

Power BI Dashboards | Beginner

Your Guides: David Zywiec and Sankhadeep Dhar

Course Outline

Overview

- What is Power BI
- Power BI Components
- Data Flow

Connecting to Data Sources

- Power BI Service Navigation
- Options to Connect to Sources
- Creating a Data Connection

Reports and Dashboards

- Dashboard Components
- Options to Share Reports and Dashboards

Creating a Report

- Power BI Desktop Navigation
- Creating a Simple Report Using Power BI Application
- Examples of Using Custom Visuals

Data Import Options

- Connecting to ODATA and Oracle Server and Extracting Data
- DirectQuery vs. Import

Data Transformation

- Data Transformation Components
- Data Modeling

Data Transformation - Advanced

- Creating your own M script for your next advanced report
- M Script Examples

DAX in Power BI

- Introduction to DAX Language
- When to use M vs. DAX
- Examples



Course Goal: To provide an overview of Power BI functionality in conjunction with CA PPM data.

Course Objectives: You will learn how to:

- Understand How Power BI Works with CA PPM Data
- Connect to Data Sources
- Create Reports and Dashboards
- Create and Modify M and DAX Scripts





Overview

- What is Power BI
- Power BI Components
- Data Flow

Power BI is a dashboard tool from Microsoft that lets you connect to variety of data sources and create visually stunning and interactive dashboards.





Overview

Power BI has 3 main components:



Overview

Data Flow





Connecting to Data Sources

- Power BI Service Navigation
- Options to Connect to Sources
- Creating a Data Connection



Power BI Service Navigation





Connecting to Data Sources

Options to Connect to Sources

- SaaS solutions
- Azure services
- Custom content packs
- Custom connections
- On-premise SAS data
- Excel workbooks
- CSV files
- Power BI Desktop files

Content Pack Library Import or Connect to Data My organization Services Files Databases Connect to live data in Browse content packs Choose content packs Bring in your reports, that other people in your from online services that workbooks, or data from Azure SQL Database and organization have Excel, Power BI Desktop you use. more. or CSV files. published. 7 7 7 Get Get Get 7 Get

Get Data

Need more guidance? Try this tutorial or watch a video

Creating a Data Connection

Import or Connect to Data

Files

Get

Bring in your reports, workbooks, or data from Excel, Power BI Desktop or CSV files.

7

Databases

Get

Connect to live data in Azure SQL Database and more.

 \overline{A}

Reports and Dashboards

- Dashboard Components
- Options to Share Reports and Dashboards

Dashboard Components

 Dashboard are high level view of some key KPI of one or more reports. Reports are pin to live page which then adds it to a new or existing Dashboard

Reports and Dashboards

Options to Share Reports and Dashboards

- Publish reports to the web
- Share dashboards
- Duplicate and modify shared dashboards (personal versions)
- Export to multiple formats
- Print
- QR code

Reports and Dashboards

Steps to Share

Publish to Web

- Do not use this option as it will allow access to dashboard to public Share
 - Grant/Revoke access to colleagues in the organization
 - It will warn you if you have entered an email address outside the organization

Create Workspaces (Pro only)

- Public or Private groups
- Edit or View only groups

Create Apps (Pro Only)

- Entire Organization
- Specific Individual or Groups

Creating a Report

- Power BI Desktop Navigation
- Creating a Simple Report Using Power BI Application
- Examples of Using Custom Visuals





Power BI Desktop Navigation

| | Untitled - Power BI Desktop iew Modeling | | | | | | | — 🗆 Sign in | × ~ (|
|---------------------------------|---|--|--------|----------------|-------------------------|---------------|---------------------------------------|-----------------|----------|
| Cut Copy * Sormat Painter | Get Recent Enter Data + Sources + Data | Solution Partner Templates Showcase | | From From File | Manage Relationships | 👘 New Measure | | | |
| Clipboard | External data | Resources | Insert | Custom visuals | Relationships | Calculations | Share | | |
| | | | | | | | Visualizations > | Fields | |
| | | | | | | | I I I I I I I I I I I I I I I I I I I | <i>P</i> Search | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | _ | | | |
| | | | | | | | | | |
| ∢ → Page 1 | + | | | | | | | | |

Dashboard Components

Chart Visualizations

- Bar chart
- Column chart
- Line chart
- Are chart
- Pie chart
- Combo chart
- Bubble chart
- Scatter chart
- Donut chart
- Gauge chart
- Funnel chart
- Waterfall chart



Creating a Report



OBS Type

Creating a report





Visuals



Examples of Custom Visuals





Data Import Options

- DirectQuery vs. Import
- Connecting to ODATA and Oracle Server and Extracting Data

Data Import Options



Connecting to ODATA and Oracle Server and Extracting Data







Data Refresh – On Premise vs On Demand CA PPM







Data Import Options

<u>**</u>

DirectQuery vs. Import [Oracle]

Data Connectivity mode 🛈

Import

DirectQuery



Data Import Options



Connecting to a Source





Data Transformation

- Data Transformation Components
- Data Modeling

Data Transformation

Data Transformation Components

- Use first row as header
- Split the field or concatenate fields
- Replace values null with blanks
- Group data
- Pivot/unpivot columns
- Aggregate functions
- Format columns



Data Transformation

Data Modeling

- Import multiple tab from Excel or Create multiple queries
- Join the tables





Data Transformation -Advanced

- Creating your Own M Script
- M Script Examples

Data Transformation - Advanced

Creating your own M script for your next advanced report





Data Transformation - Advanced



M Script Examples

| | | Advanced Editor | — |
|---|--|---|--------------------------------------|
| | | DWH_INV_INVESTMENT | |
| | Advanced Editor OVERALL_RISK_ISSUE_CR | <pre>let source = Oracle.Database("54.215.191.19/clarity", (HierarchicalNavigation=true]), PPM_DHH = Source([Schema="PPM_DHH"])(Data), DHH_DIN_INVESTMENT = PPM_DHH(Inw_INVESTMENT"])(Data), DHH_DIN_INVESTMENT_RM = Table.SelectColumns(DHH_INV_INVESTMENT,("INVESTMENT_KEY", "INVESTMENT_INVE", "SCHEDULE_START", "SCHEDULE_FINISH", "BASELIVE START",</pre> | |
| | let | "BASELINE_FINISH" } | |
| | Source = Oracle.Database("54.215.191.19/clarity", [HierarchicalHavigation+true]), PPM_DMH = Source([Schema="PM_DMH*][Data], DMH_RIM_CHANGE_REQUEST = PPM_DMH*[[Hame="DMH_RIM_CHANGE_REQUEST"]][Data], MERGED_QUERY(CR = Table.MestEdDaIn(DMH_RIM_CHANGE_REQUEST,("INVESTMENT,KeY", DMH_INV_INVESTMENT,("INVESTMENT,KeY"),"DMH_I REMOVED_DTHER_COLUMB(CR = Table.SelectColumns(MERGED_QUERY_CR,"INVESTMENT,KeY", "CHANGE_REQUEST_ID", "RIM_PRIORITY,KEY" | | × |
| ATEST_STATUS_REPORT | RENAMED_COLUMM (R = Table.RenameColumns(REMOVED_OTHER_COLUMNS_CE,{(("CHANGE_REQUEST_LO", "TO"})), ADDED_UISTOM(R = Table.AddColumn(RENAMED_COLUMN]CR, "DATA_TYPE", each "Change Request"), OVERALL_CR = ADDED_CUSTOM_CR, | DWH_INV_STATUS_REPORT_LAST6 | .{"INVESTMENT KE |
| Source = Oracle.Database("54.215.191.19/clarity", [Hierarc PPM_DMH = Source([Schema="PPM_DMH"])[Data], DMH_INV_STATUS_REPORT = PPM_DMH("Name="DMH_INV_STATUS_REP MERGED_QUERY = Table.NestedJoin(DMH_INV_STATUS_REPORT1,{"I | RENAMED_COLUMN_ISSUE = Table.RenameColumns(REMOVED_OTHER_COLUMNS_ISSUE,{{"ISSUE_ID", "ID"}}), | <pre>let let evaluation=consciences("54.215.191.19/clarity", [HierarchicalNavigation=true]), PPM_DMH = Source([Schema="PPM_DMH"])[Data], DMM_IMV_STATUS_REPORTI = PPM_DMH"[INw="TMM_ILTW_STATUS_REPORT"]][Data],</pre> | 3",JoinKind.Inne |
| <pre>//Group the data Grouped = Table.Group(MERGED_QUERY, {"INVESTMENT_KEY"}, {{ //Declare a function that adds a rank to a table</pre> | ADDED_CUSTOM_ISSUE = Table.AddColumm(RENAMED_COLUNN_ISSUE, "DATA_TYPE", each "Issue"), OVERALL_ISSUE = ADDED_CUSTOM_ISSUE, DWH_RIJM_RISK = PPM_DNH{[Name=TDMH_RIJM_RISK"])[Data], MERGED_QUERY_RISK = Table.NestedJoin(DMH_RIJM_RISK,{'INVESTMENT_KEY''),DMH_INV_INVESTMENT_("INVESTMENT_KEY"),"DMH_INV_INVEST | #"Added Custom" - Table.AddColumn(OML_TMV_STATUS_REPORTI, "OVERALL_STATUS_GREEH", each 40), #"Added CustomI" - Table.AddColumn("Added CustomI", "OVERALL_STATUS_VELLOW", each 40), #"Added CustomI" - Table.AddColumn("Added CustomI", "OVERALL_STATUS_RED", each 100), #"Added Conditional Column" - Table.AddColumn("Added CustomI", "OVERALL_STATUS_USTOM", each 16 [OVERALL_STATUS] < 40 then 10 else if [| _INV_SUMMARY_FAC JAL_TOTAL_HOURS" |
| <pre>RankFunction = (tabletorank as table) as table => let SortRows = Table.Sort(tabletorank,{{"CLARITY_C AddIndex = Table.AddIndexColumn(SortRows, "Ran in</pre> | REMOVED_OTHER_COLLUMNS_RISK = Table.SelectColumns(MERGED_QUERY_RISK,{"INVESTMENT_KEY", "RISK_ID", "RIM_PRIORITY_KEY", "RIM RENAMED_COLLUMN_RISK = Table.RenameColumns(REMOVED_OTHER_COLUMNS_RISK,{{"RISK_ID", "ID"}}), | | |
| AddIndex, //Apply the function to AllRows column AddedRank = Table.TransformColumns(Grouped, {"AllRows" //Exand the necessary columns | <pre>#*Appended Query" = Table.combine((OMERALL_CR, OMERALL_TSK, OMERALL_TSK)); #*Added Conditional Column" = Table.AddColumn(#*Appended Query", "PEIQOITY", each if [RIM_PRIORITY_KEY] = "HIGH" then "Hi #*Added Conditional Column1" = Table.AddColumn(#*Added Conditional Column", "STATUS", each if [RIM_PRIORITY_KEY] = "OPEN" t #*Redeved Columns = Table.RemoveColumns(#*Added Conditional Column", "STATUS", each if [RIM_PRIORITY_KEY] = "OPEN" t #*Remove Columns = Table.RemoveColumns(#*Added Conditional Column", "STATUS", each if [RIM_PRIORITY_KEY] = "OPEN" t #*RemoveColumns = Table.RemoveColumns(#*Added Conditional Column", "STATUS", each if [RIM_PRIORITY_KEY] = "OPEN" t #*RemoveColumns = Table.RemoveColumns(#*Added Conditional Column", "STATUS", each if [RIM_PRIORITY_KEY] = "OPEN" t #*RemoveColumns = Table.RemoveColumns(#*Added Conditional Column", "STATUS", each if [RIM_PRIORITY_KEY] = "OPEN" t #*RemoveColumns = Table.RemoveColumns(#*Added Conditional Column", "STATUS", each if [RIM_PRIORITY_KEY] = "OPEN" t #*RemoveColumns = Table.RemoveColumns(#*Added Conditional Column", "STATUS", each if [RIM_PRIORITY_KEY] = "OPEN" t #*RemoveColumns = Table.RemoveColumns(#*Added Conditional Column", "STATUS", each if [RIM_PRIORITY_KEY] = "OPEN" t #*RemoveColumns = Table.RemoveColumns(#*Added Conditional Column", "STATUS", "EXTURNES, "Added Conditional Column", "STATUS", "EXTURNES, "Added Conditional Column", "STATUS", "REMOVECOLUMNS", REMOVECOLUMNS", REMOVECOLUMNS, "REMOVECOLUMNS", "REMOVECOLUMNS", "REMOVECOLUMNS", "REMOVECOLUMNS", "REMOVECOLUMNS", "REMOVECOLUMNS", "REMOVECOLUMNS", REMOVECOLUMNS", "REMOVECOLUMNS", "REMOVECOLUMNS", "REMOVECOLUMNS</pre> | <pre>//Group the data Grouped = Table.Group(MERGED_QUERY, {"INVESTMENT_KEY"}, {{"AllRows", each _, type table}}),</pre> | > |
| Expanded = Table.ExpandTableColumn(AddedRank, "AllRows //Filter for Rank=1 FinalOutput = Table.SelectRows(Expanded, each ([Rank] = 1) #Removed Columns" = Table.RemoveColumns(FinalOutput,{"DNH | in #"Removed Columns" | <pre>//Declare a function that adds a rank to a table RankFunction = (tabletorank as table) as table >> let SortRows = Table.Sort(tabletorank,{{"CLARITY_CREATED_DATE", Order.Descending}}),</pre> | Done Cancel |
| <pre>#"Added Custom" = Table.AddColumn(#"Removed Columns", "OVE #"Added Custom1" = Table.AddColumn(#"Added Custom", "OVER #"Added Custom2" = Table.AddColumn(#"Added Custom1", "OVER #"Added Conditional Column" = Table.AddColumn(#"Added Cust</pre> | | AddIndex = Table.AddIndexColumn(SortRows, "Ramk", 1, 1) in AddIndex, //Aoply the function to AllRows column | |
| <pre>#"Added Conditional Column1" = Table.AddColumn(#"Added Con #"Added Conditional Column2" = Table.AddColumn(#"Added #"Added Conditional Column3" = Table.AddColumn(#"Added</pre> | | Addedamk = Table.TransformColumns(Grouped, ("AllRows", each RankFunction(_))), //Exapnd the necessary Columns Expanded = Table.ExpandTableColumn(AddedRank, "AllRows", ("CHANGE_EXPLANATION", "CLARITY_CREATED_DATE", "CLARITY_STATUS_REPORT_KEY", # "Changed Type1" = Table.TransformColumnTypes(Expanded,("COVERALL STATUS, Upe number), ("OVERALL STATUS_CUSTOM", type number), ("OVERALL STATUS_CUSTOM", "OVERALL STATUS_CUSTOM", "ALL STATUS_CU | |
| <pre>#"Added Conditional Column4" = Table.AddColumn(#"Added </pre> | Conditional Column3", "Effort Text", each if [SR_COST_EFFORT_STATUS_KEY] = 10 | //Filter for Rank=1 | |
| No syntax errors have been detected. | | ✓ No syntax errors have been detected. | |
| | Done Cancel | the sympa ends have been deceded. Dane Cancel | |
| | | | |
| | | | |
| | | | |



- Introduction to DAX Language
- When to use M vs. DAX
- Examples

Introduction to DAX Language

DAX: Data Analysis eXpression Language

Unlike M, DAX is an expression language and mostly uses formulas similar to **Excel**.

When to use M vs. DAX

- You will use M mostly for data transformation joining/union tables, adding columns, pivot/unpivot
- You will use DAX mostly for writing analytical queries like sum, average, running total etc.
- With M you can fetch data from source and add it to PBI data model. DAX works with data that has been already added to PBI data model.
- Because DAX and M have been built independently (M is part of power query suite and DAX is part of SSAS/power pivot suite), identical things can be achieved using both – join/union, custom column etc. My approach is to use DAX for items which cannot be easily achieved using M – example average, sum, calculating %.

There is a DAX studio which can be used to write and test DAX expressions before using in PBI. Download it <u>here</u>.



Examples

| External data resources insert custom visuals relationships calculations share | |
|--|--------|
| Timesheet_Perc = ADDCOLUMNS(CALCULATETABLE(SUMMARIZE(TimeSheet , TimeSheet[RESOURCE_KEY], TimeSheet[RESOURCE_NAME], "Perc", (IF(ISBLANK(COUNT(TimeSheet[TIMESHEET_KEY])),0,COUNT(TimeSheet[TIMESHEET_KEY]))/4) | \sim |
| *100), FILTER(TimeSheet, OR(TimeSheet[TIMESHEET_STATUS_KEY]=4, TimeSheet[TIMESHEET_STATUS_KEY]=1)), FILTER(TimeSheet, TimeSheet[PERIOD_FINISH_DATE]>= TimeSheet[CLARITY_UPDATED_DATE]), filter(TimeSheet, TimeSheet, TimeShee | |
| AND(TimeSheet[PERIOD_START_DATE] <= TODAY(), TimeSheet[PERIOD_START_DATE] >= TODAY()-28))), "Band1", 33, "Band2", 66, "End Value", 100) | |

NumberOfIssues = countrows(RELATEDTABLE(Issue))+0



Example: Calculate % of timesheets submitted on time for last 4 weeks

TIMESHEET data is added to model.

Created another table using DAX which contains the percentage data:

```
ADDCOLUMNS (
 CALCULATETABLE (
    SUMMARIZE (
     TimeSheet,
     TimeSheet[RESOURCE KEY],
     TimeSheet[RESOURCE NAME],
     "Perc", (COUNT (TimeSheet[TIMESHEET KEY])/4)
       * 100
    FILTER (
     TimeSheet,
     OR (TimeSheet[TIMESHEET_STATUS_KEY] = 4, TimeSheet[TIMESHEET_STATUS_KEY] = 1)
    FILTER (
     TimeSheet,
     TimeSheet[PERIOD FINISH DATE] >= TimeSheet[CLARITY UPDATED DATE]
    FILTER (
     TimeSheet,
     AND (
       TimeSheet[PERIOD_START_DATE] <= TODAY (),
       TimeSheet[PERIOD_START_DATE]
         >= TODAY () - 28
  "Band1". 33.
  "Band2", 66,
  "End Value", 100
```

You will mostly use **SUMMARIZE**, **CALCULATETABLE** and **FILTER** to pull information from datamodel

Example: Convert SQL to DAX



EVALUATE is not used in PBI. Instead you use = Filters in **CALCULATETABLE** are always in logical AND Filters in FILTER can be put in AND/OR.

Select ADJUSTMENT_KEY, APPROVED_BY From dwh_tme_sheet

is equivalent to:

EVALUATE SUMMARIZECOLUMNS (TimeSheet[ADJUSTMENT_KEY], TimeSheet[APPROVED_BY], TimeSheet)

You can combine CALCULATETABLE and FILTER:

EVALUATE CALCULATETABLE (TimeSheet, FILTER (TimeSheet, TimeSheet[RESOURCE_KEY] = 5003126), FILTER (TimeSheet, TimeSheet[IS_ADJUSTMENT] = 0)



Example: Summarize/Group By

| <pre>select resource_key, resource_name, count(timesheet_key)</pre> |
|---|
| from dwh_tme_sheet |
| group by resource_key, resource_name |

is written as:

EVALUATE SUMMARIZE

SQL

TimeSheet, TimeSheet[RESOURCE_KEY], TimeSheet[RESOURCE_NAME], "Count", COUNT (TimeSheet[TIMESHEET_KEY])

Now, add a few filter conditions:

select resource_key, resource_name, count(timesheet_key)
from dwh_tme_sheet
where TIMESHEET_STATUS_KEY in (4,1)
and PERIOD_FINISH_DATE >= CLARITY_UPDATED_DATE
and PERIOD_START_DATE between (sysdate-28) and sysdate
group by resource_key, resource_name

```
EVALUATE
                           CALCULATETABLE (
is written as:
                             SUMMARIZE (
                               TimeSheet,
                               TimeSheet[RESOURCE KEY],
                               TimeSheet[RESOURCE NAME],
                               "Count", COUNT ( TimeSheet[TIMESHEET KEY] )
                             ),
                             FILTER (
                               TimeSheet,
                               OR (TimeSheet[TIMESHEET STATUS KEY] = 4, TimeSheet[TIMESHEET STATUS KEY] = 1)
                             FILTER (
                               TimeSheet,
                               TimeSheet[PERIOD FINISH DATE] >= TimeSheet[CLARITY UPDATED DATE]
                             FILTER (
                               TimeSheet,
                               AND (
                                 TimeSheet[PERIOD START DATE]
                                   >= TODAY () - 28,
                                 TimeSheet[PERIOD_START_DATE] <= TODAY ()
```

Example: Use ADDCOLUMNS to add new columns

```
EVALUATE
ADDCOLUMNS (
 CALCULATETABLE (
   SUMMARIZE (
     TimeSheet,
     TimeSheet[RESOURCE KEY],
     TimeSheet[RESOURCE_NAME],
     "Count", COUNT ( TimeSheet[TIMESHEET KEY] )
    ),
    FILTER (
     TimeSheet,
     OR (TimeSheet[TIMESHEET_STATUS_KEY] = 4, TimeSheet[TIMESHEET_STATUS_KEY] = 1)
    FILTER (
     TimeSheet,
     TimeSheet[PERIOD FINISH DATE] >= TimeSheet[CLARITY UPDATED DATE]
    FILTER (
     TimeSheet,
     AND (
       TimeSheet[PERIOD START DATE]
         >= TODAY () - 28,
       TimeSheet[PERIOD START DATE] <= TODAY ()
  "COL1", 100,
  "COL2", 200
```

Number of Tasks for Each Projects:

```
EVALUATE
ADDCOLUMNS ( Project, "Number of Task", COUNTROWS ( RELATEDTABLE ( Task ) ) )
```

****RELATEDTABLE** uses the data model relationship to find the table joins. Avoid using this and use explicit join conditions.



Create a Dashboard by connecting to Odata Feed



Cancel

regoUniversity 2018

Here are the steps to create a dashboard:

- Open your Power BI Desktop instance
- Select OData feed option under GetData
- Enter the Odata URL provided by your admin and click on OK.
- Enter the username & password and click on Connect to authenticate.
- Odata Table Navigator opens up. Select the tables required for your dashboard and click on Load Navigator

| Display Options 👻 | | DWH_INV_INV Preview downloaded | | | |
|-----------------------|---|-----------------------------------|----------------------------|---------------------|--------------------------|
| | | INVESTMENT_KEY | CLARITY_INVESTMENT_KEY | INVESTMENT_ID | INVESTMENT_NAME |
| DWH_INV_INVESTMENT | ^ | 5006027 | 5006027 | IDA00275 | Annual Benefit Changes I |
| DWH_INV_INVESTMENT_LN | | 5006029 | 5006029 | IDA00277 | Annual PPO Network Cha |
| DWH_INV_OBS_MAPPING | | 5006030 | 5006030 | IDA00279 | Medicare Advantage Plar |
| DWH_INV_OTHER_WORK | | 5006032 | 5006032 | IDA00307 | New HRA vendor |
| DWH_INV_OTHER_WORK_LN | | 1 The data in the | preview has been truncated | due to size limits. | |
| DWH INV PER FACTS F V | | | | | |

• Selected Tables gets loaded and power BI canvas opens up.





Create a Dashboard by connecting to Odata Feed



regoUniversity 2018

• Select the required visualization and drag the fields from the tables on to the settings to generate a meaningful visualization.



• You can group related visualizations under a single dashboard. Save the dashboard after completion.

| rojects by Status Indicator | | S Type | OBS Path | | Project |
|------------------------------|----------|--------|--|-----------------|------------------|
| rejects by status indicator | A | | → Ali | | V All |
| | | | | | |
| | | | | | |
| = | = | E2 ··· | | | |
| Projects by Status Indicator | | | Project Name | Project Manager | Status Indicator |
| 25 | | | | | On Track |
| | | | 112017 mcm OWB Training | Calderon, Mindy | On Track |
| | | | 2017 Integration Project Cont. | Bonham, Jessica | |
| | | | 2017 New Product Development | Bonham, Jessica | |
| | | | 2017 Resource Project | Bonham, Jessica | |
| | | | 2017 Security Project | Bonham, Jessica | |
| | | | 2017 Software Implementation Project | Bonham, Jessica | |
| | | | 2017-2018 Server Maintenance | Garvey, Sara | |
| | | | A Great project | Springer, Jody | |
| | 12 | | A Great Project - Master Plan | Springer, Jody | |
| | | | A great project 2 | Ma, Elaine | |
| | | | A great project 3 | Ma, Elaine | |
| | | | Another Great Project-Use for Demo | Springer, Jody | |
| | | | Create new Marketing Campaign | Dickson, Don | On Track |
| | | | Define Cloud Migration Strategy | Dickson, Don | |
| | | | Define Cloud Strategy and Architecture | Dickson, Don | |
| | | | Genius Upgrade | Garvey, Sara | On Track |
| | | | Google Mail Transition | Garvey, Sara | On Track |
| | | | Hire New Sales Staff | Dickson, Don | On Track |
| (Blank) | On Track | | HSBC Cloud Transition Program | Dickson, Don | |

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** ٠
- Click on Maintain My Certification
- Click on Visit CCR's button under the Report PDU's ٠
- Click on Report PDU's •
- Click on **Course or Training** •
- Class Name = regoUniversity
- Course Number = Session Number
- Date Started = **Today's Date** ٠
- Date Completed = **Today's Date** ٠
- Hours Completed = 1 PDU per hour of class time ٠
- Training classes = **Technical** •
- Click on I agree and Submit





888.813.0444



Email info@regouniversity.com



Website www.regouniversity.com

