyDatabase (space)

**Table** Table to be create

**Columns** Columns to be created

**Data Type** Select Data Type for the column

**Width / Constraint** Enter Width and/or constraint for the column

**Execute DDL**

Create table in the Database

**Cancel**

Cancel table creation

**DORM Studio:**

- Add R-Aperiodic attributes table: Similar to adding Aperiodic Attributes Table.

See [DOBJ: Add/Edit Aperiodic Attributes Table](#) on page 15



**DORM Studio:**

- Create R-Aperiodic attributes table: Similar to creating Aperiodic Attributes Table

See [DOBJ: Create Aperiodic Attributes Table](#) on page 16

**DORM Studio:**

- Add R-Periodic attributes table: Similar to adding Periodic Attributes Table

See [DOBJ: Add/Edit Periodic Attributes Table](#) on page 18

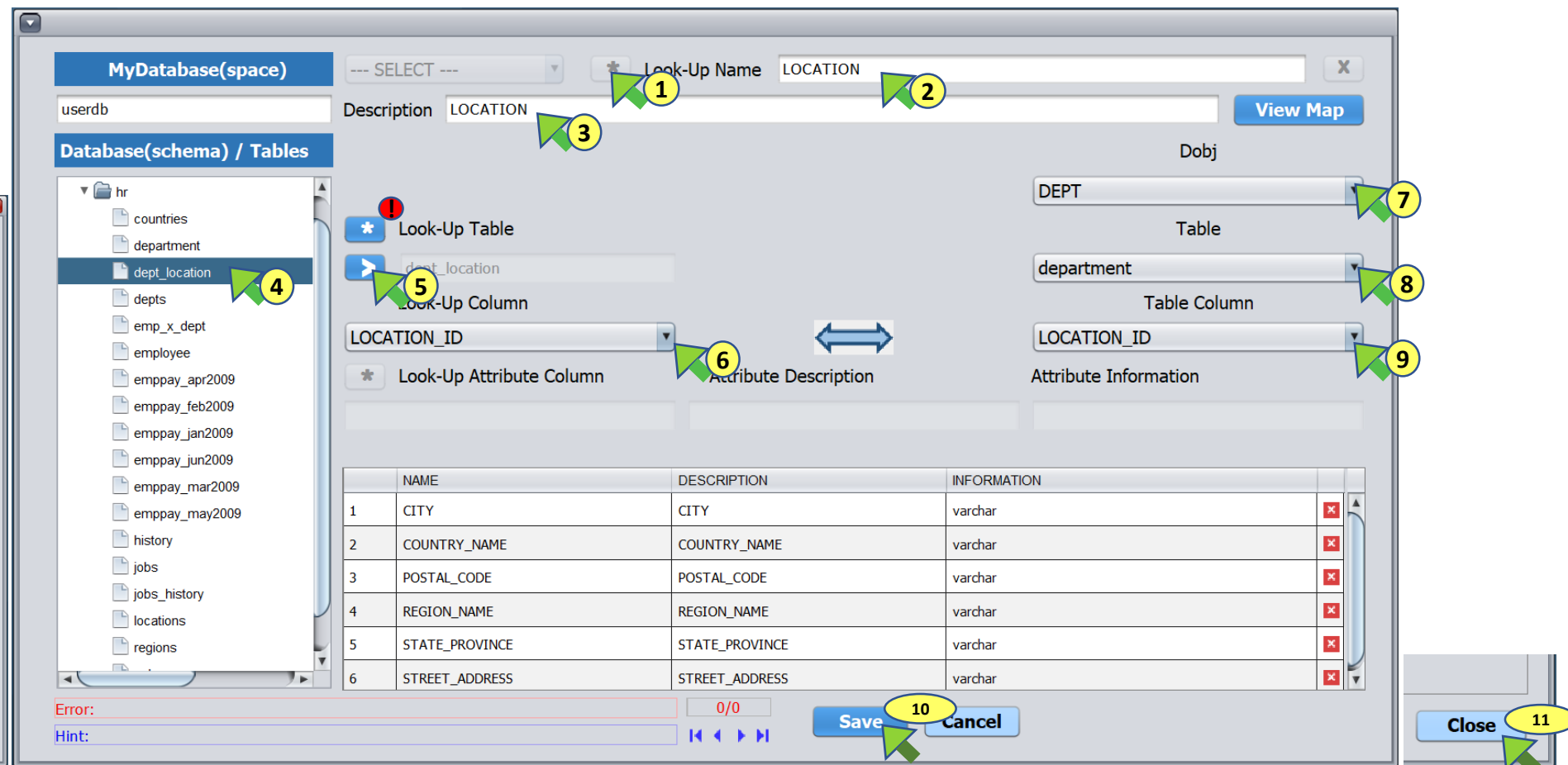
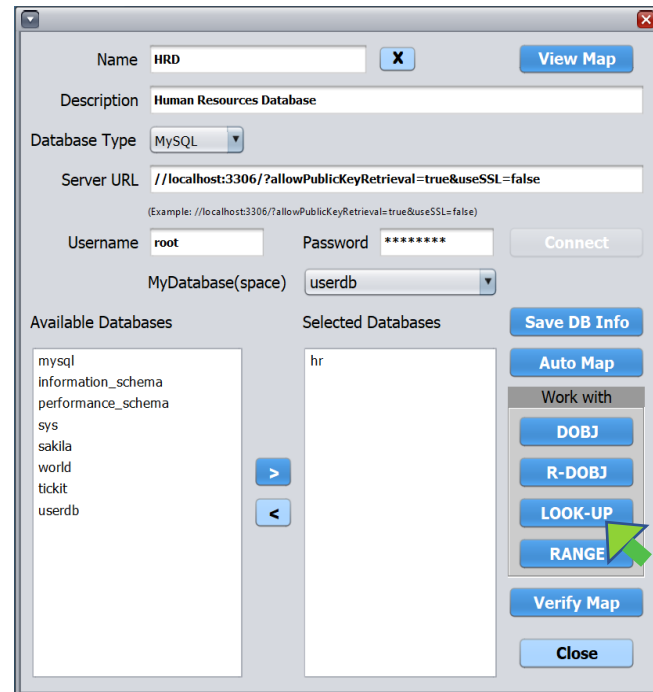
**DORM Studio:**

- Create R-Periodic attributes table: Similar to creating Periodic Attributes Table

See [DOBJ: Create Periodic Attributes Table](#) on page 19

# DORM Studio:

## - New LOOK-UP 'LOCATION'

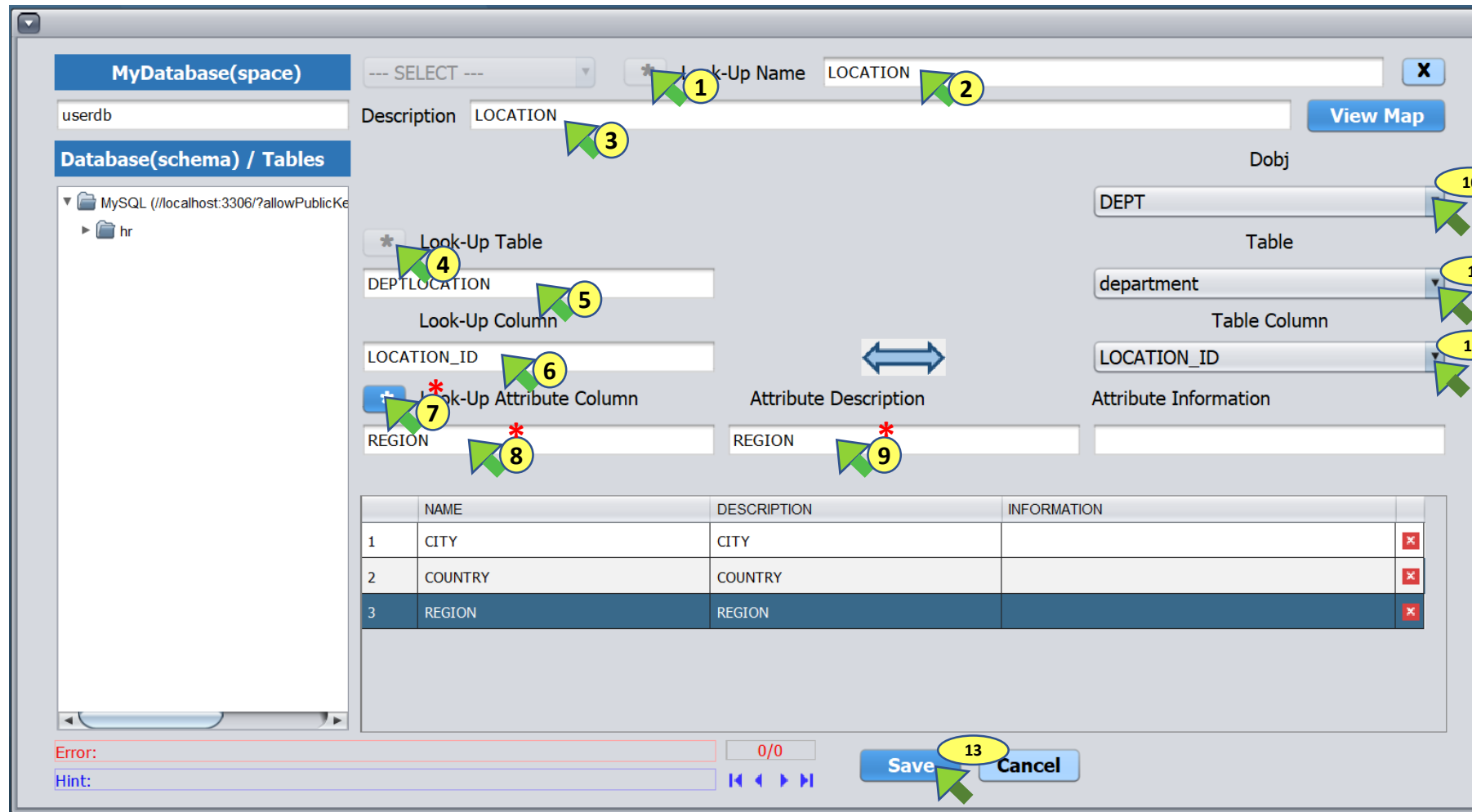


! See next page for creating a Look-Up table.

- Create new.
- Look-Up Name** Enter name of the Look-Up.
- Description** Enter small description of the Look-Up.
- Add selected table's info to Look-Up
- Delete from the map.
- Exclude attribute from the Look-Up.
- Look-Up Table** Selected look-up table
- Look-Up Column** Select look-up column
- Dobj** Select Dobj
- Table** Select a Table of selected Dobj
- Table Column** Select a column of selected Table
- Attribute Description** Description of selected (from table below) attribute (editable).
- Attribute Information** Additional info. about selected (from table below) attribute (editable).
- Save** Save add/edits.
- Cancel** Cancel add/edits.
- Close** Close DOBJ interface.
- View Map** View map in a tabular form.

# DORM Studio:

## - Create Look-Up Table



\* Repeat ( 7 8 9 )  
for each Attribute.

( Continued on next page.)

- Create new.
- Look-Up Name** Enter name of the Look-Up.
- Description** Enter small description of the Look-Up.
- Look-Up Table** New look-up table name
- Look-Up Column** New look-up column name
- Dobj** Select Dobj
- Table** Select a Table of selected Dobj
- Table Column** Select a column of selected Table
- Look-Up Attribute Column** New look-up attribute column name
- Attribute Description** Description of attribute (editable).
- Attribute Information** Additional info. about attribute (editable).
- Save add/edits.
- Cancel add/edits.
- View map in a tabular form.
- Delete from the map.
- Exclude attribute

## DORM Studio:

### - Create Look-Up Table

**New Table**

**Create Table**

Database: userdb

Table: DEPTLOCATION

Columns:

- LOCATION\_ID \*
- CITY
- COUNTRY
- REGION

Definition:

Column: REGION

Data Type: VARCHAR \*

Width / Constraint: (50) !

DDL Statement:

```
CREATE TABLE userdb.DEPTLOCATION (  
LOCATION_ID INT  
,CITY VARCHAR (50)  
,COUNTRY VARCHAR (50)  
,REGION VARCHAR (50)  
)
```

Execute DDL

Cancel

- \* For each column repeat ①, ② and ③ (for some data types such as VARCHAR)
- ! After typing Width (and/or Constraint) press Enter or click in DDL Statement box on the right.

**Database** MyDatabase (space)

**Table** Table to be create

**Columns** Columns to be created

**Data Type** Select Data Type for the column

**Width / Constraint** Enter Width and/or constraint for the column

**Execute DDL**

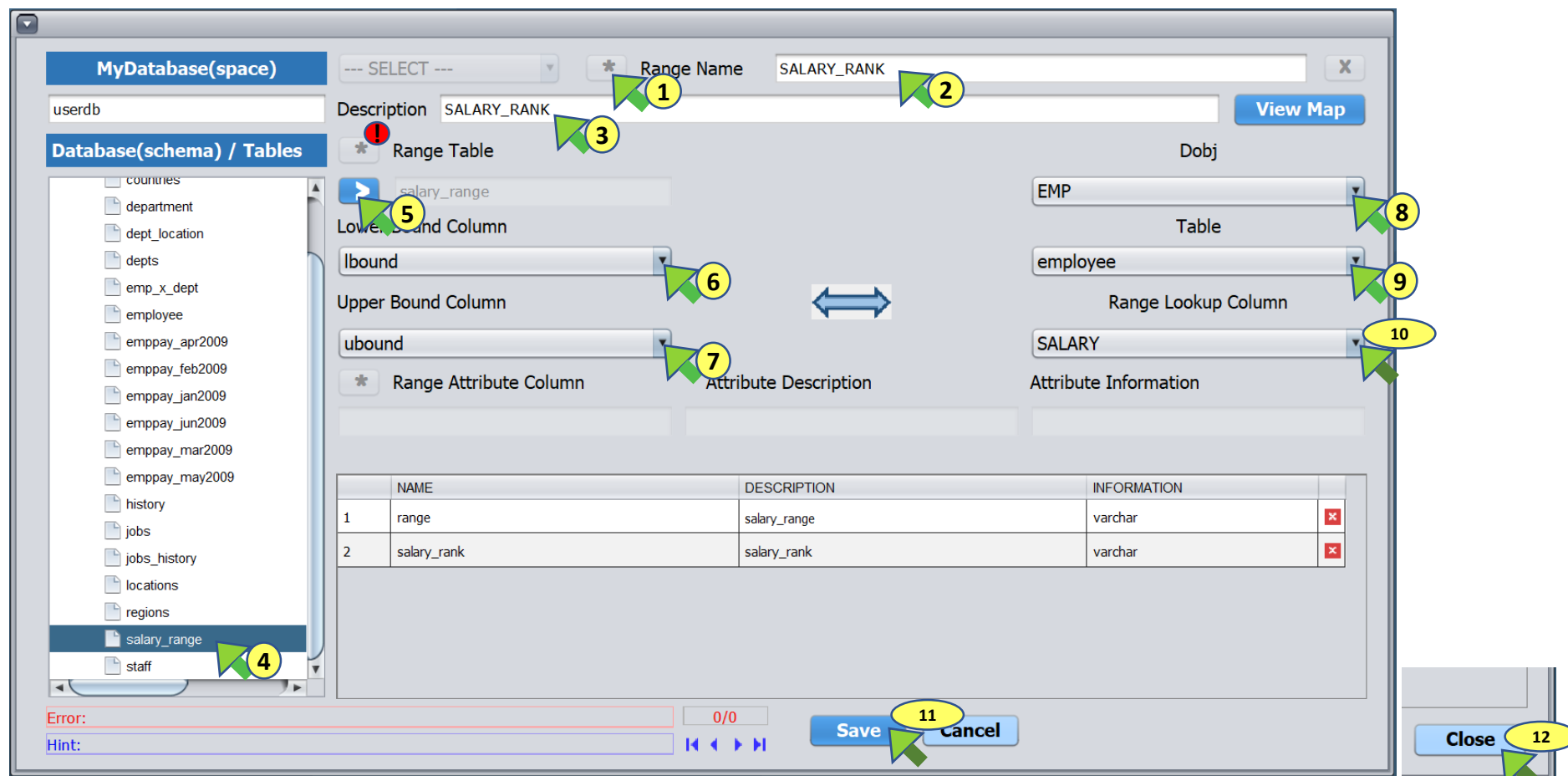
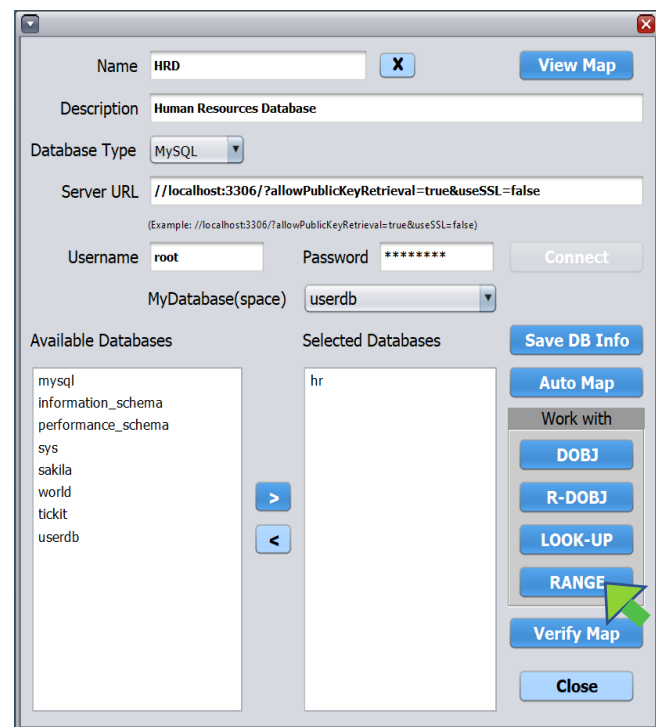
Create table in the Database

**Cancel**





Cancel table creation

# DORM Studio:

## - New RANGE 'SALARY\_RANK'



❗ See next page for creating a Range table.

-  Create new.
- Look-Up Name** Enter name of the Range.
- Description** Enter small description of the Range.
-  Add selected table's info to Range
-  Delete from the map.
-  Exclude attribute from the Range.
- Look-Up Table** Selected look-up table
- Lower Bound Column** Select lower bound column
- Upper Bound Column** Select upper bound column
- Dobj** Select Dobj
- Table** Select a Table of selected Dobj
- Table Column** Select a column of selected Table
- Attribute Description** Description of selected (from table below) attribute (editable).
- Attribute Information** Additional info. about selected (from table below) attribute (editable).
- Save** Save add/edits.
- Cancel** Cancel add/edits.
- Close** Close DOBJ interface.
- View Map** View map in a tabular form.

# DORM Studio:

## - Create Range Table

MyDatabase(space) --- SELECT --- Range Name SALARY\_RANK X

userdb Description SALARY\_RANK View Map

Database(schema) / Tables

- \* Range Table
- SALARY\_RANGE
- Lower Bound Column LOWER\_BOUND
- Upper Bound Column UPPER\_BOUND
- \* Range Attribute Column SALARY\_RANK
- Attribute Description SALARY\_RANK
- Attribute Information

Dobj EMPLOYEE

Table omployee

Range Lookup Column SALARY

	NAME	DESCRIPTION	INFORMATION
1	SALARY_RANGE	SALARY_RANGE	
2	SALARY_RANGE	SALARY_RANGE	

Error: 0/0 Save Cancel

\* Repeat ( 8 9 10 )  
for each Attribute.

( Continued on next page.)

\* Create new.

**Range Name** Enter name of the Range association.

**Description** Enter small description of the range.

**Range Table** New range table name

**Lower Bound Column** New lower bound column name

**Upper Bound Column** New upper bound column name

**Dobj** Select Dobj

**Table** Select a Table of selected Dobj

**Table Column** Select a column of selected Table

**Range Attribute Column** New range attribute column name

**Attribute Description** Description of attribute (editable).

**Attribute Information** Additional info. about attribute (editable).

**Save** Save add/edits.

**Cancel** Cancel add/edits.

**View Map** View map in a tabular form.

**X** Delete from the map.

**X** Exclude attribute.



## DORM Studio: - Create Range Table

**Create Table**

Database: userdb

Table: SALARY\_RANGE

Columns:

- LOWER\_BOUND
- UPPER\_BOUND
- SALARY\_RANGE**
- SALARY\_RANK

Definition:

Column: SALARY\_RANGE

Data Type: VARCHAR

Width / Constraint: (50)

DDL Statement:

```
CREATE TABLE userdb.SALARY_RANGE (  
LOWER_BOUND REAL  
,UPPER_BOUND REAL  
,SALARY_RANGE VARCHAR (50)  
,SALARY_RANK VARCHAR (50)  
)
```

Execute DDL

Cancel

- \* For each column repeat ①, ② and ③ (for some data types such as VARCHAR)
- ❗ After typing Width (and/or Constraint) press Enter or click in DDL Statement box on the right.

**Database** MyDatabase (space)

**Table** Table to be create

**Columns** Columns to be created

**Data Type** Select Data Type for the column

**Width / Constraint** Enter Width and/or constraint for the column

**Execute DDL**

Create table in the Database

**Cancel**

Cancel table creation

# DORM Studio:

## - View Map

Name: HRD  
Description: Human Resources Database  
Database Type: MySQL  
Server URL: //localhost:3306/?allowPublicKeyRetrieval=true&useSSL=false  
Username: root  
Password: \*\*\*\*\*  
MyDatabase(space): userdb  
Available Databases: mysql, information\_schema, performance\_schema, sys, sakila, world, tickit, userdb  
Selected Databases: hr  
Buttons: View Map, Connect, Save DB Info, Auto Map, DOBJ, R-DOBJ, LOOK-UP, RANGE, Verify Map, Close

**Map: HRD** [Details] [Export to CSV] [Close]

DEPT (L) LOCATION			*
EMP_DEPT	EMP (R) SALARY_RANK		

**Map: HRD** [Export to CSV] [Back]

DATA OBJECT	DATABASE.TABLE	TABLE CATEGORY	COLUMN	LOOKUP/RANGE COLUMN	LOOKUP/RANGE DATABASE.TABLE	LOOKUP/RANGE NAME
DEPARTMENTS	hr.department	S	DPTMGR_SALARY			
DEPARTMENTS	hr.department	S	LOCATION_ID			
DEPARTMENTS	hr.department	S	LOCATION_ID	CITY	hr.dept_location	LOCATION
DEPARTMENTS	hr.department	S	LOCATION_ID	COUNTRY_NAME	hr.dept_location	LOCATION
DEPARTMENTS	hr.department	S	LOCATION_ID	LOCATION_ID	hr.dept_location	LOCATION
DEPARTMENTS	hr.department	S	LOCATION ID	POSTAL CODE	hr.dept_location	LOCATION

**Data Object : DEPT** [Export to CSV] [Back]

DATABASE.TABLE	TABLE CATEGORY	COLUMN	LOOKUP/RANGE COLUMN	LOOKUP/RANGE DATABASE.TABLE	LOOKUP/RANGE NAME
hr.department	S	DEPARTMENT_ID			
hr.department	S	DEPARTMENT_NAME			
hr.department	S	DPTMGR_COMMISSION_PCT			
hr.department	S	DPTMGR_EMAIL			

\* DOBJs are shown in diagonal sequence (i.e. DEPT and EMP), R-DOBJs are shown at row-column intersection of the two related DOBJs.

! To view details of an object, click on the cell

**Details** Show database, tables and columns of all the data objects

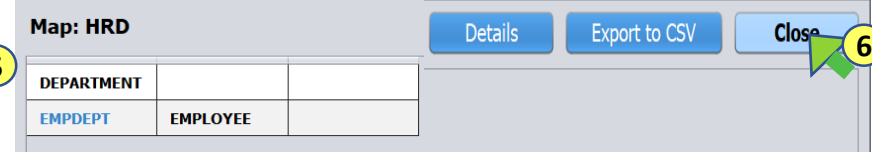
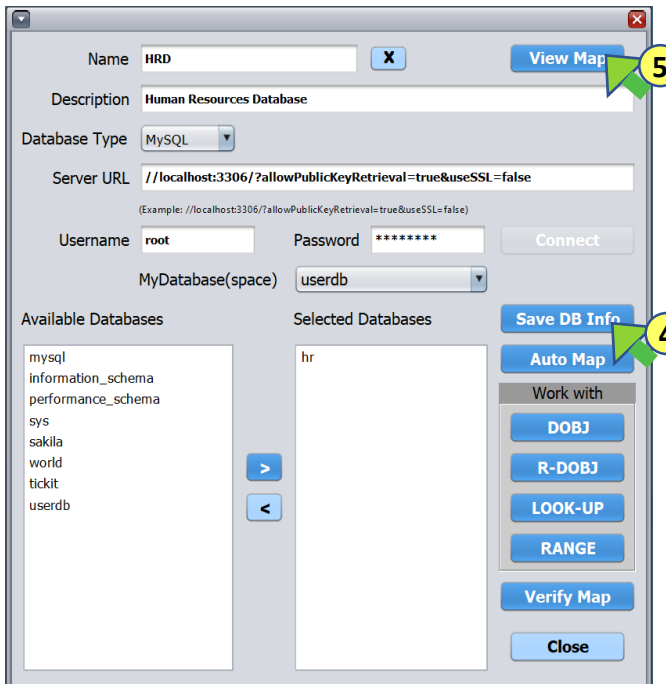
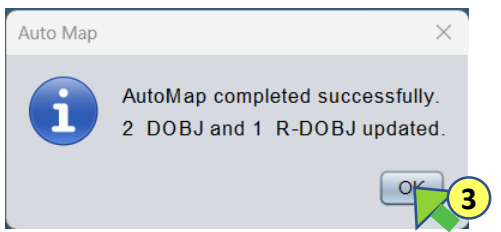
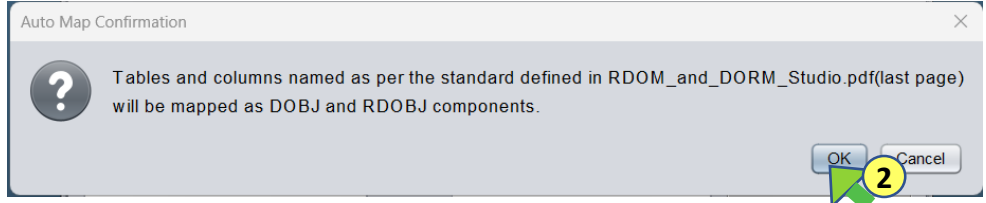
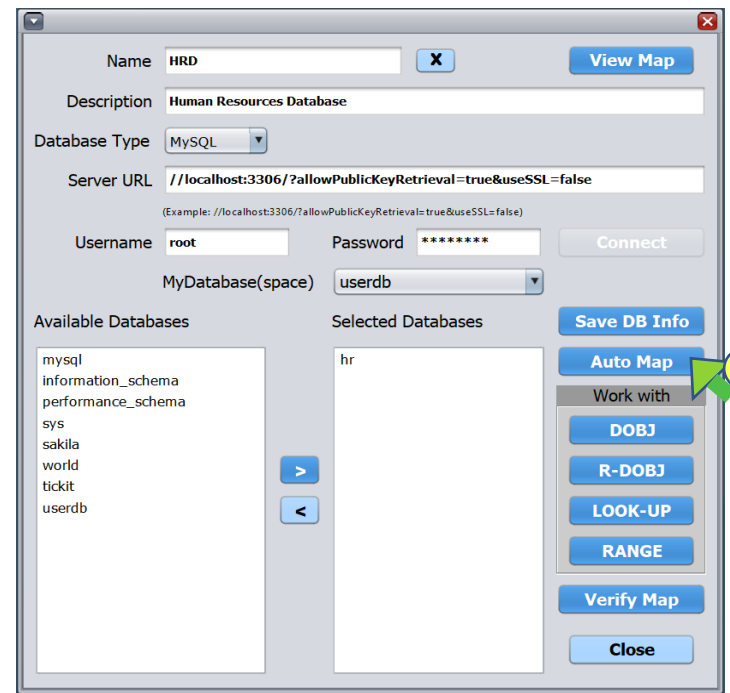
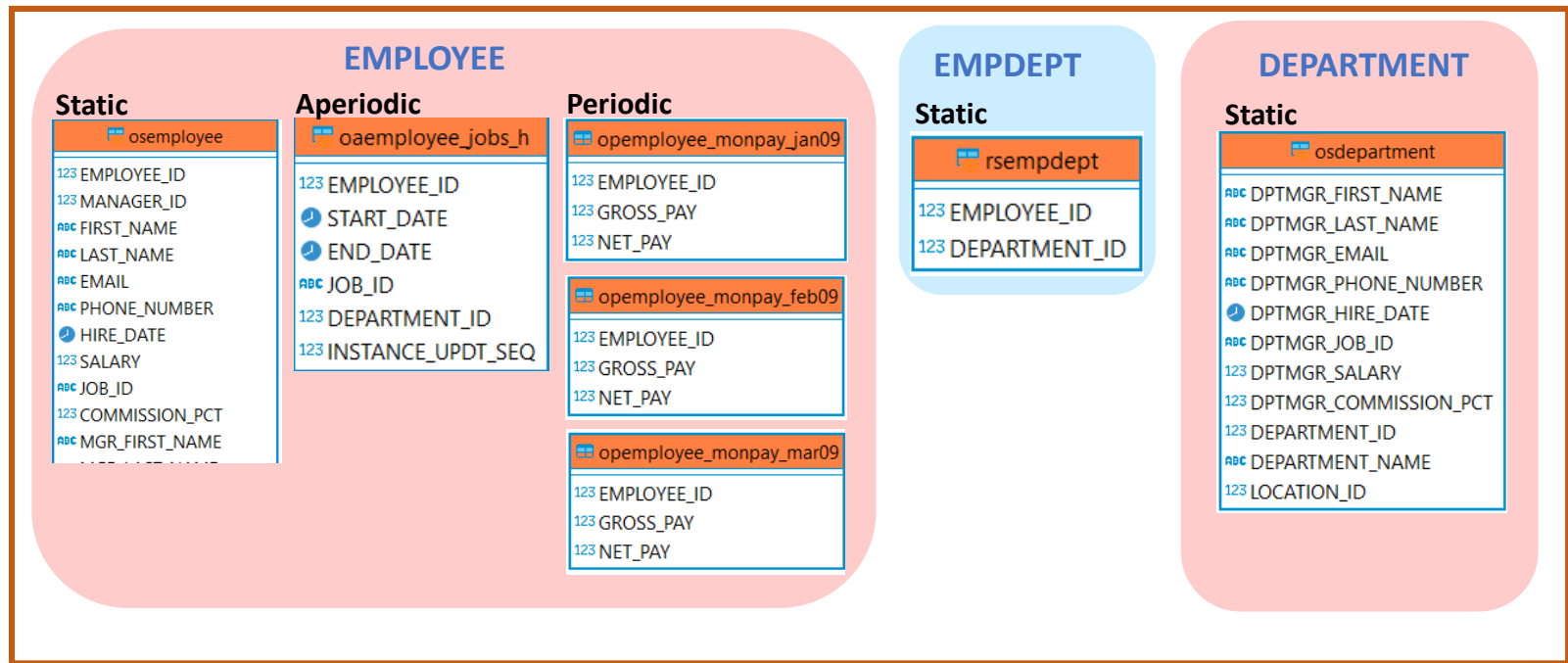
**Export to CSV** Export current view to CSV file.

**Back** Show map view

**Close** Close map view

## - AutoMap

In this example there are two DOBJs (EMPLOYEE and DEPARTMENT) and one R-DOBJ (EMPDEPT). Tables and columns are named as per the standard\*.



\* See 'Standard names for tables and columns' on page 38.

## - Verify Map

- Check for NULL values in ID columns of all tables.
- Check for duplicate values in ID columns of all tables.
- Verify integrity of object components relationships.

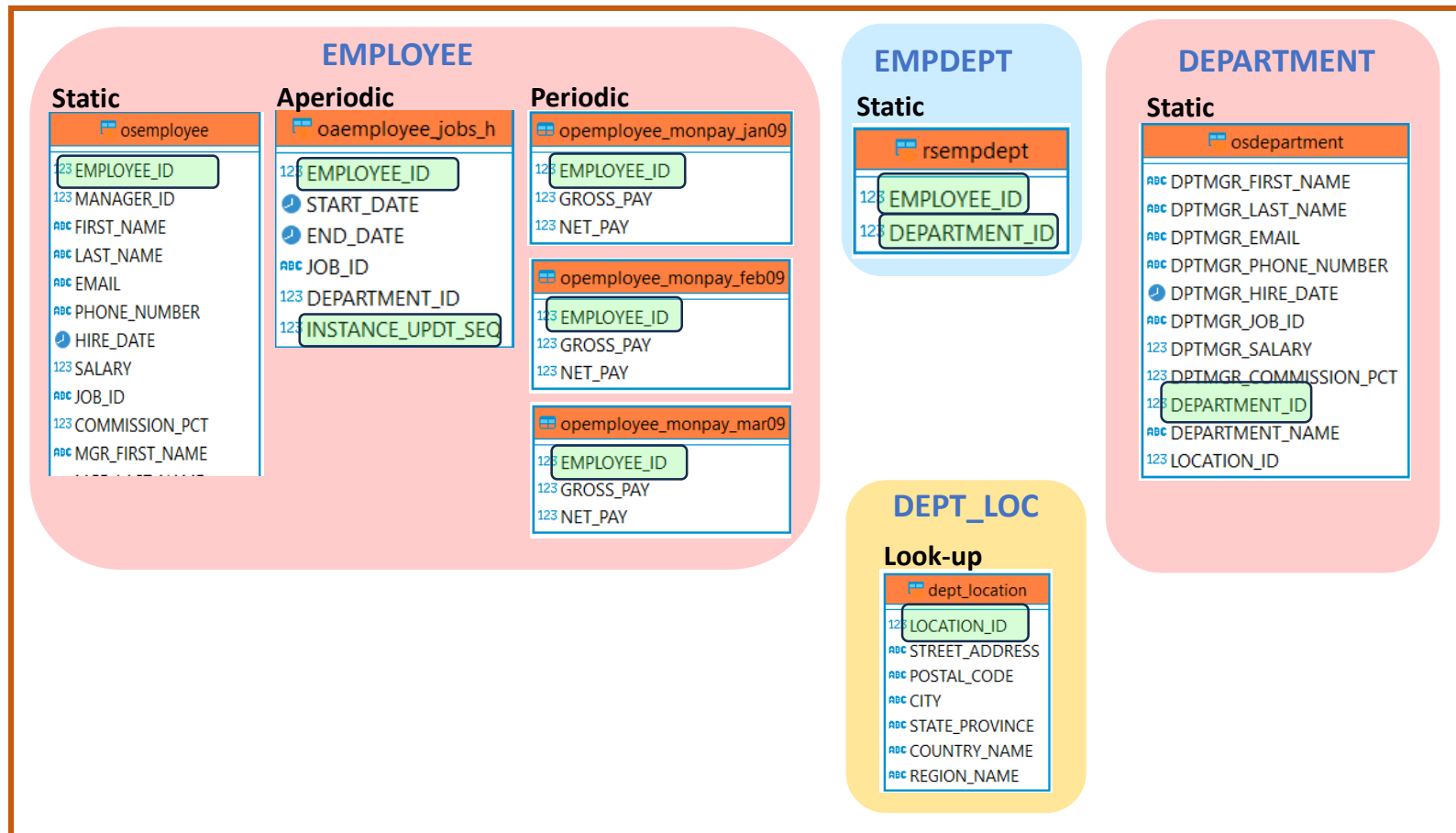
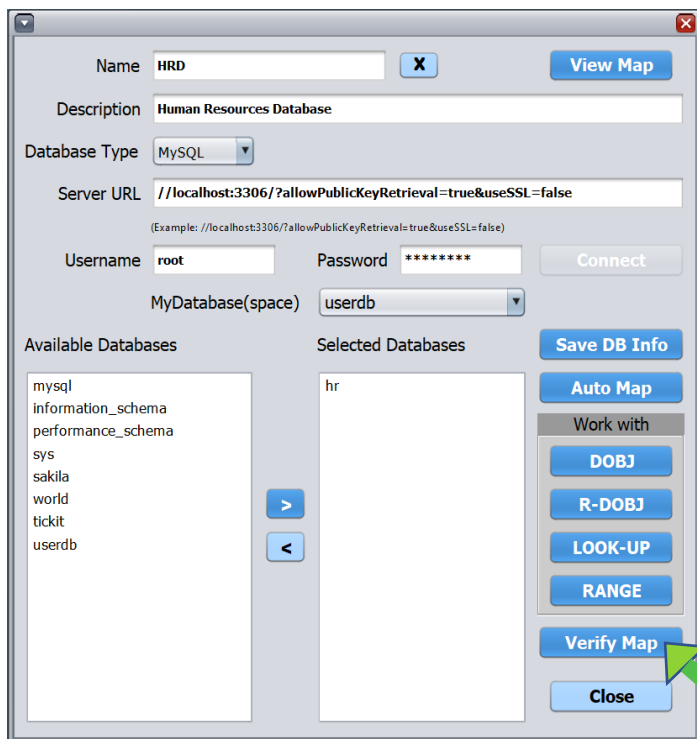
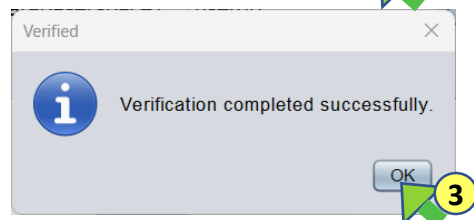
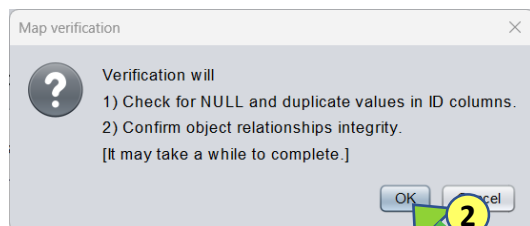


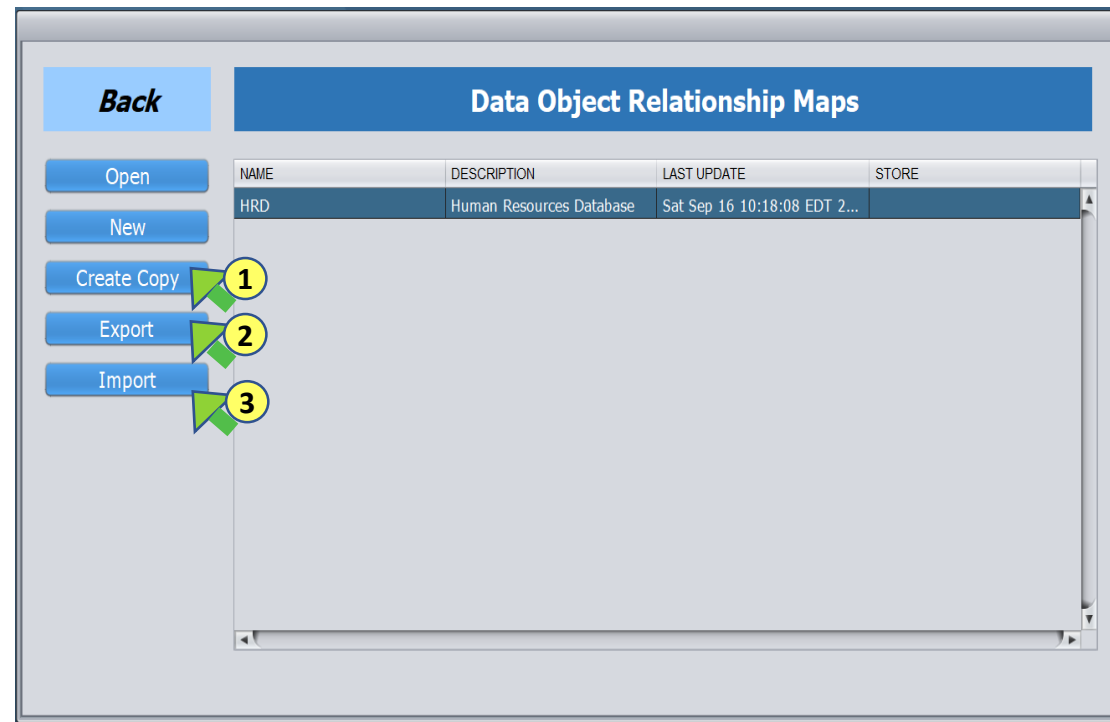
Fig. 1

## - Verification process:

- 1) Check for NULL and duplicate values in ID columns (highlighted in green in Fig 1)
- 2) Check for non-ID values in object components:
  - i) EMPLOYEE\_ID column of oaemployee\_jobs\_h table must not have value that does not exist in EMPLOYEE\_ID column of osemployee table.
  - ii) EMPLOYEE\_ID column of opemployee\_monpay\_jan09, opemployee\_monpay\_feb09 and opemployee\_monpay\_mar09 tables must not have value that does not exist in EMPLOYEE\_ID column of osemployee table.
  - iii) EMPLOYEE\_ID and DEPARTMENT\_ID columns of rsempdept table must not have value that does not exist in EMPLOYEE\_ID and DEPARTMENT\_ID columns of osemployee and osdepartment tables respectively.



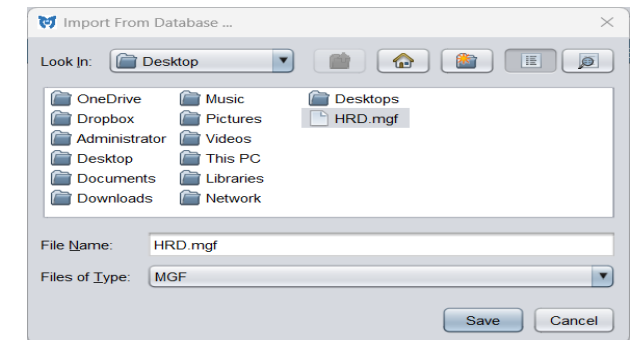
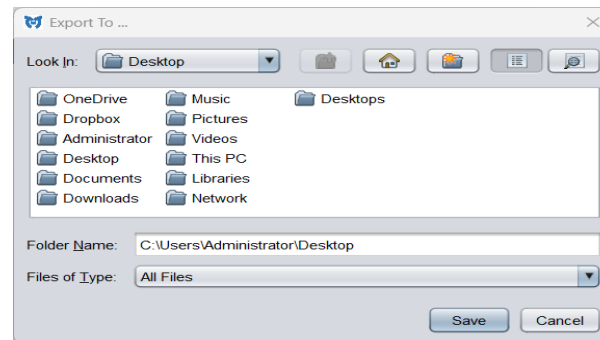
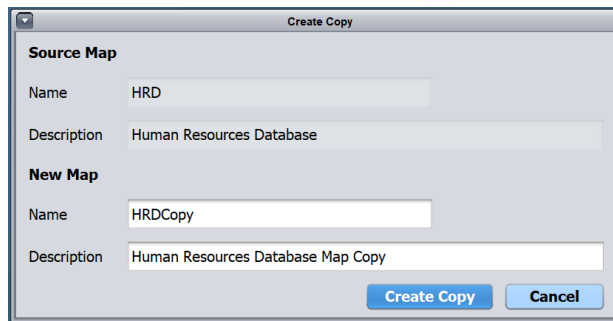
## - Export/Import/Copy:



**1. Create Copy:** Duplicates selected map.

**2. Export:** Exports selected map as a file.

**3. Import:** Imports map from a file.



**Standard names for tables and columns to auto-map:** DORM Studio's AutoMap feature creates map entries for DOBJ and R-DOBJ components from the tables and columns named using following standard (Fig. 5).

**DOBJ:**

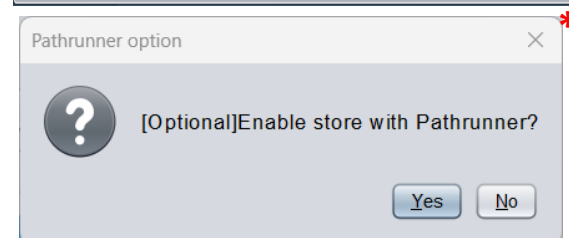
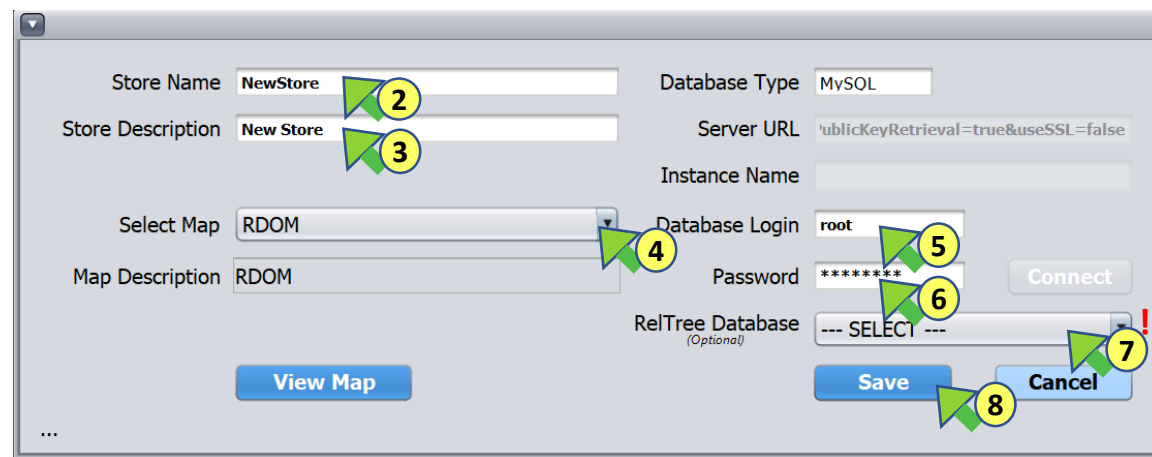
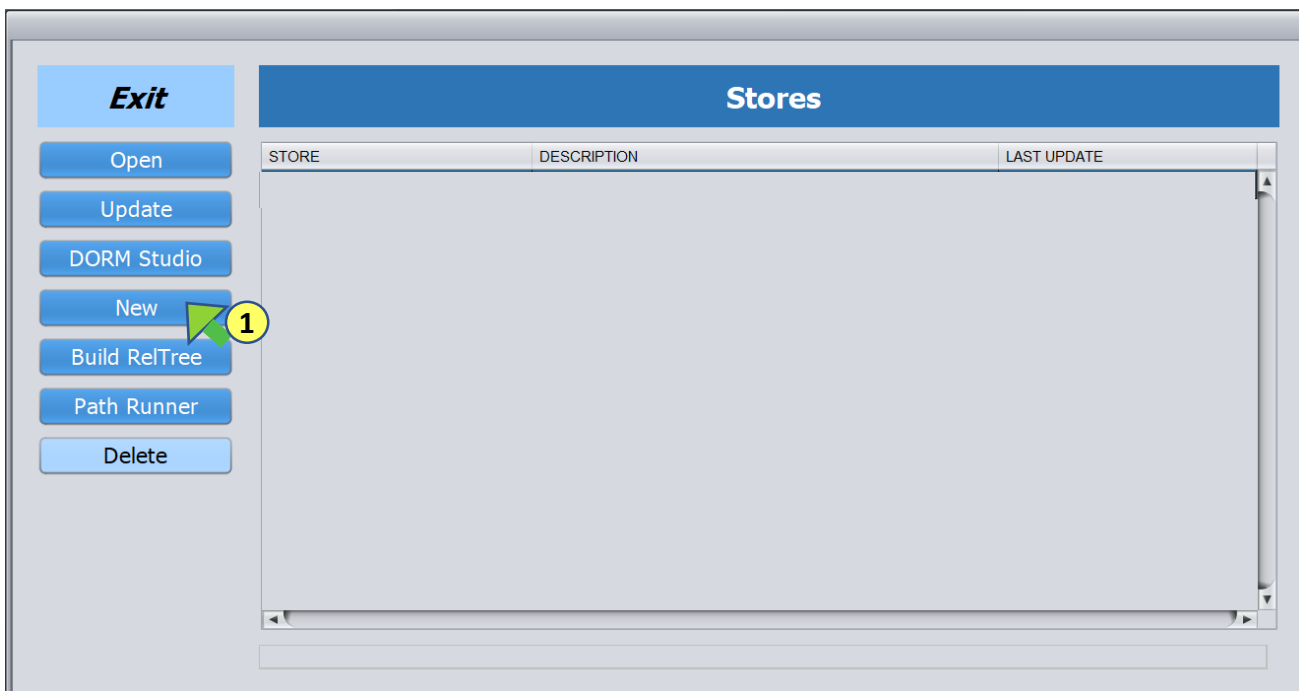
Table Type	Table Name	Column Name	Note
Static	OS<dobj>	<dobj>_ID	<dobj> is user declared DOBJ name. <dobj> must start with a character.
		INSTANCE_UPDT_DTTM	(Optional) Row update time stamp
Aperiodic	OA<dobj>_<string>	<dobj>_ID	<string> can be any set of characters.
		INSTANCE_UPDT_SEQ	Column for instance update sequence number.
		INSTANCE_UPDT_DTTM	(Optional) Row update time stamp
Periodic	OP<dobj>_<p_type>_<p_desc>	<dobj>_ID	<p_type> is user declared period type. <p_desc> is user declared period description. <p_type> and <p_desc> must not contain '_' (underscore character).
		INSTANCE_UPDT_DTTM	(Optional) Row update time-stamp.

**RDOBJ:**

Table Type	Table Name	Column Name	Note
Static	RS<rdobj>	<dobj1id> <dobj2id>	<rdobj> is user declared RDOBJ name. <rdobj> must start with a character. <dobj1id> and <dobj2id> must be respective DOBJs' ID column names.
		INSTANCE_UPDT_DTTM	(Optional) Row update time stamp
Aperiodic	RA<rdobj>_<string>	<dobj1id> <dobj2id>	<string> can be any set of characters.
		INSTANCE_UPDT_SEQ	Column for instance update sequence number.
		INSTANCE_UPDT_DTTM	(Optional) Row update time stamp
Periodic	RP<rdobj>_<p_type>_<p_desc>	<dobj1id> <dobj2id>	<p_type> is user declared period type. <p_desc> is user declared period description. <p_type> and <p_desc> must not contain '_' (underscore character).
		INSTANCE_UPDT_DTTM	(Optional) Row update time-stamp.

## Store:

### - Create new.



! Database(or Schema) where RelTree tables are to be created. If this database/schema is selected then 'Build RelTree' button, on main menu, will get activated.

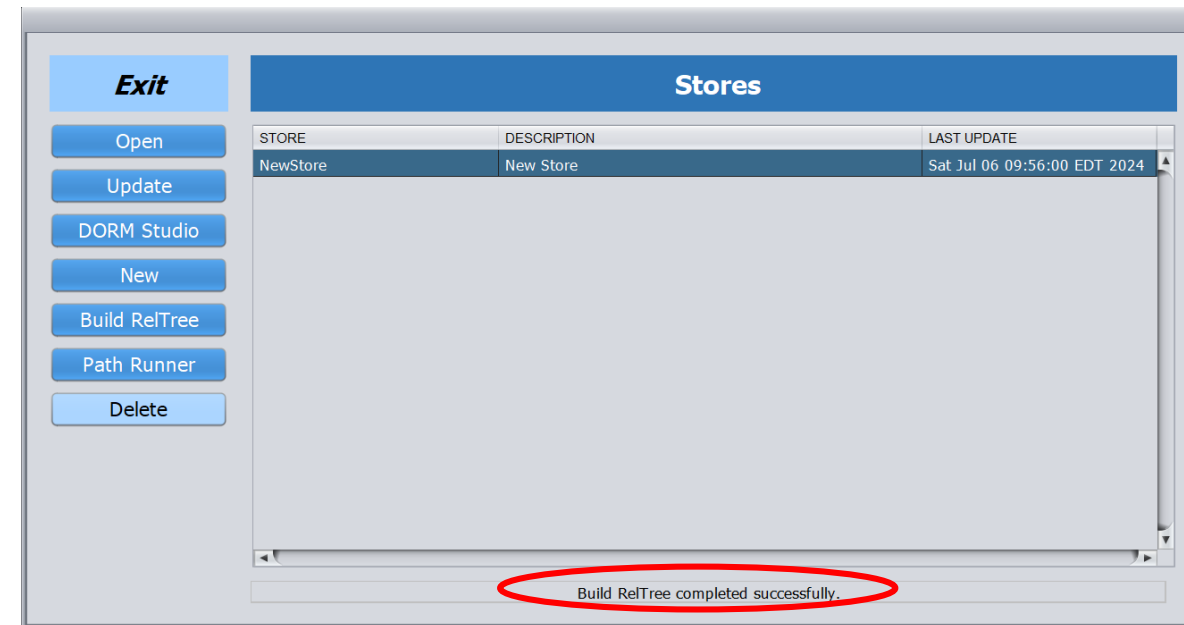
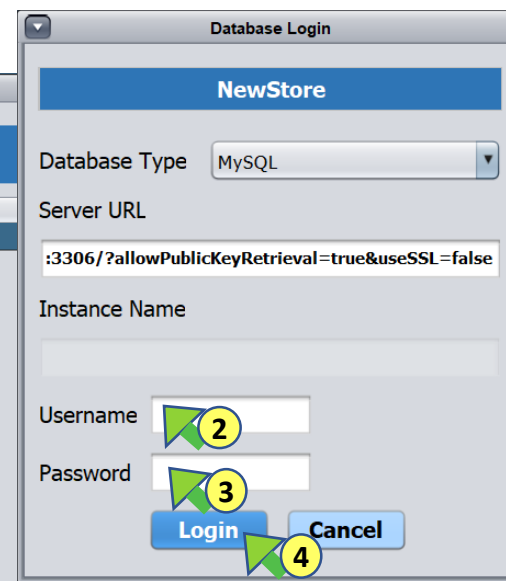
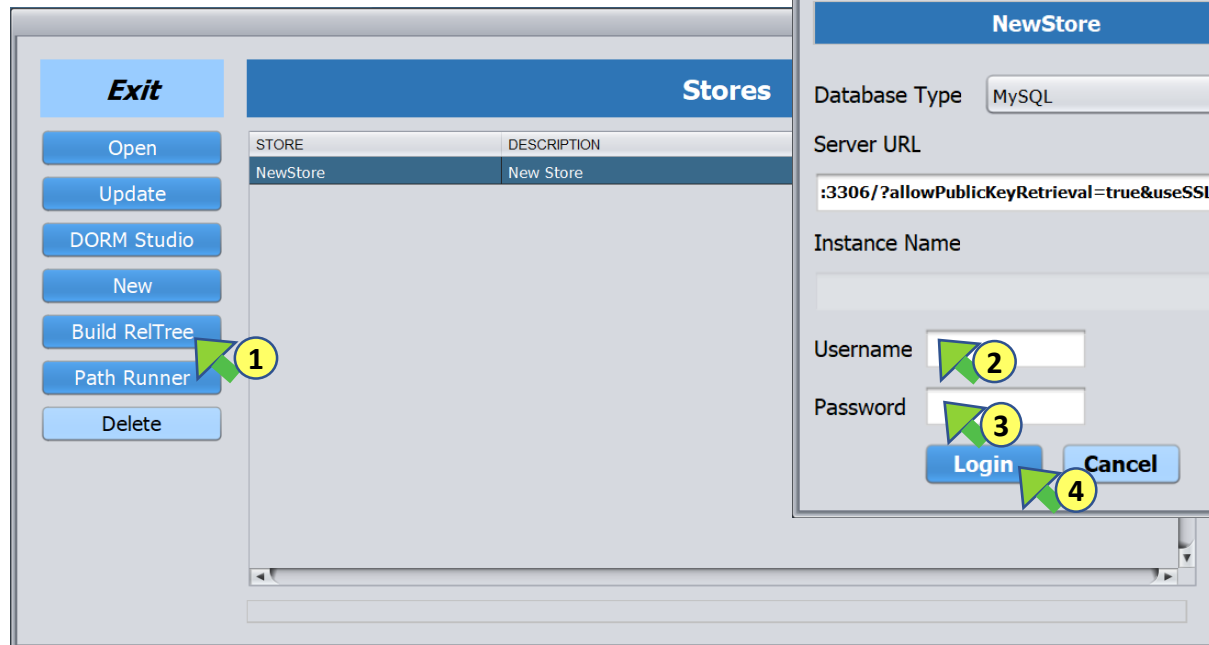
\* The prompt for Pathrunner will appear if the map contains more then two Relationship Data Objects (RDOBJ) and all Static attribute tables of RDOBJs are tables (not views).

<b>Open</b>	Open selected store.
<b>Update</b>	Update selected store after associated map update.
<b>DORM Studio</b>	Open DORM Studio.
<b>New</b>	Create new store.
<b>Build RelTree</b>	Create/update RelTree tables in RelTree database/schema.
<b>Path Runner</b>	Run PathRunner to update RDOBJ Static tables.
<b>Delete</b>	Delete selected store.
<b>Exit</b>	Exit application.

<b>Store Name</b>	Enter name of the store to be created.
<b>Store Description</b>	Enter brief description of the store.
<b>Select Map</b>	Select map to be associated with the store.
<b>Database Type</b>	Type of the database. (FYI)
<b>Instance Name</b>	MS SQL Server Instance name. (FYI)
<b>RelTree Database</b>	(Optionally) Select RelTree Database/schema.
<b>Server URL</b>	Network address of the database.(FYI)
<b>Save</b>	Create store.
<b>Cancel</b>	Cancel store creation

## Store:

### - (Optional) Run Build RelTree\*.



	Open selected store.
	Update selected store after associated map update.
	Open DORM Studio.
	Create new store.
	Create/update RelTree tables in RelTree database/schema.
	Run PathRunner to update RDOBJ Static tables.

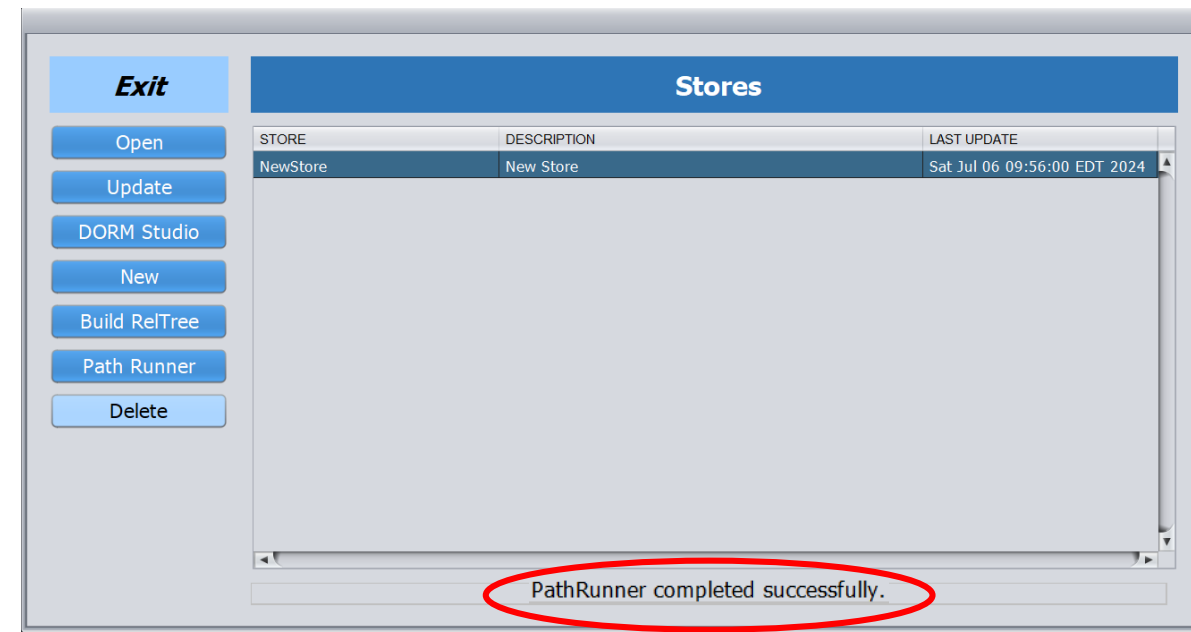
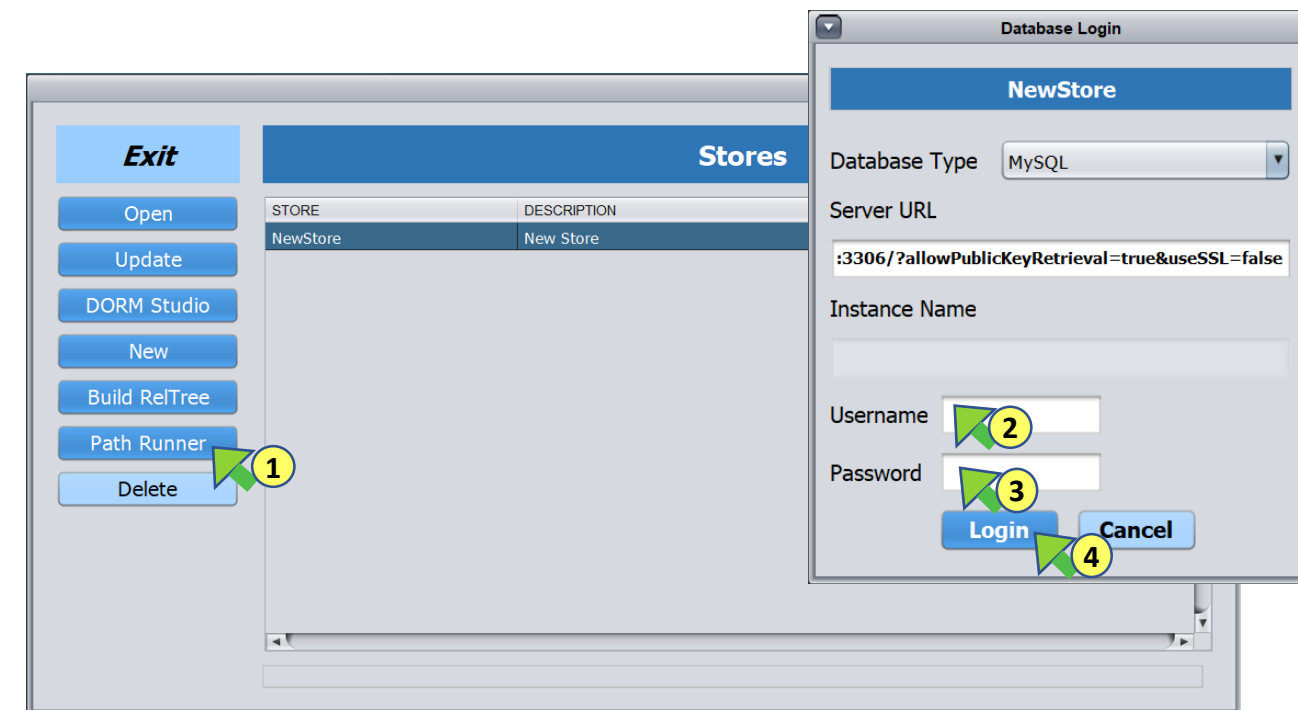
	Login and start Build RelTree
	Cancel Build RelTree run.
	Delete selected store.
	Exit application.

\* See [Appendix-2: Build RelTree](#) for what/how/why RelTree.



## Store:

### - (Optional) Run PathRunner\*.

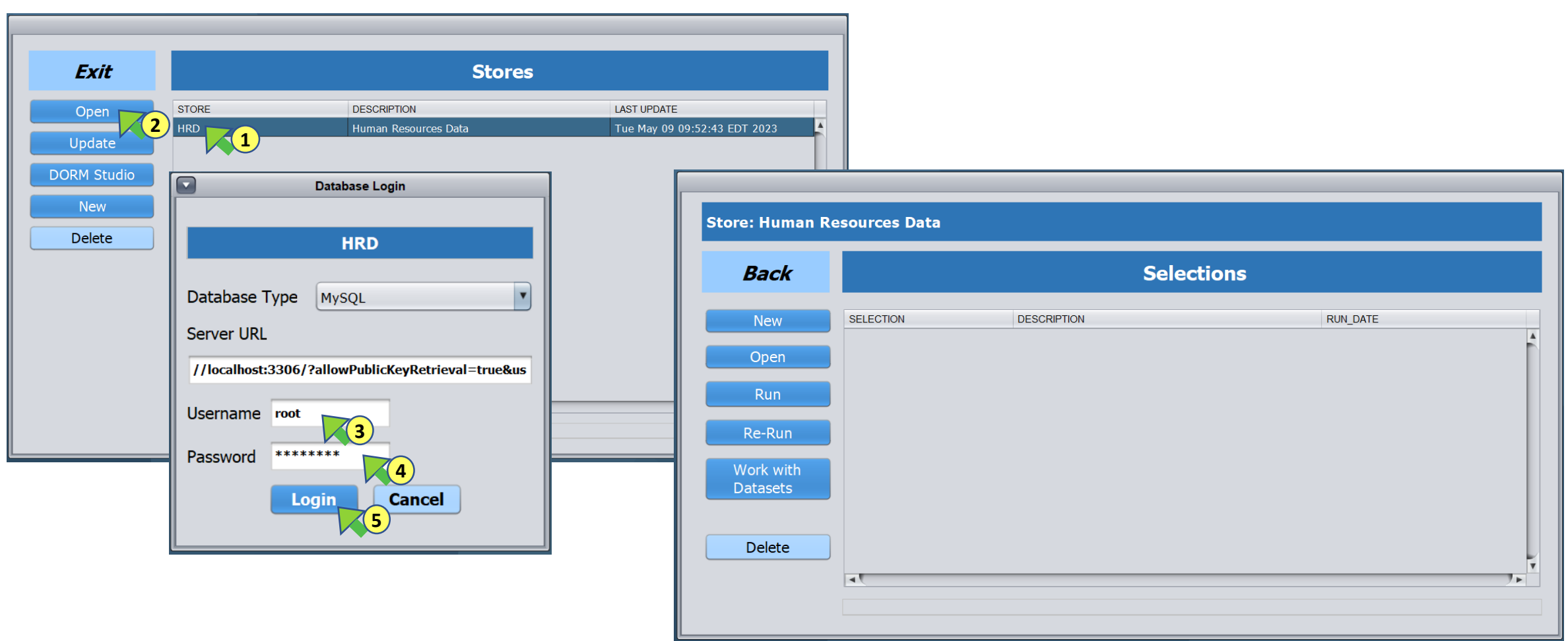


	Open selected store.
	Update selected store after associated map update.
	Open DORM Studio.
	Create new store.
	Create/update RelTree tables in RelTree database/schema.
	Run PathRunner to update RDOBJ Static Attributes tables.

	Login and start PathRunner
	Cancel Build RelTree run.
	Delete selected store.
	Exit application.

\* See [Appendix-3: Pathrunner](#) for what/how/why PathRunner.

**Store:**  
- Open

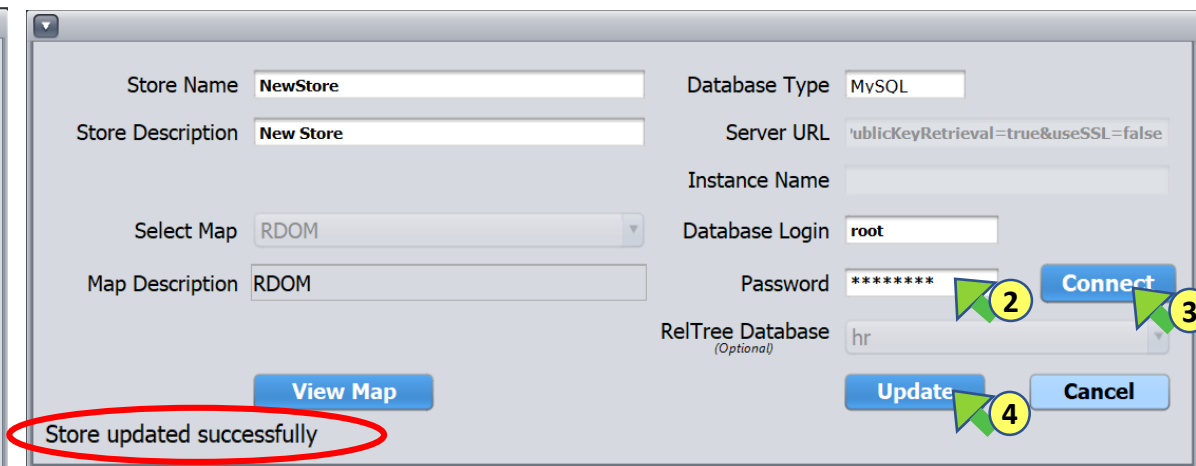
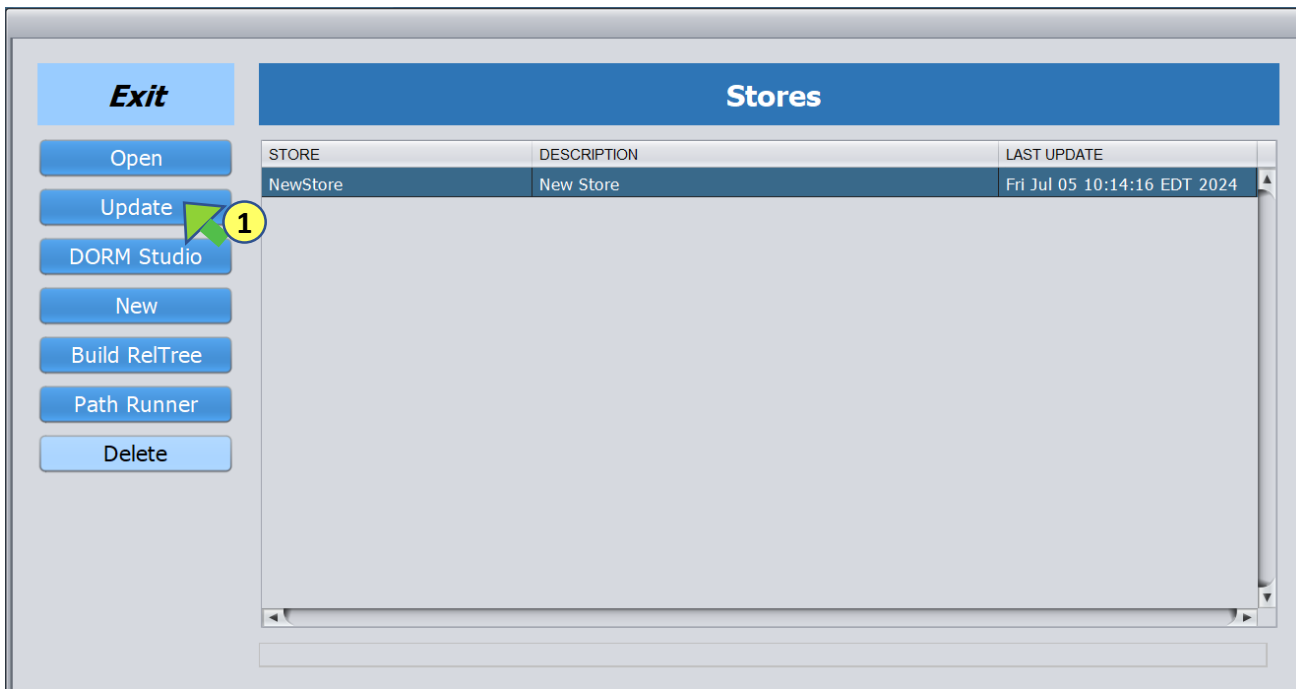


- Database Type** Type of the database associated with the store.
- Server URL** Network address of the database.
- Username** Enter database username
- Password** Enter database password
- Login** Login to the database
- Cancel** Cancel login

- New** Create new Selection.
- Open** Open selected Selection.
- Run** Run selected Selection.
- Re-Run** Re-Run selected Selection including associated analysis and subsetting.
- Work with Datasets** Open list of dataset/s created by the Selection Run (or Re-Run)
- Delete** Delete selected Selection and associated analysis and subsets.

## Store:

### - Update Store.



<b>Open</b>	Open selected store.
<b>Update</b>	Update selected store after associated map update.
<b>DORM Studio</b>	Open DORM Studio.
<b>New</b>	Create new store.
<b>Build RelTree</b>	Create/update RelTree tables in RelTree database/schema.
<b>Path Runner</b>	Run PathRunner to update RDOBJ Static tables.
<b>Delete</b>	Delete selected store.
<b>Exit</b>	Exit application.

<b>Store Name</b>	Name of the store.
<b>Store Description</b>	Description of the store.
<b>Select Map</b>	Map associated with the store.
<b>Database Type</b>	Type of the database. (FYI)
<b>Instance Name</b>	MS SQL Server Instance name. (FYI)
<b>RelTree Database</b>	RelTree Database/schema(FYI).
<b>Server URL</b>	Network address of the database.(FYI)
<b>Update</b>	Update Store based on map update/s.
<b>Cancel</b>	Cancel update.

## Store:

### - Create New Selection: Select attribute/s.

	OBJECT	ATTRIBUTE	OPTION	PERIOD	INFO	
1	DEPT_	DEPARTMENT_NAME			Department Name	X
2	EMP_	SALARY			decimal	X
3	EMP_	MONTHLY_GROSS_PAY		JAN2009	decimal	X
4	EMP_	MONTHLY_NET_PAY		JAN2009	decimal	X

! In case of aperiodic attribute, four options(ALL, CURRENT, ORIGINAL or PICK) will be available to choose from.

\* Periodic attributes are shown as <period\_type>\_<attribute\_name) with list of periods as sub-menu.

**New** Create new Selection.

**Open** Open selected Selection.

**Run** Run selected Selection.

**Re-Run** Re-Run selected Selection including analysis and subsets.

**Work with Datasets** Open the list of dataset/s created by the Selection Run (or Re-Run)

**Delete** Delete selected Selection and associated analysis and subsets.

**Name** Enter name of the selection.

**Description** Enter brief description of the selection.

**Object** Object of the selected attribute.

**Attribute** Selected attribute.

**Option** Select option for aperiodic attribute.

**Attribute Info** (optional) Enter additional information about the attribute.

**Add** Add selected attribute to selection.

X Remove attribute from the selection.

**Save** Save the selection.

**Close** Close selection interface.

CURRENT: Most recently added instance/s(row/s). For example, CURRENT 2 (i.e. select most recent two instances).  
ORIGINAL: Initially added instance/s(row/s). For example, ORIGINAL 3 (i.e. select 1<sup>st</sup>, 2<sup>nd</sup> and 3<sup>rd</sup> instances).  
ALL: All instances(rows).  
PICK: Specific instance/s (rows). For example, PICK 3 5 (i.e. select 3<sup>rd</sup> and 5<sup>th</sup> instances).

## Store:

- Create New Selection: Select Object/s. All attributes of selected object gets added to selection.

**Store: Human Resources Data**

**Objects/Attributes**

Name: DEPTSALARY Save Cancel

Description: Employee Salary and Department

Object: EMPLOYEE Attribute Info

Attribute:

Option: ALL

**Add** 3

	OBJECT	ATTRIBUTE	OPTION	PERIOD	INFO
10	EMPLOYEE_	MGR_EMAIL			E
11	EMPLOYEE_	MGR_FIRST_NAME			E
12	EMPLOYEE_	MGR_HIRE_DATE			E
13	EMPLOYEE_	MGR_JOB_ID			E
14	EMPLOYEE_	MGR_LAST_NAME			E
15	EMPLOYEE_	MGR_PHONE_NUMBER			E
16	EMPLOYEE_	MGR_SALARY			E
17	EMPLOYEE_	PHONE_NUMBER			E
18	EMPLOYEE_	SALARY			E
19	EMPLOYEE_	DEPARTMENT_ID	ALL*		E
20	EMPLOYEE_	END_DATE	ALL		E
21	EMPLOYEE_	JOB_ID_18986	ALL		E
22	EMPLOYEE_	START_DATE	ALL		E
23	EMPLOYEE_	MONPAY_GROSS_PAY		APR09	E

\* All aperiodic attributes get added with 'ALL' option.

**New** Create new Selection.

**Open** Open selected Selection.

**Run** Run selected Selection.

**Re-Run** Re-Run selected Selection including analysis and subsets.

**Work with Datasets** Open the list of dataset/s created by the Selection Run (or Re-Run)

**Delete** Delete selected Selection and associated analysis and subsets.

**Name** Enter name of the selection.

**Description** Enter brief description of the selection.

**Object** Object of the selected attribute.

**Attribute** Selected attribute.

**Option** Select option for aperiodic attribute.

**Attribute Info** (optional) Enter additional information about the attribute.

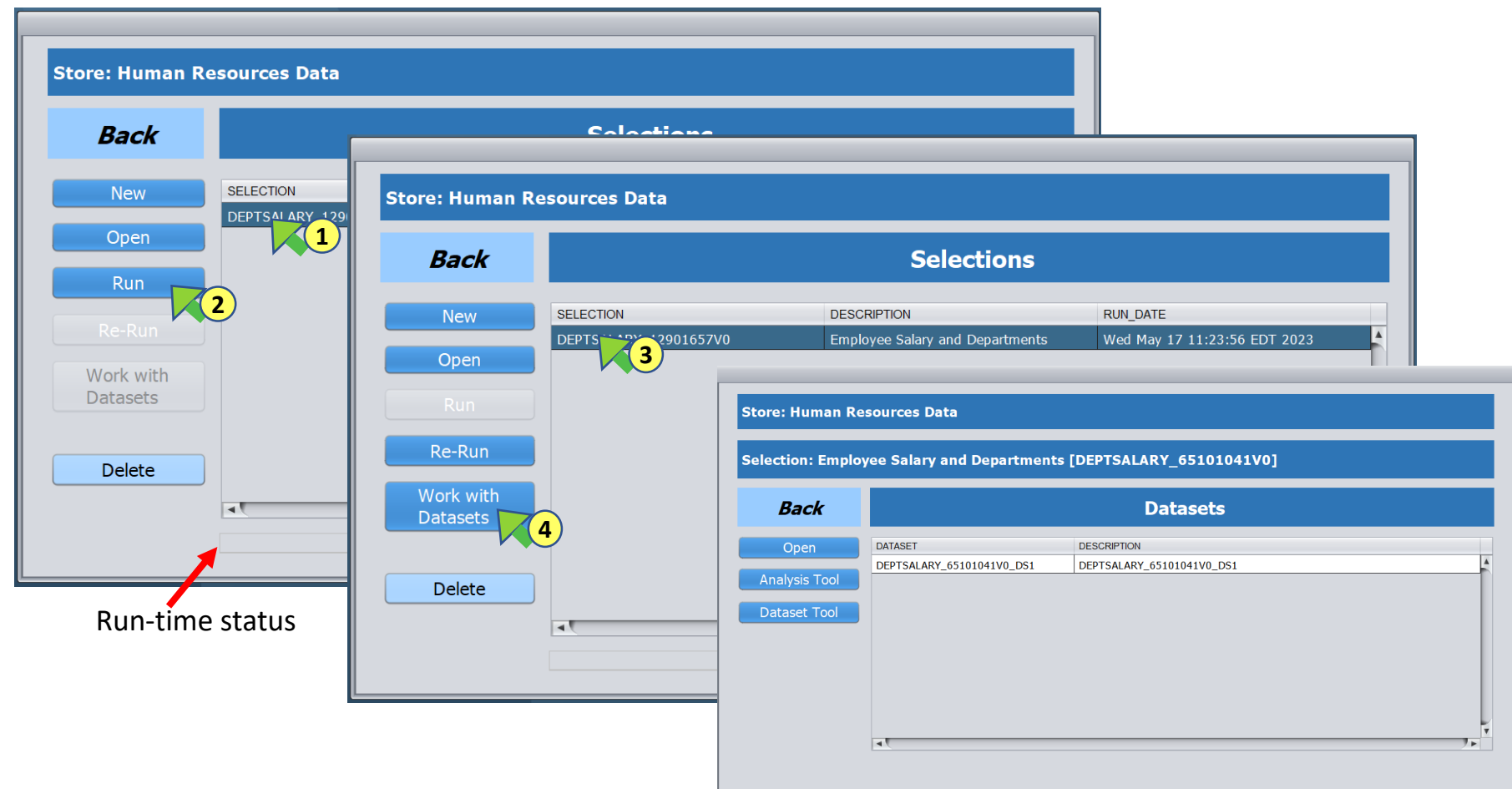
**Add** Add selected attribute to selection.

Remove attribute from the selection.

**Save** Save the selection.

**Close** Close selection interface.

**Store:**  
**- Run Selection.**



- New** Create new Selection.
- Open** Open selected Selection.
- Run** Run selected Selection.
- Re-Run** Re-Run selected Selection including analysis and subsets.
- Work with Datasets** Open the list of dataset/s created by the Selection Run (or Re-Run)
- Delete** Delete selected Selection and associated analysis and subsets.

# Work with Datasets:

- Open Dataset\*.
- View Object Relations Summary\*

The screenshot shows the 'View Dataset' application interface. On the left, there is a sidebar with buttons for 'Back', 'Open', 'Analysis Tool', and 'Dataset Tool'. The main area displays the 'Object Relations Summary' table, which includes columns for 'DEPARTMENTS', 'EMPLOYEES', and their relationships. Below the table, there are sections for 'Object Group Size', 'Objects Count(total)', and 'Related Objects Count'. On the right, there are buttons for 'Change Desc', 'Export to CSV', and 'Analysis Tool'. A red arrow points to the 'Department Name' attribute in the 'Selected attributes' section.

Selection : Employee Salary and Departments [DEPTSALARY\_12901657V0]  
Dataset : DEPTSALARY\_DS1

	DEPARTMENTS DEPARTMENT_ID	EMPLOYEES EMPLOYEE_ID	DEPARTMENTS DEPARTMENT_NAME	DEPARTMENTS REGION_NAME	EMPLOYEES FIRST_NAME	EMPLOYEES SALARY	EMPLOYEES MONTHLY_GROSS_PAY_FEB2009	EMPLOYEES MONTHLY_NET_P
1	90	100	Executive	Americas	Steven	24000.00	2000.00	1600.00
2	90	101	Executive	Americas	Neena	17000.00	1416.67	1133.33
3	90	102	Executive	Americas	Lex	17000.00	1416.67	1133.33
4	60	103	IT	Americas	Alexander	9000.00	750.00	600.00
5	60	104	IT	Americas	Bruce	6000.00	500.00	400.00

Selection: Employee Salary and Departments    Dataset: DEPTSALARY\_DS1

Object Group Size = 2    Object Group Size = 1

Objects Count(total)	
Object Count	
DEPT_	11
EMP_	107

Related Objects Count

Object Count	
Object Count	
DEPT_	11
EMP_	106

Report

DEPARTMENT DEPARTMENT_ID	EMPLOYEE EMPLOYEE_ID	DEPARTMENT DEPARTMENT	
1	90	100	Executive
2	90	101	Executive
3	90	102	Executive

This subset (ObjectRelSmry04) is added to Report Tool.

Selected attributes' information.

❗ Table (dataset) includes selected attributes plus data object IDs. Column header includes data-object names at top and attribute names.

**Object Relations Summary** View data-objects and exclusively related instances' count.

**Object Group Size** Data-object groups (combinations) in descending order(from left to right columns).

**Objects Count(total)** Total number of object instances

**Related Objects Count** Total number of related object instances

**Analysis Tool** Open Analysis Tool.

**Dataset Tool** Open Dataset Tool.

**Change Desc** Change dataset description.

**Export to CSV** Export dataset as two CSV files. ( Data file and Metadata file)

\* **Export to CSV** Export relationship summary as CSV file.

⏪ View first page (first 1000 rows).

⏴ View previous page.

⏵ View next page.

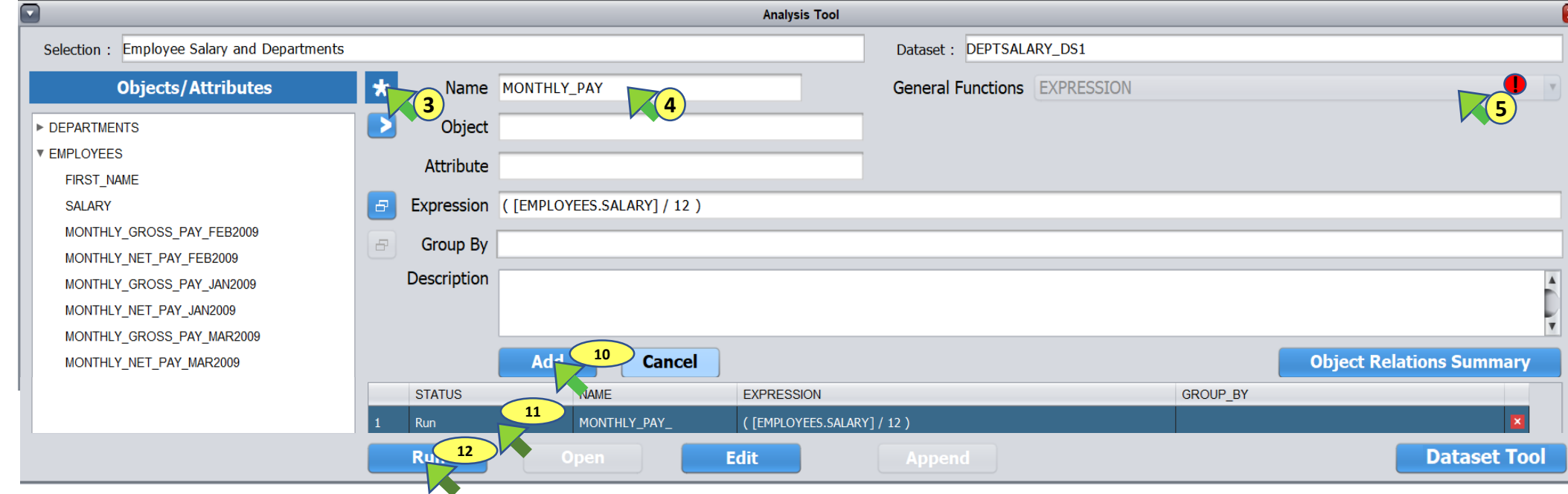
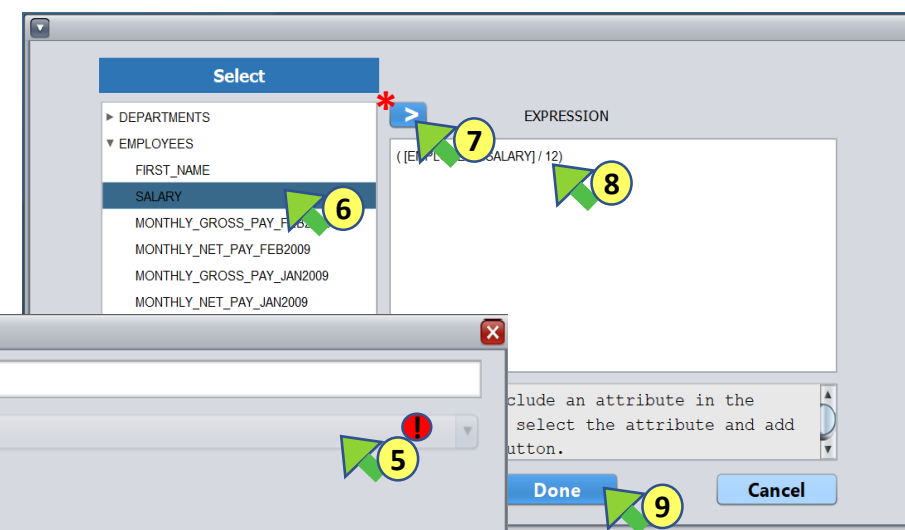
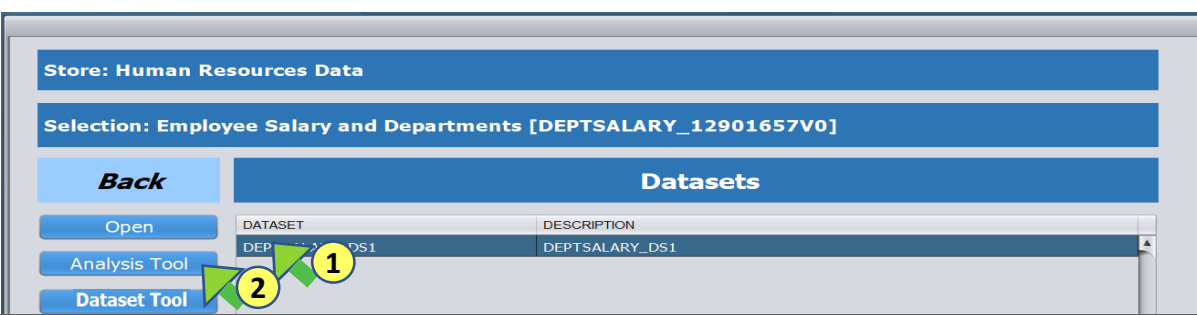
⏩ View last page.

✖ Close window.

\* See 'Appendix-4: Create Dataset' and 'Appendix-5: Object Relations Summary' for details. For metadata details see Metadata section.

# Work with Datasets:

- Analysis Tool
- General Functions
- EXPRESSION



! Select 'EXPRESSION'

\* Create new analysis.

Name Enter name.

General Functions Select a function.

> Add object/attribute to analysis.

Object Selected object.

Attribute Selected attribute.

Expression Edit expression.

Group By Edit Group By.

Description Add/Edit analysis description.

Add Add analysis to the list.

Cancel Cancel entry/edits.

Run Run selected analysis.

Open Open selected analysis result.

Edit Edit selected analysis.

Append Append statistic to the dataset.

✗ Delete selected analysis (in 'Run' or 'Saved' state).

\* > Add selected attribute to Expression.

Done Finish entry.

Object Relations Summary View data-objects and their relations count.

Dataset Tool Open Dataset Tool.



# Work with Datasets:

- Analysis Tool
- General Functions
- CASE STATEMENT

The screenshot shows the 'Analysis Tool' window with the following components:

- Store:** Human Resources Data
- Selection:** Employee Salary and Departments [DEPTSALARY\_12901657V0]
- Buttons:** Back, Open, Analysis Tool, Dataset Tool
- Datasets Table:**

DATASET	DESCRIPTION
DEPTSALARY_DS1	DEPTSALARY_DS1
- Analysis Tool Fields:**
  - Selection: Employee Salary and Departments
  - Dataset: DEPTSALARY\_DS1
  - Name: SALARY\_RANK
  - Object: (empty)
  - Attribute: (empty)
  - Case Stmt: EMPLOYEES.SALARY] < 8000 THEN "LOW" WHEN ([EMPLOYEES.SALARY] >= 8000 AND [EMPLOYEES.SALARY] < 12000) THEN "MEDIUM" ELSE "HIGH"
  - Group By: (empty)
  - Description: (empty)
- General Functions:** CASE STATEMENT (selected)
- Case Statement Editor:**
  - Select: DEPARTMENTS, EMPLOYEES
  - Case Statement: WHEN [EMPLOYEES.SALARY] < 8000 THEN "LOW" WHEN ([EMPLOYEES.SALARY] >= 8000 AND [EMPLOYEES.SALARY] < 12000) THEN "MEDIUM" ELSE "HIGH" END
- Buttons:** Add, Cancel, Run, Open, Edit, Append, Object Relations Summary, Dataset Tool
- Table:**

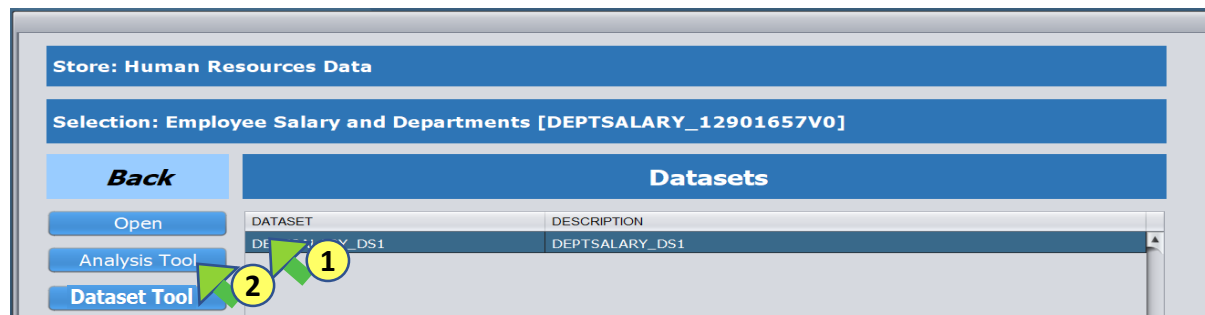
	STATUS	NAME	EXPRESSION	GROUP_BY
1	Run	SALARY_RANK_	CASE WHEN [EMPLOYEES.SALARY] < 8000 THEN 'LOW' WHEN [EMPLO...	

! Select 'CASE STATEMENT'

- \* Create new analysis.
- Name Enter name.
- General Functions Select a function.
- > Add object/attribute to analysis.
- Object Selected object.
- Attribute Selected attribute.
- Case Stmt Edit Case Statement.
- Group By Edit Group By.
- Description Add/Edit analysis description.
- Add Add analysis to the list.
- Cancel Cancel entry/edits.
- Run Run selected analysis.
- Open Open selected analysis result.
- Edit Edit selected analysis.
- Append Append statistic to the dataset.
- \* > Add selected attribute to statement.
- \* \* Add a WHEN/THEN clause.
- ✎ Edit selected clause (i.e. from statement)
- Done Finish entry.
- Object Relations Summary View data-objects and their relations count.
- Dataset Tool Open Dataset Tool.
- ✘ Delete selected analysis (in 'Run' or 'Saved' state).

## Work with Datasets:

- Analysis Tool
- Dobj Functions
- EMBED FN-CODE



! Select 'EMBED FN-CODE'

	EMPLOYEES EMPLOYEE_ID	STATISTICS FIRSTTWOCHAROFEMPID_
1	100	10
2	101	10
3	102	10
4	103	10

\* Create new analysis.

Name Enter name.

Dobj Functions Select a function.

> Add object/attribute to analysis.

Object Selected object.

Attribute Selected attribute.

Group By Edit Group By.

Description Add/Edit analysis description.

Add Add analysis to the list.

Cancel Cancel entry/edits.

Run Run selected analysis.

Open Open selected analysis result.

Edit Edit selected analysis.

Append Append statistic to the dataset.

✖ Delete selected analysis (in 'Run' or 'Saved' state).

Object Relations Summary View data-objects and their relations count.

Dataset Tool Open Dataset Tool.

## Work with Datasets:

### - Analysis Tool

### - Aggregate Functions

### - MAX with Group By

The screenshot shows the Analysis Tool interface with the following components and callouts:

- 1**: Analysis Tool button in the Datasets panel.
- 2**: Dataset Tool button in the Datasets panel.
- 3**: Create new analysis icon (\*).
- 4**: Name field (MaxSalaryByDept).
- 5**: Selected attribute (SALARY) in the Objects/Attributes list.
- 6**: Object field (EMPLOYEES).
- 7**: Aggregate Functions dropdown (MAX).
- 8**: Group By field.
- 9**: Selected object (DEPARTMENT\_NAME) in the Select dialog.
- 10**: Add button (>> Add) in the Select dialog.
- 11**: Done button in the Select dialog.
- 12**: Add button in the configuration form.
- 13**: Run button in the configuration form.
- 14**: Run button at the bottom of the configuration form.

STATUS	NAME	EXPRESSION	GROUP_BY	
1	Run	MAXSALARYBYDEPT_	MAX([EMPLOYEES.SALARY])	DEPARTMENT_NAME

**\*** Create new analysis.

**Name** Enter name.

**Obj Functions** Select a function.

**>** Add object/attribute to analysis.

**Object** Selected object.

**Attribute** Selected attribute.

**Group By** Edit Group By.

**Description** Add/Edit analysis description.

**Add** Add analysis to the list.

**Cancel** Cancel entry/edits.

**Run** Run selected analysis.

**Open** Open selected analysis result.

**Edit** Edit selected analysis.

**Append** Append statistic to the dataset.

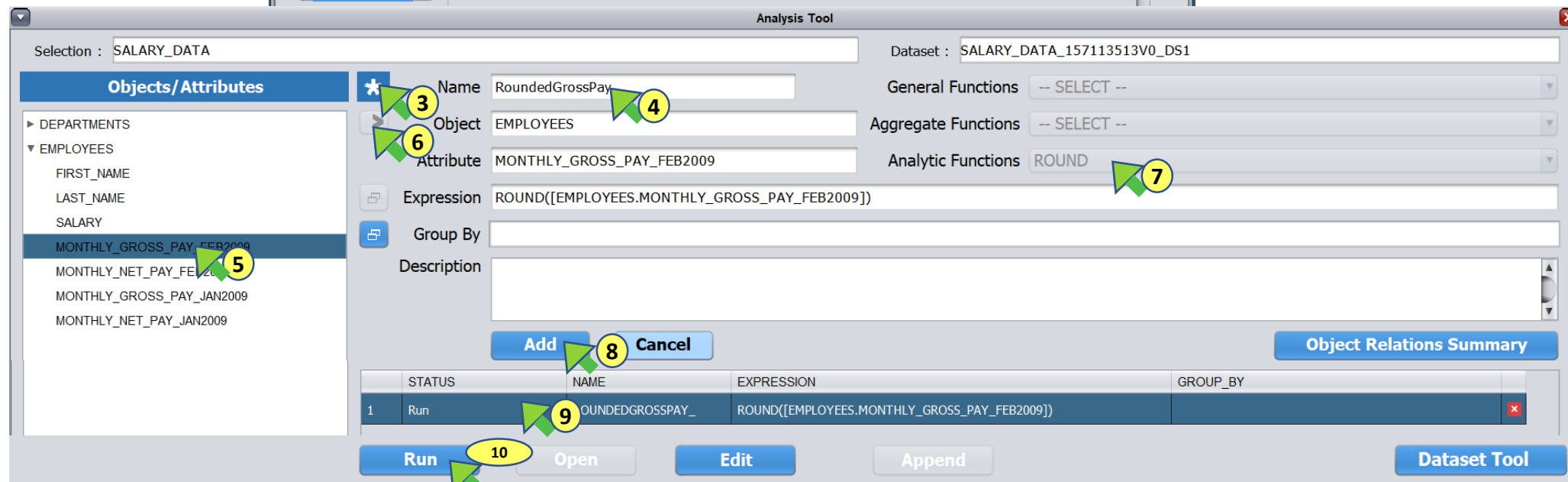
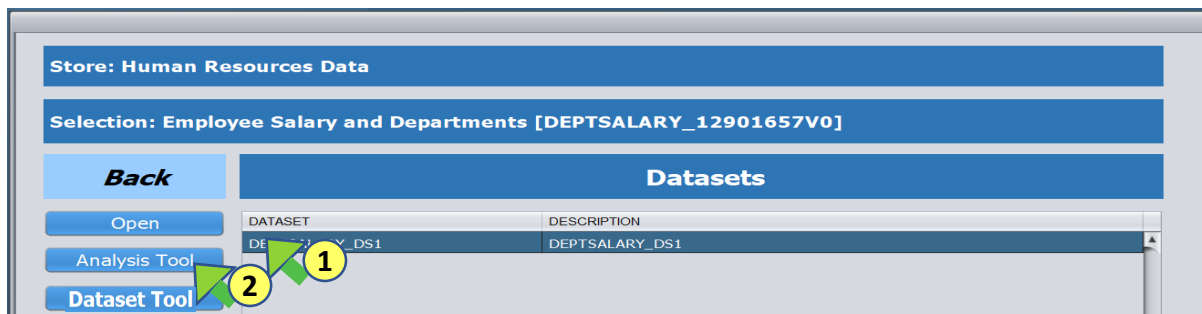
**×** Delete selected analysis (in 'Run' or 'Saved' state).

**Object Relations Summary** View data-objects and their relations count.

**Dataset Tool** Open Dataset Tool.

## Work with Datasets:

- Analysis Tool
- Analytic Functions
- ROUND



Create new analysis.

**Name** Enter name.

**Obj Functions** Select a function.

Add object/attribute to analysis.

**Object** Selected object.

**Attribute** Selected attribute.

**Description** Add/Edit analysis description.

**Add** Add analysis to the list.

**Cancel** Cancel entry/edits.

**Run** Run selected analysis.

**Open** Open selected analysis result.

**Edit** Edit selected analysis.

**Append** Append statistic to the dataset.

Delete selected analysis (in 'Run' or 'Saved' state).

**Object Relations Summary** View data-objects and their relations count.

**Dataset Tool** Open Dataset Tool.

## Work with Datasets:

### - Analysis Tool

### - String Functions

### - CONCAT\_TWO\_WS

The screenshot shows the Analysis Tool interface with the following components and callouts:

- 1**: Dataset selection button.
- 2**: Analysis Tool button.
- 3**: Create new analysis button.
- 4**: Name input field.
- 5**: Object selection button.
- 6**: Object dropdown menu.
- 7**: Attribute selection button.
- 8**: String function selection button.
- 9**: String1 input field.
- 10**: Attribute selection dropdown.
- 11**: Select button in the dropdown.
- 12**: Final Usage text area.
- 13**: Done button in the dialog.
- 14**: Add button in the analysis list.
- 15**: Cancel button in the analysis list.
- 16**: Run button in the analysis list.

Create new analysis.

**Name** Enter name.

**Obj Functions** Select a function.

Add object/attribute to analysis.

**Object** Selected object.

**Attribute** Selected attribute.

**Group By** Edit Group By.

**Description** Add/Edit analysis description.

Add analysis to the list.

Cancel entry/edits.

Run selected analysis.

Open selected analysis result.

Edit selected analysis.

Append statistic to the dataset.

Delete selected analysis (in 'Run' or 'Saved' state).

Object Relations Summary View data-objects and their relations count.

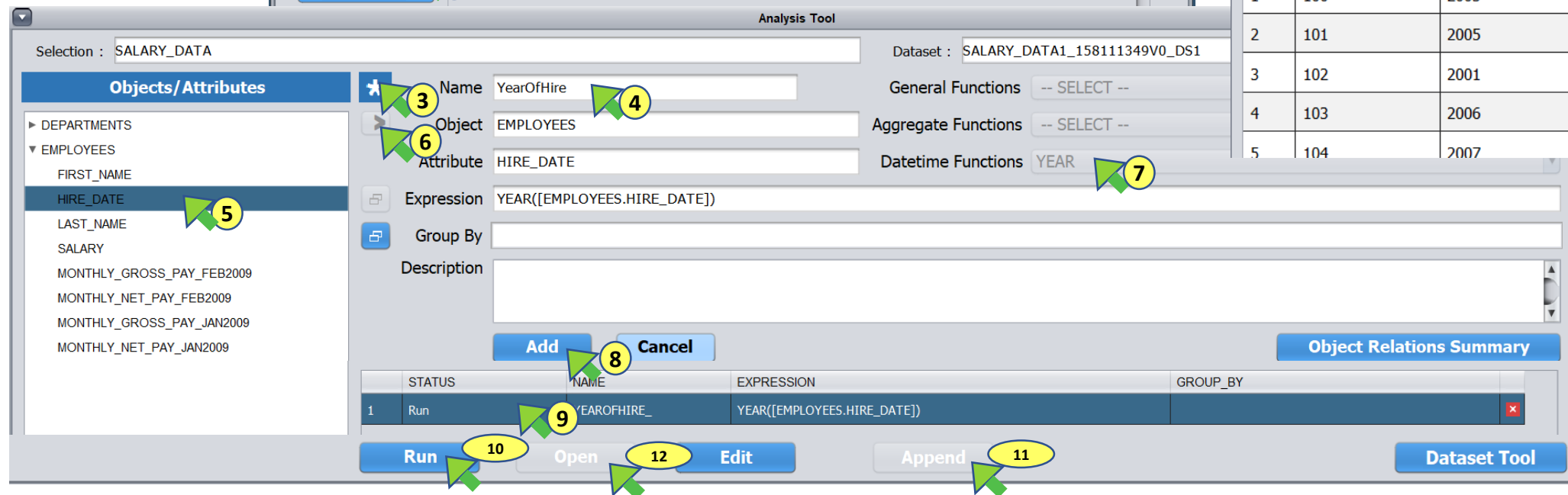
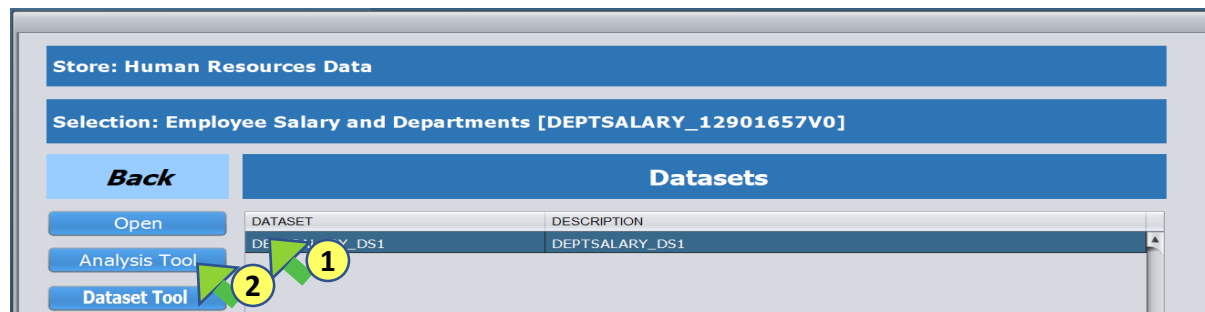
Dataset Tool Open Dataset Tool.

## Work with Datasets:

### - Analysis Tool

### - Date Functions

### - YEAR



Create new analysis.

**Name** Enter name.

**Obj Functions** Select a function.

Add object/attribute to analysis.

**Object** Selected object.

**Attribute** Selected attribute.

**Description** Add/Edit analysis description.

**Add** Add analysis to the list.

**Cancel** Cancel entry/edits.

**Run** Run selected analysis.

**Open** Open selected analysis result.

**Edit** Edit selected analysis.

**Append** Append statistic to the dataset.

Delete selected analysis (in 'Run' or 'Saved' state).

**Object Relations Summary** View data-objects and their relations count.

**Dataset Tool** Open Dataset Tool.

# Work with Datasets:

## - Dataset Tool

Selection : SALARY\_DATA

**Objects/Attributes**

- Statistics
  - MAXSALARYBYDEPT\_ (6)
  - YEAROFHIRE\_ (6)
- DEPARTMENTS
- EMPLOYEES

Name: SALARYRPT (3)

Description: SALARY REPORT (4)

Filter: (5)

Sort: [EMPLOYEES.SALARY]

Select Top: (8) [EMPLOYEES.FIRST\_NAME], [EMPLOYEES.LAST\_NAME], [EMPLOYEES.SALARY], [Statistics.MAXSALARYBYDEPT\_]

Selected attributes' information:

```
DataType=decimal(8,2) :Expression=MAX ([EMPLOYEES.SALARY]) :GroupBy=DEPARTMENT_NAME
```

STATUS	NAME	FILTER	SORT	LIMIT	SELECT
1	Preview	SALARYRPT (10)	B1_SALARY		B1_FIRST_NAME, B1_LAST_NAME, B1_SALARY...

Buttons: Add (9), Preview (11), Open (13), Edit, Save (12), Export to CSV

Secondary Window: Selection : SALARY\_DATA, Export to CSV

	EMPLOYEES FIRST_NAME	EMPLOYEES LAST_NAME	EMPLOYEES SALARY	STATISTICS MAXSALARYBYDEPT_
1	TJ	Olson	2100.00	8200.00
2	Hazel	Phitanker	2200.00	8200.00
3	Steven	Markle	2200.00	8200.00

Showing 1000 rows per page | 1/1

! If no attribute/object selected then all attributes, including object-ids, will be added to selection.

- Create new subset.
- Name** Enter name.
- Description** Add/Edit subset description.
- Filter** Add object/attribute to WHERE clause.
- Sort** Add object/attribute to ORDER BY clause.
- Select** Add object/attribute to SELECT clause.
- Top** Enter number of rows to limit the subset.
- Add** Add subset definition to the list.
- Cancel** Cancel entry/edits.
- Delete selected subset.
- Preview** Preview selected subset.
- Edit** Edit selected subset (in Preview state).
- Save** Save selected subset as database table.
- Open** Open selected (Saved) subset.
- Export to CSV** Export selected (Saved) subset to CSV files.
- View first page (first 1000 rows).
- View previous page.
- View next page.
- View last page.
- Close window.

## Store:

### - Re-Run Selection.

(i.e. re-create dataset/s, run and append statistics; and re-create subset/s)

Change Periodic Table Selection

Currently Selected Tables: [EMPLOYEE\_].[opemployee\_monpay\_jan09] (JAN09)

Change To: [hr].[opemployee\_monpay\_apr09] (APR09)

Add \*

	DATA OBJECT	CURRENT TABLE	CURRENT TABLE DESC	<->	NEW TABLE	NEW TABLE DESC
1	EMPLOYEE_	opemployee_monpay_jan09	JAN09	<->	opemployee_monpay_apr09	APR09

ReRun

Store: Human Resources Data

Back

New

Open

Run

Re-Run

Work with Datasets

Delete

Input

Change selection description for new version

Employee Salary and Departments VERS...

Continue with ReRun

Cancel ReRun

Store: Human Resources Data

Back

Selections

New

Open

Run

Re-Run

Work with Datasets

Delete

SELECTION	DESCRIPTION	RUN_DATE
DEPTSALARY_6545548V0	Employee Salary and Departments	Tue Mar 05 16:55:52 EST 2024
DEPTSALARY_6545548V1	Employee Salary and Departments VER...	Tue Mar 05 17:20:00 EST 2024

Selection(DEPTSALARY\_6545548V0) re-run completed successfully

\* (Optionally) change selected periodic table/s to different periodic table/s of the same type. The Re-Run will use the newly selected table/s, in place of originally selected tables, in creating dataset/s and subsequently in analysis as well as subset creation.

Note: Prompt to change periodic table/s selection will only appear if periodic attribute/s were in original selection.

New	Create new Selection.
Open	Open selected Selection.
Run	Run selected Selection.
Re-Run	Re-Run selected Selection including analysis and subsets.
Work with Datasets	Open the list of dataset/s created by the Selection Run (or Re-Run)
Delete	Delete selected Selection and associated analysis and subsets.



**Dataset Metadata:** Each dataset is created with two tables, a data table and a metadata table. Dataset table name/s are <selection>\_DS<num>. Metadata table name/s are <selection>\_DS<num>\_DM. Following table contains metadata table's column names and descriptions. Metadata for appended statistic is identified by ATTR\_DOBJ column's value of 'STATDOBJ'.

Name	Description	STATISTIC Description*
IQ_ATTR_NAME	Column name in dataset's table	Column name in dataset's table
RESULTSET	Dataset's table name	Dataset's table name
ATTR_DB	Database/schema of the selection attribute(column)	'STAT_DB'
ATTR_TABLE	Table of the selection attribute(column)	'STAT_TBL'
ATTR_DBTYPE	Data type of the selection attribute(column)	Data type of the statistic attribute(column)
ATTR_DOBJ	Data-object of the selection attribute	'STATDOBJ'
ATTR_DOBJ_DESC	Data-object description of the selection attribute	'Statistics'
ATTR_CATGRY	Selection attribute category (i.e. Static, Aperiodic or Periodic). Null if ID attribute.	'S'
ATTR_NAME	Selection attribute name (i.e. column name). Look-up value if LKPRNG_ATTR not NULL	Data-object/s of selection attribute/s included in the statistic's formula
ATTR_DESC	Selection attribute description. Look-Up or Range attribute description if LKPRNG_ATTR not NULL	Column name in dataset(table)
C_CATGRY_OCAP_OPTION	For selection attribute category = 'C' or 'RC' (i.e. DOBJ or R-DOBJ Aperiodic attribute)	NULL
SEQ_NUM	For selection attribute category = 'C' or 'RC' instance update sequence number.	NULL
P_CATGRY_TABLE_DESC	For selection attribute category = 'P' or 'RP' (i.e. DOBJ or R-DOBJ Periodic attribute)	NULL
LKPRNG_ATTR	Look-Up or Range attribute (column)	NULL
LKPRNG_DB	Database of the Look-Up or Range attribute	NULL
LKPRNG_TABLE	Table of the Look-Up or Range attribute	NULL
LKPRNG_ATTR_DBTYPE	Data type of the Look-Up or Range attribute	NULL
ATTR_AUX_META	I = ID Attribute, L = Look-Up attribute, R = Range attribute	ID attribute/s of the data-object/s included in the statistic's formula
ATTR_AUX_INFO	Additional selection attribute information added by user. Otherwise attribute(column) data type.	Statistic type, formula, Group By (if any) of the statistic
FROM_DOBJ	Dobj1 of the selection attribute, if ATTR_CATGRY = 'R','RC','RP' otherwise data-object of the attribute	NULL
TO_DOBJ	Dobj2 of the selection attribute, if ATTR_CATGRY = 'R','RC','RP' otherwise data-object of the attribute	NULL
VIEW_ID	NULL	NULL

\* If ATTR\_DOBJ = 'STATDOBJ' (i.e. Metadata for appended statistics)

**Statistic Metadata:** Statistic table does not get created with metadata table. When a statistic table is exported, a metadata file is created together with statistic data file in CSV format. Statistic table (and exported data file) name is <selection>\_DS<num>\_AN<num>. Statistic metadata file name is <selection>\_DS<num>\_AN<num>\_ANMETA.csv. Statistic metadata file contains one row for the statistic metadata and rows for dataset metadata for the other attributes of statistic table. Following table contains statistic metadata file columns and descriptions.

Name	Description
QINST_ID	Selection name
RESULTSET	Dataset(table) name
ANID	Statistic table name
STATUS	Statistic status in Analysis Tool (i.e. 'Saved' or 'Appended')
STATISTIC	Statistical function name
NAME	Statistic attribute name
DESC	NULL
DOBJATTR	Selected attribute (or NULL if General Function is selected)
GROUPBY	Group By attribute/s.
EXPRESSION	Syntactical expression
ANSQL	SQL statement to create statistic table
DSSQL	SQL statement to create dataset(table) input to compute the statistic. (experimental metadata, not generated for all statistics)
APNDSQL	SQL statement to append statistic (column) to dataset.
ANLEVEL	(For internal use)
USERDB	NULL
SSID	NULL
SSRMID	NULL
DOBJS	Data objects of the attributes included in the statistic
DOBJIDS	Data objects IDs of the attributes included in the statistic
ATTRLIST	Attributes included in the statistic
USEREXPR	Syntactical user expression (i.e. attributes expressed as '[<data-object>.<attribute>]')
COMPEXPR	Syntactical user expression in case of complex syntax. Parameters expressed as '{<name> = <value>}'
VIEW_ID	NULL

**Subset Metadata:** Each saved subset is created with a data and a metadata tables. Subset table name/s are <selection>\_DS<num>\_RP<num>. Metadata table name/s are <selection>\_DS<num>\_RP<num>\_DM. Metadata for appended statistic is identified by ATTR\_DOBJ column's value of 'STATDOBJ'. Following table contains metadata table column names and descriptions.

Name	Description	STATISTIC Description*
IQ_ATTR_NAME	Column name in dataset's table	Column name in dataset's table
RESULTSET	Dataset's table name	Dataset's table name
ATTR_DB	Database/schema of the selection attribute(column)	'STAT_DB'
ATTR_TABLE	Table of the selection attribute(column)	'STAT_TBL'
ATTR_DBTYPE	Data type of the selection attribute(column)	Data type of the statistic attribute(column)
ATTR_DOBJ	Data-object of the selection attribute	'STATDOBJ'
ATTR_DOBJ_DESC	Data-object description of the selection attribute	'Statistics'
ATTR_CATGRY	Selection attribute category (i.e. Static, Aperiodic or Periodic). Null if ID attribute.	'S'
ATTR_NAME	Selection attribute name (i.e. column name). Look-up value if LKPRNG_ATTR not NULL	Data-object/s of selection attribute/s included in the statistic's formula
ATTR_DESC	Selection attribute description. Look-Up or Range attribute description if LKPRNG_ATTR not NULL	Column name in dataset(table)
C_CATGRY_OCAP_OPTION	For selection attribute category = 'C' or 'RC' (i.e. DOBJ or R-DOBJ Aperiodic attribute)	NULL
SEQ_NUM	For selection attribute category = 'C' or 'RC' instance update sequence number.	NULL
P_CATGRY_TABLE_DESC	For selection attribute category = 'P' or 'RP' (i.e. DOBJ or R-DOBJ Periodic attribute)	NULL
LKPRNG_ATTR	Look-Up or Range attribute (column)	NULL
LKPRNG_DB	Database of the Look-Up or Range attribute	NULL
LKPRNG_TABLE	Table of the Look-Up or Range attribute	NULL
LKPRNG_ATTR_DBTYPE	Data type of the Look-Up or Range attribute	NULL
ATTR_AUX_META	I = ID Attribute, L = Look-Up attribute, R = Range attribute	ID attribute/s of the data-object/s included in the statistic's formula
ATTR_AUX_INFO	Additional selection attribute information added by user. Otherwise attribute(column) data type.	Statistic type, formula, Group By (if any) of the statistic
FROM_DOBJ	Dobj1 of the selection attribute, if ATTR_CATGRY = 'R','RC','RP' otherwise data-object of the attribute	NULL
TO_DOBJ	Dobj2 of the selection attribute, if ATTR_CATGRY = 'R','RC','RP' otherwise data-object of the attribute	NULL
VIEW_ID	NULL	NULL
RP_ID	Subset's table name	Subset's table name
RP_NAME	Subset name	Subset name
RP_DESC	Subset description	Subset description

\* If ATTR\_DOBJ = 'STATDOBJ' (i.e. Metadata for appended statistics)

Appendix:

## Symmetric Relational Data Object Model (RDOM)

RDOM primarily consists of symmetric data objects (DOBJs) and symmetric bilateral relationship data objects (R-DOBJs). R-DOBJ represents many-to-many relationship between two DOBJs (or one-to-many relationship between first DOBJ to R-DOBJ and one-to-many relationship between second DOBJ to the R-DOBJ). Following is an example (Fig. 1) RDOM consisting four DOBJs (Dobj1, Dobj2, Dobj3 and Dobj4) and six R-DOBJs (RDobj1-2, RDobj1-3, RDobj2-3, RDobj2-4, RDobj4-1 and RDobj3-4).

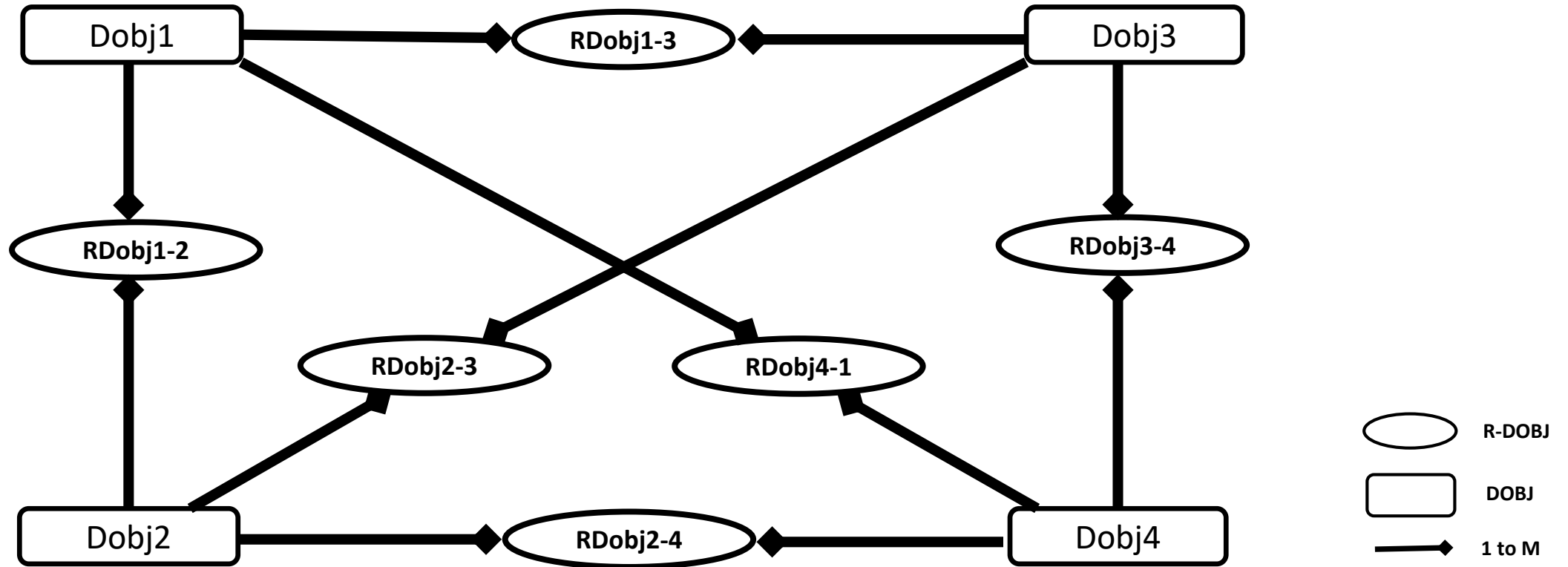


Fig. 1

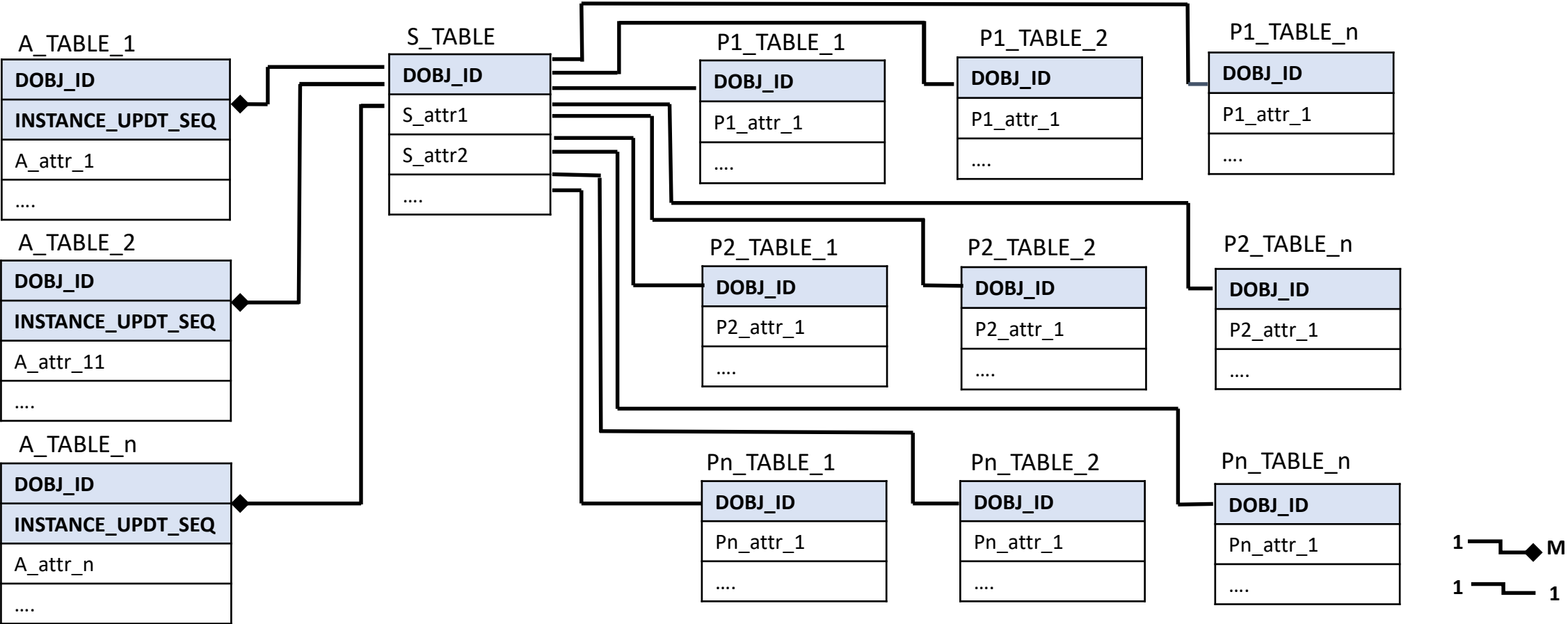
**DOBJ**: consists of three types of relational database tables; static attributes table, aperiodic attributes table/s and periodic attributes table/s.

- **Static attributes table (S-table)**: consists of a column for DOBJ's identification (ID) attribute and columns for static attributes (whose value remains static). For example, Name and Date of Birth attributes of Customer DOBJ.  
*[DOBJ must have S-table with ID column. Only one S-table is allowed per DOBJ]*
- **Aperiodic attributes table (A-table)**: consists of a column for DOBJ's ID attribute, a column for instance update sequence number (INSTANCE\_UPDT\_SEQ) and columns for aperiodic attributes (whose value changes aperiodically). INSTANCE\_UPDT\_SEQ column is used for storing sequentially increasing number for each update of aperiodic attribute/s value/s. For example, Mailing Address attribute of Customer. First instance of Customer's mailing address is to be stored with INSTANCE\_UPDT\_SEQ value of 1, next update of mailing address would be stored with INSTANCE\_UPDT\_SEQ value of 2 and so forth.  
*[DOBJ may have one or more A-tables]*
- **Periodic attributes table (P-table)**: consists of a column for DOBJ's ID attribute and columns for periodic attributes (whose value is recorded periodically – at every specific period). For example, monthly (or bi-weekly) employee payment attributes.  
*[DOBJ may have one or more sets of periodic tables]*

Similarly R-DOBJ consists of three types (Static, Aperiodic and Periodic) of relational database tables. Each R-DOBJ table consists of two ID columns (for the two related DOBJ's IDs) and other columns.

**DOBJ Data Model template (Fig. 2):**

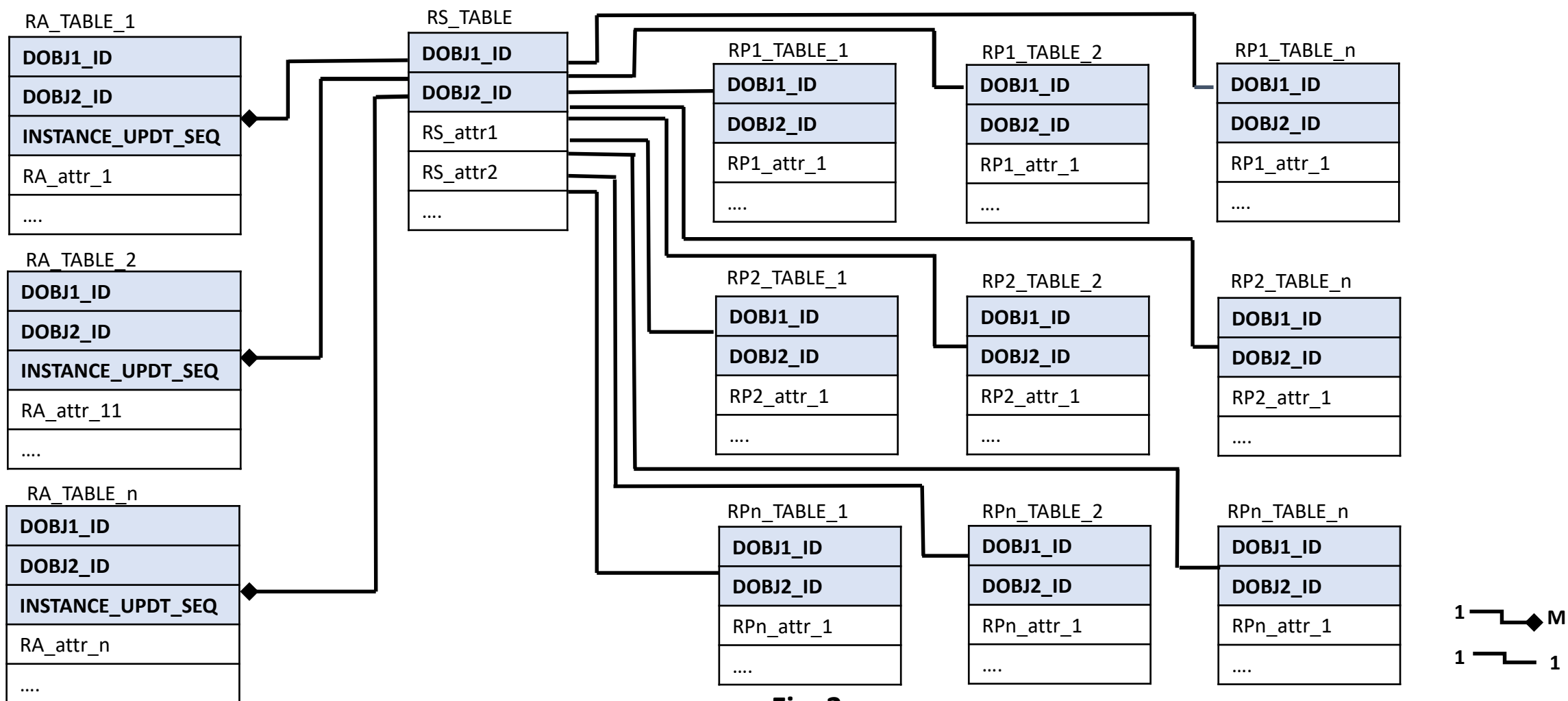
- **S TABLE (Static attributes table):** consists of ID column (DOBJ\_ID) as primary-key and columns for static attributes (S\_attr1, S\_attr2, ...).
- **A TABLEs (Aperiodic attributes tables):** consist of ID column (DOBJ\_ID) and sequence number column (INSTANCE\_UPDT\_SEQ) combined as primary-key, and columns for aperiodic attributes (A\_attr\_1, ....). Relationship from S\_TABLE to A\_TABLEs are of type one-to-many.
- **Pn TABLE ns (Periodic attributes tables):** consist of ID column (DOBJ\_ID) as primary-key and columns for periodic attributes (P1\_attr\_1, P1\_attr\_2,....). Relationships from S\_TABLE to P\_TABLEs are of type one-to-one.



**Fig. 2**

## R-DOBJ Data Model template (Fig. 3):

- **RS TABLE (Static attributes table):** consist of ID columns (DOBJ1\_ID + DOBJ2\_ID) as primary-key and columns for static attributes (RS\_attr1, RS\_attr2, ...).
- **RA TABLEs (Aperiodic attributes tables):** consist of ID columns (DOBJ1\_ID + DOBJ2\_ID) and sequence number column (INSTANCE\_UPDT\_SEQ) combined as primary-key, and columns for aperiodic attributes (RA\_attr\_1, ...). Relationship from RS\_TABLE to RA\_TABLEs are of type one-to-many.
- **RPn TABLE ns (Periodic attributes tables):** consist of ID columns (DOBJ1\_ID + DOBJ2\_ID) as primary-key and columns for periodic attributes (RP1\_attr\_1, RP1\_attr\_2,....). Relationships from RS\_TABLE to RP\_TABLEs are of type one-to-one.

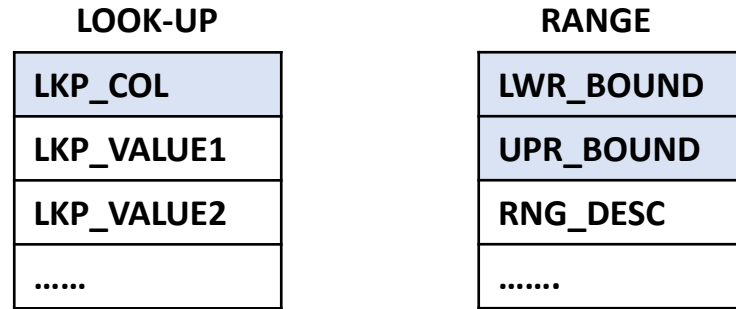


**Fig. 3**

In addition to DOBJ and R-DOBJ, RDOM may include Look-Up (LOOK-UP) and Range (RANGE) associations.

**LOOK-UP:** consists of a relational database table. The table consists of a look-up column (as primary-key) and one or more columns for look-up values (Fig.4).

**RANGE:** consists of a relational database table. The table consists of a column for lower bound of range, a column for upper bound of range and one or more columns for the range description/information (Fig.4). Lower bound column concatenated with upper bound column, is primary-key.



**Fig. 4**

- A Look-Up (and Range) may be associated with one or more non-ID columns of one or more tables of DOBJs and R-DOBJs. Also, a non-ID column may be associated with one or more Look-up (and Range).

## **RDOM Constraints:**

- Primary-key column value in all tables (of all four types of data objects) must not be NULL.
- INSTANCE\_UPDT\_SEQ column value, in Aperiodic tables, must be a positive integer starting with 1 for first (or original) row for an instance.
- In a DOBJ: DOBJ\_ID column of Aperiodic and Periodic tables must not contain value that does not exist in DOBJ\_ID column of Static table.
- In a R-DOBJ: DOBJ1\_ID and DOBJ2\_ID columns of Static table must not contain value that does not exist in DOBJ\_ID columns of the two respective DOBJs.
- In a R-DOBJ: DOBJ1\_ID and DOBJ2\_ID columns of Aperiodic and Periodic tables must not contain value that does not exist in DOBJ1\_ID and DOBJ2\_ID columns of Static table.



## RDOM as Super Model

In practice, it may be required that existing relational database structure is kept as it is . In such cases, RDOM can be implemented as Super Model, on existing relational model, in the form of combination of existing tables and views on existing tables.

Following are the four general rules for implementing RDOM as Super Model.

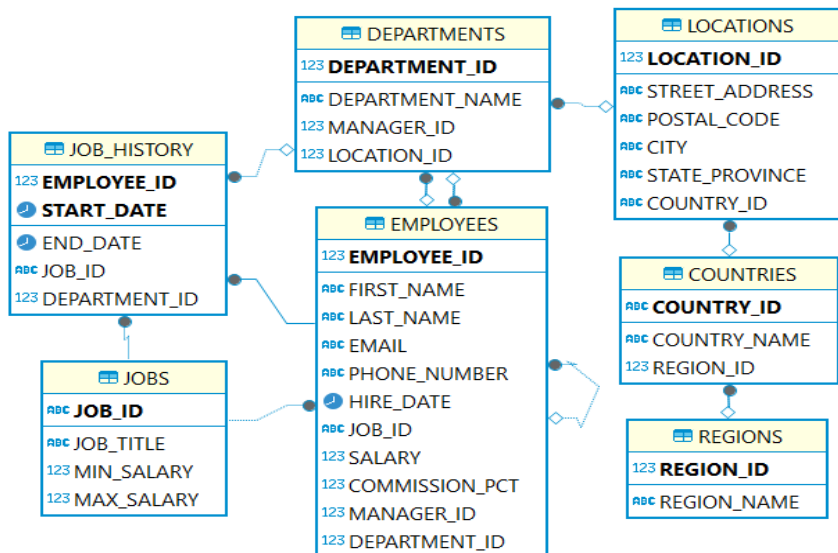
Rule-1: Minimize number of DOBJs: By consolidating relevant tables together into DOBJs; and separating DOBJ tables from LOOK-UP tables.

Rule-2: If a table contains object hierarchy then flatten the object hierarchy into a table (or a view) by means of additional columns.

Rule-3: If an object hierarchy exists in the form of multiple tables then combine the tables into one table (or a view).

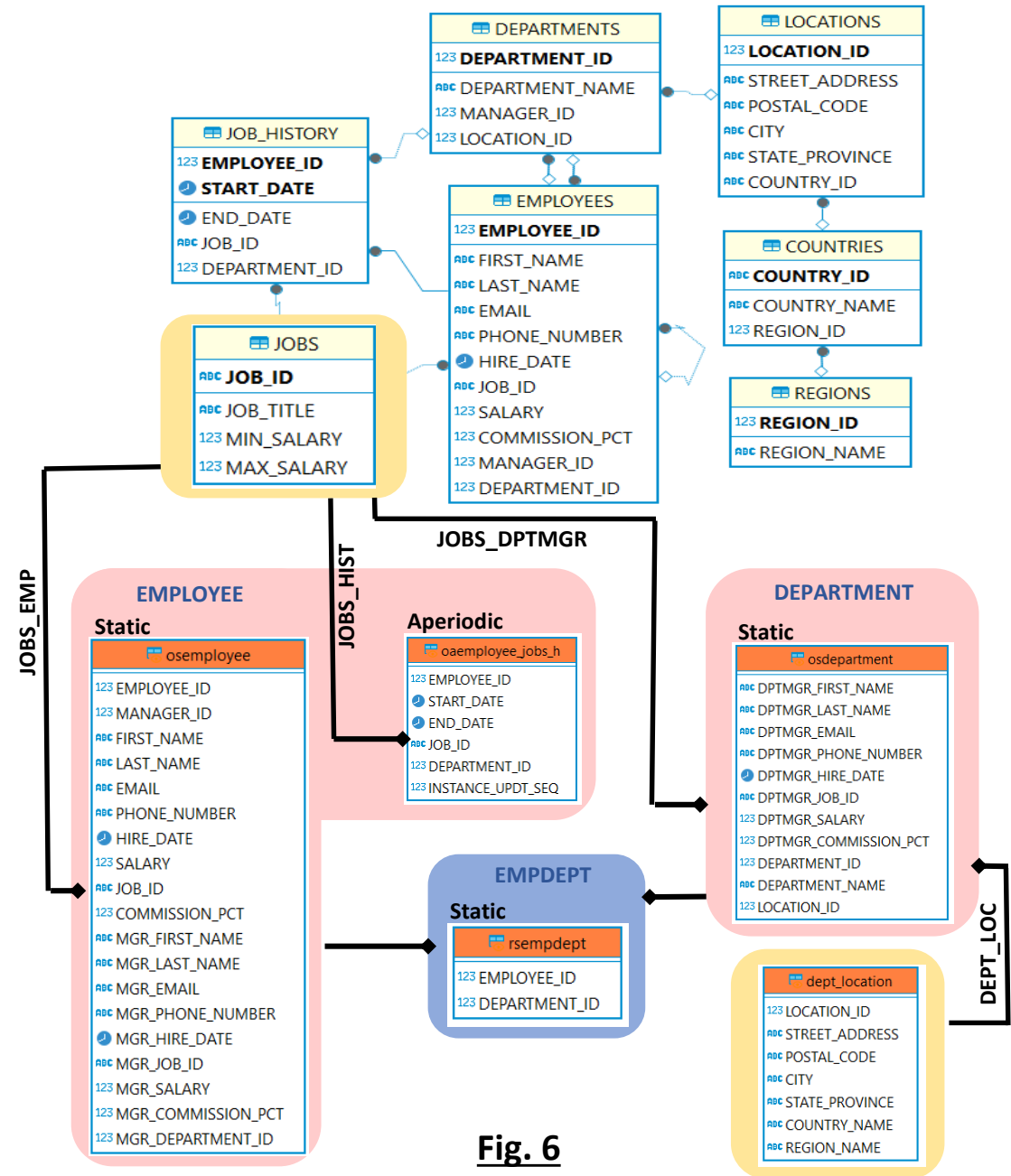
Rule-4: Maximize number of RDOBJs, possibly relating all DOBJs to each other.

**Example:** In this example (Fig. 5), HR Database contains seven tables (DEPARTMENTS, EMPLOYEES, JOB\_HISTORY, JOBS, LOCATIONS, COUNTRIES, REGIONS) for employee-manager hierarchy, department, location, jobs and employees' job history.



**Fig. 5**

## Example (continued)



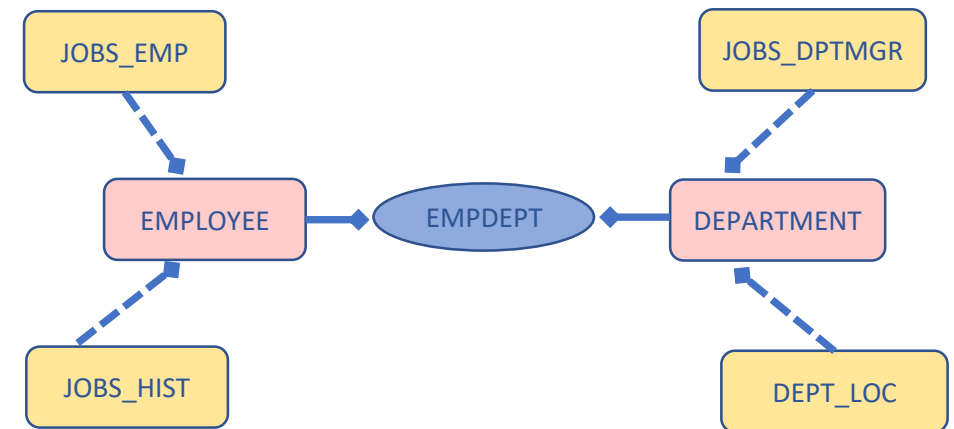
To implement RDOM as Super Model on existing relational model, five views (osemployee, oaemployee\_jobs\_h, rsempdept, osdepartment, dept\_location) are created with consideration of the four general rules.

**Rule-1: Minimize number of DOBJs: By consolidating relevant tables together into DOBJs; and distinguish DOBJ tables from LOOK-UP object tables:** OSEMPLOYEE and OAEMPLOYEE\_JOBS\_H views on EMPLOYEES and JOBS\_HISTORY tables are incorporated into EMPLOYEE data object, OSDEPARTMENT view (which includes employee columns for department manager) on DEPARTMENTS table is incorporated as DEPARTMENT data object and rest of the four tables (JOBS, LOCATIONS, COUNTRIES, REGIONS) are identified as look-up tables.

**Rule-2: If a table contains object hierarchy then flatten the object hierarchy into a table (or a view) by means of additional columns:** OSEMPLOYEE view is created with additional columns for manager, by self joining EMPLOYEES table, to flatten Employee-Manager hierarchy.

**Rule-3: If an object hierarchy exists in the form of multiple tables then combine the tables into one table (or a view):** DEPT\_LOCATION view is created representing location-country-region hierarchy, by joining LOCATIONS, COUNTRIES and REGIONS tables.

**Rule-4: Maximize number of RDOBJs, possibly relating all DOBJs to each other:** RSEMPDEPT view is created representing RDOBJ for EMPLOYEE and DEPARTMENT data objects.



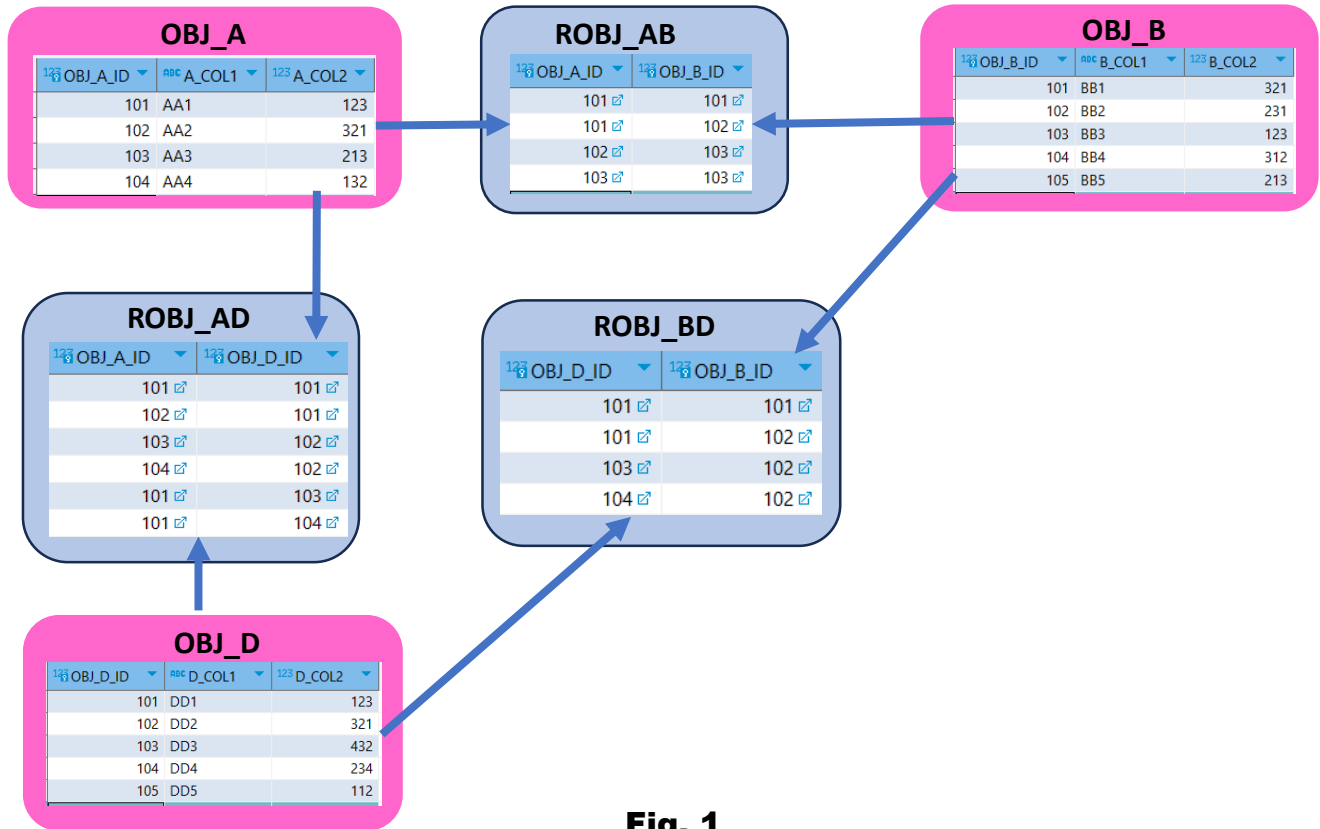
**Fig. 6(a) RDOM diagram**

# Appendix-2:

# Build RelTree:

The process of creating dataset/s of selected attributes is of two steps. First, corresponding SQL UNIONS of SQL JOINS of ID attribute columns of Static attribute tables of all combinations, from largest to smallest, of RDOBJs and DOBJs is created; than with that result set (of IDs) selected DOBJs are LEFT JOINed. The result set created from first step is called RelTree (Relation Tree). In a scenario where the database is updated periodically, for all combinations of related objects RelTree tables can be pre-built after every database update. This way for every instance of dataset creation, only second step will need to be performed. In short, pre-built RelTree tables enhances efficiency of dataset creation.

**Example:** Two attributes of OBJ\_A, one attribute of OBJ\_B and two attributes of OBJ\_D are selected. When the selection is ran the application used pre-built RelTree table (Fig.2 encircled in red) and LEFT JOINed it with each of the three data objects with their selected attributes.



**Fig. 1**

	OBJ_A OBJ_A_ID	OBJ_B OBJ_B_ID	OBJ_D OBJ_D_ID	OBJ_A A_COL1	OBJ_A A_COL2	OBJ_B B_COL1	OBJ_D D_COL1	OBJ_D D_COL2
1	101	101	101	AA1	123	BB1	DD1	123
2	101	102	101	AA1	123	BB2	DD1	123
3	101	102	103	AA1	123	BB2	DD3	432
4	101	102	104	AA1	123	BB2	DD4	234
5	102	103		AA2	321	BB3		
6	103	103		AA3	213	BB3		
7	102		101	AA2	321		DD1	123
8	103		102	AA3	213		DD2	321
9	104		102	AA4	132		DD2	321
10		104				BB4		
11		105				BB5		
12			105				DD5	112

**Fig. 2**

In practice, data object tables may get loaded from different source systems and at different frequency. In such scenario, running PathRunner will derive and update relationship data objects with any additional instances found, by iteratively and bi-directionally traversing alternate paths between each pair of related data objects.

**Example:** Original state of the database containing tables for objects Obj\_A, Obj\_B, Obj\_C, Obj\_D and relationship objects relating the four objects (Fig.1). In this state the table for relationship object Robj\_AC has two ID columns representing relationship between two objects Obj\_A and Obj\_C; the table has one row representing relation between instances of Obj\_A and Obj\_C (A\_ID = 102 <-> C\_ID = 101). Alternate paths between Obj\_A and Obj\_C are 1) Robj\_AB <-> Robj\_BC, 2) Robj\_AD <-> Robj\_DC, 3) Robj\_AB <-> Robj\_BC <-> Robj\_CD and 4) Robj\_AD <-> Robj\_BD <-> Robj\_BC. After PathRunner run is completed, state of the database is as shown in Fig.2. In this state table for Robj\_AC contains additional rows representing derived relations (A\_ID = 101 <-> C\_ID = 101, A\_ID = 101 <-> C\_ID = 102, A\_ID = 101 <-> C\_ID = 103, A\_ID = 101 <-> C\_ID = 104). Similarly tables for Robj\_AD and Robj\_CD has additional rows (encircled in red) representing derived relations between respective pairs of objects.

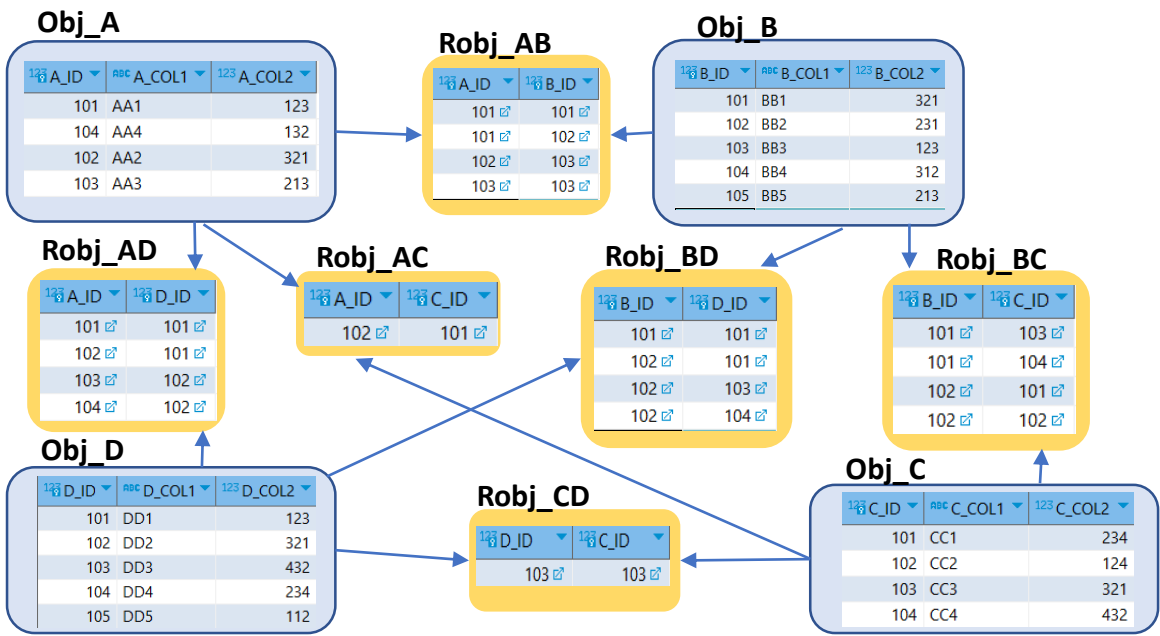


Fig. 1

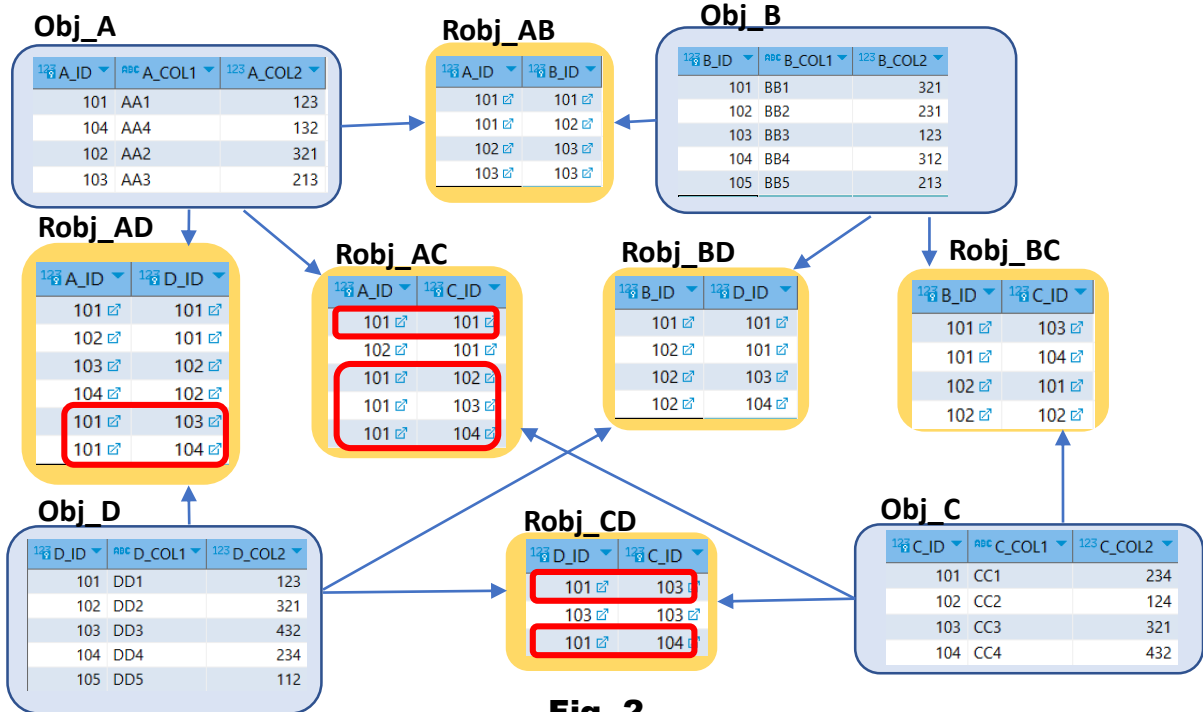


Fig. 2

Dataset is created by performing corresponding SQL UNION of SQL JOINS of all combinations, from largest to smallest combination, of ID attribute columns of Static attribute tables of relationship data objects and selected data objects; and then performing LEFT JOIN/s of selected data object attribute columns. In the following example, five attributes (A\_COL1, A\_COL2, B\_COL1, D\_COL1 and D\_COL2) are selected from three data objects (OBJ\_A, OBJ\_B and OBJ\_D). The dataset (Fig.2) is created by performing corresponding SQL UNION of SQL JOINS of ID columns from the three relationship data objects (ROBJ\_AB, ROBJ\_AD and ROBJ\_BD), each combination of two R-DOBJs and three data objects (OBJ\_A, OBJ\_B and OBJ\_D); and then performing LEFT JOIN of selected attribute columns from the three data object.

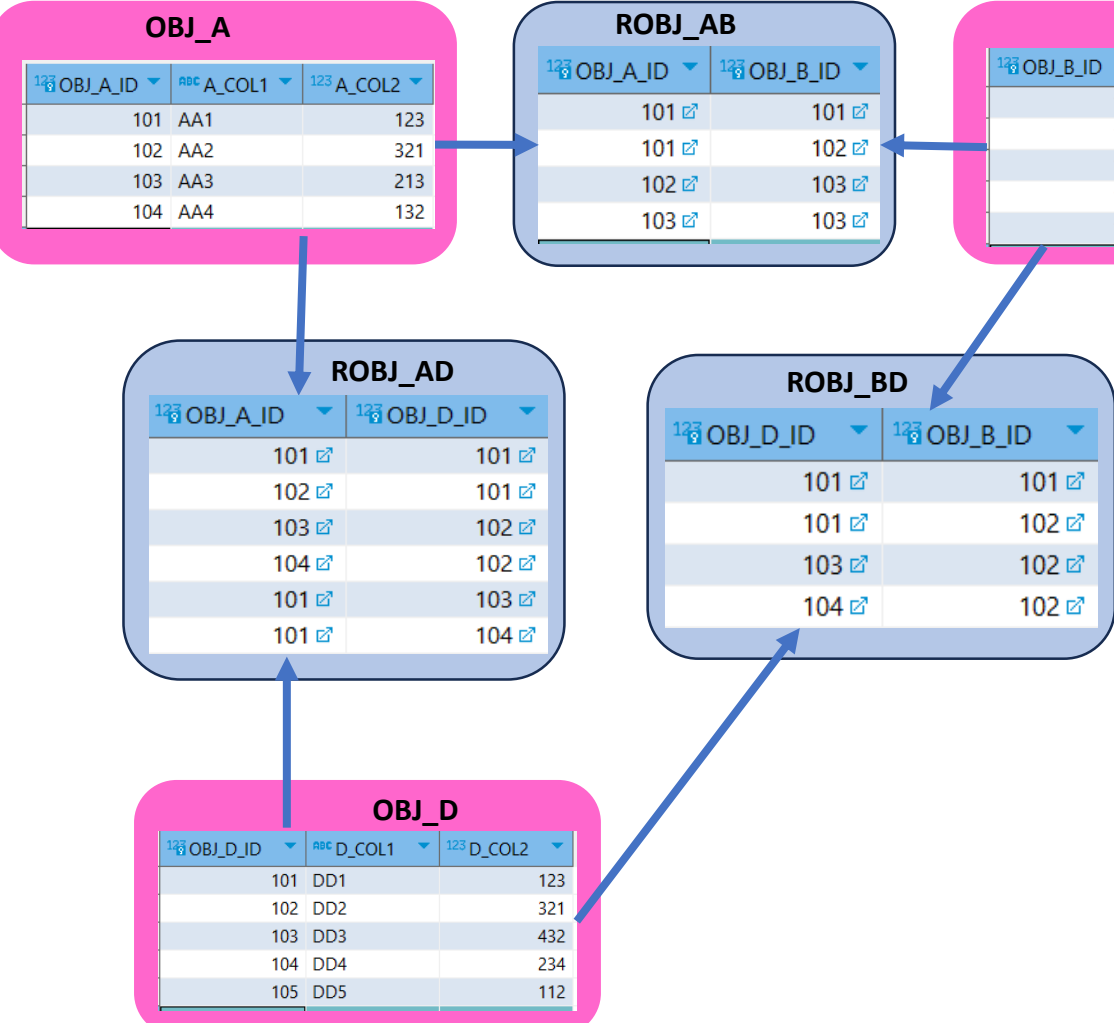


Fig. 1

	OBJ_A OBJ_A_ID	OBJ_B OBJ_B_ID	OBJ_D OBJ_D_ID	OBJ_A A_COL1	OBJ_A A_COL2	OBJ_B B_COL1	OBJ_D D_COL1	OBJ_D D_COL2
1	101	101	101	AA1	123	BB1	DD1	123
2	101	102	101	AA1	123	BB2	DD1	123
3	101	102	103	AA1	123	BB2	DD3	432
4	101	102	104	AA1	123	BB2	DD4	234
5	102	103		AA2	321	BB3		
6	103	103		AA3	213	BB3		
7	102		101	AA2	321		DD1	123
8	103		102	AA3	213		DD2	321
9	104		102	AA4	132		DD2	321
10		104				BB4		
11		105				BB5		
12			105				DD5	112

Fig. 2

# Object Relations Summary:

## Appendix-5:

The summary represents counts of object instances and related object instances in an interactive form.

Example: The dataset table (Fig.1) contains columns for three object IDs (i.e. A\_ID, B\_ID and D\_ID) and attributes of the objects.

Object relations summary (Fig. 3) shows, in the area (i.e. box) at the top, total number of objects' instances in the dataset. Boxes starting from second row and first column, shows counts for exclusively related instances of all combinations of objects in descending order of the size of object combination starting from first column on the left. In this example, the box in the first column (and second row) shows 3 instances of OBJ\_D, 1 instance of OBJ\_A and 2 instances of OBJ\_B are related to each other; similarly second column boxes shows counts for exclusively related instances among each combination of two objects; and the third column shows non-related instance counts for each object.

Selecting (clicking) the box (at 2<sup>nd</sup> column and 2<sup>nd</sup> row) shows dataset rows and columns of the exclusively related instances (Fig. 3) in the form of a table.

	OBJ_A A_ID	OBJ_B B_ID	OBJ_D D_ID	OBJ_A A_COL1	OBJ_B B_COL1	OBJ_B B_COL2	OBJ_D D_COL1
1	101	101	101	AA1	BB1	321	DD1
2	101	102	101	AA1	BB2	231	DD1
3	101	102	103	AA1	BB2	231	DD3
4	101	102	104	AA1	BB2	231	DD4
5	102	103		AA2	BB3	123	
6	103	103		AA3	BB3	123	
7	102		101	AA2			DD1
8	103		102	AA3			DD2
9	104		102	AA4			DD2
10		104			BB4	312	
11		105			BB5	213	
12			105				DD5

Fig. 7

Object Group Size = 3		Object Group Size = 2		Object Group Size = 1	
<b>Objects Count(total)</b>					
<b>Object</b>	<b>Count</b>				
OBJ_D_	5				
OBJ_A_	4				
OBJ_B_	5				
<b>Related Objects Count</b>					
<b>Object</b>	<b>Count</b>	<b>Object</b>	<b>Count</b>	<b>Object</b>	<b>Count</b>
OBJ_D_	3	OBJ_A_	2	OBJ_A_	0
OBJ_A_	1	OBJ_B_	1		
OBJ_B_	2				
		<b>Object</b>	<b>Count</b>	<b>Object</b>	<b>Count</b>
		OBJ_D_	2	OBJ_B_	2
		OBJ_A_	3		
		<b>Object</b>	<b>Count</b>	<b>Object</b>	<b>Count</b>
		OBJ_D_	0	OBJ_D_	1
		OBJ_B_	0		

Fig. 8

	OBJ_A A_ID	OBJ_B B_ID	OBJ_A A_COL1	OBJ_B B_COL1	OBJ_B B_COL2
1	102	103	AA2	BB3	123
2	103	103	AA3	BB3	123

Fig. 9

**Appendix-6:**

**Create and Append Statistic:**

Creating statistic is creating a table containing results of running a statistical function by means of SQL statement. The SQL statement is of two parts, inner SQL is to create a subset containing ID columns and selected attribute columns of the objects included in the defined statistic; outer SQL is to run the selected statistical function on the subset. Appending statistic to the dataset is to create a SQL LEFT JOIN of the dataset and table containing statistic, on ID and other attributes of selected objects. Aggregate statistic is appended to the entire dataset, aggregate statistic with GROUP BY is appended by LEFT JOINing with the dataset on the GROUP BY columns, scalar statistic is appended by LEFT JOINing with dataset on the ID columns of the selected object/s.

**Example:** A dataset table (Fig. 1) contains selection from three objects (OBJ\_A, OBJ\_B and OBJ\_D). The dataset contains three ID columns (OBJ\_A\_ID, OBJ\_B\_ID and OBJ\_D\_ID) and four attribute columns (A\_COL1, B\_COL1, B\_COL2 and D\_COL1). A statistic table for AVG(B\_COL2) with GROUP BY A\_COL1 is created (Fig.2) by generating and running a SQL statement that creates a subset containing four columns and six rows (Fig.1 encircled in orange) and then calculates AVG(B\_COL2) with GROUP BY A\_COL1. The statistic is then appended to the dataset (Fig. 3 encircled in green) by generating and running a SQL statement that performs LEFT JOIN of the dataset with statistic table on A\_COL1 (Fig.3 encircled in orange) with condition (OBJ\_A\_ID IS NOT NULL AND OBJ\_B\_ID IS NOT NULL). Metadata about the appended statistic is added to the metadata table for the dataset (see [Dataset Metadata](#) for details).

	OBJ_A OBJ_A_ID	OBJ_B OBJ_B_ID	OBJ_D OBJ_D_ID	OBJ_A A_COL1	OBJ_B B_COL1	OBJ_B B_COL2	OBJ_D D_COL1
1	101	101	101	AA1	BB1	321	DD1
2	101	102	101	AA1	BB2	231	DD1
3	101	102	103	AA1	BB2	231	DD3
4	101	102	104	AA1	BB2	231	DD4
5	102	103		AA2	BB3	123	
6	103	103		AA3	BB3	123	
7	102		101	AA2			DD1
8	103		102	AA3			DD2
9	104		102	AA4			DD2
10		104			BB4	312	
11		105			BB5	213	
12			105				DD5

**Fig. 1**

	STATISTICS AVG_B_COL2_	OBJ_A A_COL1
1	276.0000	AA1
2	123.0000	AA2
3	123.0000	AA3

**Fig. 2**

	OBJ_A OBJ_A_ID	OBJ_B OBJ_B_ID	OBJ_D OBJ_D_ID	OBJ_A A_COL1	OBJ_B B_COL1	OBJ_B B_COL2	OBJ_D D_COL1	STATISTICS AVG_B_COL2_
1	101	101	101	AA1	BB1	321	DD1	276.0000
2	101	102	101	AA1	BB2	231	DD1	276.0000
3	101	102	103	AA1	BB2	231	DD3	276.0000
4	101	102	104	AA1	BB2	231	DD4	276.0000
5	102	103		AA2	BB3	123		123.0000
6	103	103		AA3	BB3	123		123.0000
7	102		101	AA2			DD1	
8	103		102	AA3			DD2	
9	104		102	AA4			DD2	
10		104			BB4	312		
11		105			BB5	213		
12			105				DD5	

**Fig. 3**