PPDM Association

Reference Guide

Load of the Rings

Last updated for PPDM 3.7

Version 1.0

How to improve data quality and integrity by loading PPDM 3.7.1 with foreign key constraints enabled.

1. Table of Contents

2. PPDM Association Disclaimer

USER agrees on receipt of the PPDM model and supporting material that its use by USER constitutes acceptance by the USER that:

1. The PPDM products and supporting material are provided “as is” without any warranty or condition of any kind, either statutory, express or implied, including, without limiting the foregoing, implied warranties of merchantability of fitness for a particular purpose, or warranty as to non-infringement with existing or future copyright, patent of other intellectual property rights claimed or asserted by any third party;
2. PPDM shall not be liable to User nor to anyone else who uses or otherwise obtains the PPDM products and supporting material, either directly or indirectly from User, for direct, indirect, special or consequential damages or other loss or cost however arising;
3. User agrees to indemnify and hold PPDM harmless from any third party claims arising directly or indirectly, or however arising, from the PPDM products and supporting material provided to the User;
4. The user acknowledges that the PPDM products and supporting material are under copyright notice and that all rights are reserved by PPDM, the Public Petroleum Data Model Association.

3. About This Document

This document is designed for data managers and administrators, database analysts, programmers and system designers. It will help guide you through the process of improving the quality of data loads into PPDM 3.7.1 by showing you how to load your data with constraints enabled.

4. Introduction

As long as databases have existed, data managers have struggled to balance the need for speed with the desire for high quality results. Virtually every dataset that needs to be loaded contains errors of some kind, from spelling and typographic errors to invalid or incorrect values.

In the mid 1990's, relational database technology advanced to include integrated support for Referential Integrity (RI), allowing data managers to apply many useful validation processes to their data. For example, rather than allowing users to create many versions of the same country name (Canada, CA, CAN, Ca, Canda ..), they could be forced to select a valid country from a pre-existing list of values. Even better, the task of managing these values could be handled by the users who were most qualified to do the work, instead of by a DBA.

Loading data with RI enabled also can ensure that critical business dependencies are maintained. For example, a well test that cannot be associated with a well should not be loaded into your database. Seismic points that are not referenced to their owning seismic line or 3D survey should be held for review and correction.

Exploration and Production data contains, because of its complex nature, a rich network of relationships that reflect real business practices and are necessary to support business processes. PPDM 3.7.1 contains a very rich set of tables, columns and constraints.

While it's possible to fully load a PPDM database with all foreign key constraints enabled, figuring out what order the tables have to be loaded in is complicated and can be frustrating. In 2004, the PPDM Association Data Management work group recommended that the load order for PPDM 3.7.1 be determined and made available to the membership. This document will help you through the process.

5. Some Things to Think About

5.1. Disclaimer

The PPDM Association does not promise that the material described in this document will work perfectly for you, but we do feel that it will provide you with important guidelines and recommendations.

The PPDM Association works mostly in Oracle, but also with SQL*Server, MySQL and PostGres. As much as possible, we try to produce deliverables in ANSII SQL, that can be run in any SQL 92 compliant RDBMS. Please consider the material provided to be guidelines only; you should review the results and modify them to suit your implementation as necessary and prudent.

We urge implementers to feed back any revisions to the PPDM Association, so that the industry standard can grow and mature.

5.2. Theory vs Reality

The Load of the Rings assumes that you will be fully populating a PPDM 3.7.1 database. In most real life situations, you are most likely to only populate a portion of the data model at a time, since you will probably be loading from a non-PPDM 3.7.1 source. We are including enough information in the Load of the Rings to allow you to dynamically subset and re-calculate the ring order once you know which columns you will load.

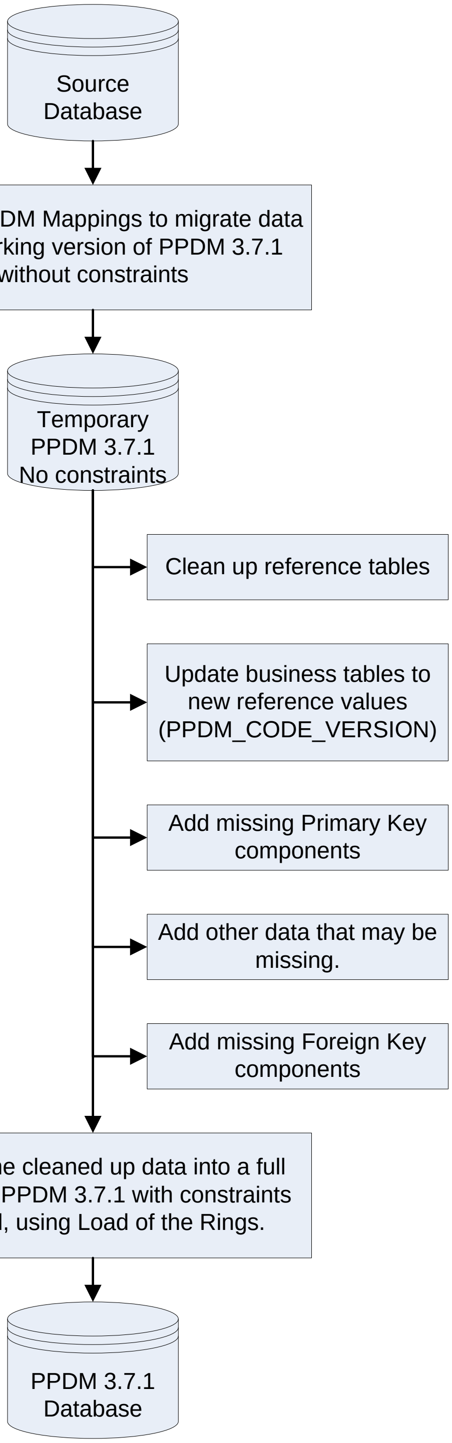
In order to calculate which columns you will load, remember to take these issues into consideration:

- Include the columns you have mapped into PPDM 3.7.1.
- The initial load of the rings order includes all foreign keys, including Units of Measure and PPDM ROW QUALITY. If you won't be using those foreign keys, your load order may change.
- Include all the primary key columns for all the tables you will be loading. Note that some of the primary key columns may not be populated through your mapping. You'll have to write loaders for those columns.
- Include all the necessary audit information for the tables you are moving into PPDM 3.7.1. When you do this, you'll have to decide how to move the data. As yourself these questions
    - Will I keep the all or part of the audit history from my original data source
    - Will I delete all or some of the original audit history and replace it with new information?

5.3. Using the Load of the Rings for Migrations

You can use the Load of the Rings to help you with your data migration in ways that will help you ensure that you have your data in the best possible condition prior to the final load. There are many ways you can do this, depending on your data and the decisions made by your implementation team.

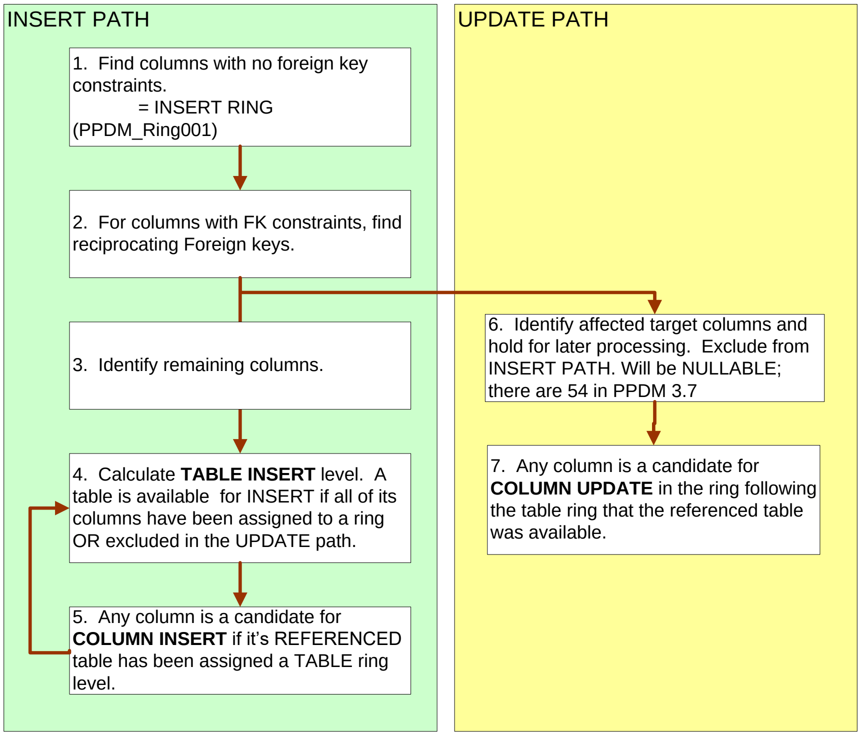
The diagram below shows one possible work flow that could be used; think of this as a starting point.


Source Database
Use the PPDM Mappings to migrate data into a working version of PPDM 3.7.1 without constraints
Temporary PPDM 3.7.1 No constraints
Clean up reference tables
Update business tables to new reference values (PPDM_CODE_VERSION)
Add missing Primary Key components
Add other data that may be missing.
Add missing Foreign Key components
Migrate the cleaned up data into a full version of PPDM 3.7.1 with constraints enabled, using Load of the Rings.
PPDM 3.7.1 Database

6. Design

6.1. Overview

The process for calculating the load order is illustrated below. Each step is described in the following sections.



INSERT PATH

1. Find columns with no foreign key constraints.
= INSERT RING
(PPDM_Ring001)

2. For columns with FK constraints, find reciprocating Foreign keys.

3. Identify remaining columns.

4. Calculate **TABLE INSERT** level. A table is available for INSERT if all of its columns have been assigned to a ring OR excluded in the UPDATE path.

5. Any column is a candidate for **COLUMN INSERT** if it's REFERENCED table has been assigned a TABLE ring level.

UPDATE PATH

6. Identify affected target columns and hold for later processing. Exclude from INSERT PATH. Will be NULLABLE; there are 54 in PPDM 3.7

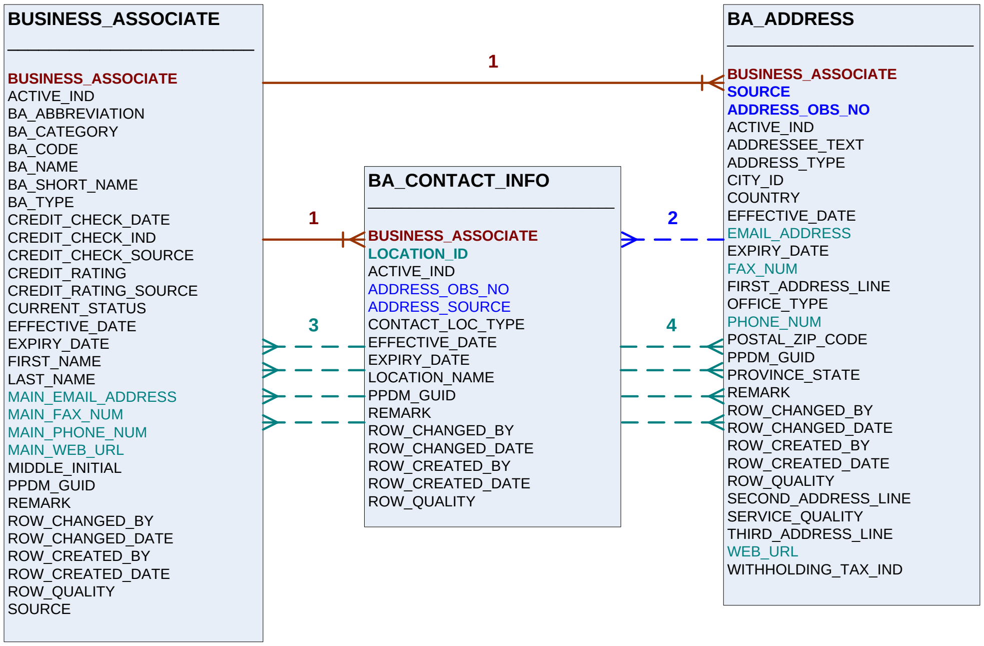
7. Any column is a candidate for **COLUMN UPDATE** in the ring following the table ring that the referenced table was available.

6.2. Calculate reciprocating foreign key columns

First, it's necessary to understand what a reciprocating foreign key is and why they exist in PPDM. Reciprocal foreign keys exist when two database tables each contain one or more foreign keys that reference the other. These may be shown as a "pig's ear" or recursive relationship, or as foreign keys from a child table that are referenced in the parent table.

In PPDM, reciprocal foreign keys are most often created to support specific kinds of denormalization requested by the work group. Denormalized columns are added to promote query efficiency and to simplify some kinds of calculations.

For example, the Business Associates module in PPDM 3.7.1 contains the tables BUSINESS_ASSOCIATE, BA_ADDRESS and BA_CONTACT_INFO.



In the diagram, you will see three sets of foreign keys, each shown in a different color. Most of the foreign keys are compound keys. For more information about compound keys, see the Constraints Reference Guide on the PPDM Web Site.

We'll describe each set and how they affect the load order:

FK Set	Description	Load Order
1	The tables BA_CONTACT_INFO and BA_ADDRESS are child tables of BUSINESS_ASSOCIATE. You cannot load any row into the child table unless you can validate the column BUSINESS_ASSOCIATE in the child table against a matching value in the BUSINESS_ASSOCIATE table.	The table BUSINESS_ASSOCIATE must be populated before you can load either of the child tables. BUSINESS ASSOCIATE must exist in a ring prior to the load ring for either of the child tables.
2	The table BA_CONTACT_INFO has a foreign key from BA_ADDRESS that allows you to define which address each row of contact information is valid for. This is important when a business associate has more than one office, and you want to associate the proper contact information with each office.	You cannot load the referenced columns (ADDRESS_OBS_NO and ADDRESS_SOURCE) in BA_CONTACT_INFO until the necessary values have been loaded into BA_ADDRESS.
3	The BUSINESS_ASSOCIATE table contains 4 denormalized values from BA_CONTACT_INFO. These values represent the Main phone number, main fax number, main email address and main URL for the business associate. This information can be derived from BA_CONTACT_INFO by using a select	You cannot fully populate the reciprocal columns (MAIN WEB URL, MAIN PHONE NUM, MAIN FAX NUM or MAIN EMAIL ADDRESS) in BUSINESS ASSOCIATE until you have loaded the necessary values in BA_CONTACT_INFO.

	query that obtains values having the appropriate LOCATION_TYPE.	
4	The table BA_ADDESS has four foreign keys from BA_CONTACT_INFO. Each foreign key references two columns in BA_ADDRESS. One is always BUSINESS_ASSOCIATE. This other column is derived from BA_CONTACT_INFO by using a select query that obtains values having the appropriate LOCATION TYPE.	You cannot fully populate the reciprocal columns (WEB URL, PHONE NUM, FAX NUM or EMAIL ADDRESS) in BUSINESS ASSOCIATE until you have loaded the necessary values in BA_CONTACT_INFO.

The PPDM Meta Model can be used to identify the tables that have reciprocating constraints:

```
select  distinct c.table_name, c.column_name
  from  PPDM_CONS_COLUMN c, PPDM_CONS_COLUMN c1
 where  c.referenced_table_name is not null
   and  c.TABLE_NAME  =  c1.REFERENCED_TABLE_NAME

   and  c.REFERENCED_TABLE_NAME = c1.TABLE_NAME
```

6.3. Exclude the Reciprocal Target Columns

From this, it is clear that you cannot fully load any of the example three tables in a single pass.

The columns that are referenced by the foreign key constraint in the target tables are compound (each constraint references more than one column). In each case, BUSINESS_ASSOCIATE is one of them. Since the foreign key is compound, the constraint will not fire until ALL of the components have been populated. That means that you can populate the BUSINESS_ASSOCIATE column in each table once its parent in the BUSINESS_ASSOCIATE table has been populated; the compound foreign key will not be enforced until the other column in the constraint is also populated.

In our example, the reciprocal foreign key target columns are the second component of the foreign keys from BA_CONTACT_INFO that contain phone numbers, fax numbers, email addresses and web URL’s. There are a total of 8 columns.

Having gotten that far, stop and think for a moment or two. All of the columns that we just identified are optional and they are all derived using some pretty simple logic (where LOCATION_TYPE = ‘whatever’). You can always calculate them later if your data is any good. If you don’t need to create them during the bulk load, don’t. Denormalized data such as these columns always creates some maintenance overhead – is the overhead worth the value to you? You can re-calculate them later if necessary.

If you decide to populate a reciprocal column, you will need to exclude these table and column names from the next step. You will not be able to populate them using an INSERT statement. They must be populated in another UPDATE step.

6.4. The Insert Rings

6.4.1. Rules

The PPDM Load of the Rings data contains rings that are numbered and named sequentially. Each ring is either an INSERT ring or an UPDATE ring. We could have combined them, but we felt that it would just be confusing.

- Every ring is named as PPDM RING nni or PPDM RING nnu.
  - Rings that end in the letter "i" are insert rings.
  - Rings that end in the letter "u" are update rings.
- The rings are numbered in the order they can be run.
- The order in which you process the tables within the same ring is not relevant. Load them in any order you choose.
- In order to do the insert, all preceding rings must be run and committed.
- The only columns that are eligible for population as an update are the 54 reciprocating columns identified in the last step.

6.4.2. Tables and Columns

There are several considerations to think about when you make your final decision about your final load order. Let's look at BA_SERVICE as an example:

Column Name	Primary Key	Foreign Key	Ring Order Rule
BUSINESS_ASSOCIATE	✓	✓	After BUSINESS_ASSOCIATE
BA_SERVICE_TYPE	✓	✓	After R_BA_SERVICE_TYPE
BA_SERVICE_SEQ_NO	✓		Must be created on insert
ACTIVE_IND			Ring 0 (no dependencies)
EFFECTIVE_DATE			Ring 0 (no dependencies)
EXPIRY_DATE			Ring 0 (no dependencies)
PPDM_GUID			Ring 0 (no dependencies)
REMARK			Ring 0 (no dependencies)
ROW_CHANGED_BY			Ring 0 (no dependencies)
ROW_CHANGED_DATE			Ring 0 (no dependencies)
ROW_CREATED_BY			Ring 0 (no dependencies)
ROW_CREATED_DATE			Ring 0 (no dependencies)
ROW_QUALITY		✓	After R_PPDM_ROW_QUALITY
SERVICE_QUALITY		✓	After R_SERVICE_QUALITY
SOURCE		✓	After R_SOURCE

PRIMARY KEYS: In order to insert into any table, all of the components in the Primary Key must be created during the insert (since they are all NOT NULL, you can't leave them empty). That means:

- Each target PK column that is part of a foreign key must have parent data available for validation via the foreign key. These parent columns are prerequisites for inserting each target column. The tables and columns that are referenced by the foreign key will be created in an earlier ring.
- One of the components of the Primary Key (BA_SERVICE_SEQ_NO) must be inserted procedurally during the load (unless you have been able to map that column to a column in the original database).

FOREIGN KEYS: In order to insert into a column that is validated by a foreign key, the necessary parent table and column information must be loaded into the database. In the Load of the Rings, these pre-requisites will exist in a prior ring.

PPDM_COLUMN_GROUP contains the ring order for every column in each table. A column is available for INSERT after its pre-requisite table and column have been populated. Any column that is not part of a foreign key constraint can be populated at any time; these columns are all referenced as RING00i.

PPDM_TABLE_GROUP contains the ring in which the table INSERT must occur, assuming that you intend to fully populate the table. The sequence of this ring is set at the ring of the ***last*** column to be available for INSERT for that table. If you are not planning to load all the columns in a table, you may be able to change the order to exclude the columns you are not populating.

6.4.3. The Table Ring Data

The table ring for each PPDM 3.7.1 table in PPDM_TABLE_GROUP shows the ring number when that table can be fully INSERTED. Once that table has been inserted, it becomes a candidate for being referenced by other tables that depend on its existence (by a foreign key). It is not necessary for the UPDATE rings to be completed before a parent table is available for referencing by its child tables.

6.4.4. The Column Ring Data

The Column Ring for each PPDM 3.7.1 column in PPDM_COLUMN_GROUP shows the insert or update ring level when each column could be populated. There are several things in this table that you should know:

1. Some columns have no foreign key constraints. You can populate those columns any time you wish. The sample data shows these columns in Ring00i.
2. Some columns have one or more foreign key constraints on it. A foreign key will fire when ALL the columns in the constraint are populated. You can populate one column in a two part foreign key without populating the other. The sample data lists all of the foreign keys on a column in the REMARK column. You can also calculate the list using the table PPDM_CONS_COLUMN.
3. The 54 reciprocating foreign key columns (NOT NULL only) in PPDM must be populated in an UPDATE ring. It is not necessary for an UPDATE column to be populated before that table is available for referencing by other tables.

6.5. The Update Rings

6.5.1. Rules

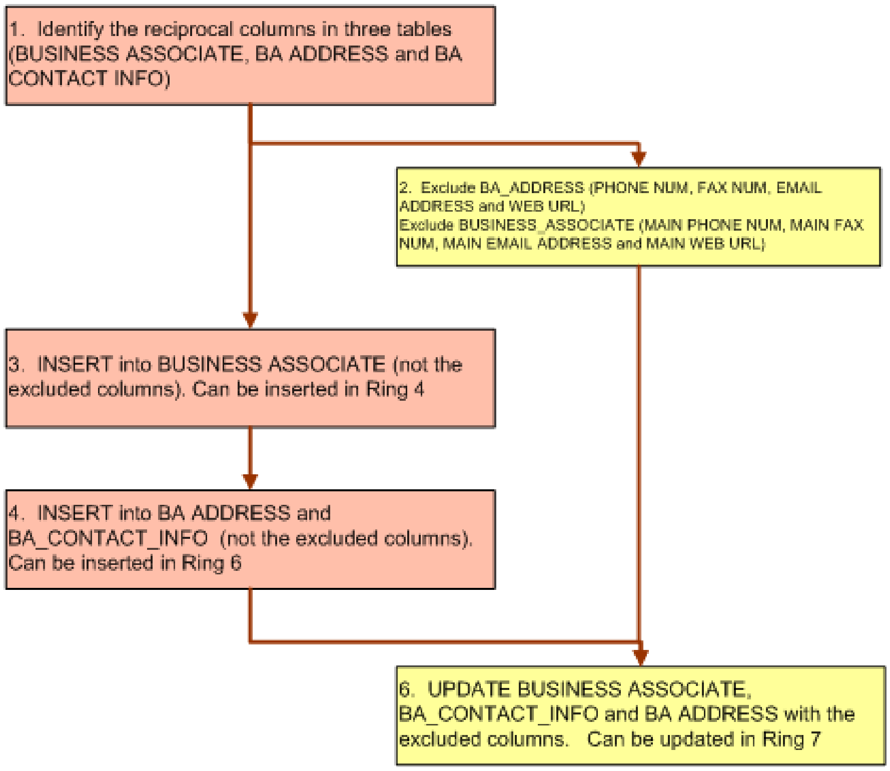
Each update ring is placed in the sequence so that the necessary parent tables have already been populated. In order for an UPDATE to be eligible, only the Primary Key columns in the source table need to be populated (for the foreign keys that are referenced to the primary key).

- The order in which you process the tables within the same ring is not relevant. Update them in any order you choose.
- In order to do the update, all preceding rings must be run and committed.

6.5.2. Things to Think About

You need to look at every reciprocating constraint to determine whether the value is calculated and derivable or whether it is important information that should not be lost. While our example contains derivable information, others are not derivable. Do not be deceived, or you may regret it.

This is what the final process (just for the three tables we're talking about) would look like:


1. Identify the reciprocal columns in three tables (BUSINESS ASSOCIATE, BA ADDRESS and BA CONTACT INFO)
2. Exclude BA_ADDRESS (PHONE NUM, FAX NUM, EMAIL ADDRESS and WEB URL)
Exclude BUSINESS_ASSOCIATE (MAIN PHONE NUM, MAIN FAX NUM, MAIN EMAIL ADDRESS and MAIN WEB URL)
3. INSERT into BUSINESS ASSOCIATE (not the excluded columns). Can be inserted in Ring 4
4. INSERT into BA ADDRESS and BA_CONTACT_INFO (not the excluded columns). Can be inserted in Ring 6
6. UPDATE BUSINESS ASSOCIATE, BA_CONTACT_INFO and BA ADDRESS with the excluded columns. Can be updated in Ring 7

6.5.3. Business Associate Example

This example shows how the rings look after they are calculated. You will see that every column has been assigned to an insert ring level except the blue columns.

The BUSINESS_ASSOCIATE table is available for use during INSERTS into other tables once all of its columns (except the excluded reciprocating columns) have been assigned to a ring. The last eligible column can be populated in RING04i, so the table ring is RING04.

BUSINESS_ASSOCIATE can be used by another column insert requirement for other tables in RING06i.

RING00i	BUSINESS_ASSOCIATE	ACTIVE_IND	
RING00i	BUSINESS_ASSOCIATE	BA_ABBREVIATION	
RING04i	BUSINESS_ASSOCIATE	BA_CATEGORY	where ppdm_constraint.constraint_name in ('BA_R_BC_FK');
RING00i	BUSINESS_ASSOCIATE	BA_CODE	
RING00i	BUSINESS_ASSOCIATE	BA_NAME	
RING00i	BUSINESS_ASSOCIATE	BA_SHORT_NAME	
RING04i	BUSINESS_ASSOCIATE	BA_TYPE	where ppdm_constraint.constraint_name in ('BA_R_BT_FK');
RING00i	BUSINESS_ASSOCIATE	BUSINESS_ASSOCIATE	where ppdm_constraint.constraint_name in ('BA_BACI_FK', 'BA_BACI_FK2', 'BA_BACI_FK3', 'BA_BACI_FK4');
RING00i	BUSINESS_ASSOCIATE	CREDIT_CHECK_DATE	
RING00i	BUSINESS_ASSOCIATE	CREDIT_CHECK_IND	
RING02i	BUSINESS_ASSOCIATE	CREDIT_CHECK_SOURCE	where ppdm_constraint.constraint_name in ('BA_R_S_FK2');
RING00i	BUSINESS_ASSOCIATE	CREDIT_RATING	
RING00i	BUSINESS_ASSOCIATE	CREDIT_RATING_SOURCE	
RING04i	BUSINESS_ASSOCIATE	CURRENT_STATUS	where ppdm_constraint.constraint_name in ('BA_R_BS_FK');
RING00i	BUSINESS_ASSOCIATE	EFFECTIVE_DATE	
RING00i	BUSINESS_ASSOCIATE	EXPIRY_DATE	
RING00i	BUSINESS_ASSOCIATE	FIRST_NAME	
RING00i	BUSINESS_ASSOCIATE	LAST_NAME	

RING07u	BUSINESS_ASSOCIATE	MAIN_EMAIL_ADDRESS	where ppdm_constraint.constraint_name in ('BA_BACI_FK2');
RING07u	BUSINESS_ASSOCIATE	MAIN_FAX_NUM	where ppdm_constraint.constraint_name in ('BA_BACI_FK3');
RING07u	BUSINESS_ASSOCIATE	MAIN_PHONE_NUM	where ppdm_constraint.constraint_name in ('BA_BACI_FK4');
RING07u	BUSINESS_ASSOCIATE	MAIN_WEB_URL	where ppdm_constraint.constraint_name in ('BA_BACI_FK');
RING00i	BUSINESS_ASSOCIATE	MIDDLE_INITIAL	
RING00i	BUSINESS_ASSOCIATE	PPDM_GUID	
RING00i	BUSINESS_ASSOCIATE	REMARK	
RING00i	BUSINESS_ASSOCIATE	ROW_CHANGED_BY	
RING00i	BUSINESS_ASSOCIATE	ROW_CHANGED_DATE	
RING00i	BUSINESS_ASSOCIATE	ROW_CREATED_BY	
RING00i	BUSINESS_ASSOCIATE	ROW_CREATED_DATE	
RING02i	BUSINESS_ASSOCIATE	ROW_QUALITY	where ppdm_constraint.constraint_name in ('BA_R_PRQ_FK');
RING02i	BUSINESS_ASSOCIATE	SOURCE	where ppdm_constraint.constraint_name in ('BA_R_S_FK');

6.5.4. Exceptions

The % GEOMETRY tables in PPDM contain foreign keys that do not reference the primary key of the source table. However, none of these tables contain reciprocating constraints, so they can all be loaded in the INSERT rings.

6.6. The Sample Data in PPDM 3.7.1

We have populated the sample data in PPDM as follows:

1. Table rings have their own groups. All tables rings are INSERT RINGS
2. Column rings have their own groups. There are two kinds of groups (insert and update).
3. Each column has been loaded into PPDM_COLUMN_GROUP once for every group it is in.
4. The REMARK column lists the constraints that govern each column in PPDM_COLUMN_GROUP.
5. Each occurrence of the column in PPDM_COLUMN_GROUP shows the column insert ring in which the referenced table will allow it to be inserted. All occurrences must be available for the column to be ready to use.

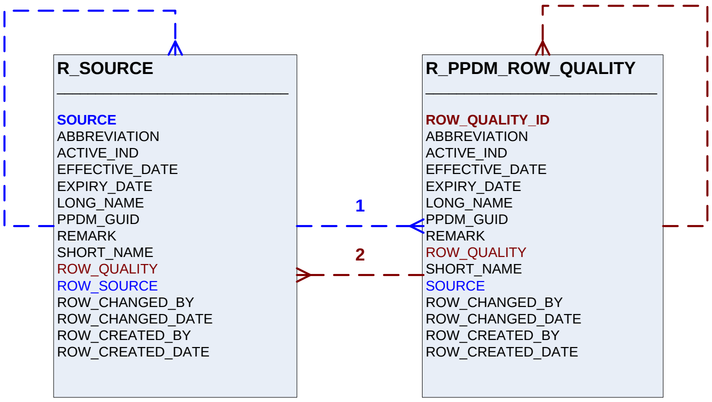
7. PPDM Load of the Rings

7.1. Getting Started

By this time, you've probably got a good understanding of how to calculate the Load of the Rings. There is one more very important issue to understand before you can go ahead and get started. You have to figure out where to start.

There are two tables that have to be populated before you can fully populate any other table in PPDM 3.7.1. These tables are difficult to identify because they each have a foreign key from the other and they each contain a "pig's ear" foreign key back to itself.

The two tables are R_SOURCE and R_PPDM_ROW_QUALITY. The diagram below shows these relationships.



These two tables have to comprise the first two rings in PPDM Load of the Rings.

7.1.1. The Insert Ring

An INSERT into both tables can be done in the same ring, if you exclude the reciprocal columns:

- R_SOURCE.ROW_SOURCE
- R_SOURCE.ROW_QUALITY
- R_PPDM_ROW_QUALITY.SOURCE
- R_PPDM_ROW_QUALITY. ROW_QUALITY

7.1.2. The Update Ring

Once you have inserted the primary key columns for the two tables, you will be able to UPDATE both tables to include the insert exclusions. Note that in this case, the reciprocating constraints are NOT calculated or derived. In this case, you must do the update, or you will lose important information.

RING00i	R_PPDM_ROW_QUALITY	ABBREVIATION	
RING00i	R_PPDM_ROW_QUALITY	ACTIVE_IND	
RING00i	R_PPDM_ROW_QUALITY	EFFECTIVE_DATE	
RING00i	R_PPDM_ROW_QUALITY	EXPIRY_DATE	
RING00i	R_PPDM_ROW_QUALITY	LONG_NAME	
RING00i	R_PPDM_ROW_QUALITY	PPDM_GUID	
RING00i	R_PPDM_ROW_QUALITY	REMARK	
RING00i	R_PPDM_ROW_QUALITY	ROW_CHANGED_BY	
RING00i	R_PPDM_ROW_QUALITY	ROW_CHANGED_DATE	
RING00i	R_PPDM_ROW_QUALITY	ROW_CREATED_BY	
RING00i	R_PPDM_ROW_QUALITY	ROW_CREATED_DATE	
RING01u	R_PPDM_ROW_QUALITY	ROW_QUALITY	where ppdm_constraint.constraint_name in ('R_PRQ_R_PRQ_FK');
RING00i	R_PPDM_ROW_QUALITY	ROW_QUALITY_ID	
RING00i	R_PPDM_ROW_QUALITY	SHORT_NAME	
RING01u	R_PPDM_ROW_QUALITY	SOURCE	where ppdm_constraint.constraint_name in ('R_PRQ_R_S_FK');
RING00i	R_SOURCE	ABBREVIATION	
RING00i	R_SOURCE	ACTIVE_IND	
RING00i	R_SOURCE	EFFECTIVE_DATE	
RING00i	R_SOURCE	EXPIRY_DATE	

RING00i	R_SOURCE	LONG_NAME	
RING00i	R_SOURCE	PPDM_GUID	
RING00i	R_SOURCE	REMARK	
RING00i	R_SOURCE	ROW_CHANGED_BY	
RING00i	R_SOURCE	ROW_CHANGED_DATE	
RING00i	R_SOURCE	ROW_CREATED_BY	
RING00i	R_SOURCE	ROW_CREATED_DATE	
RING01u	R_SOURCE	ROW_QUALITY	where ppdm_constraint.constraint_name in ('R_S_R_PRQ_FK');
RING01u	R_SOURCE	ROW_SOURCE	where ppdm_constraint.constraint_name in ('R_S_R_S_FK');
RING00i	R_SOURCE	SHORT_NAME	
RING00i	R_SOURCE	SOURCE	

7.2. Exclusions

The PPDM calculated Load of the Rings takes into account the foreign keys related to **Units of Measure**. If you are using the foreign key constraints provided in the files PPDM37.UOM and PPDM37.OUOM, you will need to add these into your load sequence. If you are not using the foreign keys, you will need to drop these from your load sequence.

The PPDM calculated Load of the Rings does not take into account the **Unique Keys** in the %GEOMETRY tables, or the Unique keys created if you use the PPDM_GUID. If you are using these constraints, you will need to add these into your load sequence.

8. More Tips and Tricks

8.1. Plan

Discuss the process with everyone involved before you try this for the first time. Most data loading practitioners are accustomed to loading with constraints disabled. They hope that when the constraints are re-enabled the data will allow them all to be enabled immediately. If your data is that good, you might want to present a paper bragging about it at a PPDM Conference.

Loading data with constraints enabled will involve a learning process for everyone, and will definitely be more complicated than doing it the old way. Be patient and plan to get the most benefit out of the load possible.

8.2. Performance

The PPDM Load of the Rings does not take performance into consideration. We recommend that you test the process outside of peak user hours. Your system administrator and DBA will be important players in designing the process for success.

Make sure you have adequate system resources dedicated to the process. Shared pool size may need to be quite large, particularly if you are dynamically cleaning up your data as you load.

8.3. Concurrency

Be sure to commit during the load frequently, and ensure you have allocated enough shared pool size and roll back space.

8.4. Prepare the data

- If you're interested in cleaning up reference tables during the load, consider using the table PPDM_CODE_VERSION to automate the conversion from old values to new values during the load.
- The ALIAS tables that are associated with most of the major business objects (wells, seismic sets, business associates) can be pre-populated and used as tools to help clean up identifiers for those objects during the load. If you do this, keep track of the conversions, so you don't accidentally find an orphan in a later ring!

8.5. Using the Rings in the Real World

If you are migrating data from a legacy system into PPDM 3.7.1, we recommend that you consider staging the data into a PPDM data model without any constraints. This will allow you to examine your data and ensure that all the necessary components for primary key validation and your foreign keys are in place.

This data repository should be considered a temporary store only and should never be used for transactional processing!!! Once you have the data in the temporary repository in place, migrate it into PPDM 3.7.1 with all constraints enabled using the Load of the Rings. Any errors that remain in your data will be exposed through processing errors; you can then go back and correct these errors.

9. Conclusions

We feel that the added value from populating your PPDM 3.7.1 data base using the Load of the Rings will add substantially to the quality and value of your data. The process may take a little time to fully understand and will add to the time necessary for the data migration. The added value you get from higher quality data should ultimately offset your costs of development.