regoUniversity 2018

Data Model | Advanced

Your Guides: David Matzdorf and Ben Rimmasch

Introductions

- Take 5 Minutes
- Turn to a Person Near You
- Introduce Yourself
- Business Cards

Agenda

- OBS Tables
  - Associations and Type
  - Filtering
- Portfolio Tables
- Baseline and Project Hierarchy
- Admin Tables
  - Notifications & Captions
  - Custom Attributes on Objects
  - Portlet Tables
  - Security
- Process and Job Logs

OBS Tables

regoUniversity 2018

OBS Association Table

- PRJ_OBS_ASSOCIATIONS
  - Table contains the association for a particular record to an obs through the unit_id and obs_type.
  - It is important to check the column table_name and the table are of the same type.
  - Common practice when finding an investment's OBS is to use the odf_objects table to connect to the column table_name on the PRJ_OBS_ASSOCIATIONS table. However if connecting a resource the SRM_RESOURCES table will always only connect to the table_name of 'SRM_RESOURCES'.

PRJ_OBS_ASSOCIATIONS

```
SELECT *
FROM INV_INVESTMENTS INVI
JOIN ODF_OBJECTS OBJS ON OBJS.CODE = INVI.ODF_OBJECT_CODE
JOIN PRJ_OBS_ASSOCIATIONS POA ON POA.RECORD_ID = INVI.ID AND POA.TABLE_NAME = OBJS.OBS_CODE
```

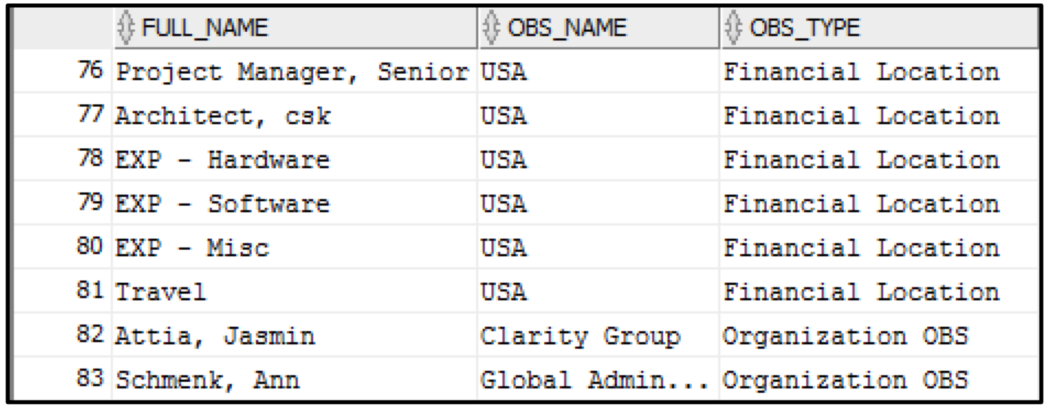
PRJ OBS Tables

- PRJ_OBS_TYPES
  - Contains the information for specific OBS types. This table is important for distinguishing which obs association you are looking at.
    - Ex: Department vs Location
- PRJ_OBS_UNITS
  - Table contains the base details for the node in the OBS

PRJ_OBS_TYPES

PRJ_OBS_UNITS

```
SELECT SRM.FULL_NAME
,POU.NAME OBS_NAME
,POT.NAME OBS_TYPE
FROM SRM_RESOURCES SRM
JOIN PRJ_OBS_ASSOCIATIONS POA ON POA.RECORD_ID = SRM.ID
   AND POA.TABLE_NAME = 'SRM_RESOURCES'
JOIN PRJ_OBS_UNITS POU ON POU.ID = POA.UNIT_ID
JOIN PRJ_OBS_TYPES POT ON POT.ID = POU.TYPE_ID
```



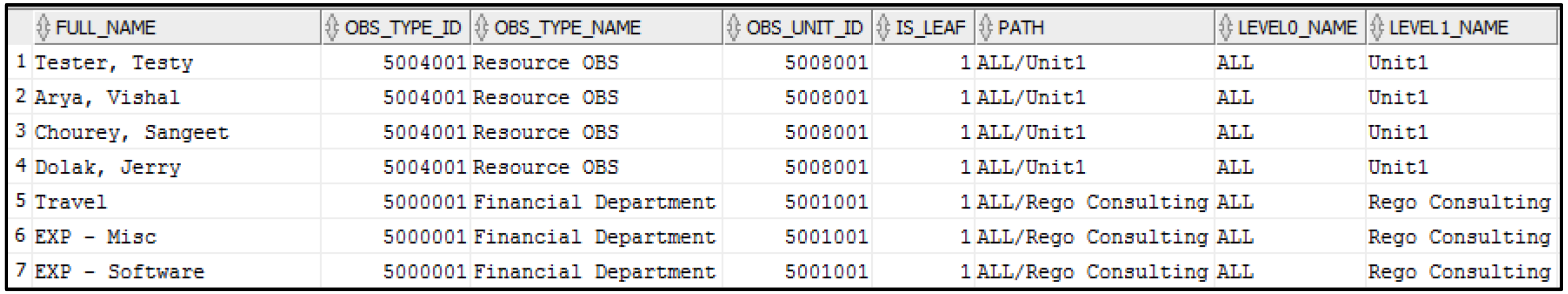
	FULL_NAME	OBS_NAME	OBS_TYPE
76	Project Manager, Senior	USA	Financial Location
77	Architect, csk	USA	Financial Location
78	EXP - Hardware	USA	Financial Location
79	EXP - Software	USA	Financial Location
80	EXP - Misc	USA	Financial Location
81	Travel	USA	Financial Location
82	Attia, Jasmin	Clarity Group	Organization OBS
83	Schmenk, Ann	Global Admin...	Organization OBS

NBI OBS Table

- NBI_DIM_OBS
  - Contains the details for a specific unit in the OBS. This table includes the path, and all level's associated.

NBI_DIM_OBS

```
SELECT SRM.FULL_NAME
, NDO.*
FROM SRM_RESOURCES SRM
JOIN PRJ_OBS_ASSOCIATIONS POA ON POA.RECORD_ID = SRM.ID
   AND POA.TABLE_NAME = 'SRM_RESOURCES'
JOIN NBI_DIM_OBS NDO ON NDO.OBS_UNIT_ID = POA.UNIT_ID;
```



	FULL_NAME	OBS_TYPE_ID	OBS_TYPE_NAME	OBS_UNIT_ID	IS_LEAF	PATH	LEVEL0_NAME	LEVEL1_NAME
1	Tester, Testy	5004001	Resource OBS	5008001	1	ALL/Unit1	ALL	Unit1
2	Arya, Vishal	5004001	Resource OBS	5008001	1	ALL/Unit1	ALL	Unit1
3	Chourey, Sangeet	5004001	Resource OBS	5008001	1	ALL/Unit1	ALL	Unit1
4	Dolak, Jerry	5004001	Resource OBS	5008001	1	ALL/Unit1	ALL	Unit1
5	Travel	5000001	Financial Department	5001001	1	ALL/Rego Consulting	ALL	Rego Consulting
6	EXP - Misc	5000001	Financial Department	5001001	1	ALL/Rego Consulting	ALL	Rego Consulting
7	EXP - Software	5000001	Financial Department	5001001	1	ALL/Rego Consulting	ALL	Rego Consulting

How To Filter On OBS

- OBS_UNITS_FLAT_BY_MODE
  - Table contains the flat hierarchy of each OBS. This is important because it allows filtering by a specific unit with the unit_mode on the record.
  - Each record contains a linked_unit_id which represents the child, and it's relationship to the unit id which represents the parent id. The relationship is represented by the unit_mode.
  - Example on left: Unit and Children will get all records that are a descendent of the Unit_ID specified on the OBS_UNITS_FLAT_BY_MODE by the parameter :OBS_ID

OBS_UNITS_FLAT_BY_MODE

```
SELECT SRMR.FULL_NAME
FROM SRM_RESOURCES SRMR
WHERE (OBS_ID IS NULL OR
    EXISTS (SELECT 1
        FROM OBS_UNITS_FLAT_BY_MODE OBSM
        JOIN PRJ_OBS_ASSOCIATIONS OBSA ON
            OBSM.LINKED_UNIT_ID = OBSA.UNIT_ID AND
            OBSA.TABLE_NAME = 'SRM_RESOURCES'
        WHERE OBSM.UNIT_ID = OBS_ID AND
            OBSM.UNIT_MODE = NVL(OBS_MODE,
            'OBS_UNIT_AND_CHILDREN')
        AND OBSA.RECORD_ID = SRMR.ID))
```

Portfolio Tables

regoUniversity 2018

Let Rego be your guide.

Portfolio Tables

- PFM_PORTFOLIOS
  - Table contains the base information for portfolios
- PFM_INVESTMENTS
  - Table contains the investments and investment fields associated to the portfolio. This connects to the INV_INVESTMENTS table
  - Portfolio_id = pfm_portfolios.id
  - Investment_id = inv_investments.id
- PFM_PROJECTS/IDEAS
  - Table contains the project/idea fields that are associated to the portfolio. These records should only be used if you are purposely not synching the portfolio to create a baseline

Baseline and Project Hierarchy

regoUniversity 2018

Baseline Tables

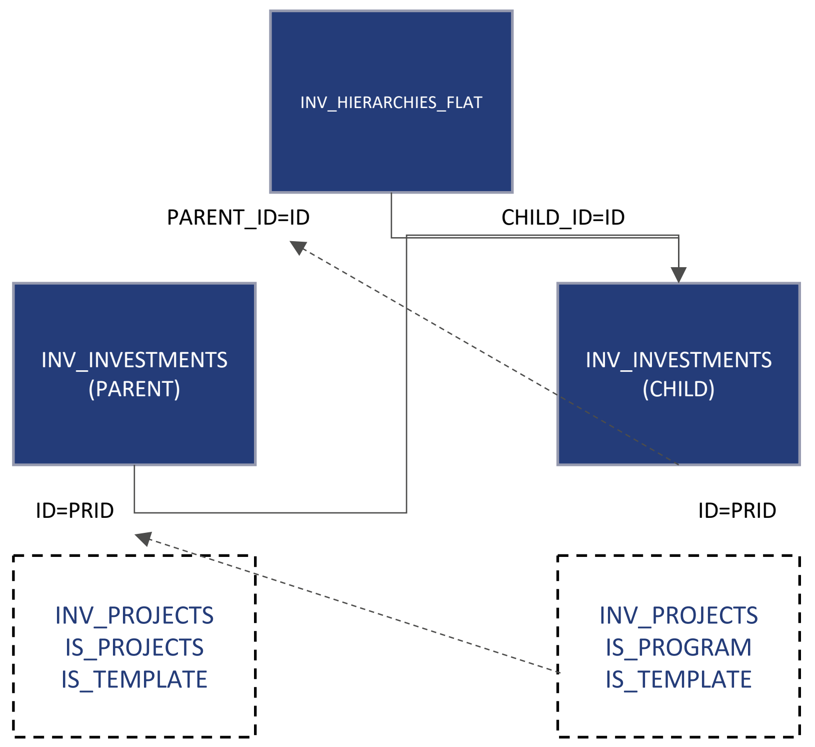
- PRJ_BASELINES
  - Table contains the baselines stored in the database. These baselines connect to the investment being baselined
- PRJ_BASELINE_DETAILS
  - Table contains the drilldown of the baseline. This table will contain baseline data based on Task, Assignment, and Project
  - **<u>Important Defintions</u>**
  - USAGE_SUM = Baselined Effort (Act + Remaining Effort) in seconds
  - COST_SUM = Baselined Cost
  - DURATION = Effort duration
  - Baseline_id = prj_baselines.id
  - **Note:** The details can be linked to the timeslice table

Master / Sub Tables

- INV_HIERARCHIES_FLAT
  - This table enables rapid retrieval of all descendants within a hierarchy.
  - Table contains the relationships associated to each investment
  - CHILD_ID = INV_INVESTMENTS.ID
  - PARENT_ID = INV_INVESTMENTS.ID

- Same table is used for multiple purposes
  - Filter for Program
  - INV_PROJECTS . IS_PROGRAM

- The link_source_id contains the ID of the immediate parent of the child. By examining the link_source_id, the original hierarchical order can also be retrieved



Activity

<u>Primary Activity</u>

- Display all the projects associated to programs. Program, Project, Project ID

<u>Additional Activity</u>

- Display all of the projects that have a sub-project, but are not programs. Project, Project ID

Admin Tables

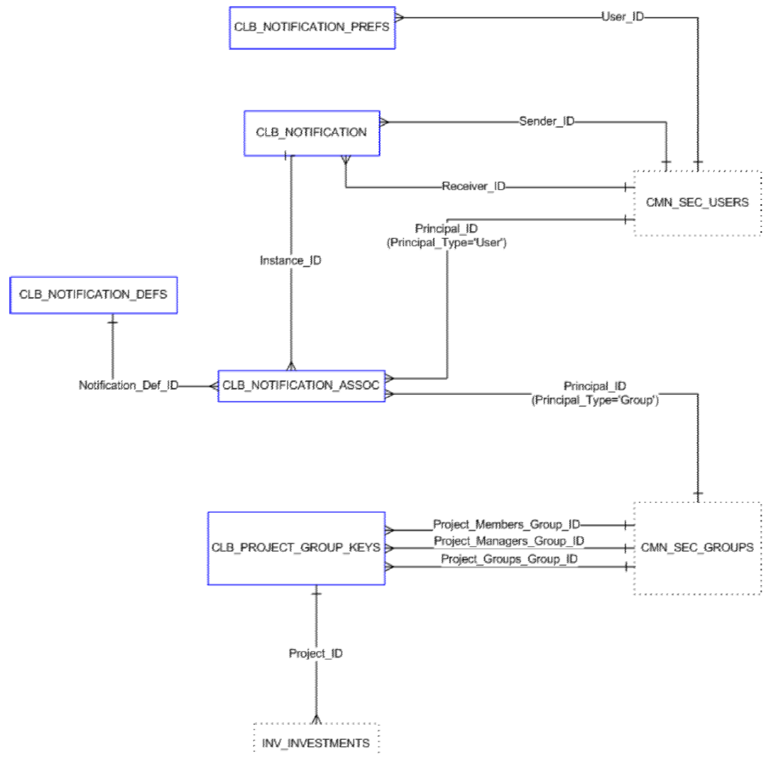
regoUniversity 2018

Notification User Settings

- CLB_NOTIFICATION_PREFS
  - Table contains the records for users who do not want notifications based on the settings in “Account Settings”. Insert records into this table in order to quickly change notification settings.

CLB_NOTIFICATION_PREFS

Notifications



CMN_CAPTIONS_NLS

- CMN_CAPTIONS_NLS
  - Table contains the name labels by language for specific values in CA PPM. This table will can connect to over 70 tables in CA PPM.

cmn_captions_nls

```
SELECT CCNOCA.NAME ATTRIBUTE_NAME
,OCA.INTERNAL_NAME ATTRIBUTE_ID
,CCNOCA.DESCRIPTION  DESCRIPTION
FROM ODF_CUSTOM_ATTRIBUTES OCA
JOIN CMN_CAPTIONS_NLS CCNOCA ON CCNOCA.PK_ID = OCA.ID
AND CCNOCA.TABLE_NAME = 'ODF_CUSTOM_ATTRIBUTES'
AND CCNOCA.LANGUAGE_CODE = 'en'
```

Custom Attributes On Objects

- ODF_CUSTOM_ATTRIBUTES
  - Table contains the custom attributes on objects in CA PPM
- ODF_OBJECTS
  - Table contains the objects and the tables associated to the objects in CA PPM
- CMN_LOOKUP_TYPES
  - Table contains the lookup values associated to the attribute

odf_custom_attributes

odf_objects

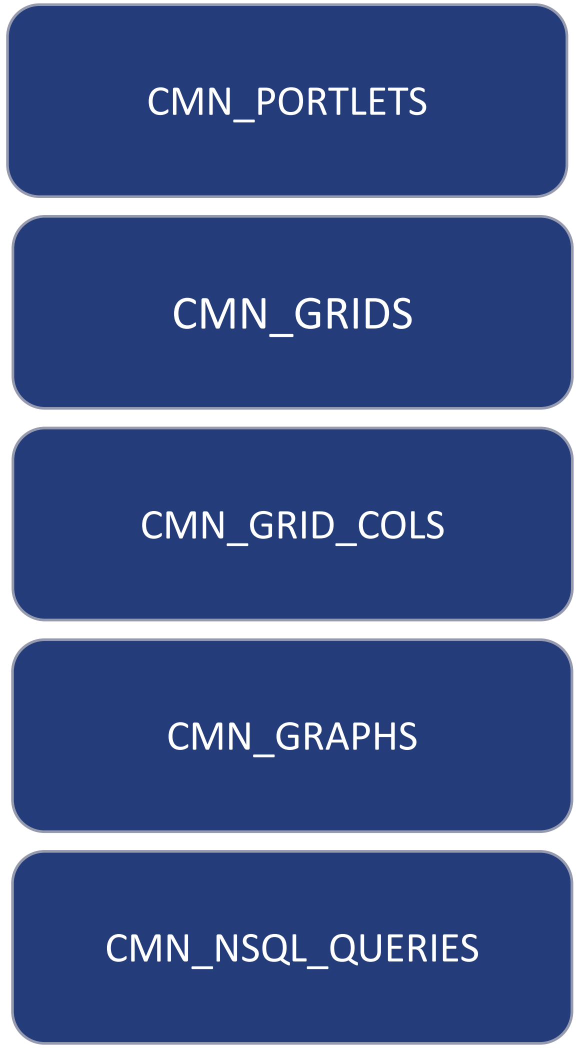
cmn_lookup_types

How to connect lookup types to the attribute table:

- LEFT OUTER JOIN (cmn_lookup_types clt
- INNER JOIN cmn_captions_nls ccnclt ON ccnclt.pk_id = clt.id
- AND ccnclt.table_name = 'CMN_LOOKUP_TYPES'
- AND ccnclt.language_code = 'en') ON clt.lookup_type = ODF_CUSTOM_ATTRIBUTES..lookup_type

Portlet Tables

- CMN_PORTLETS
  - Table contains the basic information on the portlets, such as the name, id, and query id
- CMN_GRIDS
  - Table contains the portlets with a “Grid” type
- CMN_GRID_COLS
  - Table contains the specific portlets columns that can be configured onto the portlet
- CMN_GRAPHS
  - Table contains the portlets with a “Graph” type
- CMN_NSQL_QUERIES
  - Table contains the “Query” behind the portlet
  - CMN_GRAPHS.DAL_ID = CMN_NSQL_QUERIES.ID
  - CMN_GRID.DAL_ID = CMN_NSQL_QUERIES.ID
- CMN_GG_NSQL_QUERIES
  - Table contains the base information for the query

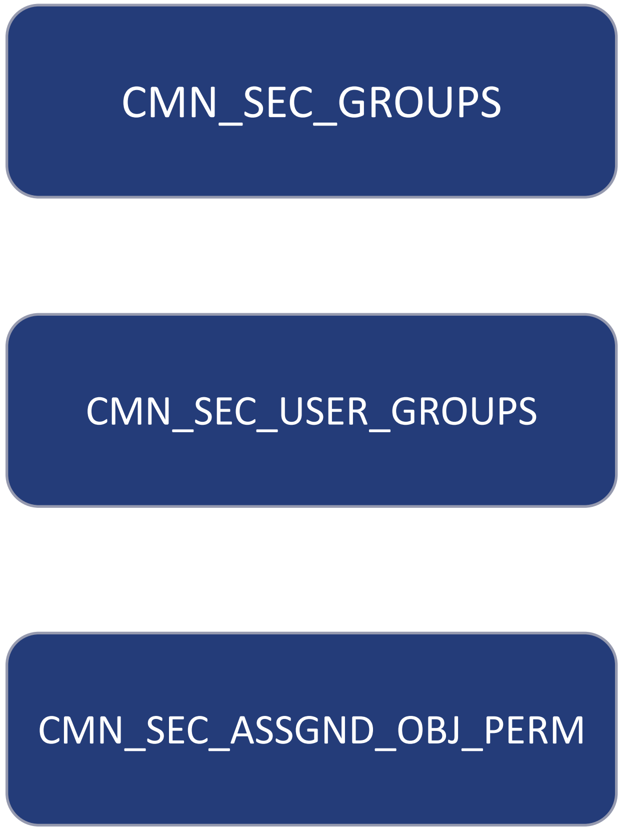


Example Portlet Query

```
SELECT P.ID PORTLET_ID
            ,P.PORTLET_CODE PORTLET_CODE
            ,PN.NAME PORTLET_NAME
            ,PT.NAME PORTLET_TYPE
            ,GQ.QUERY_CODE QUERY_CODE
            ,QN.NAME QUERY_NAME
            ,TO_CHAR(SUBSTR(Q.NSQL_TEXT, 0, 4000)) NSQL
FROM CMN_PORTLETS P
            JOIN (SELECT G.ID, G.PORTLET_ID, G.DAL_ID FROM CMN_GRIDS G WHERE G.PRINCIPAL_TYPE = 'SYSTEM'
               UNION ALL
               SELECT G.ID, G.PORTLET_ID, G.DAL_ID FROM CMN_GRAPHS G WHERE G.PRINCIPAL_TYPE = 'SYSTEM‘
            ) G ON P.ID = G.PORTLET_ID
             JOIN CMN_NSQL_QUERIES Q ON G.DAL_ID = Q.ID
             JOIN CMN_GG_NSQL_QUERIES GQ ON Q.ID = GQ.CMN_NSQL_QUERIES_ID
             JOIN CMN_CAPTIONS_NLS PN ON P.ID = PN.PK_ID AND PN.TABLE_NAME = 'CMN_PORTLETS'
                        AND PN.LANGUAGE_CODE = 'en'
             JOIN CMN_LOOKUPS_V PT ON P.PORTLET_TYPE_CODE = PT.LOOKUP_CODE
                        AND PT.LOOKUP_TYPE = 'PORTLET_TYPE' AND PT.LANGUAGE_CODE = 'en'
             JOIN CMN_CAPTIONS_NLS QN ON GQ.ID = QN.PK_ID AND QN.TABLE_NAME = 'CMN_GG_NSQL_QUERIES'
                        AND QN.LANGUAGE_CODE = 'en'
 WHERE 1=1 AND P.SOURCE = 'customer'
```

Security Tables

- CMN_SEC_GROUPS
  - Table contains the groups **AND** rights associated to the group, or user.
- CMN_SEC_USER_GROUPS
  - Table is the connecting table for groups and users. Users are assigned groups in Clarity in a many-to-many relationship.
  - GROUP_ID = CMN_SEC_GROUPS.ID
  - USER_ID = CMN_SEC_USERS.ID
- CMN_SEC_ASSGND_OBJ_PERM
  - Table connects to the CMN_SEC_GROUPS as a more detailed record of the instance rights attached to users and groups.
  - Example: Edit Timesheet for a specific user is 3 records in this table. **Access the timesheet, Read** the timesheet, **Write** the timesheet.



Activity

Primary Activity

- Get Resource and User details. Resource ID, Name, Username, User Status.

Additional Activity

- Get list of resources who have enabled Project email notification. Resource Name, Resource ID

Process and Job Logs

regoUniversity 2018

Process Logs

- BPM_RUN_PROCESSES
  - Lists all process instances that have been started
  - Includes run stats such as start/end times, status and initiator
  - process_version_id column points to the ID in the next table

- BPM_DEF_PROCESS_VERSIONS
  - One record for each saved process in PPM
  - Contains things like validation status, active/draft/on-hold
  - Does not provide the process name or code
  - process_id column points to the ID in the next table

Process Logs

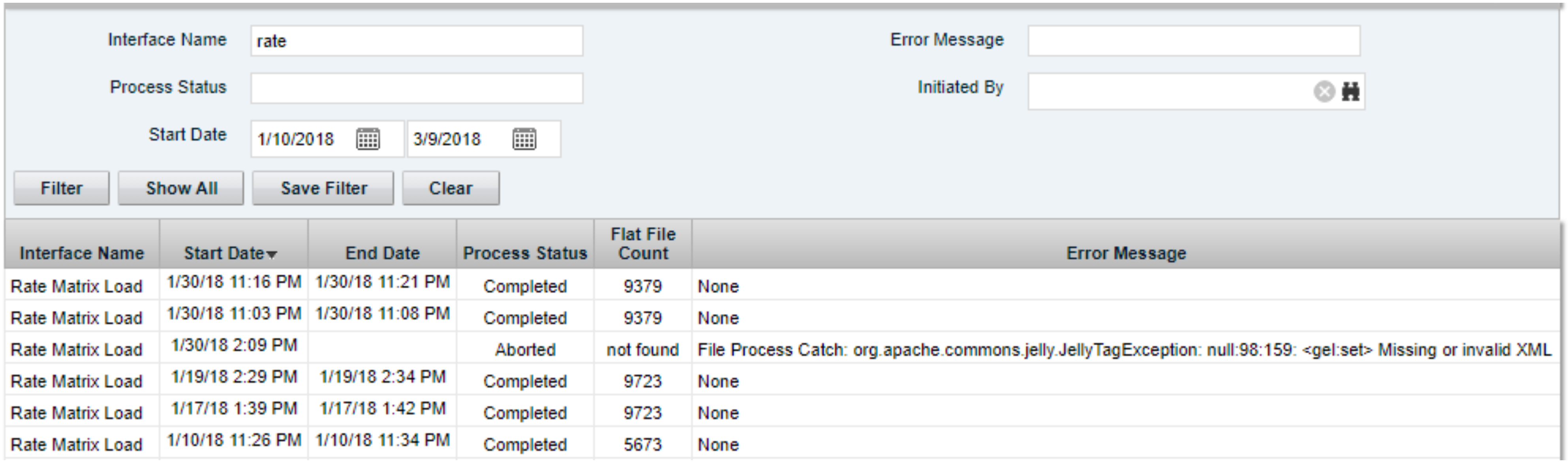
- BPM_DEF_PROCESSES
  - Contains the process code as shown in the UI (Process ID)
  - If the code is all you need you're good to go. If you want the actual process name you need to look it up in the captions table:

```
JOIN cmn_captions_nls cap ON cap.pk_id=bpm.id AND cap.table_name='BPM_DEF_PROCESSES' AND cap.language_code='en'
```

- BPM_ERRORS
  - Even though it's call "ERRORS" it actually contains all log messages generated by a process (ERROR, WARN and INFO).
  - Depending on how many processes are run, how many messages are generated and how much history is kept, this could be a large file.
  - Column process_instance_id links back to the ID in BPM_RUN_PROCESSES

Process Logs

- Using information in the preceding tables allows you to track critical processes via a query-based portlet, for example:



Interface Name	rate	Error Message	
Process Status		Initiated By	
Start Date	1/10/2018 – 3/9/2018		

Interface Name	Start Date	End Date	Process Status	Flat File Count	Error Message
Rate Matrix Load	1/30/18 11:16 PM	1/30/18 11:21 PM	Completed	9379	None
Rate Matrix Load	1/30/18 11:03 PM	1/30/18 11:08 PM	Completed	9379	None
Rate Matrix Load	1/30/18 2:09 PM		Aborted	not found	File Process Catch: org.apache.commons.jelly.JellyTagException: null:98:159: <gel:set> Missing or invalid XML
Rate Matrix Load	1/19/18 2:29 PM	1/19/18 2:34 PM	Completed	9723	None
Rate Matrix Load	1/17/18 1:39 PM	1/17/18 1:42 PM	Completed	9723	None
Rate Matrix Load	1/10/18 11:26 PM	1/10/18 11:34 PM	Completed	5673	None

- Or send the information in an automated notification script

Job Logs

- CMN_SCH_JOB_RUNS
  - Lists all jobs that have been started
  - Includes start/end times, status and the processing engine it ran on
  - job_id column points to the ID in the next table
- CMN_SCH_JOBS
  - One record for each defined job
  - Contains the descriptive job name and the job status
  - Lists all of the scheduling information (hours, days, months, etc)
- CMN_SCH_JOB_LOGS
  - The messages generated by each job execution (job_run_id)

Questions?

regoUniversity 2018

Let Rego be your guide.

Thank You For Attending regoUniversity

Instructions for PMI credits

- Access your account at pmi.org
- Click on **Certifications**
- Click on **Maintain My Certification**
- Click on **Visit CCR's** button under the **Report PDU's**
- Click on **Report PDU's**
- Click on **Course or Training**
- Class Name = **regoUniversity**
- Course Number = **Session Number**
- Date Started = **Today's Date**
- Date Completed = **Today's Date**
- Hours Completed = **1 PDU per hour of class time**
- Training classes = **Technical**
- Click on **I agree** and **Submit**

Let us know how we can improve!
Don't forget to fill out the class survey.

Phone
888.813.0444

Email
info@regouniversity.com

Website
www.regouniversity.com