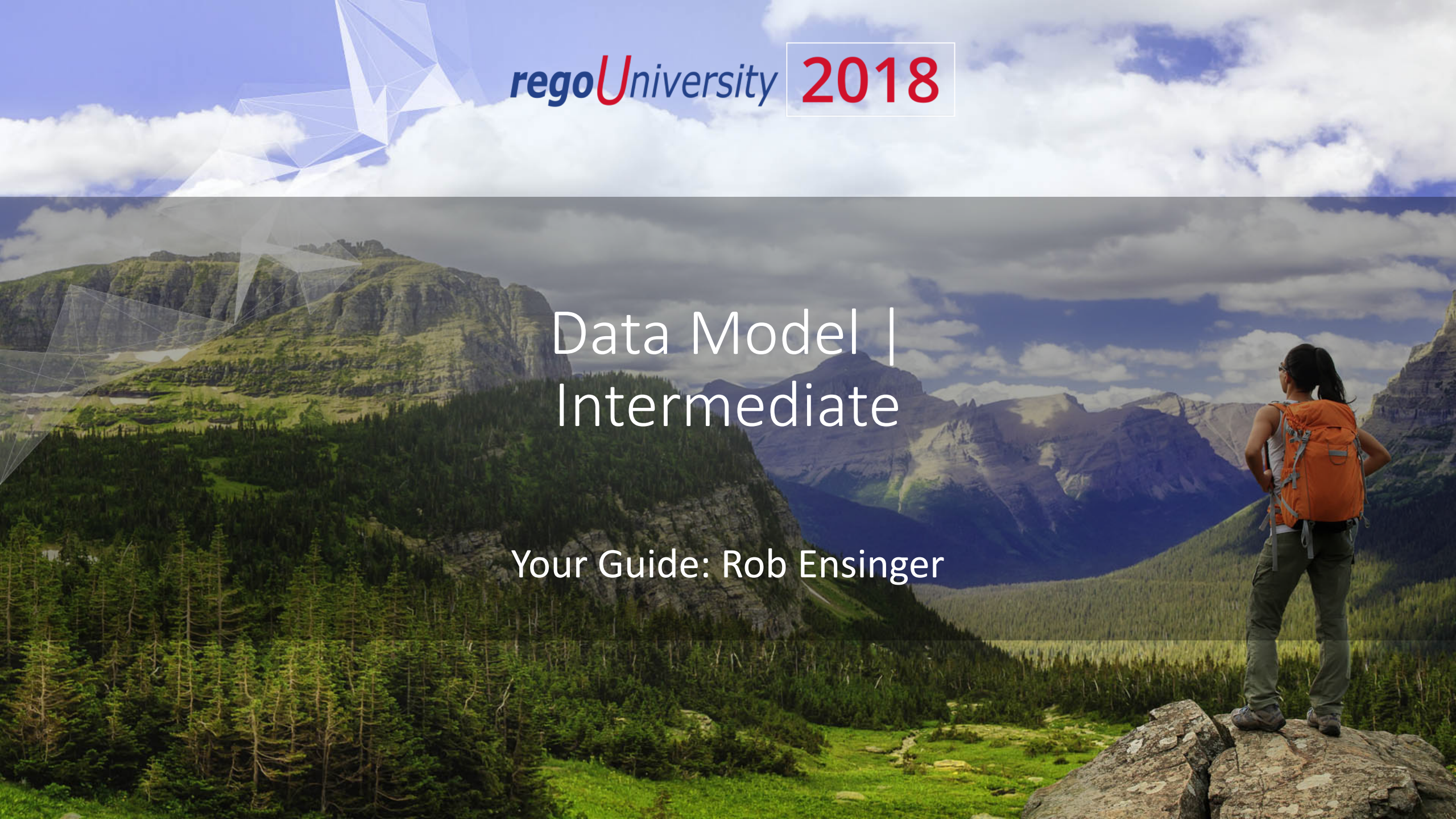
regoUniversity 2018

Data Model | Intermediate

Your Guide: Rob Ensinger



Introductions

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

Agenda

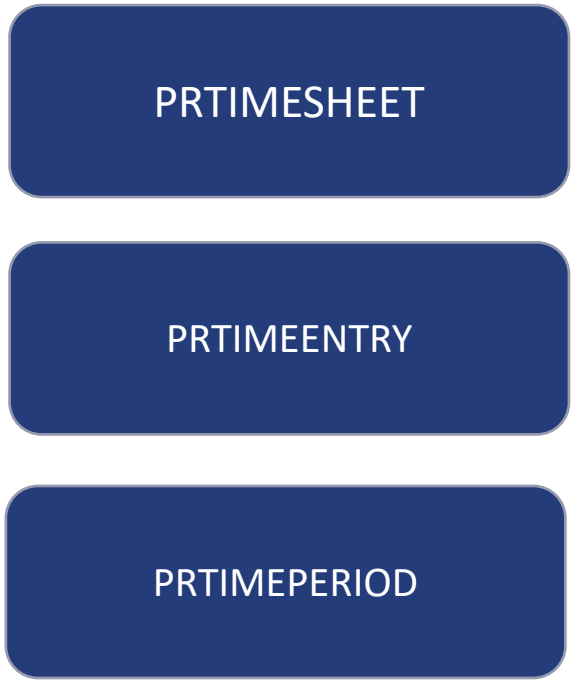
- Timesheets
- Financial Tables
  - Financial Plans
  - WIP Transactions
- Time Slices
- Lookup Tables
  - Single and Multi-Valued
- Data Warehouse

Timesheets

regoUniversity 2018

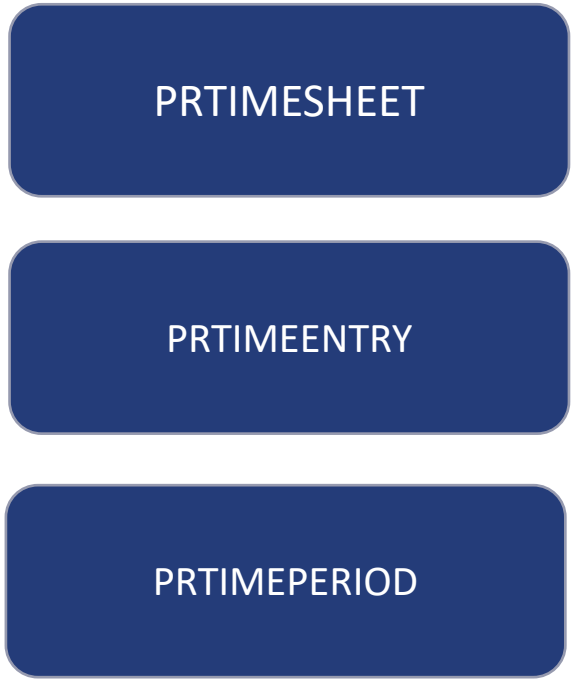
Timesheet Tables

- PRTIMESHEET
  - Contains one record for every timesheet that has been opened and saved by a user.
  - Timesheets with no activity do not exist in the table, even though their status shows “Open” in the UI.
  - This means that to identify users who have not submitted their time, 2 searches are required:
    - Find records where the status is Open, Returned, etc.
    - Discover which records are completely missing from the table.
  - Key columns:
    - PRTIMEPERIODID – indicates the timesheet time period value
    - PRRESOURCEID – the associated resource
    - PRSTATUS – numerical value (0 = Open, 4 = Posted, etc.)



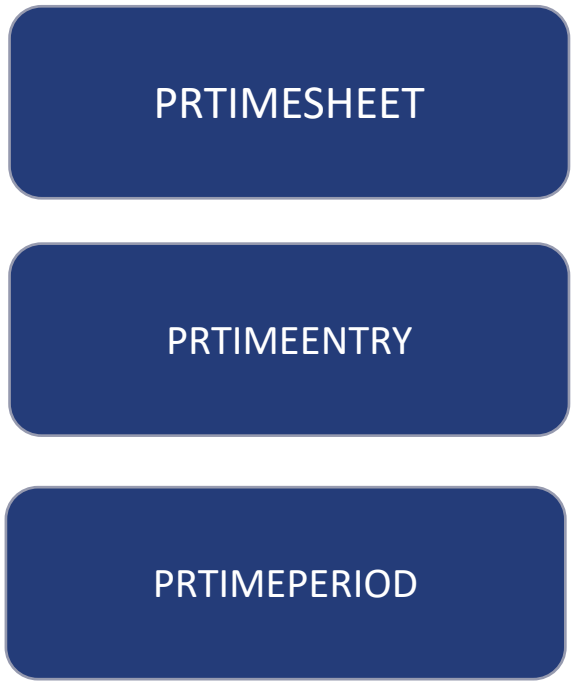
Timesheet Tables

- PRTIMEENTRY
  - The records in this table represent each line on a specific timesheet.
  - The associated Task and Investment can be determined via the following path:
    - Column PRASSIGNMENTID -> column PRID in table PRASSIGNMENT
    - Table PRASSIGNMENT.PRTASKID -> column PRID in table PRTASK
    - Table PRTASK.PRPROJECTID -> column ID in table INV_INVESTMENTS
  - Key columns:
    - PRTIMESHEETID – indicates the timesheet that this entry is tied to
    - PRASSIGNMENTID – the associated assignment as described above
    - PRACTSUM – the number of seconds entered (divide by 3600 for hours)



Timesheet Tables

- PRTIMEPERIOD
  - Contains all of the time periods that have been created in the system.
  - Each timesheet points to one of these time period entries via column PRTIMEPERIODID.
  - Key columns:
    - PRSTART – time period start date
    - PRFINISH – time period finish date
    - PRISOPEN – indicates if the time period is open for time entry
  - The above entries are typically managed through the UI (Administration -> Time Reporting Periods).



Activity

Primary Activity

1. Display all timesheets that have been posted in the system within the past 3 months.
   Resource, Time Period, Timesheet Status

Additional Activity

1. Display the timesheet and the task assigned to the timesheet with the actuals on the line.
   Resource, Time Period, Task Name, Total Actuals

Financial Tables

regoUniversity 2018

Let Rego be your guide.

Financial Tables (Plans)

- FIN_PLANS
  - Table contains the base information for financial plans. These are the base properties such as the time periods, the time period type, name, etc.

FIN_PLANS

- FIN_COST_PLAN_DETAILS
  - Table contains the financial plan details such as the groupings and the values for each grouping (e.g. type of Transaction Class or Cost Type). This table contains both the cost and budget plan details.
  - PLAN_ID = FIN_PLANS.ID

FIN_COST_PLAN_DETAILS

- FIN_BENEFIT_PLAN_DETAILS
  - Table contains benefit plan details. This includes the type and values for each group.
  - PLAN_ID = FIN_PLANS.ID

FIN_BENEFIT_PLAN_DETAILS

Financial Tables (Plans)

- FIN_FINANCIALS
  - Table contains the simple budget details. These are the attributes used when the Project/Idea is not financially open.

FIN_FINANCIALS

- ODF_SSL_CST_DTL_UNITS
  - Table contains the financial plan details slices. This slice represents the “Unit” value on the cost plan. These connect to the details on the financial cost plan.
  - PRJ_OBJECT_ID = FIN_COST_PLAN_DETAILS.ID

ODF_SSL_CST_DTL_UNITS

- ODF_SSL_CST_DTL_COST
  - Table contains the financial plan details slices. This slice represents the “Cost” value on the cost plan per the time period value assigned. These connect to the details on the financial cost plan.
  - PRJ_OBJECT_ID = FIN_COST_PLAN_DETAILS.ID

ODF_SSL_CST_DTL_COST

Financial Tables (WIP)

- PPA_WIP
  - Table contains the transaction details, such as the resource, investment, transaction class, charge code, cost type, status, etc.
  - **Note:** When querying this table, make sure to look at all transactions that have a status of 0, which indicates it's a final transaction.

PPA_WIP

- PPA_WIP_VALUES
  - Table contains the transaction cost. There are many rows for one Transaction in this table due to the currency type. It is best practice to use the "HOME" currency type when querying.

PPA_WIP_VALUES

```
WHERE PPA_WIP.TRANSNO = PPA_WIP_VALUES.TRANSNO
AND PPA_WIP_VALUES.CURRENCY_TYPE = 'HOME'
AND PPA_WIP.STATUS = 0
```

Financial Tables

- ENTITY
  - Table contains the entity's basic information.

- BIZ_COM_PERIODS
  - Table contains the fiscal time periods created on the entity. These are the time periods that are used to create details on financial plans.
  - It is important to use these time periods as they hold the start and finish dates for fiscal time periods and not calendar.
  - BIZ_COM_PERIODS.ENTITY = ENTITY.ID

ENTITY

BIZ_COM_PERIODS

Quick Cleanup Of Invalid Transactions

- IMP_TRANSACTIONIMPORT
  - Table contains the transactions that are current pending posting. New transactions and transactions that were invalid are stored as records here.
- IMP_TRANSACTIONIMPORTERRORS
  - Table contains the details on the invalid transaction errors.
  - IMP_TRANSACTIONIMPORT.errorcode = IMP_TRANSACTIONIMPORTERRORS.errorcode

IMP_TRANSACTIONIMPORT

IMP_TRANSACTIONIMPORTERRORS

These tables allow for the user to identify the error and create a mass fix based on the error found. Common errors include Input Type Code invalid or not found. With this information you can use the Resource code found on the IMP_TRANSACTIONIMPORT table and UPDATE the resources input type.

Activity

Primary Activity

1. Display all actual transactions within the 2017 fiscal time period by fiscal month with cost type, transaction class, and amount

Additional Activity

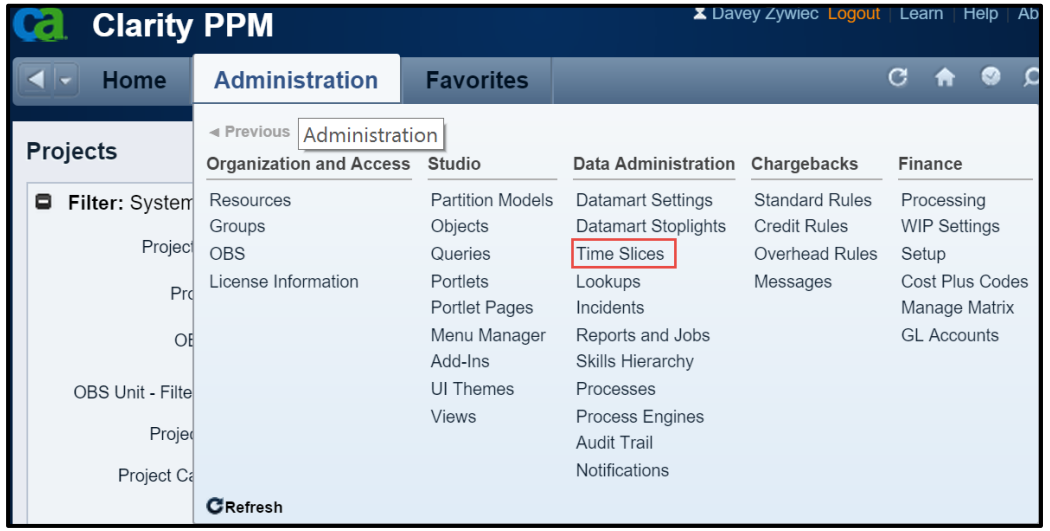
1. Display the current plan of record cost plan amounts in addition to the actual amounts

Time Slices

regoUniversity 2018

Time Slice Tables

- Time and Hour metrics are stored as blobs and are unreadable in the day to day production tables
- Time Slice tables open a window to this data for viewing
- These views allow PPM to group data into Weeks, Months, Quarters etc…
- This grouping allows for more efficient queries
- Need to tell it what slice you are going after
- Keep your daily slices to a minimum



Time Slice Tables

- PRJ_BLB_SLICEREQUESTS
  - Contains the Time Slices defined by the PPM Administrator
  - Includes the type of slice it is (actuals, allocation, etc) and the associated object (resource, investment, etc)
  - Also specifies for each request the period type (weekly, monthly, etc) and the number of periods to maintain.
  - The ID column in this table is the key to the actual sliced data in the next table (PRJ_BLB_SLICES.slice_request_id).
  - The ID column values are also shown on the Time Slices admin screen within the UI.

Time Slice Tables

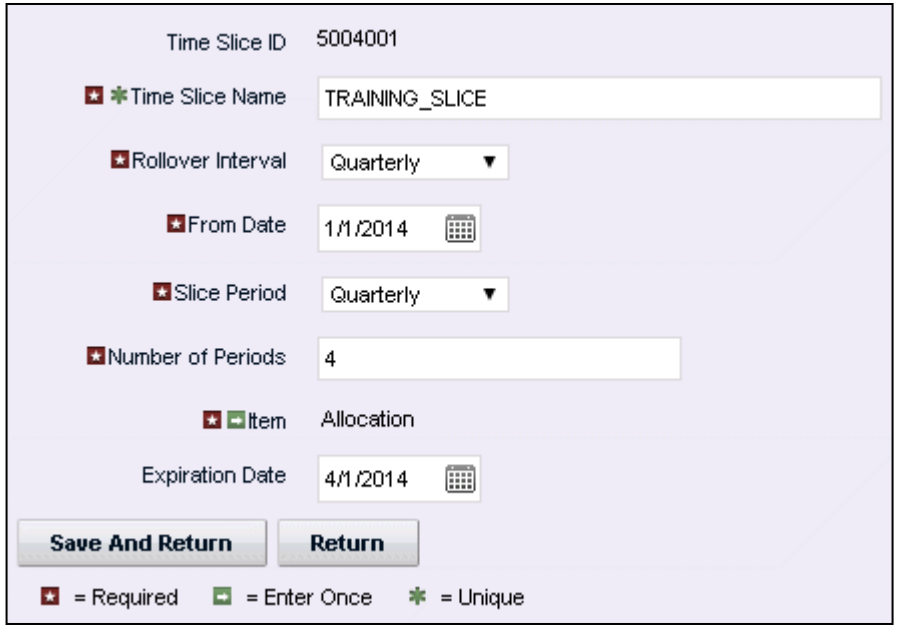
- PRJ_BLB_SLICES
  - This is a very large table (often tens of millions of rows) with just a handful of columns.
  - To pull the right data you need to know two things:
    - The appropriate slice request (slice_request_id)
    - The associated PPM table for that request (e.g. PRJ_RESOURCES, PRTEAM, PRTIMEENTRY, PRASSIGNMENT, etc).
  - The column prj_object_id points to a specific entry in one of the above tables.
  - If your query returns no rows you may be joining to the wrong PPM core table.

    Some examples:
    - xxxRESOURCEACTCURVE joins to PRTIMEENTRY
    - xxxRESOURCEAVAILCURVE joins to PRJ_RESOURCES
  - Just as with timesheets, the numerical value of the slice column is in seconds.

PRJ_BLB_SLICEREQUESTS

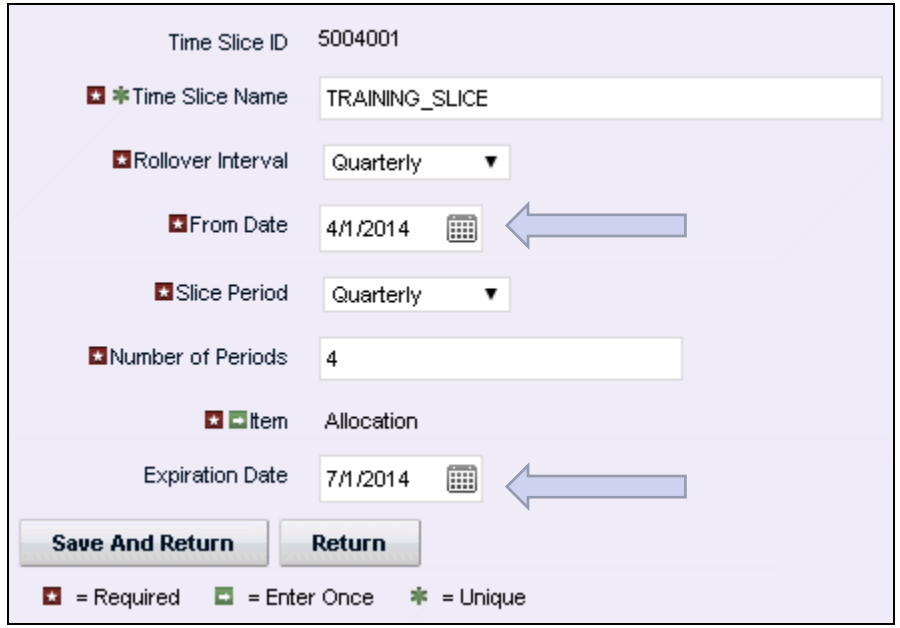
PRJ_BLB_SLICES

Time Slice Tables



Slice Range																									
Qtr			Qtr			Qtr			Qtr																
Jan-14	Feb-14	Mar-14	Apr-14	May-14	Jun-14	Jul-14	Aug-14	Sep-14	Oct-14	Nov-14	Dec-14	Jan-15	Feb-15	Mar-15	Apr-15	May-15	Jun-15	Jul-15	Aug-15	Sep-15	Oct-15	Nov-15	Dec-15	Jan-16	Feb-16

Time Slice Tables



Rollover

			Slice Range																					
			Qtr			Qtr			Qtr			Qtr													
Jan-14	Feb-14	Mar-14	Apr-14	May-14	Jun-14	Jul-14	Aug-14	Sep-14	Oct-14	Nov-14	Dec-14	Jan-15	Feb-15	Mar-15	Apr-15	May-15	Jun-15	Jul-15	Aug-15	Sep-15	Oct-15	Nov-15	Dec-15	Jan-16	Feb-16

Activity

Primary Activity

1. Display the time for a month for each project. Project Name, Project ID, Actuals

Additional Activity

1. Display the Resource's actuals by project by month, Resource, Project, Actuals, Month

Lookup Tables

regoUniversity 2018

Lookup Tables

- CMN_LOOKUPS
  - This table contains the base data for a lookup and its values.
- CMN_LOOKUPS_V
  - This table contains the base data for a lookup and its values but it also includes the labels for each value.
- ODF_MULTI_VALUED_LOOKUPS
  - This table contains the object instance's attribute, value_id (to connect to the lookups table) for each value stored in the attribute.
  - Example: My attribute for IT Council on a project has 3 names. ODF_MULTI_VALUED_LOOKUPS would contain a record with the project instance, for the attribute "IT Council", for each name associated.

CMN_LOOKUPS

CMN_LOOKUPS_V

ODF_MULTI_VALUED_LOOKUPS

```
SELECT INVI.NAME, L1.NAME
FROM ODF_MULTI_VALUED_LOOKUPS MVL1
JOIN CMN_LOOKUPS_V L1 ON MVL1.VALUE = L1.LOOKUP_CODE AND L1.LANGUAGE_CODE = 'en'
JOIN INV_INVESTMENTS INVI ON INVI.ID = MVL1.PK_ID AND INVI.ODF_OBJECT_CODE = 'project' AND INVI.CODE = 'DZ_TEST'
WHERE MVL1.OBJECT = 'PROJECT' AND MVL1.ATTRIBUTE = 'rego_it_council';
```

Multi-Value Lookup Aggregate Example

Oracle: LISTAGG function

```
SELECT INVI.NAME PROJECT_NAME
, MVL.MVL_VALUE
FROM INV_INVESTMENTS INVI
JOIN (SELECT MVL.OBJECT, MVL.PK_ID, LISTAGG(V.NAME, ', ')
WITHIN GROUP (ORDER BY V.NAME) MVL_VALUE
    FROM ODF_MULTI_VALUED_LOOKUPS MVL
    JOIN CMN_LOOKUPS_V V ON MVL.VALUE =
V.LOOKUP_CODE AND V.LOOKUP_TYPE =
'DLM_MVL_EXAMPLE' AND V.LANGUAGE_CODE = 'en'
    WHERE MVL.ATTRIBUTE = 'dlm_values'
    GROUP BY MVL.OBJECT, MVL.PK_ID) MVL ON
INVI.ODF_OBJECT_CODE = MVL.OBJECT AND INVI.ID =
MVL.PK_ID
```

MSQL: STUFF and XML PATH() function

```
select distinct INVI.NAME,
 STUFF((SELECT DISTINCT "" + V.NAME MVL_VALUE
    FROM ODF_MULTI_VALUED_LOOKUPS MVL
    JOIN CMN_LOOKUPS_V V ON MVL.VALUE = V.LOOKUP_CODE AND
V.LOOKUP_TYPE = 'DLM_MVL_EXAMPLE' AND V.LANGUAGE_CODE =
'en'
    WHERE MVL.ATTRIBUTE = 'dlm_values'
            AND INVI.ID = PK_ID
    GROUP BY MVL.OBJECT, MVL.PK_ID
        FOR XML PATH(''), TYPE
        ).value('.', 'NVARCHAR(MAX)')
     ,1,0,'') data
from INV_INVESTMENTS INVI
```

Activity

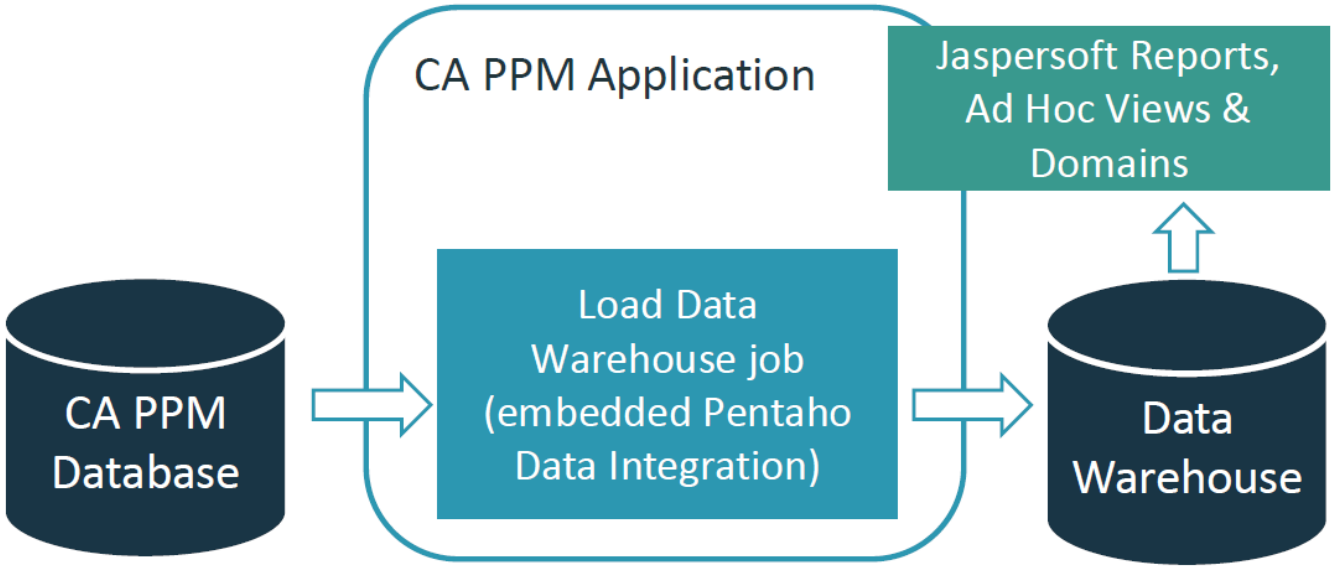
- Primary Activity
  - Get List of investments, its type and associated OBS. Project ID, Name, Object Type, OBS Full Path

- Additional Activity
  - Get List of Projects and associated OBS Path on or under a specific OBS node. (Try to use OBS UNIT BY MODE).

Data Warehouse

regoUniversity 2018

Data Architecture



- Lightweight, drag and drop business user reporting capability
- Out of the box reports and domains for Investments, Resources, Financials and Timesheets.

- The Data Warehouse is modeled on a STAR schema, with Dimensions covering the major areas in CA PPM and their associated Facts.

- Pentaho Data Integrator is embedded within CA PPM. The data transformation and load runs as a CA PPM job.

Data Warehouse Key Benefits

- Data Model is easy to query
- Having the data warehouse on separate server minimizes the impact on the CA PPM Application
- The Data Warehouse carries keys and descriptive values in the dimension tables so that fewer joins are required. Facts are combined into summary and period tables.
- Similar tables are grouped together by the table prefix, and the names are descriptive.
- The ability to query the data warehouse and CA PPM transactional database within the same report is seamless
- Report output is returned faster using the data warehouse

Data Warehouse “DWH” Prefix Standards

Prefix	Table
DWH_CFG	Configuration tables used to supply the DWH with log and audit information
DWH_CMN	Common database objects used across most areas
DWH_CMP	Company database objects
DWH_FIN	Financial management database objects
DWH_INV	INV Investment management database objects
DWH_LKP	Lookup database objects
DWH_META	Meta data tables that help determine the DW structure
DWH_ODF	Customer specific database objects
DWH_PFM	Portfolio management database objects
DWH_RES	Resource management database objects
DWH_RIM	Risk and Issue management database objects
DWH_TME	Time management database objects
DWH_X	Internal database objects used to help populate the DW fact tables

Data Warehouse Content

Data Warehouse Content		
Change Request Management	Issue Management	WBS Structure
Exchange Rates	OBS Hierarchy	WIP Transactions
Financial Benefit Plan	Portfolio (High Level)	Facts by Weekly/Monthly/Fiscal
Financial Budget/Cost Plans	Resource Assignments	Summary Facts
Investment – Applications	Resources	All Associated Lookups
Investment – Assets	Resource User Security	Customer Specific Attributes
Investment – Ideas	Risk Management	TSV Values
Investment – Other Work	Team Allocations	Summary Facts
Financial Budget/Cost Plans	Resource Assignments	Summary Investment EV Data
Investment – Products	Time Entry	Current Baseline Data
Investment – Projects	Time Entry Notes	PMO Accelerator
Investment – Services	Time Sheets	DB Link for Missing Data
Investment – User Security	Time Sheet Notes	New Cost Slices – ETC/Allocations

Example

Team Details SQL:

Simple SQL
Very few JOINS
No Lookup table JOINS

```
SELECT I.INVESTMENT_MANAGER,
                I.INVESTMENT_NAME,
                T.RESOURCE_NAME,
                T.ROLE_NAME,
                TL.BOOKING_STATUS,
                TL.REQUEST_STATUS,
                P.PERIOD_START_DATE,
                TF.ALLOC_HOURS,
                TF.ALLOC_COST
FROM DWH_INV_TEAM T
                INNER JOIN DWH_INV_TEAM_LN TL ON T.TEAM_KEY = TL.TEAM_KEY
                INNER JOIN DWH_INV_INVESTMENT I ON T.INVESTMENT_KEY = I.INVESTMENT_KEY
                INNER JOIN DWH_INV_TEAM_PERIOD_FACTS TF ON T.TEAM_KEY = TF.TEAM_KEY
                INNER JOIN DWH_CMN_PERIOD P ON TF.PERIOD_KEY = P.PERIOD_KEY
WHERE SYSDATE BETWEEN P.YEAR_START_DATE AND P.YEAR_END_DATE
                AND P.PERIOD_TYPE_KEY = 'MONTHLY'
                AND TL.LANGUAGE_CODE = 'en';
```

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