Fusion Technical Training

Front end and Back end - Front end is User interface and Back end means where we have various tables of data base.

Report:

It enables businesses to provide instant access to information to all levels within and outside of organization in a structured and well defined manner.

Types of reports:
1.BIP - Business Intelligence Process
2.OTBI - Oracle Transactional Business Intelligence
3.Interactive report

BIP Report:

Bip report is the customize reporting tool. We have a Data Model, which is a place where we write sql query. BIP reports are created by writing queries.

Navigation to generate reports :

Tools-> Reports and Analytics-> Browse catalog

My Folder: Reports in personal folders can only be accessed by the user who created and saved the content into the folder.

Shared Folder: Reports in shared folders can accessed by all the users in the environment. But in Shared Folder we cannot make any customization (creation of folders) in standard areas, we have a folder/area called custom in shared folder where we can customize(create more no. of folders).

OTBI - from subject area through collection of multiple tables and views reports are created.

Steps involved to create a OTBI report-
Analysis
Dash boards
Filter
Dashboard prompt

Infolets:
An infolets is seeded objects that display key information about your organization and which redirected to the detailed information of the thing which is displaying in infolets.

Purchase order:

A purchase order is a formal document from buyer to supplier, indicating the details of the actual purchase of items or services.

--Standard report which we cannot make any customization, if we have privileges to the reports we can make customization.

Data Model Properties

Description: The description that you enter here displays in the catalog. This description is translatable.

Default Data Source : Select the data source from the list. Data models can include multiple data sets from one or more data sources. The default data source you select here is presented as the default for each new SQL data set you define

Oracle DB Default Package : If you define a query against an Oracle Database, then you can include before or after data triggers (event triggers) in your data model. Event triggers make use of PL/SQL packages to execute RDBMS level functions. For data models that include event triggers or a PL/SQL group filter, you must enter a default PL/SQL package here

Query Time Out :If the SQL query is still processing when the time out value is met, the error Failed to retrievedata xml.is returned. If you do not enter a value for this data model, the server property value is used.

Enable SQL Pruning : Applies to Oracle Database queries only that use Standard SQL. If your query returns many columns but only a subset are used by your report template, SQL pruning returns only those columns required by the template. Setting this property enhances processing time and reduces memory usage

Backup Data Source : If you have set up a backup database for this data source, select Enable Backup Connection to enable the option; then select it when you want Oracle BI Publisher to use the backup.

➢ To use the backup data source only when the primary is down, select **Switch to Backup Data Source when Primary Data Source is unavailable**. Note that when the primary data source is down, the data engine must wait for a response before switching to the backup.

➢ To always use the backup data source when executing this data model, select **Use Backup Data Source Only**. Using the backup database may enhance performance.

XML OUTPUT OPTIONS:

Include Parameter Tags — If you define parameters for your data model, select this check box to include the parameter values in the XML output file. Enable this option when you want to use the parameter value in the reporT.

Include Empty Tags for Null Elements— Select this check box to include elements with null values in your output XML data. When you include a null element, then a requested element that contains no data in your data source is included in your XML output as an empty XML tag. If you do not select this option, no entry appears for COLUMN.

Include Group List Tag— Select this check box to include the rowset tags in your output XML data. If you include the group list tags, then the group list appears as another hierarchy within your data.

XML Tag Display— Select this check box to generate the XML data tags in upper case, in lower case, or to preserve the definition you supplied in the data structure.

Attachments: The Attachment region of the page displays data files that you have uploaded or attached to the data model.

Sample Data: After you build your data model, you must attach a small, but representative set of sample data generated from your data model. The sample data is used by BI Publisher's layout editing tools.

Attaching Schema:

Data Files: If you have uploaded a local Microsoft Excel, CSV, or XML file as a data source for this report, the file displays here

Data Source Options:

Categorized into 3 types:

1.The Oracle BI Publisher can retrieve metadata information from the SQL queries submitted against Oracle BI Server, an Oracle Database, or other supported databases.

- ➢ SQL query
- ➢ Multidimensional (MDX) query
- ➢ LDAP Query
- ➢ Microsoft Excel file
- ➢ XML data file
- ➢ CSV (comma separated value) file

2.BI Publisher can retrieve column names and data type information from the data source but it cannot process or structure the data.

- ➢ Analysis or Oracle BI Analysis
- ➢ View objects

3.BI Publisher retrieves data that was generated and structured at the source

- ➢ HTTP XML feed
- ➢ Web services

1. SQL QUERY:

In this Sql Quer data set, we write the Sql query to retrieve the data. Also we have query builder to build a query for making of report.

2. MDX QUERY:

Multidimensional Expressions (MDX) is the query language that you use to work with and retrieve multidimensional data in Microsoft SQL Server Analysis Services in OLAP databases.

In MDX , we have attributes like Dimensions, Members, Facts/Measures.. To finish MDX Query we must any one of the MDX attribute.

With MDX we can create KPI's, Calculations etc,.

SQL Server Analysis Services (SSAS) offers OLAP.

3. ANALYSIS or ORACLE BI ANALYSIS:

We can directly use the analysis created by using OTBI. Here we attached the OTBI Analysis and extract data in BI Publisher.

4.VIEW OBJECT:

BI Publisher enables us to connect to your custom applications built with Oracle Application desktop framework and use view objects in our appliications to create reports.

5.WEB SERVICES:

BI Publisher supports data sets that use Web service data sources to return valid XML data.

It uses WSDL URL to extract the data from data source. WSDL URL automatically populated from the Web Service Data Source.

There is no metadata available from Web service data sets, therefore grouping and linking are not supported.

6. LDAP Query:

BI Publisher supports queries against Lightweight Directory Access protocol (LDAP) data sources. You can query user information stored in LDAP directories and then use the data model editor to link the user information with data retrieved from other data sources.

7. XML FILE:

By uploading xml file into the data set we can create a report.

We can upload XML File by 2 ways:

Place the XML file in a directory that your administrator has set up as a data source.

Upload the XML file to the data model from a local directory.
The XML files that you use as input to the BI Publisher data engine must be UTF-8 encoded.
After uploading the file, it is displayed on the Properties pane of the data model under the Attachments region .

8.CONTENT SERVER:

You can set up connections to Content Server data source on the Administration page and then use that in multiple data models.
We must set up the connection before you create a data model. Create a data model by creating the SQL Query data set (required) first and then create the Content Server data set.

1.   **9.MICROSOFT EXCEL FILE:**
By uploading MICROSOFT EXCEL FILE into the data set we can create a report.
We can upload MICROSOFT EXCEL FILE by 2 ways:

Place the MICROSOFT EXCEL FILE in a directory that your administrator has set up as a data source.

Upload the MICROSOFT EXCEL FILE to the data model from a local directory.
After uploading the file, it is displayed on the Properties pane of the data model under the Attachments region.

10.CSV File:

BI Publisher supports data sets that use CSV file data sources to return valid XML data.
 We can upload CSV File by 2 ways:
You can use a CSV file that is located in a directory that your administrator has set up as a data source.
You can upload a file from a local directory.
The supported CSV file delimiters are Comma, Pipe, Semicolon, and Tab.

11.HTTP XML Feed:
Using the HTTP (XML Feed) data set type, you can create data models from RSS and XML feeds over the Web by retrieving data through the HTTP GET method.

You can set up an HTTP (XML Feed) data sources in two different ways.

On the Administration page: Connections to HTTP data sources can be set up on the Administration page and then used in multiple data models.

As a private data source: You can also set up a private connection accessible only to you.

Data Model:

A data model is an object that contains a set of instructions for BI Publisher to retrieve and structure data for a report.

Components in Data Model:

➤ **DATA SET:** A data set contains the logic to retrieve data from a single data source. A data set can retrieve data from a variety of data sources (for example, a database, an existing data file, a Web service call to another application, or a URL/URI to an external data provider). A data model can have multiple data sets from multiple sources.

➤ **Event triggers:** When the event occurs the trigger runs the PL/SQL code associated with it. The data model editor supports before data and after data triggers as well as schedule triggers. Before data and after data triggers consist of a call to execute a set of functions defined in a PL/SQL package stored in an Oracle database. A schedule trigger is executed for scheduled reports and tests for a condition that determines whether or not to run a scheduled report job.

➤ **Lists of values:** A list of values is a menu of values from which report consumers can select parameter values to pass to the report.

➤ **Parameters:** A parameter is a variable whose value can be set at runtime. The data model editor supports several parameter types

➤ **Flexfields:** A flexfield is a structure specific to Oracle Applications. The data model editor supports retrieving data from flexfield structures defined in your Oracle Application database tables.

A flexfield is a data field that your organization can customize to your business needs without programming.

1.key flexfields

A key flexfield is a field you can customize to enter multi-segment values such as part numbers, account numbers, and so on.

2.descriptive flexfields

A descriptive flexfield is a field you customize to enter additional information for which your Oracle applications product has not already provided a field.

➤ **Bursting Definitions:** Bursting is a process of splitting data into blocks, generating documents for each data block, and delivering the documents to one or more destinations. A single bursting definition provides the instructions for splitting the report data, generating the document, and delivering the output to its specified destinations.

TASKS ON CATALOG OBJECT:

1.Expand - displays the folder contents.

2.**Delete**- removes a folder or object from the catalog.

3.**Copy**- duplicates a folder or object. To paste the object in a different folder, navigate to the folder location and click Paste To place the copied item in the desired folder. To paste a copy of the item into the same folder, click Paste. The copied object is renamed with the prefix *Copy_of_*.

4.**Rename**- renames a folder or object.

5.**Permissions**- sets object-level permissions like read only, custom, modify, view output only etc..

6.**Archive:** Archiving enables you to bundle the entire catalog, specific folders, or multi-component objects (for example, scorecards) as a catalog file and upload the catalog file to unarchive the data to another location in the catalog. It is like zipping of multiple objects.

7. **Unarchive:** Unzipping the archived or bundled catalog/multiple objects.

BIP HOME PAGE: The Home page is divided into sections that enable you to quickly begin a specific task, locate an object, or access documentation. Whatever the things we have done so far, we can easily see in the home page, it's quite like recent history.

10-02-2023

DATA MODEL:

It is the collection of invoke (from table) and trigger(expected results).A data model is an object that contains a set of instructions for BI Publisher to retrieve and structure data for a report.

DATA SET:

It is the multiple components and it refers to a file that contains one or more records.

Query Building:

The query builder is an interface that helps you construct and modify queries in a structured format. Using Query Builder, you can search and filter database objects, select objects and columns, create relationships between objects, view formatted query results, and save queries with little or no SQL knowledge.

Parameter:

Adding parameters to your data model enables users to interact with data when they submit or view reports.It is a placeholder for a variable that contains some value of some type when executing a general-purpose query, or arguments and return values when a stored procedure is executed.

NVL:

The NVL function allows you to replace null values with a default value. If the value in the first parameter is null, the function returns the value in the second parameter. If the first parameter is any value other than null, it is returned unchanged.It converts null value to a actual value.

Default:

The DEFAULT constraint is used to set a default value for a column. The default value will be added to all new records, if no other value is specified.

JOIN:

A Join clause is used to combine rows from two or more tables, based on a related column between them. A join clause is a SQL command used to combine records from multiple tables or retrieve data from these tables based on the existence of a common field between them. A join condition and SELECT statement can be used to join the tables.

Types of Join:

1) INNER JOIN

2) LEFT JOIN

3) RIGHT JOIN

4) CROSS JOIN

Inner Join:

The Inner Join selects records that have matching values in both tables.

Left Join:

The Left Join returns all records from the left table (table1), and the matching records from the right table (table2). The

result is 0 records from the right side, if there is no match.

Right Join:

The Right Join returns all records from the right table (table2), and the matching records from the left table (table1). The result is 0 records from the left side, if there is no match.

Cartesian or Cross Join:

The Cross Join returns all records from both tables i.e. each row of one table to each row of another table.

DATASET JOIN:

A data join is when two data sets are combined in a side by side manner, therefore at least one column in each data set must be the same.

DATA SOURCE:

A data source contains details about the database server you will be connecting to, the login to use and the database to use

PRIMARY KEY:

A primary key is the column or columns that contain values that uniquely identify each row in a table. We cannot save NULL values.

FOREIGN KEY:

Foreign key is a column or combination of columns that creates a relationship between two tables, whose values must match values of a column in some other table.

UNIQUE:

A unique key in SQL is the set of fields or columns of a table that helps us uniquely identify records. The unique key guarantees the uniqueness of the columns in the database. It is similar to the primary key but can accept a null value, unlike it.

CASCADE:

If a User deletes one or more rows from the Parent table, Oracle should automatically delete the corresponding rows from the Child table.

13-02-2023

Creation of Reports:

1. Interactive Report:

➔ Go to the Browse Catalog in Reports & Analytics.

➔ Go to the Data Model option in Create tab

➔ Build the SQL query to extract the data in report.

➔ View the Data

➔ Save the Data Model.

➔ Click on Save as Sample Data.

➔ Click on Create Report. (It navigates to **INTERACTIVE REPORT**)
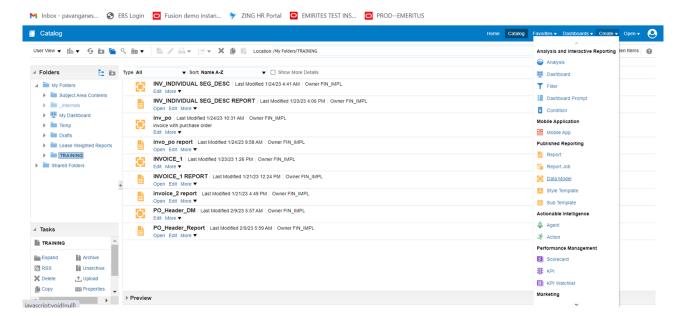
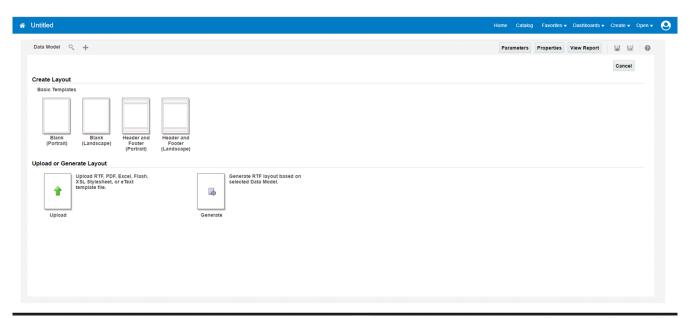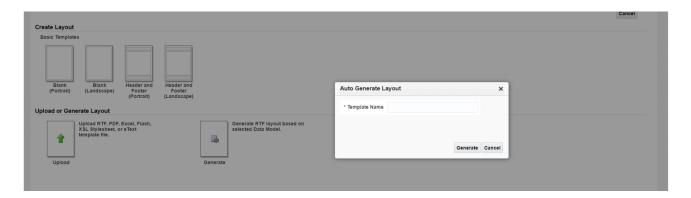➔ Save the Report.

➔ Click on View Report to see the report.

2. BIP Report:

a. Generate auto layout.

➔ Go to the Browse Catalog in Reports & Analytics.

➔ Go to the Data Model option in Create tab

➔ Build the SQL query to extract the data in report.

➔ View the Data

➔ Save the Data Model.

➔ Click on Save as Sample Data.

➔ Go to the Catalog and Click on report.

➔ Navigates to report creation page.

➔ Select the Data Model on which we want to create a report.
➔ Then, click on 'Generate RTF layout based on selected layout.'

➔ Name the layout. (Here a layout is automatically generated for the report)
➔ It displays the layout created for the data model as below.

➔ Save the Report and click in 'View Report' to see the report.

➔ We can able to see the report output in multiple formats.
➔ We can able to add multiple template a single report. Just click on 'Add New Layout', it redirects to Report creation page and layout.

➔ In terms of multiple layouts, have to choose only one layout to see the report.
➔ Go to 'View a list', then select the layout we want to see for the report.

➔ Save the changes and Click on 'View report' to see the report.

Configure Parameter Settings for the Report

➔ **Parameter Location - This property controls where the parameter region is displayed in the report viewer. The options are:**

· Horizontal Region - displays the parameters horizontally across the top of the report viewer.

· Vertical Region - displays the parameters vertically along the left side of the report viewer.

· Full Page - displays the parameters on a separate page in the report viewer. After a user enters parameter values, the page is dismissed. To change parameter values, click the report viewer Parameters button to display the Parameters page again.

· Dialog - invokes a dialog box to display the parameters. After a user enters parameter values, the dialog is dismissed.

➔ **Parameter Label Location - This property controls where the parameter labels are displayed. The options are:**

· Place label on side - places the parameter label to the left side of the entry box.

· Place label on top - places parameter label on top of the entry box.

➔ **Show Apply Button -**When set to True, reports with parameter options display the Apply button in the report viewer. When a user changes the parameter values, he must click Apply to render the report with the new values.

- ➔ **Show- This property controls whether the parameter is displayed to the user. Disable the Show property if you don't want the user to see or change the parameter values that're passed to the data model.**

- ➔ **Type- This property is customizable for menu type parameters only. We have different types to display the LOV's while runtime.**
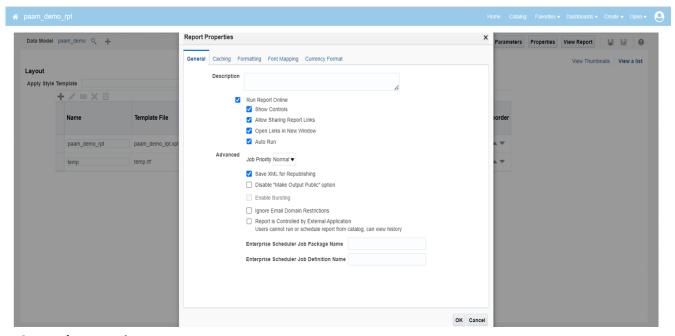
1. **Check Box**

2. **Radio Button**

- ➔ **Multiple- This property is display only; it indicates whether multiple values may be selected for a menu parameter.**

- ➔ **Display Label- Use this property to edit the display labels shown for each parameter. The default values are defined in the data model.**

- ➔ **Default Value- Use this property to configure the default value for the parameter specifically for this report.**

- ➔ **Row Placement-Use this property to configure the number of rows for displaying the parameters and in which row to place each parameter.**

Configuring Report Properties:

We have several report properties to configure,
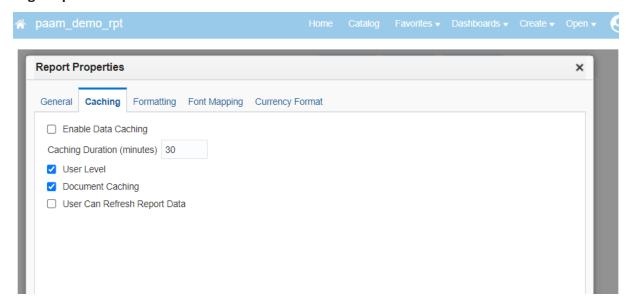

General Properties

- ➔ **Description:** Enter a description to display with the report in the catalog. This text is translatable.

- ➔ **Run Report Online:** Disable this property if you do not want users to view this report in the online Report Viewer. When disabled, users can Schedule the report only.

- ➔ **Show controls :** This property controls the display of the control region of the report. The Control region consists of the Template list, Output list, and Parameter lists.

- ➔ **Allow Sharing Report Links** : The Actions menu of the Report Viewer includes the option Share Report Link, which enables users to display the URL for the current report.

- ➔ **Open Links in New Window** : This property controls how links contained within a report are opened. By default links open in a new browser window.

➔ **Asynchronous Mode**: Reports run in asynchronous mode use a unique thread to execute the report when run in the report viewer.

➔ **Auto Run :** When this property is enabled the report will automatically run when the user selects the Open link for the report. When Auto Run is disabled, selecting the Open link for the report displays the online viewer but does not run the report.

Advanced Options

➔ **Job Priority** : Specifies the priority for the report.

➔ **Enable Bursting** : When a user schedules the report, the selected bursting definition will be enabled in the Scheduler.

➔ **Ignore Email Domain Restrictions** : To ignore the values set in the Allowed Email Recipient Domains property in the delivery configuration page

➔ **Report is Controlled by External Application. Users cannot run or schedule report from catalog, can view history**.
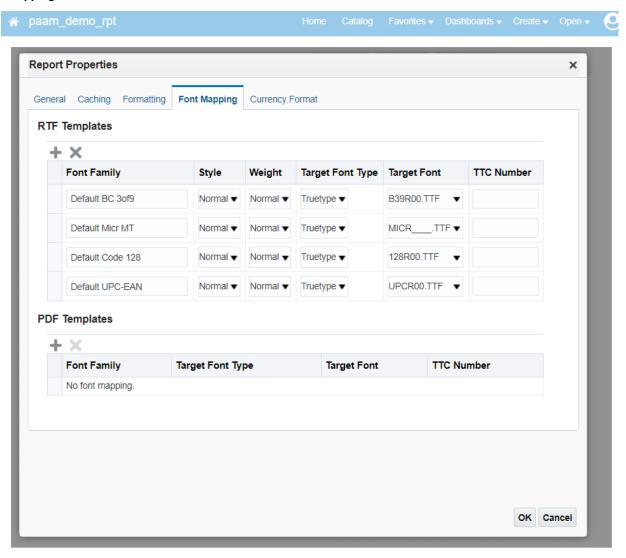
Caching Properties

➔ **Enable Data Caching** :When this property is enabled, the data generated by the online submission of this report is stored in the cache.

➔ **Caching Duration** : Enter the time limit for a report dataset or document to remain in cache.

➔ **User Level** : This property stores a separate cache for each user. The report data shown to each user comes only from the private cache.

➔ **Document Caching** : W**hen a user views the report online, the document is placed in the cache.**

➔ **User Can Refresh Report Data : When the user clicks Refresh in the report viewer, Publisher generates a fresh dataset for the report.**
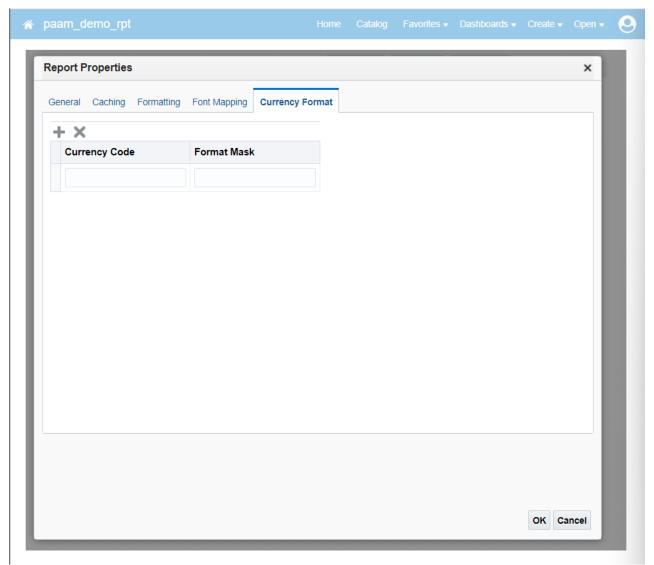
Formatting Properties

  Formatting properties tab enables you to set runtime properties at the report level.

Font Mapping

⌂ paam_demo_rpt	Home Catalog Favorites ▾ Dashboards ▾ Create ▾ Open ▾

Report Properties   ✕

General   Caching   Formatting   **Font Mapping**   Currency Format

RTF Templates

➕ ✕

Font Family	Style	Weight	Target Font Type	Target Font	TTC Number
Default BC 3of9	Normal ▾	Normal ▾	Truetype ▾	B39R00.TTF ▾	
Default Micr MT	Normal ▾	Normal ▾	Truetype ▾	MICR____.TTF ▾	
Default Code 128	Normal ▾	Normal ▾	Truetype ▾	128R00.TTF ▾	
Default UPC-EAN	Normal ▾	Normal ▾	Truetype ▾	UPCR00.TTF ▾	

PDF Templates

➕ ✕

Font Family	Target Font Type	Target Font	TTC Number
No font mapping.			

OK   Cancel

➔ **You can map base fonts in RTF or PDF templates to target fonts to be used in the published document**

Currency Formats

➔ We can map a number format mask to a specific currency so that reports can display with their own corresponding formatting.

Configuring Layouts Using the List View:

After creating or uploading the layouts for the report, you can configure settings for the layout from the List View

Applying a Style Template: A style template contains style definitions that are applied to the paragraphs, headings, tables, and headers and footers of a report.

L**ayouts Toolbar:**

Create, Edit, Properties, Delete, Profiling.

Configuring the Layout Settings

Name: We can name for the layout.

Template File: Displays the name of the file that was saved to the report definition. Click the template file name to download it.

Type: Extension of the template file.

Output Formats : More no. of output formats we have.

Default Format: Default it select the layout for the report we can able to change the layout.

Apply Style Template : Select this box to apply the style template to this layout.

Active: When you want to keep the layout as part of the report definition, but no longer make it available.

View Online: By default, a layout is available for report consumers who open the report in the Report Viewer.

Locale: Displays the locale selected when the layout was uploaded. This field is not updatable.

14-02-2023

Oracle Transactional Business Intelligence (OTBI):
To navigate to the subject areas.

1.   In the Navigator, click **Reports and Analytics**, and then click on **Browse Catalog**. This will take you to the Oracle Business Intelligence dashboard.

2.   Click **New** to access the options in the drop-down list.

3.   Click **Analysis** in the drop-down list. You will see the Select Subject Area dialog box.
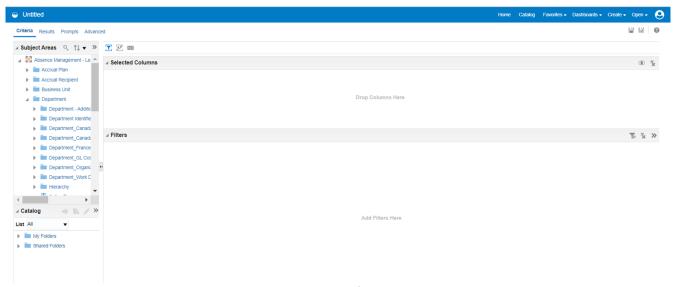

4.   Click the subject area that you want to analyze. The page is refreshed to display the columns available under that subject area.

5.   You can now proceed to create your analysis.

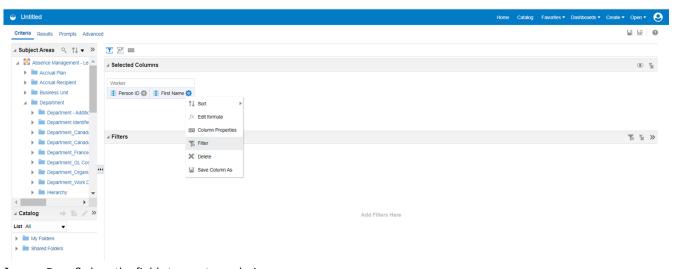You can create new reports using these subject areas or add them to your existing reports.

Criteria tab— By using the Criteria Tab you build up a report. It lets you specify the criteria for an analysis, including columns and filters.

Results tab —The Results tab is where you can view the results of the OTBA report based on the columns that were extracted onto the Criteria tab.

Prompts tab— The prompts tab is where you can build various kinds of prompts. This tab lets you create prompts that allow users to select values to filter an analysis or analyses on a dashboard. Prompts allow users to select values that dynamically filter all views within the analysis or analyses.
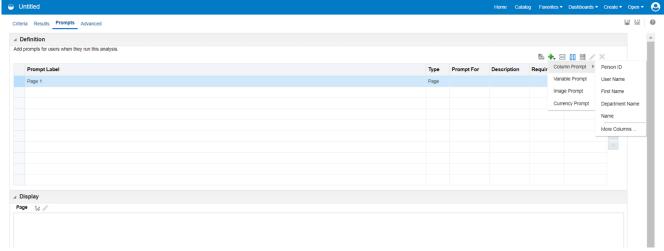
Advanced tab— This tab lets you edit XML code and examine the logical SQL statement that was generated for an analysis. You can use the existing SQL statement as the basis for creating a new analysis. Additional filters for the report can be added in the Advanced Tab. Using this tab, you can generate XML and save into the system. This particular XML can be used as a data source in BI Publisher Report and create complex report and further customization can be done in BI Publisher Report.

➔　　　　Drag & drop the fields to create analysis.

➔　　　　Here we have filter option to limit the results that are displayed when an analysis is run.
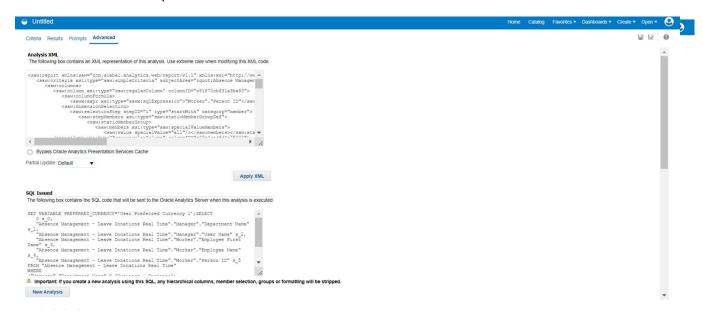
➔　　　　Click on RESULT tab to see the analysis.

➔ Table prompt makes the analysis to choose the variable at runtime.

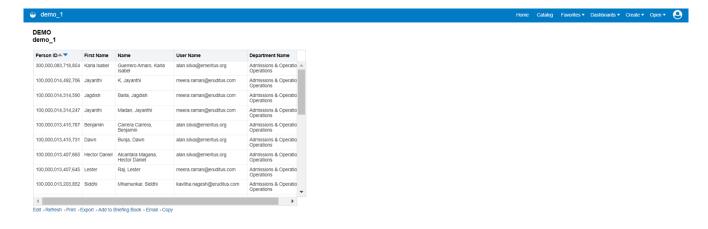➔ Excluded makes us to eliminate the column which we selected at the time of analysis.

➔ We can create a new prompt in prompts tab.

➔ Advanced tab lets you edit XML code and examine the logical SQL statement that was generated for an analysis.
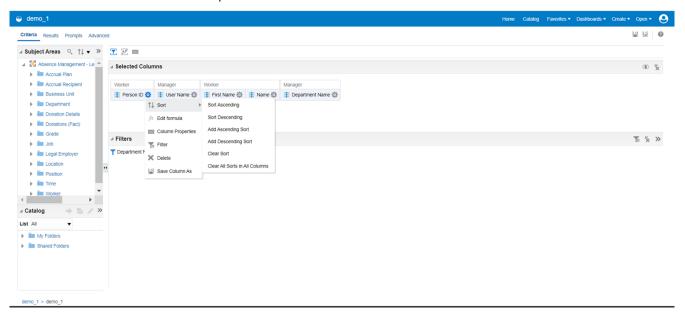

➔ Prompts in layout, automatically select variable at runtime show the result for that variable. Able to change the variable for analysis.

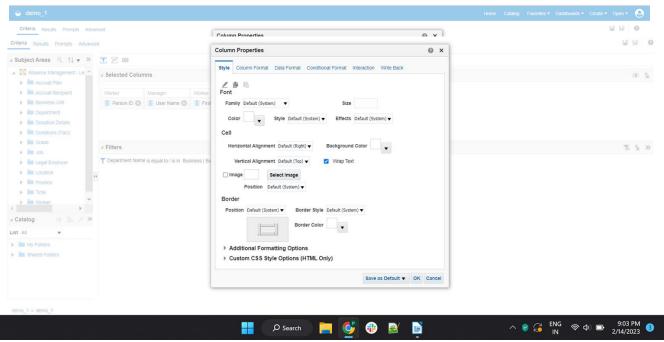➔ If we create a prompt at PROMPTS tab, it will also able to choose variable at runtime only once.

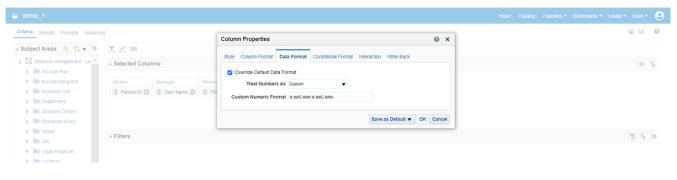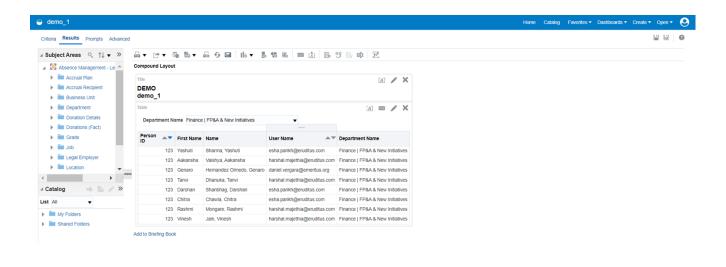➔ We have different actions to perform on column.

➔ We have sorting, Edit formula, Properties, Filters, Delete, Save Column as….
➔ Sorting to sort the rows and columns either from highest to lowest data values or lowest to highest.
➔ Edit formula, We can change the default formula for the measures.
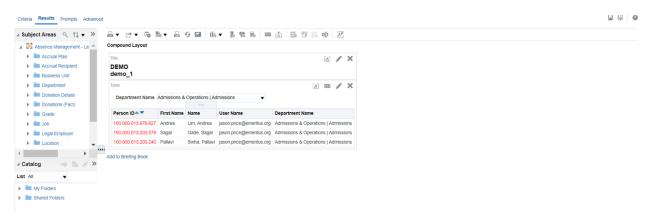➔ Column Properties….

➔ Column Format

➔ We can make changes for that specific column we want.

➔ In Data Format, we can override the default data as per our requirement.

➔ Conditional Format, for the specific condition/prompt we can format the appearance of any column we want.

➔ Interactions, User can make views more interactive by adding hyperlinks, drill down actions etc.