

ORACLE FUSION OTBI

PART 1

Nikita Shantaram Khetade©

4/10/2020

INTRODUCTION

1. ORACLE FUSION APPLICATION (SAAS) USER CREATION WITH BASE TABLES
2. CREATING DATA MODEL.
3. ORACLE FUSION BI PUBLISHER - FIRST REPORT
4. RUNNING THE REPORT FROM CATALOG
5. RUNNING THE REPORT FROM FUSION APPLICATION BY CREATING ESS JOB
6. REPORT LAYOUT CREATION USING MANUAL METHOD 1
7. REPORT LAYOUT CREATION USING MANUAL METHOD 2
8. REPORT LAYOUT PROPERTIES EXPLAINED - GENERAL
9. REPORT LAYOUT PROPERTIES EXPLAINED - CACHING
10. REPORT LAYOUT PROPERTIES EXPLAINED - FORMATTING
11. REPORT LAYOUT PROPERTIES EXPLAINED - CURRENCY FORMAT
12. STYLE SHEET TEMPLATE
13. MASTER DETAIL REPORT USING ELEMENT LEVEL LINK
14. MASTER DETAIL REPORT USING GROUP LEVEL LINK
15. MASTER DETAIL REPORT USING SUBGROUPS
16. AGGREGATE FUNCTIONS
17. ARITHMETIC OPERATION ON COLUMNS
18. EXPRESSIONS - IF (EXP) WITH EXAMPLE
19. EXPRESSIONS - NOT WITH XDO_USER_REPORT_LOCALE
20. EXPRESSIONS - AND FUNCTION
21. EXPRESSIONS - OR FUNCTION
22. EXPRESSIONS - MIN AND MAX FUNCTIONS
23. EXPRESSIONS - ROUND , SUM FUNCTION
24. WORKING WITH BIND PARAMETERS
25. BIND PARAMETERS WITH TABLE VALUESET AND \$FLEX\$ CONCEPT IN FUSION
26. SUBTEMPLATE
27. MULTILANGUAGE REPORT OR TRANSLATIONS
28. MULTI LANGUAGE REPORT OR TRANSLATIONS. -TESTING
29. BURSTING - DELIVERY CHANNEL - EMAIL AND FTP AND SFTP AND FILE
30. BURSTING REPORT TESTING FIXING THE ISSUES IN BURSTING QUERY
31. AR INVOICE REPORT DEVELOPMENT FROM SCRATCH - SQL QUERY DEVELOPMENT
32. WORKING WITH EXCEL TEMPLATES - PART 1

1. ORACLE FUSION APPLICATION (SAAS) USER CREATION WITH BASE TABLES

CREATING A REPORT OF ORACLE SALES CLOUD USER ACCOUNTS

TO EFFECTIVELY ENABLE ORACLE SALES CLOUD USERS' ACCESS ORACLE JAVA CLOUD SERVICE - SAAS EXTENSION, YOU SHOULD PERIODICALLY SYNCHRONIZE THOSE USERS WITH ORACLE JAVA CLOUD SERVICE - SAAS EXTENSION. THIS SYNCHRONIZATION IS A MANUAL, TWO-STEP PROCESS: CREATING THE LIST OF USERS FROM THE ORACLE SALES CLOUD SERVICE AND IMPORTING THE USERS INTO ORACLE JAVA CLOUD SERVICE - SAAS EXTENSION INSTANCE.

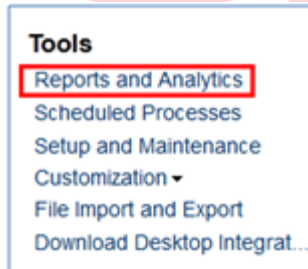
THE FOLLOWING PROCEDURE CREATES A REPORT CONTAINING A COMMA-SEPARATED VALUES (CSV) LIST OF THE FUSION APPLICATION USER ACCOUNT INFORMATION REQUIRED TO SET UP EACH USER ACCOUNT. ONCE YOU CREATE THE CSV FILE, YOU CAN IMPORT THESE ACCOUNTS TO ORACLE CLOUD. TO EXTRACT THE FUSION APPLICATIONS USER ACCOUNT INFORMATION:

1. FROM THE FUSION APPLICATIONS HOME PAGE, CLICK **NAVIGATOR**.



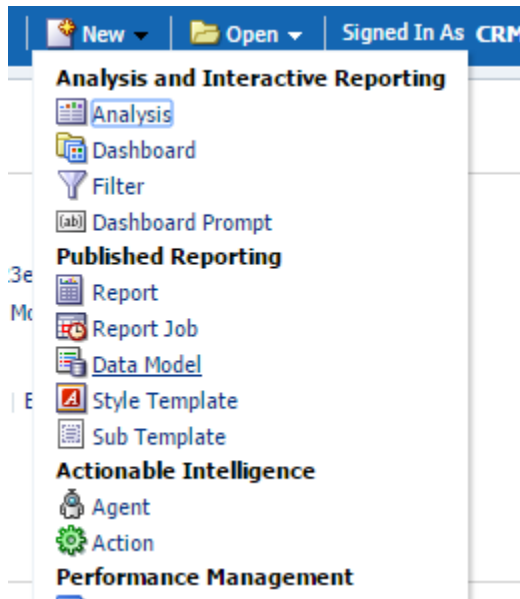
DESCRIPTION OF THE ILLUSTRATION 1_FUSION_NAVIGATOR.PNG

2. UNDER TOOLS, SELECT **REPORTS AND ANALYTICS**.



DESCRIPTION OF THE ILLUSTRATION 2_TOOLS_MENU.PNG

3. CLICK THE **BROWSE CATALOG** ICON TO OPEN THE ORACLE BI CATALOG, THEN SELECT **NEW**, THEN **DATA MODEL**.



DESCRIPTION OF THE ILLUSTRATION 3_BI_CATALOG_DATA_MODEL.PNG

4. UNDER THE DIAGRAM TAB, CLICK **NEW**, THEN **SQL QUERY** TO CREATE A NEW SQL QUERY DATA SET.
5. IN THE NEW DATA SET - SQL QUERY DIALOG, ENTER A **NAME**, SELECT **APPLICATIONDB_HCM** AS THE DATA SOURCE AND SELECT **STANDARD SQL** AS THE TYPE OF SQL. ENTER THE FOLLOWING SQL QUERY IN THE SQL QUERY SECTION, AND THEN CLICK OK.

```

6. --HCM

7. SELECT E.EMAIL_ADDRESS AS EMAIL

8. FROM FUSION.PER_USERS U, FUSION.PER_ROLES_DN R, FUSION.PER_USER_ROLES UR,

9. FUSION.PER_ALL_PEOPLE_F F

10. JOIN FUSION.PER_EMAIL_ADDRESSES E ON E.PERSON_ID = F.PERSON_ID

11. AND E.EMAIL_ADDRESS_ID = F.PRIMARY_EMAIL_ID AND E.EMAIL_TYPE = 'W1'

12. WHERE TRUNC(SYSDATE) BETWEEN F.EFFECTIVE_START_DATE AND F.EFFECTIVE_END_DATE

13. AND U.PERSON_ID = F.PERSON_ID AND U.ACTIVE_FLAG = 'Y' AND R.ROLE_COMMON_NAME =
:BIND

14. AND R.ROLE_GUID = UR.ROLE_GUID AND UR.ACTIVE_FLAG = 'Y' AND UR.TERMINATED_FLAG !=
'Y'

15.

16. --TCA

17. UNION SELECT C.EMAIL_ADDRESS AS EMAIL

18. FROM FUSION.PER_USERS U, FUSION.PER_ROLES_DN R, FUSION.PER_USER_ROLES UR,

19. FUSION.HZ_PERSON_PROFILES P

20. JOIN FUSION.HZ_CONTACT_POINTS C ON C.OWNER_TABLE_ID = P.PARTY_ID

```

21. AND C.OWNER_TABLE_NAME = 'HZ_PARTIES'

22. AND C.OVERALL_PRIMARY_FLAG = 'Y' AND C.CONTACT_POINT_TYPE = 'EMAIL' AND C.STATUS = 'A'

23. AND TRUNC(SYSDATE) BETWEEN C.START_DATE AND C.END_DATE

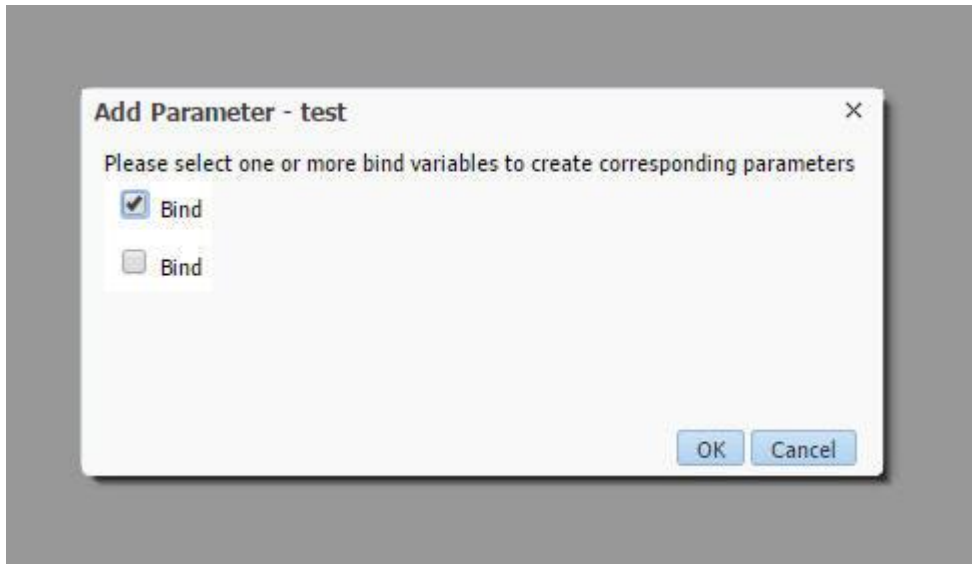
24. WHERE U.PARTY_ID = P.PARTY_ID AND TRUNC(SYSDATE) BETWEEN P.EFFECTIVE_START_DATE

25. AND P.EFFECTIVE_END_DATE AND P.STATUS = 'A' AND U.ACTIVE_FLAG = 'Y'

26. AND R.ROLE_COMMON_NAME = :BIND AND R.ROLE_GUID = UR.ROLE_GUID

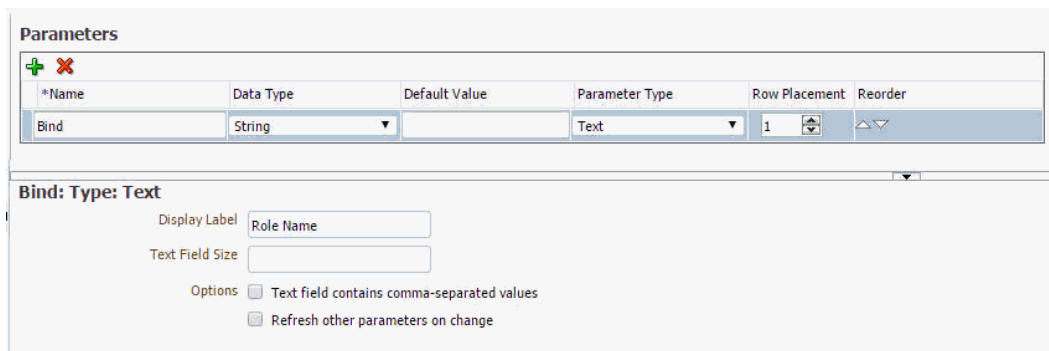
27. AND UR.ACTIVE_FLAG = 'Y' AND UR.TERMINATED_FLAG != 'Y'

28. IN THE ADD PARAMETER DIALOG, SELECT THE FIRST **BIND** AND CLICK **OK**.
THIS PARAMETER IS USED AS THE INPUT TO THE REPORT FOR GETTING ALL USERS FOR A ROLE.



DESCRIPTION OF THE ILLUSTRATION 6_ADD_PARAMETER_DIALOG.PNG

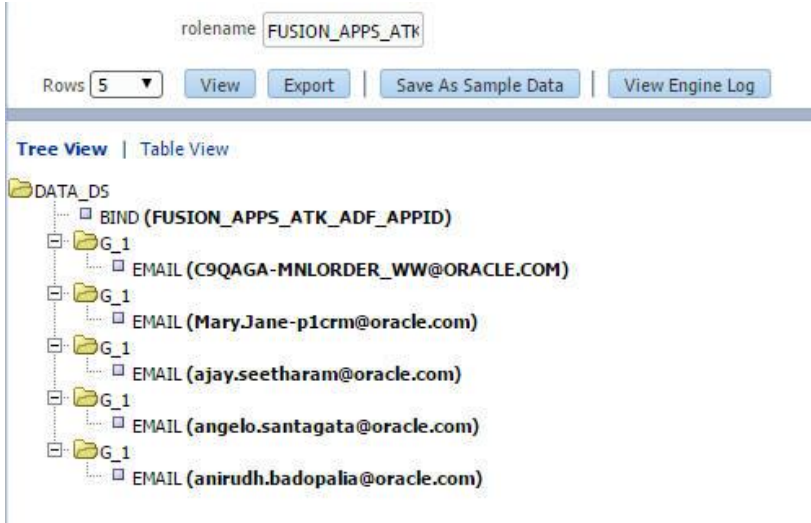
29. ENTER A NAME AND DISPLAY LABEL FOR THE BIND PARAMETER.



DESCRIPTION OF THE ILLUSTRATION USER_LOAD.GIF

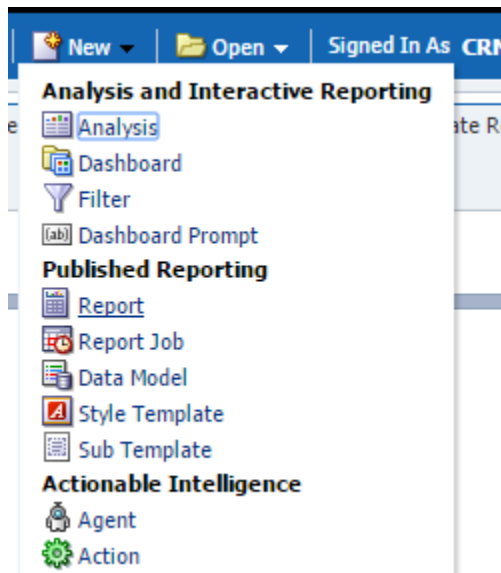
30. CLICK **VIEW DATA** TO DISPLAY THE DATA TAB, THEN ENTER A VALUE FOR ROLE NAME PARAMETER, FOR EXAMPLE, FUSION_APPS_HCM_ADF_APPID.

- CLICK **VIEW**.
- ONCE THE DATA APPEARS, CLICK **SAVE AS SAMPLE DATA**.
- CLICK **SAVE** TO SAVE THE DATA MODEL IN THE DRAFTS FOLDER (UNDER MY FOLDERS).



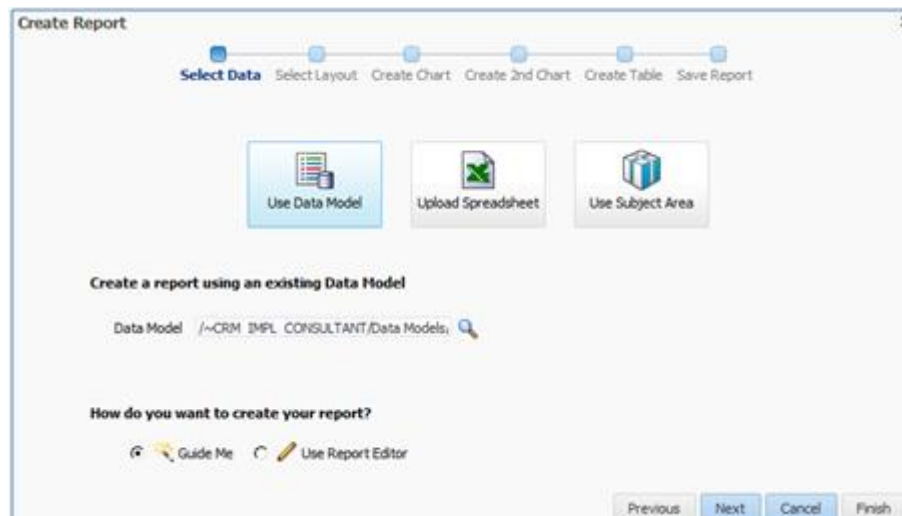
DESCRIPTION OF THE ILLUSTRATION 8_DATA_MODEL_TREE_VIEW.PNG

31. AT THE TOP OF THE WINDOW, CLICK **NEW**, AND THEN SELECT **REPORT**.



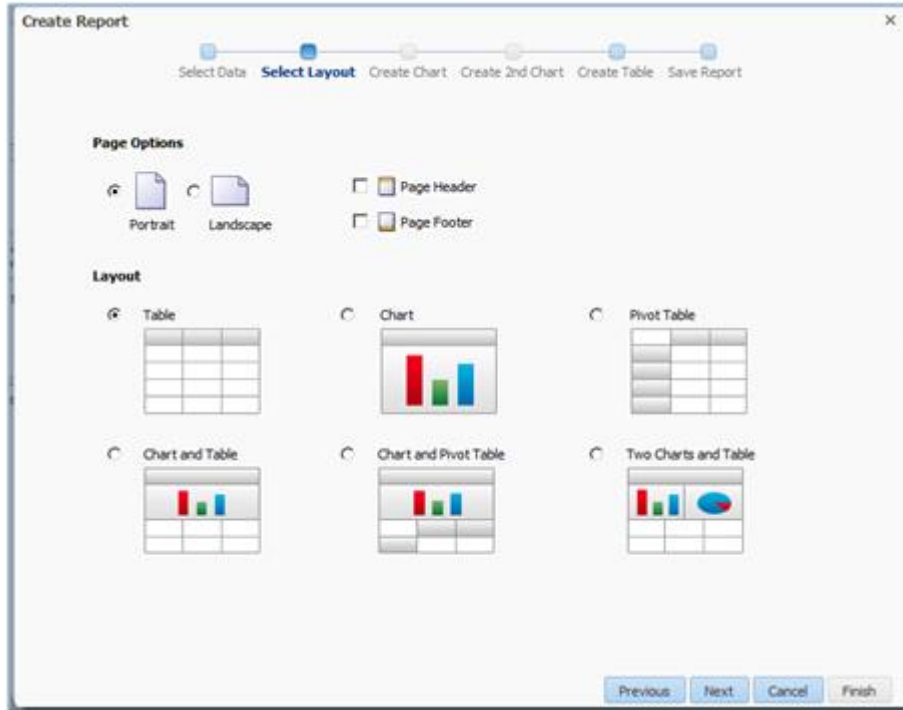
DESCRIPTION OF THE ILLUSTRATION 9_BI_CATALOG_NEW_REPORT.PNG

32. IN THE SELECT DATA DIALOG, FOR DATA MODEL, SELECT THE DATA MODEL YOU CREATED, ACCEPT THE OTHER DEFAULT SELECTIONS, AND THEN CLICK **NEXT**.



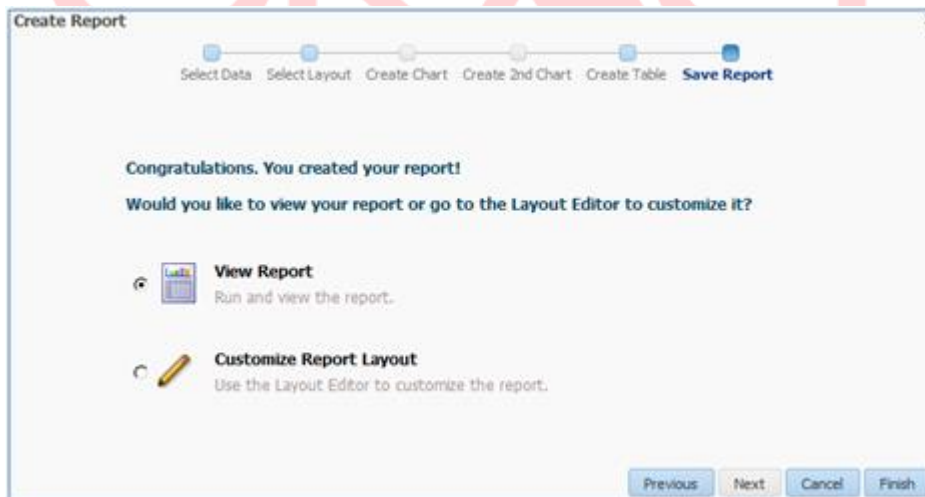
DESCRIPTION OF THE ILLUSTRATION 10_SELECT_DATA_DIALOG.PNG

33. IN THE SELECT LAYOUT DIALOG, ACCEPT THE DEFAULT REPORT LAYOUT SELECTIONS, AND THEN CLICK **NEXT**.



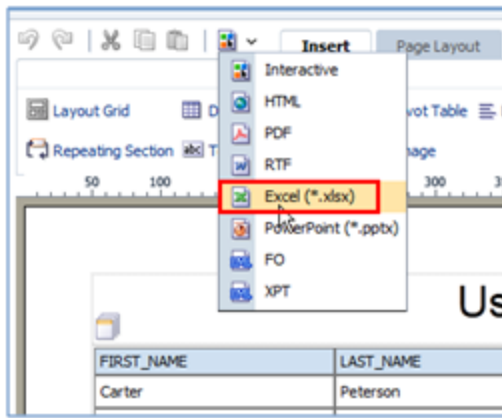
DESCRIPTION OF THE ILLUSTRATION 11_SELECT_LAYOUT_DIALOG.PNG

34. IN THE CREATE TABLE DIALOG, DRAG AND DROP THE **EMAIL** COLUMN.
35. DESELECT THE **SHOW GRAND TOTALS ROW** CHECK BOX, AND THEN CLICK **NEXT**.
36. IN THE SAVE REPORT DIALOG, SELECT **VIEW REPORT** AND THEN CLICK **FINISH**.



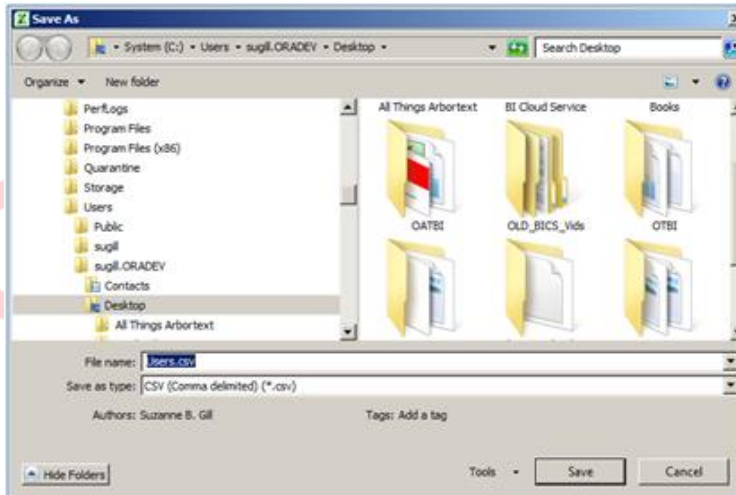
DESCRIPTION OF THE ILLUSTRATION 14_SAVE_REPORT_DIALOG.PNG

37. SAVE THE REPORT.
- a. IN THE SAVE AS DIALOG, NAME THE REPORT **USERS**, AND SAVE IT IN THE DRAFTS FOLDER (UNDER MY FOLDERS), THEN CLICK **OK**
- THE LAYOUT EDITOR AUTOMATICALLY DISPLAYS THE USERS REPORT.



DESCRIPTION OF THE ILLUSTRATION 15A_SAVE_AS_OUTPUT_DROPDOWN.PNG

- b. FROM THE OUTPUT DROP-DOWN LIST, SELECT YOUR DESIRED SPREADSHEET APPLICATION; FOR EXAMPLE, **EXCEL (*.XLSX)**.
- c. IN THE OPEN DIALOG, OPEN THE REPORT WITH THE DEFAULT APPLICATION, MICROSOFT EXCEL.
- d. IN MICROSOFT EXCEL, FROM THE SAVE AS MENU, SELECT SAVE AS TYPE **CSV**, SELECT WHERE YOU WANT TO STORE THE FILE, AND THEN CLICK **SAVE**.



DESCRIPTION OF THE ILLUSTRATION 15D_SAVE_AS_DIALOG.PNG

38. IN THE CSV FILE, RENAME THE COLUMN TO **EMAIL** TO COMPLY WITH THE ORACLE CLOUD REQUIREMENTS FOR IMPORTING THIS FILE.
39. IMPORT THE CSV FILE WITH THE FUSION APPLICATIONS ACCOUNT USERS TO ORACLE CLOUD. FOR INSTRUCTIONS, SEE **IMPORTING A BATCH OF USER ACCOUNTS IN GETTING STARTED WITH ORACLE CLOUD**.

2. CREATING DATA MODEL

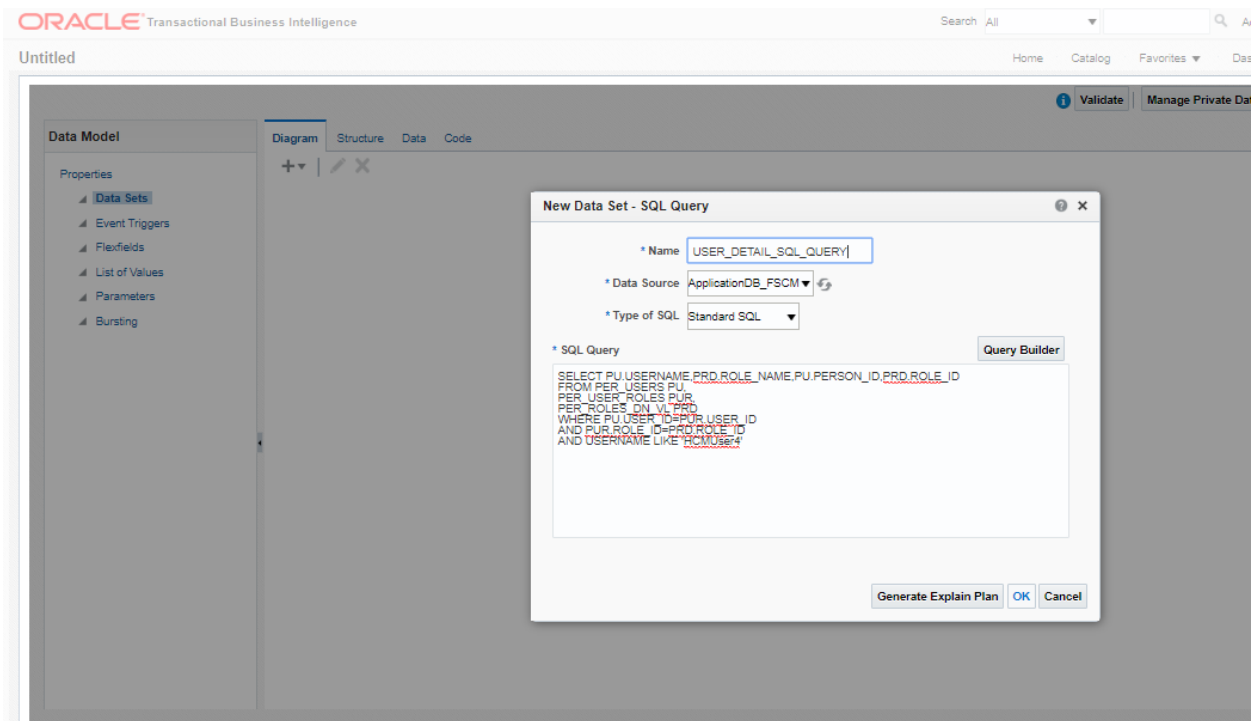
REPORT AND ANALYTIC > BROWSE CATALOG > NEW > DATA MODEL > SQL QUERY

STANDARD SQL THEN PASTE THIS

USER_DETAIL_SQL_QUERY:

```
SELECT      PU.USERNAME, PRD.ROLE_NAME, PU.PERSON_ID, PRD.ROLE_ID
FROM        PER_USERS PU,
            PER_USER_ROLES PUR,
            PER_ROLES_DN_VL PRD
WHERE       PU.USER_ID=PUR.USER_ID
AND        PUR.ROLE_ID=PRD.ROLE_ID
AND        USERNAME LIKE 'HCMUser4'
```

--HCMUser4 is the userid which opens the oracle cloud.

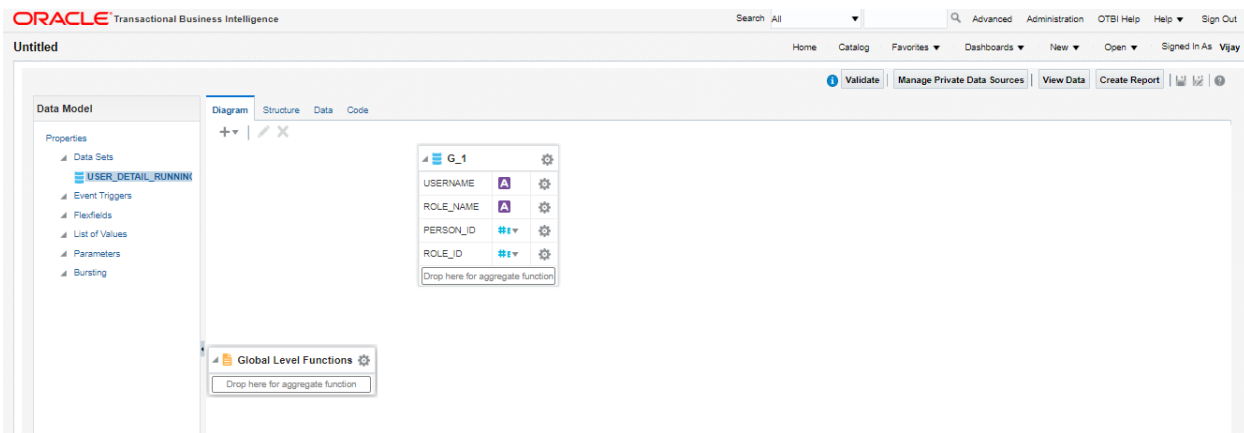


DATA >VIEW>SAVE IT IN A FOLDER AS **USER_DETAIL_SQL_QUERY** >SAVE AS SAMPLE DATA

3. CREATING REPORT

CREATE REPORT> select data model>

>GUIDE ME>NEXT>SELECT ANY >NEXT>



NEXT>DRAG THE DATA>UNCHECK THE BOX AS IT GIVES TOTAL ROWS>NEXT>VIEW REPORT>FINISH

SAVE IN FOLDER AS **USER_DETAIL_SQL_QUERY_LAYOUT**

SETTINGS IN WHATEVER FORMAT YOU WANT OR EDIT THE NAME..SAVE

YOU WILL SEE THE LAYOUT /REPORT NOW

ORACLE®

4. RUNNING THE REPORT USING CATALOG

GO TO THE CATALOG THERE U FIND THIS FILE **USER_DETAIL_SQL_QUERY_LAYOUT**

SEE MORE OPTION>SCHEDULE>

YOU WILL SEE MANY OPTIONS

GENERAL IS YOUR LAYOUT FILE

OUTPUT IS SET AS GIVEN BELOW

ORACLE Transactional Business Intelligence

Search All

Advanced Administration OTBI Help Help Sign Out

Home Catalog Favorites Dashboards New Open Signed In As Vijay

Schedule Report Job

Overview General Created by homuser4
Report Name ~/HCMUser4/Nikita/USER_DETAIL_SQL_QUERY_LAYOUT.xdo
Schedule Start immediately

Outputs Output1
Destination Notification

Return Submit

General Output Schedule Notification Diagnostic

☐ Make Output Public
☒ Save Data for Republishing
☐ Compress output prior to delivery

Output

Name	Layout	Format	Locale	Timezone	Calendar	Save Output
Output1	USER_DETAIL_SQL_Ql	PDF	English (India)	(UTC+05:30) Kolkata - India Time (IT)	Gregorian	<input checked="" type="checkbox"/>

Destination

There are five destination types: Email, Printer, Fax, FTP and Web folder. You can add multiple destinations as you need.

Destination Type Email Add Destination

SUBMIT>NAME THE REPORT>OK> GO TO THE CATALOG THERE U FIND THIS FILE **USER_DETAIL_SQL_QUERY_LAYOUT**

ORACLE Transactional Business Intelligence

Search All

Advanced Administration OTBI Help Help S

Home Catalog Favorites Dashboards New Open Signed In A

Report Job History

Last Refreshed Thu Mar 26, 2020 04:40:31 PM India Standard Time

Return

General Information

General Information

Report Job ID 2281
Report Job Name USER_DETAILS_SQL_QUERY_RUNNINGQ_REPORT__CATALOG
Owner homuser4
Report Name USER_DETAIL_SQL_QUERY_LAYOUT
Report Scope Private
Report Job Schedule 3/26/20 4:37:16 PM IST
Active Start Date
Active End Date
Trigger Data Model
Trigger Name
Trigger Retry Limit
Trigger Pause Time
Trigger Parameters

Report Job Execution Information

Report Job Status ✔ Success
Start Processing Time 3/26/20 4:37:26 PM IST
End Processing Time 3/26/20 4:37:26 PM IST
Time Elapsed 0.044 seconds

No parameters available

Output & Delivery

XML Data Diagnostic Log Republish

Output Name	Template	Format	Locale	Time Zone	Calendar	Status	Send
USER_DETAIL_RUNNING_REPO...	USER_DETAIL_SQL_QU...	PDF	English (India)	(UTC+05:30) Kolkata - India Time (IT)	Gregorian	✔	

SEE MORE OPTION>HISTORY>

CLICK ON REPORT NAME AS OUTPUT

ORACLE[®] Transactional Business Intelligence

Search All Advanced Administration

Report Job History Home Catalog Favorites Dashboards New

USERNAME	ROLE_NAME	PERSON_ID	ROLE_ID
HCMUser4	Application Diagnostics Regular User		3.00000001246388E 14
HCMUser4	IT Security Manager		3.0000000124639E 14
HCMUser4	Application Diagnostics Viewer		3.00000001246397E 14
HCMUser4	Application Implementation Consultant		3.00000001246353E 14
HCMUser4	Contingent Worker		3.00000001246371E 14
HCMUser4	Employee		3.00000001246372E 14
HCMUser4	Line Manager		3.00000001245373E 14
HCMUser4	Application Administrator		3.00000001246386E 14
HCMUser4	Compensation Administrator		3.00000001246449E 14
HCMUser4	Compensation Manager		3.00000001246453E 14
HCMUser4	Human Capital Management Integration Specialist		3.00000001246454E 14
HCMUser4	Payroll Manager		3.00000001246459E 14

Mar 26, 2020

USERNAME	ROLE_NAME	PERSON_ID	ROLE_ID
HCMUser4	Payroll Administrator		3.00000001246461E 14
HCMUser4	Recruiting Administrator		3.00000001246467E 14
HCMUser4	HRAAnalyst_ViewAll		3.0000000124801E 14
HCMUser4	HRSpecialist_ViewAll		3.00000001248034E 14

GO BACK..

SEE THE TIME ..

YOU CAN ALSO SEND THE INFORMATION IN THE MAIL.

ORACLE[®]

5. RUNNING THE REPORT FROM FUSION APPLICATION BY CREATING ESS JOB

GO TO THE CATALOG THERE U FIND THIS FILE **USER_DETAIL_SQL_QUERY_LAYOUT**

SEE MORE OPTION>SCHEDULE> NOTE THE LINK AS

1. REPORT ID: **/~HCMUser4/Nikita/USER_DETAIL_SQL_QUERY_LAYOUT.xdo**
2. GO TO NAVIGATOR >SETUP AND MAINTAINANCE >TASKS>MANAGE IMPLEMENTATION PROJECTS>
3. TAKE ANY PROJECT>

The screenshot displays the Oracle Fusion Applications user interface. At the top, a banner reads "You have a new home page!" with instructions to use the home icon or company logo. A "Settings and Actions" menu is open on the right, listing options like Personalization, Administration, Troubleshooting, and Print Me. The main navigation bar shows "Setup: Financials" and "Actions". Below this, the "Functional Areas" list includes "Initial Users", "Enterprise Profile", "Legal Structures", "Financial Reporting Structures", "General Ledger", "Organization Structures", "Resources", "Workforce Structures", and "Users and Security". The "Initial Users" task is selected, showing a table with columns for "Task" and "Scope". The "Task" column lists "Run User and Roles Synchronization Process". The "Scope" column is empty. A "Columns Hidden: 4" message is displayed. On the right side of the task list, a "Tasks" button is visible. Below the task list, a "Manage Implementation Projects" task is highlighted, with a list of related tasks: "Manage Implementation Projects", "Manage Export and Import Processes", "Manage Setup Content", "Review Topology", and "Search".

Implementation Projects

Search

Name

Advanced Saved Search All Implementation Projects

Search Reset Save...

Search Results

Actions View Format Freeze Detach Wrap

Name	Status	Assigned To	Start Date	Finish Date	Last Updated	Last Updated By
Implementation Project-Recruitment(SS)		HCMUser4	12-Mar-2020		12-Mar-2020	HCMUser4
Profile management Implementation Project		HCMUser4	01-Jan-1951		10-Mar-2020	HCMUser4
Implementation Project Demo Recruiting_2		HCMUser4	09-Mar-2020		09-Mar-2020	HCMUser4
Implementation Project Demo Recruiting		HCMUser4	09-Mar-2020		09-Mar-2020	HCMUser4
Implementation Project Recruiting		HCMUser4	09-Mar-2020		09-Mar-2020	HCMUser4
Qatar Corporation		HCMUser4	01-Jan-2019		10-Feb-2020	HCMUser4
SRY Fin Implementation project		sry.homuser	25-Dec-2019		25-Dec-2019	sry.homuser
Payroll Implementation Project		HCMUser4	01-Jan-2019		24-Dec-2019	HCMUser4
TS HCM Cloud Implementation1		HCMUser2	30-Mar-2017		03-Dec-2019	HCMUser4
Sry Implementation Project		sry.homuser	09-Oct-2019		10-Oct-2019	sry.homuser

Columns Hidden 2 Columns Frozen 1

4. TASK: MANAGE ENTERPRISE SCHEDULER JOB >SEARCH.
5. GO TO TASK>+ CREATE > DISPLAY REPORT : **USER_DETAIL_SQL_QUERY_REPORT**
6. PATH : MORE>PRPERTIES OF LAYOUT
7. GIVE REQUIRED FIELD.
8. JO APPLICATION NAME: FINANCIAL
9. JOB TYPE BIP
10. OUTPUT: PDF
11. REPORT ID:MENTION ABOVE

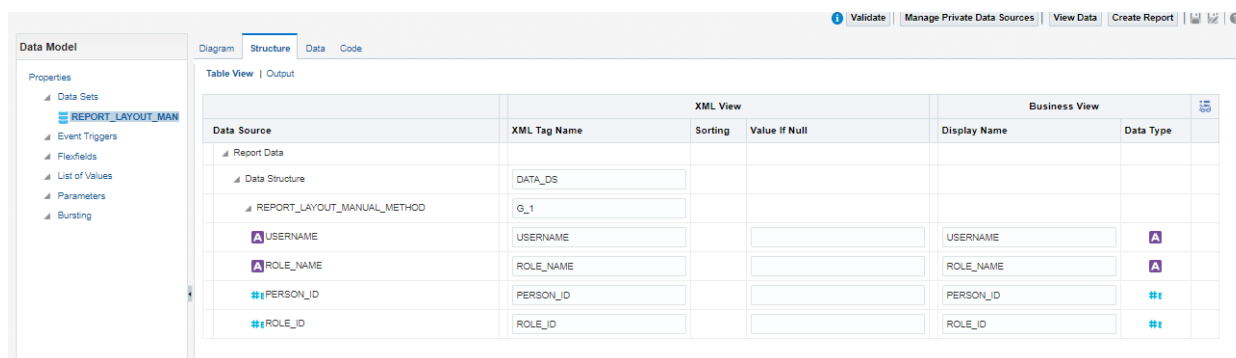
SAVE>

GO TO NAVIGATOR > SCHEDULE PROCESS>SCHEDULE NEW PROCESS>OK>OK>ADVANCE >OUTPUT FORMAT: PDF>PRINT OUTPUT>SUBMIT>CLICK ON JOB> REFRESH AND SEE OUTPUT>SAVE>USER_DETAIL_REPORT AS ALREADY SAVED.

6. REPORT LAYOUT CREATION USING MANUAL METHOD 1

USER_DETAILS_MANUAL_LAYOUT:

```
SELECT PU.USERNAME,
       PRD.ROLE_NAME,
       PU.PERSON_ID,
       PRD.ROLE_ID
FROM   PER_USERS PU,
       PER_USER_ROLES PUR,
       PER_ROLES_DN_VL PRD,
WHERE  PU.USER_ID = PUR.USER_ID
AND    PUR.ROLE_ID = PRD.ROLE_ID
AND    USERNAME LIKE 'HCMuser4'
```



GO TO STRUCTURE

TAKE WORD FILE AND DRAW AS SHOWN USING GROUP ID SAVE AS **RTF FILE**

USERNAME	ROLENAME	ROLE ID
<?for-each:G_1?><?USERNAME?>	<?ROLE_NAME?>	<?ROLE_ID?><?end for-each?>

<?for-each:G_1?> <?USERNAME?>

<?ROLE_NAME?>

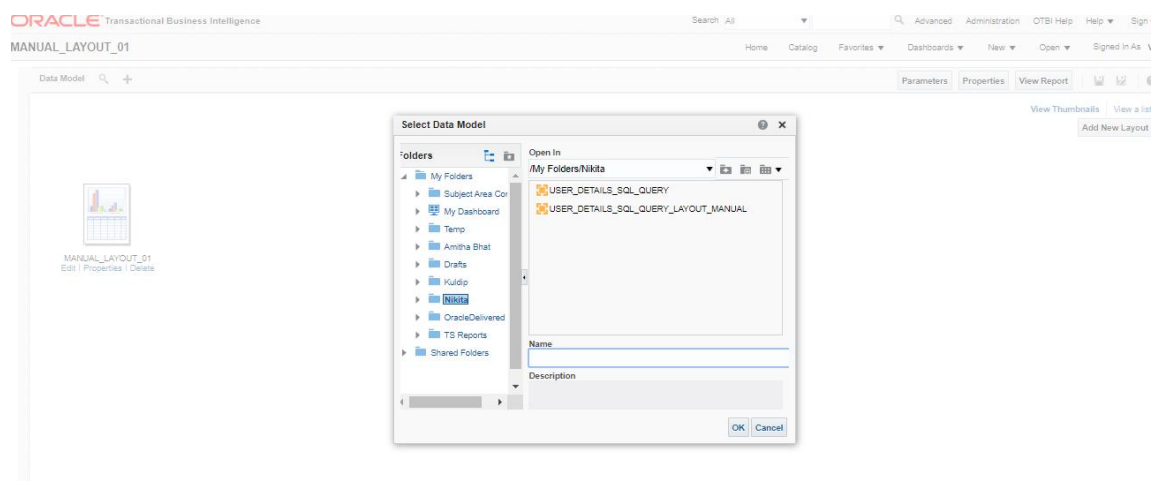
<?ROLE_ID?><?end for-each?>

ADD INS > VALIDATE>

OR WE CAN JUST GO TO DATA>VIEW>SAVE IT>

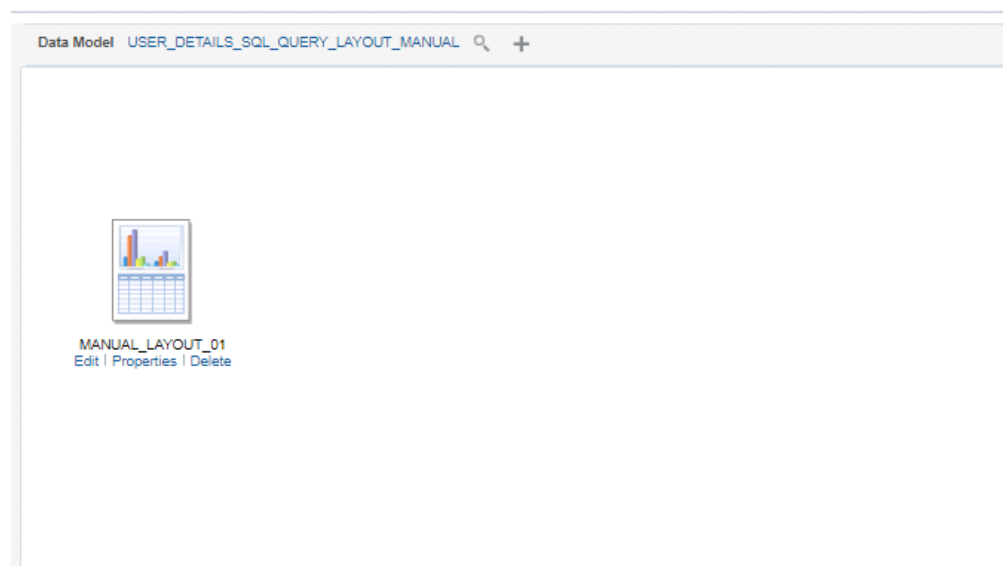
NOW

CREATE REPORT>CANCEL>UPLOAD **RTF FILE** >AS RTF>ENGLISH>UPLOAD.

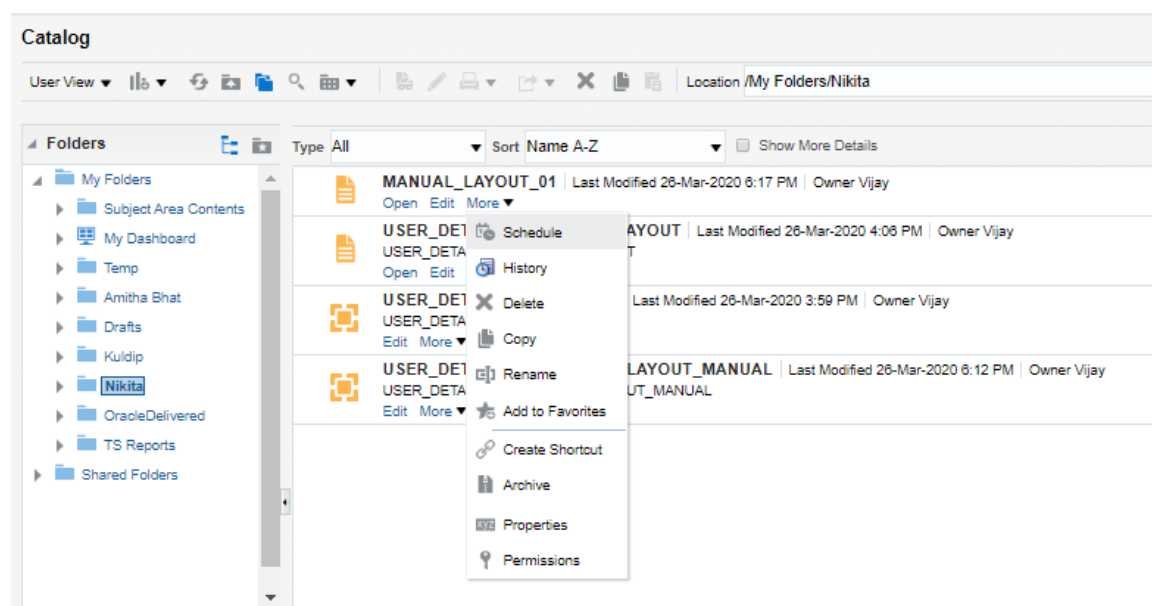


SAVE IT>

AGAIN CHOOSE DATAMODEL IT IS LINKED TO



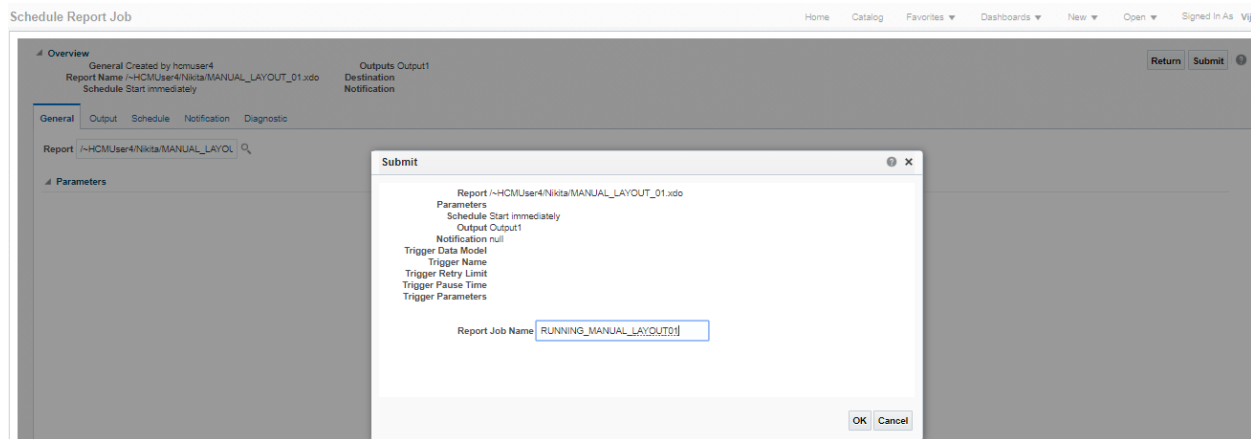
CATALOG>SCHEDULE THE LAYOUT FOR RUNNING



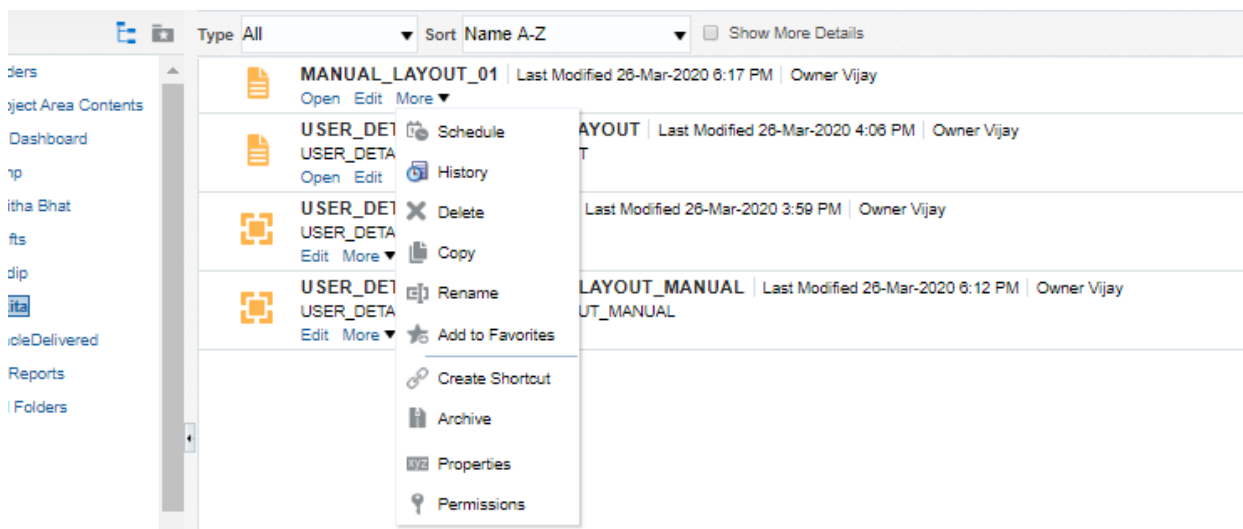
COPY THE REPORT ID: /~HCMUser4/Nikita/MANUAL_LAYOUT_01.xdo

FOR ESS JOB CREATION

SUBMIT> NAME IT>OK TO SEE THE OUTPUT.



GO TO CATALOG>LAYOUT>HISTORY>RUN THE OUTPUT



CLICK ON OUTPUT

ORACLE Transactional Business Intelligence

Search All

Report Job History Home Catalog Favorites

Last Refreshed Thu Mar 26, 2020 06:32:49 PM India Standard Time

General Information

General Information

Report Job ID: 2283
 Report Job Name: RUNNING_MANUAL_LAYOUT01
 Owner: homuser4
 Report Name: MANUAL_LAYOUT_01
 Report Scope: Private
 Report Job Schedule: 3/26/20 6:28:30 PM IST
 Active Start Date:
 Active End Date:
 Trigger Data Model:
 Trigger Name:
 Trigger Retry Limit:
 Trigger Pause Time:
 Trigger Parameters:

Report Job Execution Information

Report Job Status: ✔ Success
 Start Processing Time: 3/26/20 6:28:40 PM IST
 End Processing Time: 3/26/20 6:28:41 PM IST
 Time Elapsed: 0.000 seconds

No parameters available

Output & Delivery

XML Data Diagnostic Log **Republish**

Status: All

Output Name	Template	Format	Locale	Time Zone	Calendar	Status	Send
. Output1	MANUAL_LAYOUT_01	HTML	English (India)	(UTC+05:30) Kolkata - India Time (IT)	Gregorian	✔	

CLICK ON OUTPUT TO SAVE IN SYSTEM

reports and analytics - reports - Oracle BI Publisher - Report Job

ee1-test-fa.ap4.oraclecloud.com/analytics/saw.dll?bipublisherEntry&Action=history&iten

ORACLE Transactional Business Intelligence

Report Job History

Last Refreshed Thu Mar 26, 2020 06:32:49 PM India Standard Time

General Information

General Information

Report Job ID: 2283
 Report Job Name: RUNNING_MANUAL_LAYOUT01
 Owner: homuser4
 Report Name: MANUAL_LAYOUT_01
 Report Scope: Private
 Report Job Schedule: 3/26/20 6:28:30 PM IST
 Active Start Date:
 Active End Date:
 Trigger Data Model:
 Trigger Name:
 Trigger Retry Limit:
 Trigger Pause Time:
 Trigger Parameters:

No parameters available

Output & Delivery

XML Data Diagnostic Log **Republish**

Status: All

Output Name: . Output1
 Template: MANUAL_LAYOUT_01
 Format: HTML
 Locale: English (India)
 Time Zone: (UTC+05:30) Kolkata - India Time (IT)
 Calendar: Gregorian
 Status: ✔
 Send:

Downloads

File Home Share View

Users > abc > Downloads >

MANUAL_LAYOUT_01

175 items 1 item selected 10.5 KB

Open

- Scan with ByteFence Anti-Malware...
- Share files via SHAREit
- Edit with Notepad++
- Share
- Open with
- Give access to
- Add to archive...
- Add to "MANUAL_LAYOUT_01.rar"
- Compress and email...
- Compress to "MANUAL_LAYOUT_01.rar" and email
- Restore previous versions
- Send to
- Cut
- Copy
- Create shortcut
- Delete
- Rename
- Properties

Open with

- Code Writer
- Firefox
- Google Chrome
- Internet Explorer
- Microsoft Edge
- Search the Store
- Choose another app

OUTPUT

USERNAME	ROLE NAME	ROLE ID
HCMUser4	Application Diagnostics Regular User	300000001246388
HCMUser4	IT Security Manager	300000001246390
HCMUser4	Application Diagnostics Viewer	300000001246397
HCMUser4	Application Implementation Consultant	300000001246353
HCMUser4	Contingent Worker	300000001246371
HCMUser4	Employee	300000001246372
HCMUser4	Line Manager	300000001246373
HCMUser4	Application Administrator	300000001246386
HCMUser4	Compensation Administrator	300000001246449
HCMUser4	Compensation Manager	300000001246453
HCMUser4	Human Capital Management Integration Specialist	300000001246454
HCMUser4	Payroll Manager	300000001246459
HCMUser4	Payroll Administrator	300000001246461
HCMUser4	Recruiting Administrator	300000001246467
HCMUser4	HRAnalyst_ViewAll	300000001248010
HCMUser4	HRSpecialist_ViewAll	300000001248034
HCMUser4	GFI ESS Monitor Role	300000004087469
HCMUser4	Gfi Report and Analytics Custom(FSCM)	300000004087546

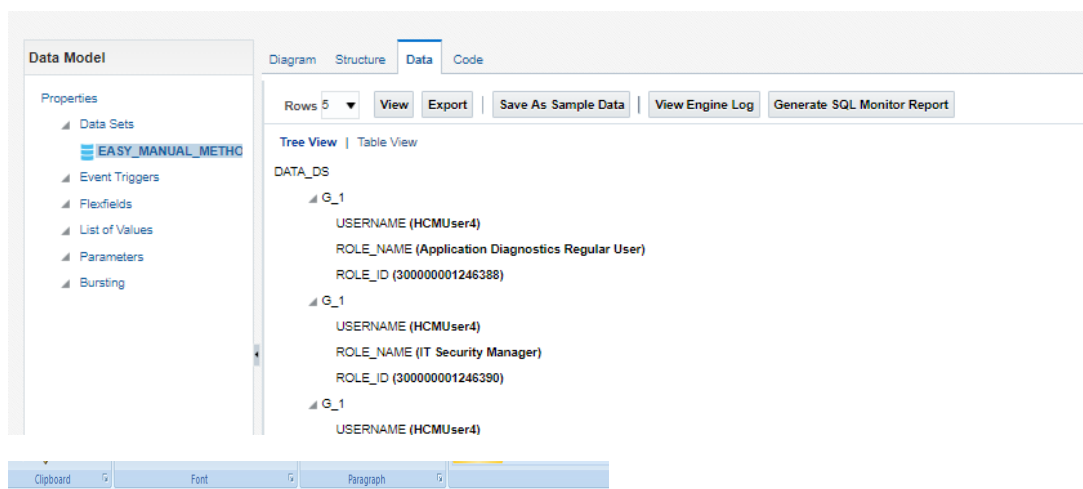
TRY WITH ESS JOB.

ORACLE®

7. RUNNING FOR MORE THAN ONE ROW MANUAL METHOD 02 XML TO RTF

SAME METHOD AS ABOVE BUT YOU NEED TO EXPORT THE FILE AS XML

SAVE THIS DATA MODEL>VIEW>SAVE AS SAMPLE DATA> EXPORT>SAVE>



```

<DATA_DS>
  <G_1>
    <USERNAME>HCMUser4</USERNAME>
    <ROLE_NAME>Application Diagnostics Regular User</ROLE_NAME>
    <ROLE_ID>300000001246388</ROLE_ID>
  </G_1>
  <G_1>
    <USERNAME>HCMUser4</USERNAME>
    <ROLE_NAME>IT Security Manager</ROLE_NAME>
    <ROLE_ID>300000001246390</ROLE_ID>
  </G_1>
  <G_1>
    <USERNAME>HCMUser4</USERNAME>
    <ROLE_NAME>Application Diagnostics Viewer</ROLE_NAME>
    <ROLE_ID>300000001246397</ROLE_ID>
  </G_1>
  <G_1>
    <USERNAME>HCMUser4</USERNAME>
    <ROLE_NAME>Application Implementation Consultant</ROLE_NAME>
    <ROLE_ID>300000001246353</ROLE_ID>
  </G_1>
</DATA_DS>

```

OPEN THE XML FILE WITH WORD

NOW TAKE A NEW WORD FILE>ADD-INS>DATA>LOAD XML DATA>SELECT>IF THE DATA IS SUCCESSFULLY LOADED THEN SELECT>ADD-INS>INSERT

OR IF YOU WANT TO INSERT MULTIPLE THEN CLICK INSERT>FIELD>SAVE AS RTF.

CREATE THE REPORT>CANCEL>UPLOAD>BROWSE>OK

SEARCH DATA MODEL>SAVE>CATALOG>MORE>SCHEDULE>OUTPUT>PDF>SUBMIT>JOB NAME>OK>CATALOG>HISTORY>OUTPUT.

8. REPORT LAYOUT PROPERTIES EXPLAINED - GENERAL

REPORT LAYOUT PROPERTIES.

TAKE ANY LAYOUT > EDIT > PROPERTIES>GENERAL

1. RUN REPORT: FOR RUNNING REPORT ONLINE, IF UNCHECKED THEN YOU CAN'T SCHEDULE THE RUN.
2. SHOW CONTROLS: YOU DON'T SEE CONTROLS IF UNCHECKED.
3. ALLOWS SHARING REPORT LINK: VIEW REPORT > SETTING > SHARE LINK ENABLED, IF UNCHECKED THEN NOT ENABLES.
4. OPEN LINKS IN NEW WINDOW: SO THIS WILL ALLOW LINK IN NEW WINDOW OTHERWISE ON SAME WINDOW.
5. AUTO RUN: CATALOG>OPEN> AUTORUN

ADVANCE: IF 3RD OPTION IS UNCHECKED THEN ONLY YOU ARE ABLE TO RUN THE REPORT.

9. REPORT LAYOUT PROPERTIES EXPLAINED - CACHING

1. ENABLE DATA CACHING MEANS AFTER THAT TIME YOU MAY NOT BE ABLE TO ACCEPT THE DATA FROM CACHING YOU HAVE TO TAKE IT FROM DATABASE. USUALLY DISABLED BUT USED DURING PERFORMANCE ISSUES.
2. USER LEVEL: FOR EVERY RUN, CACHING IS DONE ON USER LEVEL (USER 1, USER 2, USER 3) NOT ACROSS THE APPLICATION.
3. DOCUMENT CACHING: WHEN USER USES THE OUTPUT THE DOCUMENT IS CACHED.
4. USER CAN REFRESH THE USER DATA: AUTOMATU=ICALLY WE CAN RUN AND USER CAN REFRESH THE DATA OTHERWISE ONCE FETCHED THE DATA CAN'T BE REFRESHED IF UNCHECKED.

10. REPORT LAYOUT PROPERTIES EXPLAINED - FORMATTING

1. ENABLE PDF SECURITY : TRUE
 2. SERVER VALUE : FALSE
 3. DOCUMENT PASSWORD
 4. MODIFICATION PASSWORD
 5. ENABLE DIGITAL SIGNATURE : FALSE
- >OK>VIEW REPORT>DOCUMENT TYPE PDF>PASSWORD> BUT OUTPUT WILL NOT GET IN ONLINE MODE
 CATALOG>LAYOUT>MORE>SCHEDULE>OUTPUT>PDF>SUBMIT>RENAME IT /SAVE>OK
 CATALOG>REPORT LAYOUT>HISTORY>REMOVE DATE>SEARCH>SUCCESS>CLICK >PASSWORD

11. REPORT LAYOUT PROPERTIES EXPLAINED - CURRENCY FORMAT

CURRENCY CODE: INR

56,789.12

FORMAT MASK: 99G999D99

G: GROUPING, D: DECIMAL

<?fOrmat-currency:COLUMNNAME;' INR' ?>

SO FORMAT MASK WILL BE DEFAULTED HERE:

<?fOrmat-currency:COLUMNNAME;' 99G999D99' ?>

12. STYLE SHEET TEMPLATE

1. CREATE RTF FILE WITH HEADER AND FOOTER IN IT.

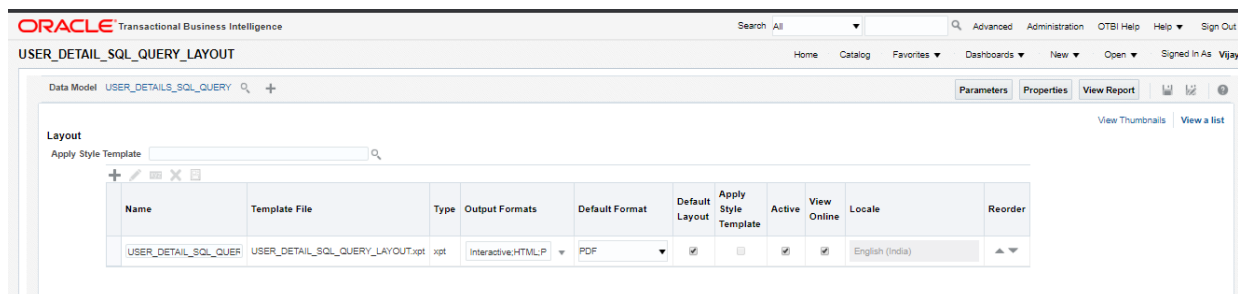
WORD>NEW>NEW STYLE SHEET>SAVE IT IN THE FORMAT AS RTF.

2. UPLOAD TO CATALOG TO ANY FOLDER AS STYLE SHEET.

NEW>STYLE TEMPLATE>UPLOAD>BROWSE>RTF>ENGLISH>SAVE AS.>OK

3. GO TO ANY REPORT LAYOUT AND APPLY THIS NEWLY CREATED STYLE SHEET.

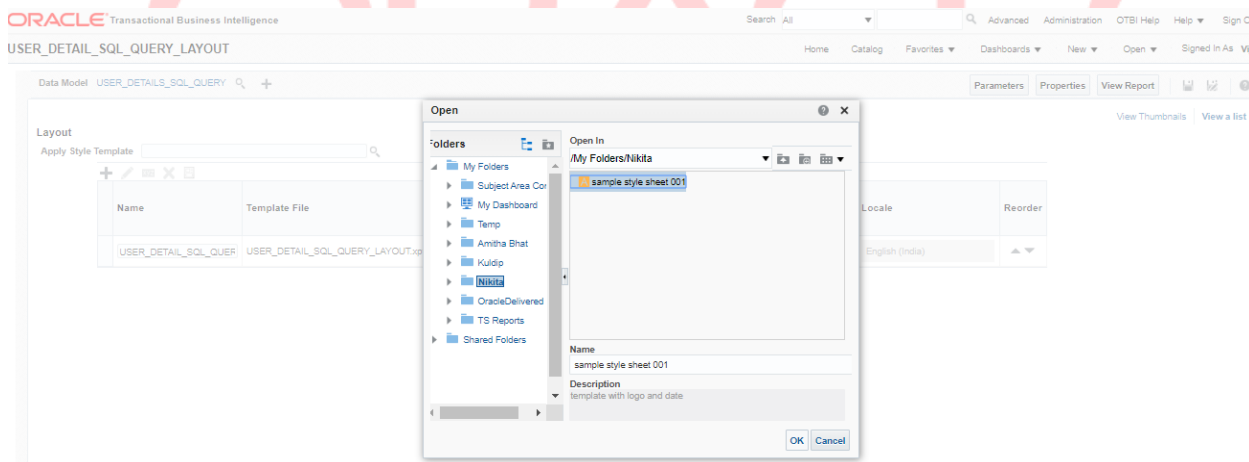
CATALOG AND EDIT ANY OF THE LAYOUT.> VIEW A LIST>APPLY STYLE SHEET >PDF>APPLY STYLE TEMPLATE SHOULD BE CHECKED.



4. RUN THE REPORT.

CATALOG>SCHEDULE>OUTPUT>RENAME>OK

CATALOG>HISTORY>REMOVE DATE>OUTPUT>RENAME>OK



MORE THAN ONE LAYOUT:

CATALOG>EDIT>ADD LAYOUT>PORTRAIT>INSERT>DATA TABLE>SELECT>HIGHLIGHT FOR MANIPULATION >DONE>OK

OR UPLOAD NEW RTF> REPEAT AS ABOVE

RUN WILL NOT SHOW BOTH LAYOUTS

13. MASTER DETAIL REPORT USING ELEMENT LEVEL LINK

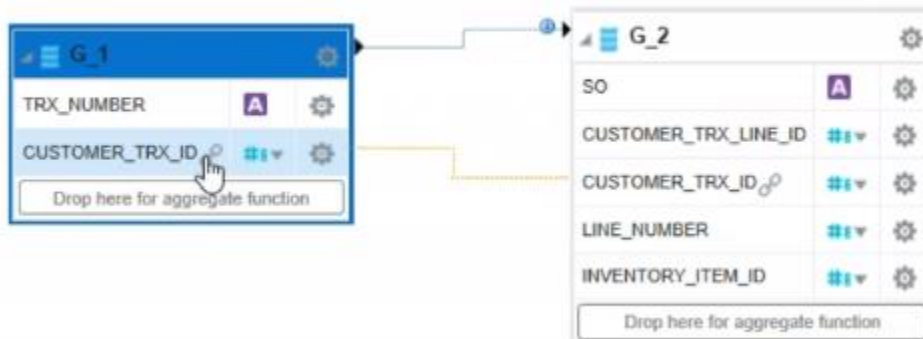
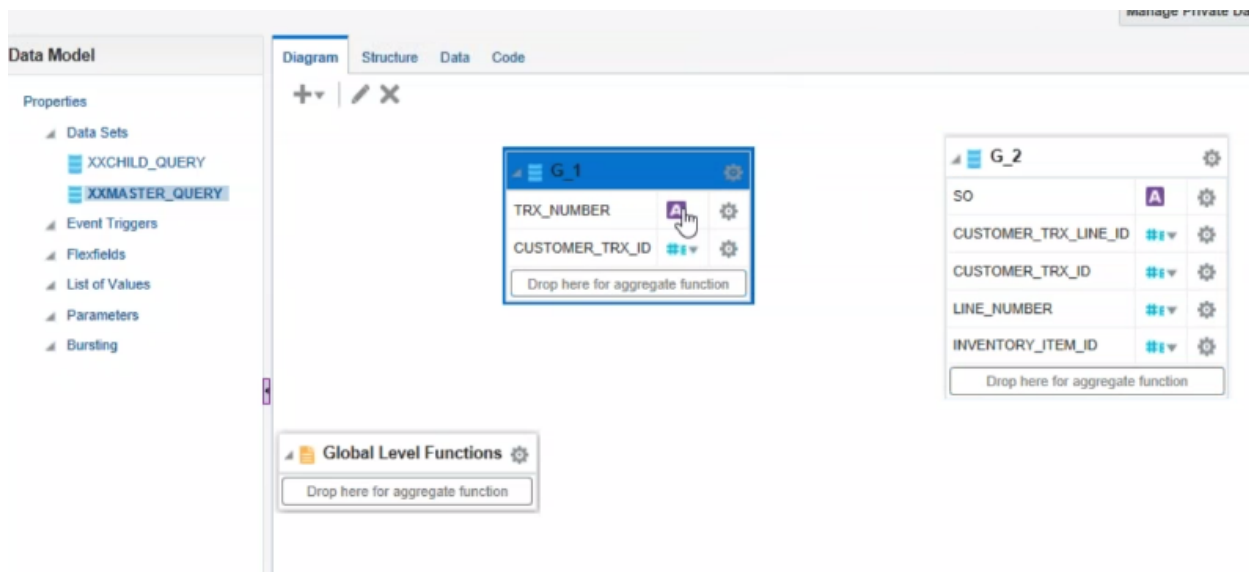
NEW DATA MODEL>

XXMASTER_QUERY:

```
SELECT TRX_NUMBER, CUSTOMER_TRX_ID, CUSTOMER_TRX_ALL WHERE ROWNUM <= 1;
```

XXCHILD_QUERY:

```
SELECT INTERFACE_LINE_ATTRIBUTE1 SO,  
       CUSTOMER_TRX_LINE_ID,  
       CUSTOMER_TRX_ID,  
       LINE_NUMBER,  
       INVENTORY_ITEM_ID  
FROM RA_CUSTOMER_TRX_LINES_ALL
```



14. MASTER DETAIL REPORT USING GROUP LEVEL LINK

MASTER QUERY:

```
SELECT    TRX_NUMBER, CUSTOMER_TRX_ID
FROM      RA_CUSTOMER_TRX_ALL CUST_ID
WHERE     ROWNUM<=1
```

CHILD QUERY:

```
SELECT    INTERFACE_LINE_ATTRIBUTE1 SO,
          A.CUSTOMER_TRX_LINE_ID,
          A.CUSTOMER_TRX_ID CUST_ID,
          A.LINE_NUMBER,
          A.INVENTORY_ITEM_ID
FROM      RA_CUSTOMER_TRX_LINES_ALL A
WHERE     CUSTOMER_TRX_ID = CUST_ID
```

GROUP LEVEL MENU>CREATE GROUP LINK>SAVE>DATA>VIEW>SAVE AS>CREATE REPORT>CANCEL>SELECT DATA MODEL>TEMP NAME: **XXMASTER_REPORT_GROUP_LEVEL**>GENERATE>SAVE>VIEW REPORT.

ORACLE®

15. MASTER DETAIL REPORT USING SUBGROUPS

NEW>DATA MODEL>FOR SINGLE QUERY

XXMASTER_QUERY_METHOD2

```
SELECT      A.INTERFACE_LINE_ATTRIBUTE1 SO,
            A.CUSTOMER_TRX_LINE_ID,
            A.CUSTOMER_TRX_ID,
            A.LINE_NUMBER,
            A.INVENTORY_ITEM_ID
            B.TRX_NUMBER,
            B.CUSTOMER_TRX_ID
FROM        RA_CUSTOMER_TRX_LINES_ALL A
            RA_CUSTOMER_TRX_ALL B
WHERE       A.CUSTOMER_TRX_ID = B.CUSTOMER_TRX_ID
AND         A.CUSTOMER_TRX_ID = 1005
```

SO TAKE MASTER COLUMNS: CLICK MENU OF ELEMENTS > GROUP BY

YOU WILL SEE DIFFERENT GROUPS THERE SELECT ELEMENT > GROUP BY

OTHER ELEMENT>MORE >MOVE TO.. YOU WILL SEE OTHER COLUMNS ALSO ACCORDINGLY.

DATA>VIEW>SAMPLE DATA>SAVE>CREATE REPORT>CANCEL>SELECT DATA MODEL>OK>GENERATE>SAVE>VIEW REPORT.

DATA TO MOVE ACROSS PARENT AND CHILD: EDIT

ORACLE®

16. AGGREGATE FUNCTION

AVAILABLE AGGREGATE FUNCTIONS ARE AS FOLLOWS:

FUNCTIONS	EXAMPLE	DESCRIPTION
AVG	AVG(SALES)	CALCULATING AVERAGE VALUE
BIN	UNIT PRICE BY PRODUCT NAME	SELECTS ANY NUMERIC ATTRIBUTE FROM A DIMENSION, FACT TABLE OR MEASURE CONTAINING DATA VALUES AND PLACES THEM INTO DISCRETE NUMBER OF BINS. THIS FUNCTION IS TREATED LIKE NEW DIMENSION ATTRIBUTE FOR PURPOSES LIKE AGGREGATION, FILTERING, DRILLING.
COUNT	PRODUCTS	DETERMINES THE NO. OF ITEMS WITH A NON NULL VALUES
FIRST	SALES	SELECTS FIRST NOT NULL RETURNED VALUE
LAST	SALES	SELECTS LAST NOT NULL RETURNED VALUE
MAX	REVENUE	CALCULATE MAX HIGHEST VALUE OF THE ROW
MEDIAN	SALES	CALCULATE MIDDLE VALUE OF THE ROW
MIN	REVENUE	CALCULATE LOWEST MINIMUM VALUE OF THE ROW
Std_DEV	SALES DISTINCT SALES	RETURNS STANDARD DEVIATION FOR A SET OF VALUES. RETURN TYPE IS ALWAYS DOUBLE
Std_DEV_P OP	SALES DISTINCT SALES	RETURNS STANDARD DEVIATION FOR A SET OF VALUES USING A COMPUTATIONAL FORMULA FOR POPULATION VARIANCE STANDARD DEVIATION.
SUM	REVENUE	CALCULATES THE SUM OBTAINED.

DATA MODEL>

```

SELECT      A.INTERFACE_LINE_ATTRIBUTE1 SO,
            A.CUSTOMER_TRX_LINE_ID,
            A.CUSTOMER_TRX_ID CUST_ID,
            A.LINE_NUMBER,
            A.INVENTORY_ITEM_ID
            B.TRX_NUMBER,
            B.CUSTOMER_TRX_ID,
            A.QUANTITY_INVOICED,
            A.UNIT_SELLING_PRICE,
            A.LINE_TYPE,
            A.EXTENDED_AMOUNT,
            A.REVENUE_AMOUNT,
            A.TAX_RATE,
FROM        RA_CUSTOMER_TRX_LINES_ALL A
            RA_CUSTOMER_TRX_ALL B
WHERE       A.CUSTOMER_TRX_ID = B.CUSTOMER_TRX_ID
AND         A.CUSTOMER_TRX_ID = 1005

```

DRAW AND DROP THE COLUMN TO GLOBAL LEVEL

MENU>PROPERTIES>ALIAS SHOULD BE IN CAPS>FUNCTION: TAKE ANY
DATA>VIEW>TABLE>TOTAL DISPLAYED AT THE BOTTOM.

SO NEAR THE GLOBAL THERE IS A DROP DOWN FOR FUNCTIONS.

FOR MIN VALUE AND MAX VALUE ALSO

SAVE SAMPLE>SAVE > (CREATE REPORT/NEW >REPORT)>CANCEL>SELECT DATA
MODEL>GENERATE>NAME IT>EDIT TO DOWNLOAD>EDIT WITH HEADER FOOTER AND ALSO KEEP
THE TABLES IN PROPER MANNER>NO DELETE OLD LAYOUT>ADD NEW
LAYOUT>UPLOAD>SAVE>CATALOG>LAYOUT>SCHEDULE AND HISTORY>OUTPUT>SUBMIT>OUTPUT
NEW SQL LAYOUT IN SAME DATA MODEL AND REPEAT ABOVE SUBGROUP PROCEDURE.

17. ARITHMETIC OPERATION ON COLUMNS

WE CAN PERFORM ARITHMETIC OPERATIONS DIRECTLY ON OR MORE COLUMNS. BY CREATING NEW COLUMN IN DATA SET SIMILAR TO PLACE HOLDER COLUMN IN RTF.

DATA MODEL: **XXEXP_1**

```

SELECT      A.INTERFACE_LINE_ATTRIBUTE1 SO,
            A.CUSTOMER_TRX_LINE_ID,
            A.CUSTOMER_TRX_ID CUST_ID,
            A.LINE_NUMBER,
            A.INVENTORY_ITEM_ID
            B.TRX_NUMBER,
            B.CUSTOMER_TRX_ID,
            A.QUANTITY_INVOICED,
            A.UNIT_SELLING_PRICE,
            A.LINE_TYPE,
            A.EXTENDED_AMOUNT,
            A.REVENUE_AMOUNT,
            A.TAX_RATE,
FROM        RA_CUSTOMER_TRX_LINES_ALL A
            RA_CUSTOMER_TRX_ALL B
WHERE       A.CUSTOMER_TRX_ID = B.CUSTOMER_TRX_ID
AND         A.CUSTOMER_TRX_ID = 1005

```

G_1>MENU>ADD ELEMENT BY EXPRESSION>E_XX_CALCULATED>RETURN TYPE STRING

```

COLUMN1*COLUMN2 == C_MUL
COLUMN1+COLUMN2 == C_SUM
COLUMN1/COLUMN2 == C_DIV
QUANTITY_INVOICED * UNIT_SELLING_PRICE = E_MUL_CALCULATED
QUANTITY_INVOICED + UNIT_SELLING_PRICE = E_SUM_CALCULATED
QUANTITY_INVOICED - UNIT_SELLING_PRICE = E_SUB_CALCULATED

```

DATA>VIEW>TABLE VIEW

18. EXPRESSIONS - IF (EXP) WITH EXAMPLE

WE CAN PERFORM ARITHMETIC OPERATIONS DIRECTLY ON OR OR MORE COLUMNS. BY CREATING NEW COLUMN IN DATA SET SIMILAR TO PLACE HOLDER COLUMN IN RTF.

DATA MODEL: **XXEXP_1**

```

SELECT      A.INTERFACE_LINE_ATTRIBUTE1 SO,
            A.CUSTOMER_TRX_LINE_ID,
            A.CUSTOMER_TRX_ID CUST_ID,
            A.LINE_NUMBER,
            A.INVENTORY_ITEM_ID
            B.TRX_NUMBER,
            B.CUSTOMER_TRX_ID,
            A.QUANTITY_INVOICED,
            A.UNIT_SELLING_PRICE,
            A.LINE_TYPE,
            A.EXTENDED_AMOUNT,
            A.REVENUE_AMOUNT,
            A.TAX_RATE,
FROM        RA_CUSTOMER_TRX_LINES_ALL A
            RA_CUSTOMER_TRX_ALL B
WHERE       A.CUSTOMER_TRX_ID = B.CUSTOMER_TRX_ID
AND         A.CUSTOMER_TRX_ID = 1005

```

G_1>MENU>ADD ELEMENT BY EXPRESSION>E_IF_TAX >RETURN TYPE STRING

IF (BOOLEAN_EXPR, TRUE_RETURN, FALSE_RETURN)

CLICK f(...)

EXAMPLE: IF (G_1.DEPARTMENT_ID == 10, 'VALID', 'NOT VALID')

**--RETURNS 'VALID' IF DEPARTMENT_ID = 10,
OTHERWISE RETURNS 'NOT VALID'**

IF (G_1.TAX_RATE<=5, 'LOW', 'HIGH')
IF (G_1.LINE_TYPE='LINE', 'INVOICE LINE', 'TAX LINE')
 SAVE>DATA>VIEW>TABLE VIEW.

WE CAN PERFORM BELOW FUNCTIONS ON ONE OR MORE COLUMNS.

Edit Properties - ExpressionElement

*Name: ExpressionElement

Display Name: ExpressionElement

Data Type: String

Available:

- G_1
- PERSON_NUMBER
- Parameters
- ComparisonValue

*Express:

IF(G_1.F... than , param.Cc... param.Cc...

f(...) ▼

IF
NOT
AND
OR
MAX
MIN
ROUND
FLOOR
CEILING
ABS
AVG
LENGTH
SUM
NVL
CONCAT
STRING
SUBSTRING
INSTR
DATE
FORMAT_DATE
FORMAT_NUMBER
DECODE

19. EXPRESSIONS - NOT WITH XDO USER REPORT LOCALE

NOT LOGICAL NOT OPERATOR
EVALUATES BOOLEAN_EXPR, AND RETURNS TRUE IF BOOLEAN_EXPR IS FALSE.
STRING(NOT(BOOLEAN_EXPR))
STRING((NOT(G_1.JOB_ID == 'MANAGER')) RETURNS 'TRUE' OTHERWISE 'FALSE'

REQUIREMENT 1:

IF USER LANGUAGE IS US THEN I WANT TO RETURN TRUE
:XDO_USER_REPORT_LOCALE

IF E_NOT_USER = 'TRUE' THEN
CALL TEMPLATE1
ELSE CALL TEMPLATE 2
END IF

20. EXPRESSIONS - AND FUNCTION

A	B	AND
0	0	0
0	1	0
1	0	0
1	1	1

EVALUATES BOOLEAN_EXPR1 AND BOOLEAN_EXPR2, AND RETURNS TRUE IF BOTH BOOLEANS ARE TRUE

```
STRING (AND (BOOLEAN_EXPR1, BOOLEAN_EXPR2,...))
STRING (AND (G_1.JOB_ID == 'MANAGER, G_1.DEPARTMENT_ID ==10))
STRING (BOOLEAN_EXPR1 && BOOLEAN_EXPR2,...)
STRING (G_1.JOB_ID == 'MANAGER && G_1.DEPARTMENT_ID ==10)
```

21. EXPRESSIONS - OR FUNCTION

A	B	OR
0	0	0
0	1	1
1	0	1
1	1	1

EVALUATES BOOLEAN_EXPR1 AND BOOLEAN_EXPR2, AND RETURNS TRUE IF EITHER OF BOOLEANS IS TRUE.

```
OR: STRING (OR (BOOLEAN_EXPR1, BOOLEAN_EXPR2,...))
STRING (OR (G_1.JOB_ID == 'MANAGER, G_1.DEPARTMENT_ID ==10))
||: STRING (BOOLEAN_EXPR1 || BOOLEAN_EXPR2,...)
STRING (G_1.JOB_ID == 'MANAGER || G_1.DEPARTMENT_ID ==10)
```

22. EXPRESSIONS - MIN AND MAX FUNCTIONS

MAXIMUM:

RETURNS THE MAXIMUM VALUE OF ELEMENT IN THE SET.

```
MAX (EXPR1, EXPR2, EXPR3...)
```

```
MAX (G_1.SALARY, 10000)
```

RETURNS MAX OF SALARY OR 10000

MINIMUM:

RETURNS THE MINIMUM VALUE OF ELEMENT IN THE SET.

```
MIN (EXPR1, EXPR2, EXPR3...)
```

```
MIN (G_1.SALARY, 2000)
```

RETURNS MIN OF SALARY OR 2000

23. EXPRESSIONS - ROUND , SUM FUNCTION

ROUND:

RETURNS A NUMBER ROUNDED TO THE INTEGER PLACES RIGHT OF THE DECIMAL POINT.

```
ROUND (NUMBER1, INTEGER1)
```

IF INTEGER IS OMITTED THEN THE NUMBER IS ROUNDED TO 0.

INTEGER CAN BE NEGATIVE TO ROUND OF DIGITS LEFT IF THE DECIMAL POINTS.

INTEGER MUST BE AN INTEGER

```
ROUND (2.777)
```

RETURNS 3

```
ROUND (2.777, 2)
```

RETURNS 2.78

SUM:

RETURNS SUM OF THE VALUE OF EXPRESSIONS

SUM (EXPR1, EXPR2, EXPR3,...)

SUM (G_1.SALARY, G_1.COMMISSION_PCT*G_1.SALARY)

RETURNS SUM OF SALARY AND COMMISSION.

FOR EXAMPLE IF SALARY IS 14000, COMMISSION_PCT=0.4, THE EXPRESSION EVALUATES:

SUM (14000, 100) RETURNS 1500.

ALL EXPRESSIONS:**NVL:**

REPLACES NULL (RETURNED AS A BLANK) WITH A STRING IN THE RESULTS OF A QUERY.

NVL (EXPR1, EXPR2)

IF EXPR1 IS NULL, THEN **NVL** RETURNS EXPR2,

IF EXPR1 IS NOT NULL THEN **NVL** RETURNS EXPR1,

NVL (G_1.COMMISSION_PCT, 0.3) RETURNS 0.3 WHEN G_1.COMMISSION_PCT IS NULL

CONCAT:

RETURNS CHAR1 CONCATENATED WITH CHAR2

CONCAT (CHAR1, CHAR2)

CONCAT (CONCAT (FIRST_NAME, ' '), LAST_NAME)

WHERE FIRST_NAME = JOE AND LAST_NAME = SMITH

RETURNS JOE SMITH

CONCAT ('HI', ' THIS IS NIKITA')

STRING:

RETURNS CHAR AS A STRING DATA TYPE.

STRING (EXPR)

STRING (G_1.SALARY)

WHERE SALARY = 4400

RETURNS 4400 AS A STRING

SUBSTRING

EXTRACTS A SUBSTRING FROM STRING

SUBSTRING (STRING, START_POS, END_POS)

STRING IS THE SOURCE STRING.

START_POS IS THE POSITION TO START THE EXTRACTION

END_POS IS THE POSITION OF THE STRING TO EXTRACT (OPTIONAL).

SUBSTRING ('THIS IS A TEST', 5, 7) RETURNS "IS" (CHARACTER 6 IS 'I' 7 IS 'S')

INSTR:

RETURNS THE POSITION/LOCATION OF THE FIRST CHARACTER OF THE SUBSTRING IN A STRING

INSTR (STRING1, STRING2)

STRING 1: STRING TO SEARCH

STRING 2: SUBSTRING TO SEARCH FOR STRING1

INSTR ('THIS IS A TEST', 'IS A')

RETURNS 5

DATE:

CONVERTS A VALID JAVA DATE STRING TO A DATE DATA TYPE IN CONICAL FORMAT.

DATE (CHAR, FORMAT_STRING)

WHERE CHAR IS ANY VALID JAVA DATE STRING (FOR EG: 09-APR-2020)

DATE ('01-JAN-2020', 'dd-MMM-yyyy')

RETURNS 2020-01-01T08:00:00.000+00:00

DATE FORMAT WITH MM/dd/yyyy : 09/23/2020

DATE FORMAT WITH dd-M-yyyy hh:mm:ss : 23-9-2020 10:59:28

DATE FORMAT WITH dd MMMM yyyy : 23 September 2020

DATE FORMAT WITH dd MMMM yyyy zzzz : 23 September 2020 India Standard Time

DATE FORMAT WITH E, dd MMMM yyyy HH:mm:ss z : Mon, 23 Sept 2020 22:59:26 IST

FORMAT_DATE:

CONVERTS A DATE ARGUMENT IN JAVA DATE FORMAT TO A FORMATTED STRING

FORMAT_DATE(DATE,FORMAT_STRING)

FORMAT_DATE(SYSDATE, 'dd-mmm-yyyy')

WHERE THE VALUE OF SYSDATE = 2020-09-23T16:32:45.000-08:00 RETURNS 23-SEPT-2020

FORMAT_NUMBER :

CONVERTS A NUMBER OR NUMERIC STRING TO A STRING IN THE SPECIFIED NUMBER FORMAT

FORMAT_NUMBER(NUMBER, FORMAT_STRING)

FORMAT_NUMBER(SOME_NUMBER,'\$9,999.00')

WHERE THE VALUE OF SOME_NUMBER = 12345.678 RETURNS \$12,345.68

DECODE:

REPLACES THE VALUE OF THE EXPRESSION WITH ANOTHER VALUE

DECODE(EXPR, SEARCH, RESULT [,SEARCH, RESULT]...[, DEFAULT])

DECODE(PROD_FAMILY_CODE, 100, 'COLAS',200,'BEER',300,'SODA')

REPLACE:

REPLACES A SEQUENCES OF CHARACTER IN A STRING WITH ANOTHER SET OF CHARACTERS

REPLACE (EXPR, STRING1, STRING2)

WHERE

STRING1 IS THE STRING TO SEARCH FOR,

STRING2 IS THE STRING TO REPLACE

REPLACE(G_1.FIRST_NAME,'B','L')

WHERE G_1.FIRST_NAME = BARRY RETURNS LARRY

ORACLE®

24. WORKING WITH BIND PARAMETERS

THERE ARE 2 TYPES OF PARAMETERS:

1. BIND PARAMETER = > ;
2. LEXICAL PARAMETER = > &

LIST OF VALUES : IN ORACLE APPS VALUE SET
NONE

INDEPENDENT

TRANSLATED INDEPENDENT

TABLE

DEPENDENT

TRANSLATED DEPENDENT

create the new data model> **XX_BIND_PARA**

```
SELECT A.INTERFACE_LINE_ATTRIBUTE1 SO,
       A.CUSTOMER_TRX_LINE_ID,
       A.CUSTOMER_TRX_ID CUST_ID,
       A.LINE_NUMBER,
       A.INVENTORY_ITEM_ID
       B.TRX_NUMBER,
       B.CUSTOMER_TRX_ID,
       A.QUANTITY_INVOICED,
       A.UNIT_SELLING_PRICE,
       A.LINE_TYPE,
       A.EXTENDED_AMOUNT,
       A.REVENUE_AMOUNT,
       A.TAX_RATE,
       B.TAX_CLASS,
       C.NAME,
       C.DESCRPTION,
       C.TYPE
FROM   RA_CUSTOMER_TRX_LINES_ALL A,
       RA_CUSTOMER_TRX_ALL B,
       RA_CUST_TRX_TYPES_ALL C
WHERE  A.CUSTOMER_TRX_ID = B.CUSTOMER_TRX_ID
AND    B.CUST_TRX_TYPE_SEQ_ID = C.CUST_TRX_TYPE_SEQ_ID
AND    B.TRX_CLASS = C.TYPE
AND    A.CUSTOMER_TRX_ID =:P_CUST_TRX_ID
```

TESTING PURPOSE WE CAN USE THIS QUERY TO CHECK WHETHER THE TABLE AND CONTENT MATCHES THE ACTUAL PARAMETERS OR NOT LET THIS BE NAMED AS **XX_TEST**.

```
SELECT * FROM RA_CUST_TRX_TYPES_ALL
```

OK > IT WILL SHOW WHETHER **P_CUST_TRX_ID** IS A BIND PARAMETER OR NOT.
SELECT YES.

NOTE : YOU SHOULD **UNCHECK** IT WHEN YOU LINK THE GROUPS.

DATATYPE : INTEGER

DEFAULT VALUE : 1

PARAMETER TYPE : TEXT

DISPLAY LABEL : CUST_TRX_ID

TEXT FIELD SIZE : 20

SAVE> **XX_TEST_BIND_PARAMETERS** > SAVE AS SAMPLE DATA (FOR GENERATING REPORT) >
CREATE REPORT > CANCEL > SELECT DATA MODEL > **XX_TEST_BIND_PARAMETERS** > GENERATE >
XXTEST_BIND_PARAMETERS > OK > SAVE > **XXTEST_BIND_PARAMETERS_LAYOUT** > OK > CHANGE OR
MANUPULATE CUST_TRX_ID = 1005 AND SEE THE CHANGES.

RUN THE REPORT FROM CATALOG

CATALOG> **XXTEST_BIND_PARAMETERS_LAYOUT** > MORE > SCHEDULE > OUTPUT > AS A PDF >
SUBMIT > **XXTEST_BIND_PARAMETER_001** > OK > RETURN > **_BIND_PARAMETERS_LAYOUT** > MORE >
HISTORY > CLICK ON OUTPUT.

25. BIND PARAMETERS WITH TABLE VALUESET AND \$FLEX\$ CONCEPT IN FUSION

TABLE:

XXTEST_BIND_PARAMETER > EDIT>

WE NEED TO CREATE THE LIST OF VALUES AND THEN WE WILL ATTACH IT WITH THE **P_CUST_TRX_ID**

NEW VALUE SET>

NAME: **TRX_ID**

TYPE: SQL QUERY

DATA SOURCE: FUSIONDB

CACHE RESULT

QUERY: SELECT **CUSTOMER_TRX_ID** FROM **RA_CUSTOMER_TRX_ALL**

DATAMODEL WE SEE PARAMTERS>**P_CUST_TRX_ID**> CHANGE PARAMETER TYPE AS MENU.

IN THE DOWNSIDE: LIST OF VALUES (LOV)= **TRX_ID**

AND DESELECT EVERYTHING

SAVE>VIEW REPORT>DATA >VIEW>SEE THE VALUES>SAVE AS SAMPLE>CATALOG>LAYOUT>OPEN>RUN ONLINE >CHANGE PARAMETER>APPLY

TO GET THE VALUES FOR SOME OTHER PARAMETERS

CATALOG> **XXTEST_BIND_PARAMETER** > EDIT> **TRX_ID**

QUERY: **SELECT CUSTOMER_TRX_ID FROM RA_CUSTOMER_TRX_ALL WHERE TRX_CLASS = P_TRX_TYPE**>SAVE>

--**P_TRX_TYPE HAS ALL THE INVOICE TYPES.**

DATAMODEL WE SEE PARAMTERS>**P_CUST_TRX_ID**> DRAG UP ABOVE > WE SEE EVERYTHING IS UNCHECKED

BUT IN **P_TRX_TYPE**

WHEN DELECTING ALL PARAMETERS AND RUNNING BY ONLINE MODE

CATALOG>LAYOUT >OPEN : WE SEE THAT CHANGING TRANSACTION TYPE DOESN'T CHANGE THE **CUSTOMER_TRX_ID**.

CATALOG>LAYOUT>EDIT> **P_TRX_TYPE**

WHEN SELECTING REFRESH OTHER PARAMETER ON CHANGE

CATALOG>LAYOUT >OPEN : WE SEE THAT CHANGING TRANSACTION TYPE CHANGES THE **CUSTOMER_TRX_ID**.

26. SUBTEMPLATE

STEPS TO CREATE SUBTEMPLATE

1. CREATE DATA MODEL

XXAMSUBTEMP

```

SELECT A.INTERFACE_LINE_ATTRIBUTE1 SO,
       A.CUSTOMER_TRX_LINE_ID,
       A.CUSTOMER_TRX_ID CUST_ID,
       A.LINE_NUMBER,
       A.INVENTORY_ITEM_ID
       B.TRX_NUMBER,
       B.CUSTOMER_TRX_ID,
       A.QUANTITY_INVOICED,
       A.UNIT_SELLING_PRICE,
       A.LINE_TYPE,
       A.EXTENDED_AMOUNT,
       A.REVENUE_AMOUNT,
       A.TAX_RATE,
       B.TRX_CLASS,
       C.NAME,
       C.DESCRPTION,
       C.TYPE
FROM   RA_CUSTOMER_TRX_LINES_ALL A
       RA_CUSTOMER_TRX_ALL B
WHERE  A.CUSTOMER_TRX_ID = B.CUSTOMER_TRX_ID
AND    B.CUSTOMER_TRX_TYPE_SEQ_ID = B.CUSTOMER_TRX_TYPE_SEQ_ID
AND    B.TRX_CLASS = C.TYPE
AND    A.CUSTOMER_TRX_ID =:P_CUST_TRX_ID
--AND C.TYPE = :P_TRX_TYPE - (CM,INV,CB)

```

OK> YES TO BINDING PARAMETERS>

DATATYPE: INTEGER

DEFAULT VALUE: 1/2/3 ANY

DRAG UP> DISPLAY LABEL> TRX_ID

TEXT_SIZE=5

DATAMODEL>DATA>VIEW DATA>SAVE AS SAMPLE>TABLE VIEW>SAVE>CREATE
FOLDER>MYREPORTS>XXAM_MAIN_DATASET>OK

2. CREATE MAIN TEMPLATE

NEW>REPORT>CANCEL>ATTACH DATA MODEL>GENERATE>XXAM_MAIN_DATASET_TEMP

3. CREATE SUBTEMPLATE AND SAVE AS RTF AND UPLOAD TO CATALOG AND ITS EXTENSION WILL BE .XSB

WORD>

<?TEMPLATE:US?>



<?END:TEMPLATE?>

SAVE AS RTF: XXAM_SUBTEMP

NEW>SUBTEMPLATE>UPLOAD>OK>USE SAME NAME IN DESCRIPTION AS OF RTF>SAVE AS XXAM_SUBTEMP

CATALOG>XXAM_SUBTEMP > MORE>PROPERTIES>TYPE:RTF AND COMPOSITE SIGNAL AS .XSB
IF YOU GO TO MORE> ARCHIEVE > SAVE>OPEN & CHECK EXTENSION .XSB

4. IN MAIN TEMPLATE CALL SUBTEMPLATE
 IN THE MAIN TEMPLATE WE SHOULD HAVE 2 SECTIONS
 A. IMPORT SECTION

ASSUME YOU UPLOADED SUBTEMPLATE INTO SHARED FOLDER
 <?IMPORT:XDOS1:///PATH OF SUBTEMPLATE/SUBTEMPLATENAME.XSB?>
 CALL SECTION
 <?CALL-TEMPLATE:TEMPLATENAME?>

CATALOG> **XXAM_SUBTEMP** >PROPERTIES>COPY PATH >/**SHARED/CUSTOM/MYREPORTS**.
 CATALOG> **XXAM_MAIN_DATASET_TEMP** > EDIT> EDIT TO DOWNLOAD YOUR TEMPLATE>
 SAVE>MODIFY CHANGES > IMPORT SUBTEMPLATE INTO THAT
 AS >

- <?IMPORT:XDOXSL:///CUSTOM/MYREPORTS/XXAM_SUBTEMP.XSB?>
- <?CALL-TEMPLATE:US?>

-FOR CHECKING ERROR
 ADD INS>TOOLS>VALIDATE
 --UPLOAD TEMPLATE TO CATALOG
XXAM_MAIN_DATASET_TEMP>EDIT>DELETE OLD TEMPLATE>ADD NEW LAYOUT>UPLOAD>
XXAM_MAIN_DATASET_TEMP>UPLOAD>SAVE>VIEW REPORT>PDF.

5. TEST REPORT ONLINE OR FROM LOCALE MACHINE:
 NOTE: TESTING FROM LOCALE MACHINE

IN MAIN TEMPLATE CHANGE IMPORT STATEMENT AS BELOW
 <?IMPORT:FILE:DRIVE_NAME:///PATH/SUBTEMPLATE_NAME.RTF?>
 CATALOG>DATA MODEL OF **XXAM_MAIN_DATASET** >EDIT>DATA>VIEW>SAVE AS
 SAMPLE>EXPORT>SAVE> **_CUSTOM_MYREPORT**.

WORD> PASTE THIS :

- <?IMPORT:FILE:D//XXAM_SUBTEMP.RTF?>
- <?CALL-TEMPLATE:US?>

ADD-INS>DATA>LOAD XML DATA> **_CUSTOM_MYREPORT**>PREVIEW>PDF.

CATALOG> **XXAM_MAIN_DATASET_TEMP** >OPEN >VIEW AS PDF

ASSUME YOU UPLOADED SUBTEMPLATE INTO MY FOLDER
 <?IMPORT:XDOS1:///~USER_NAME/PATH OF SUBTEMPLATE/SUBTEMPLATENAME.XSB?>
 CALL SECTION
 <?CALL-TEMPLATE:TEMPLATENAME?>

CATALOG> **XXAM_MAIN_DATASET_TEMP** (TEMPLATE FILE)>EDIT>EDIT>SAVE AS
XXAM_SUBTEMP_LOCALE.RTF.

CATALOG>MYFOLDER> SUBTEMPLATES>EXPAND> NEW>SUBTEMPLATE>UPLOAD>OK>USE SAME NAME IN
 DESCRIPTION AS OF RTF>SAVE AS **XXAM_SUBTEMP_001**

CATALOG>**XXAM_SUBTEMP_001**>MORE>PROPERTIES>COPY
 PATH>MYFOLDER/USERS/CLOUD_IMPL/SUBTEMPLATES/
 MODIFY THAT WORD FILE: **XXAM_SUBTEMP_LOCALE**.RTF

<?IMPORT:XDOS1:///~CLOUD_IMPL/SUBTEMPLATES/ **XXAM_SUBTEMP_001.XSB**?>

<?CALL-TEMPLATE:TEMPLATENAME?>

SHARED
 FOLDER/CUSTOM/MYREPORTS>**XXAM_MAIN_DATASET_TEMP**>EDIT>DELETE>ADD_NEW_LAYOUT>UPLOAD>
_SUBTEMP_LOCALE > VIEW REPORT>PDF.

SO WE FINISHED IT USING SUBTEMPLATES FROM SHARED FOLDERS,MY FOLDER AND LOCAL DESKTOP.

27. MULTILANGUAGE REPORT OR TRANSLATIONS

WE WILL CREATE RTF OF OUR LANGUAGE AND WILL CHANGE ITS OUTPUT ACCORDING TO THE ALNGUAGE REQUIRED.

1. CREATE DATA TEMPLATE

XXTEST FOLDER > NEW > DATAMODEL : **XXTEST_TRANSLATION**

```

SELECT      A.INTERFACE_LINE_ATTRIBUTE1 SO,
            A.CUSTOMER_TRX_LINE_ID,
            A.CUSTOMER_TRX_ID CUST_ID,
            A.LINE_NUMBER,
            A.INVENTORY_ITEM_ID
            B.TRX_NUMBER,
            B.CUSTOMER_TRX_ID,
            A.QUANTITY_INVOICED,
            A.UNIT_SELLING_PRICE,
            A.LINE_TYPE,
            A.EXTENDED_AMOUNT,
            A.REVENUE_AMOUNT,
FROM        RA_CUSTOMER_TRX_LINES_ALL A,
            RA_CUSTOMER_TRX_ALL B
WHERE       A.CUSTOMER_TRX_ID = B.CUSTOMER_TRX_ID
AND         A.CUSTOMER_TRX_ID =1005 --:P_CUST_TRX_ID

```

2. CREATE LAYOUT AND ATTACH TO DATAMODEL

SAVE>DATA>VIEW>SAVE AS SAMPLE DATA > **XXTEST** FOLDER > **DATAMODEL** FOLDER > **XXINVOICE_REPORT_MULTILANG** > OK > CREATE REPORT > CANCEL > SELECT DATA MODEL > **XXINVOICE_REPORT_MULTILANG** > OK > GENERATE > **XXINVOICE_MULTI** > GENERATE > SAVE > **XXTEST** FOLDER > **XXINVOICE_MULTILAYOUT**.

3. GO TO REPORT LAYOUT PROPERTIES AND DOWN LOAD TRANSLATION FILE (XLIFE FILE) UNDER TRANSLATION SECTION (EXTENSION.XLF)
LAYOUT WILL BE SEEN > **XXINVOICE_MULTI** > PROPERTIES > BY DEFAULT WE CAN SEE THE ENGLISH LANGUAGE > AND IN THE TRANSLATIONS > EXTRACT TRANSLATIONS > DOWNLOAD AND SAVE >

4. NOW EDIT ALL THE FOLLOWING FIELDS
OPEN FILE > RENAME IT AS **TRANSLATION_SPAIN_LANGUAGE** > EDIT WITH NOTEPAD++ > **TARGET ID AND TARGET FIELDS NEED TO BE CHANGED.**

BEFORE THAT RETURN > VIEW REPORT > ALL COLUMNS ARE STATIC ONLY WE HAVE TO CHANGE OR TRANSLATE THE FILES.

ENGLISH	SPAIN
INVOICE	INFORME DE FACTURAS
DATE	FECHA
PAGE	PAGINA
SO	ORDENED DE VENTA
CUSTOMER_TRX_LINE_ID	CUSTOMER_TRX_LINE_ID
CUST_ID	CUST_ID
LINE NUMBER	NUMERO DE LINEA
INVENTORY_ITEM_ID	INVENTORY_ITEM_ID
TRX_NUMBER	NUMERO DE FACTURA
CUSTOMER_TRX_ID	CUSTOMER_TRX_ID
QUANTITY	CANTIDAD
UNIT SELLIMG PRICE	PRECIO DE VENTA UNITARIO
EXTENDED AMOUNT	MONTO EXTENDIDO
REVENUE_AMOUNT	CANTIDAD DE INGRESOS

CHANGE **TARGET-LANGUAGE="EN-US"** --ENGLISH TO **TARGET-LANGUAGE="ES"** - SPAIN
 CHANGE **<TARGET> ORACLE BI PUBLISHER </TARGET>** TO **<TARGET> ORACLE BI PUBLISHER SPANISH</TARGET>**
 CHANGE **<TARGET> DATE : ... </TARGET>** TO **<TARGET> FETCHA : ...</TARGET>**
 AND CHANGE THE PARAMETERS MENTIONED IN THE TABLE ABOVE.

5. UPLOAD TRANSLATED FILE WITH TARGET LANGUAGE

XXINVOICE_MULTILAYOUT > EDIT > PROPERTIES > UPLOAD > BROWSE > **TRANSLATION_SPAIN_LANGUAGE**
 > SPANISH LOCALE > OK > SAVE

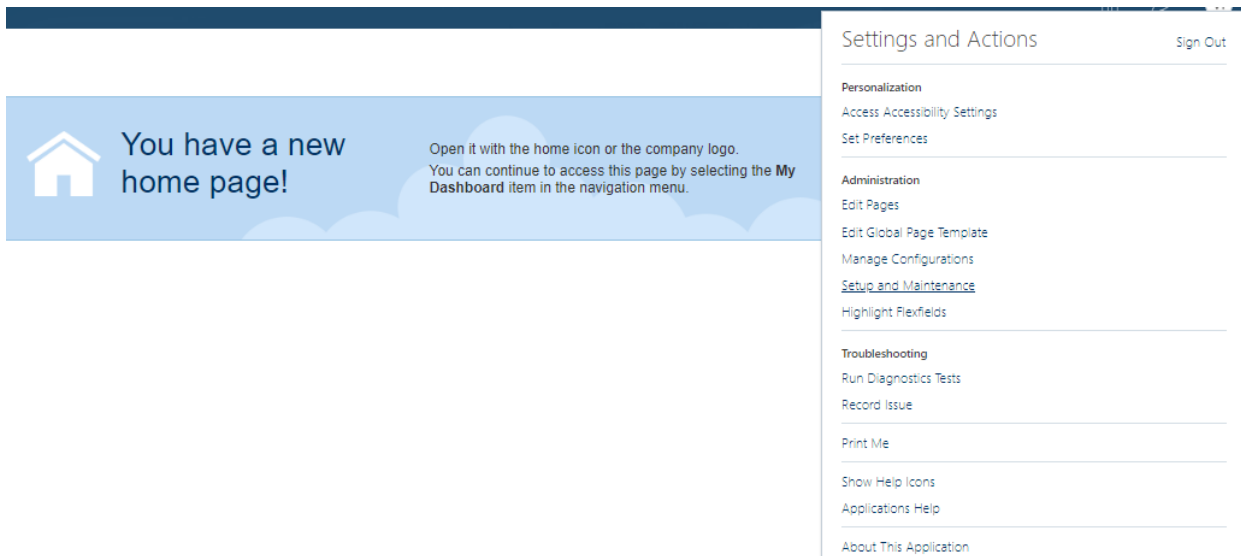
6. RUN THE REPORT FROM CATALOG OR BY CREATING ESS JOB

TO SEE THE REPORT IN SPANISH LANGUAGE

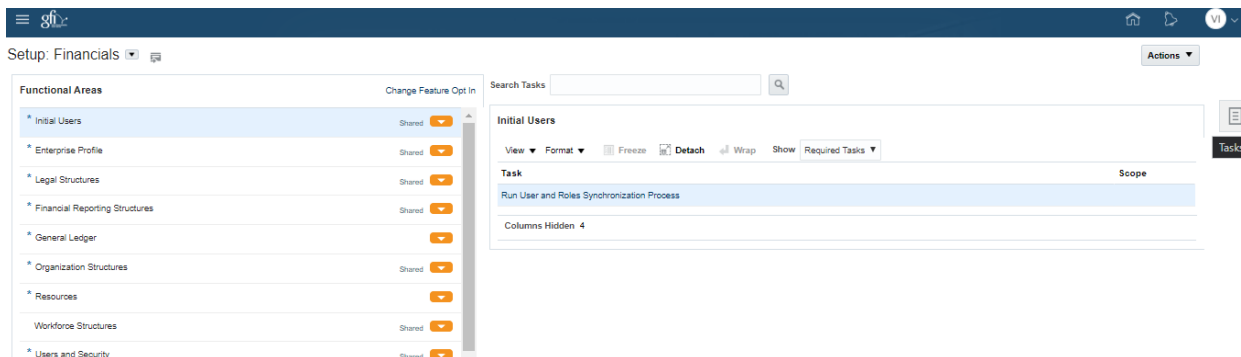
CATALOG > **XINVOICE_MULTILAYOUT** > MORE > SCHEDULE > OUTPUT > PDF > SPANISH > SUBMIT > **SPANISH_001** > CATALOG > **XINVOICE_MULTILAYOUT** > MORE > HISTORY > OUTPUT .

RUN THE REPORT USING CONCURRENT JOB

CATALOG > **XINVOICE_MULTILAYOUT** > MORE > SCHEDULE > GENERAL > COPY **REPORT ID : >**



SETUP AND MAINTAINANCE > TASK > MANAGE IMPLEMENTATION PROJECT > IMPLEMENT PROJECT 1 > SEARCH > TASK > MANAGE%ENTERPRISE% > SEARCH > OBSOLETE : MANAGE CUSTOM ENTERPRISE SCHEDULER JOBS FOR PRODUCT MANAGEMENT AND RELATED APPLICATIONS > GO TO -> >



Change Feature Opt In

Search Tasks

Initial Users

View Format Freeze Detach Wrap Show Required Tasks

Task

Run User and Roles Synchronization Process

Columns Hidden 4

Manage Implementation Projects

Manage Implementation Projects

Manage Export and Import Processes

Manage Setup Content

Review Topology

Search

Implementation Projects

Search

Name

Advanced Saved Search All Implementation Projects

Search Reset Save

Search Results

Actions View Format Freeze Detach Wrap

Name	Status	Assigned To	Start Date	Finish Date	Last Updated	Last Updated By
Implementation Project-Recruitment(SS)		HCMUser4	12-Mar-2020	12-Mar-2020	HCMUser4	
Profile management Implementation Project		HCMUser4	01-Jan-1951	10-Mar-2020	HCMUser4	
Implementation Project Demo Recruiting_2		HCMUser4	09-Mar-2020	09-Mar-2020	HCMUser4	
Implementation Project Demo Recruiting		HCMUser4	09-Mar-2020	09-Mar-2020	HCMUser4	
Implementation Project Recruiting		HCMUser4	09-Mar-2020	09-Mar-2020	HCMUser4	
Qatar Corporation		HCMUser4	01-Jan-2019	10-Feb-2020	HCMUser4	
SRY Fin Implementation project		sryhomuser	25-Dec-2019	25-Dec-2019	sryhomuser	
Payroll Implementation Project		HCMUser4	01-Jan-2019	24-Dec-2019	HCMUser4	
TS HCM Cloud Implementation1		HCMUser2	30-Mar-2017	03-Dec-2019	HCMUser4	
Sry Implementation Project		sryhomuser	09-Oct-2019	10-Oct-2019	sryhomuser	

Columns Hidden 2 Columns Frozen 1

CLICK + NEW >

- DISPLAY NAME: `XXARINVOICE_MULTI_LANG_REPORT`
- NAME: `XXARMULTIRERORT`
- PATH: `/CUSTOM/XXTEST/`
- APPLICATION : APPLICATION TOOLKIT
- DESCRIPTION: `XXAR_INVOICE_MULTI_LANG_REPORT`
- JOB APPLICATION NAME: FACMESS
- JOB TYPE: BIP JOB TYPE
- DEFAULT OUTPUT FORMAT : PDF
- REPORT ID: THE ONE WHICH WE COPIED.
- ENABLE SUBMISSION.
SAVE.

NAVIGATOR > SCHEDULE PROCESSES > SCHEDULE NEW PROCESSES > `XXARINVOICE_MULTI_LANG_REPORT` > OK > OK > PRINT OUTPUT > PROCESS OPTIONS > SPANISH > OK > ADVANCED > SCHEDULE > AS SOON AS POSSIBLE > OUTPUT > + > PDF > SUBMIT > REFRESH > CLICK ON `XXARINVOICE_MULTI_LANG_REPORT` > CLICK ON OUTPUT.

SCHEDULE PROCESSES > SCHEDULE NEW PROCESSES > `XXARINVOICE_MULTI_LANG_REPORT` > OK > OK > PRINT OUTPUT > PROCESS OPTIONS > AMERICAN ENGLISH > OK > ADVANCED > SCHEDULE > AS SOON AS POSSIBLE > OUTPUT > + > PDF > SUBMIT > REFRESH > CLICK ON `XXARINVOICE_MULTI_LANG_REPORT` > CLICK ON OUTPUT.

28. MULTI LANGUAGE REPORT OR TRANSLATIONS. -TESTING

ORACLE FUSION WEBSITE >

USERID: HCMuser4

PASSWORD: *****

SELECT THE LANGUAGE: SPANISH

NAVIGATION > HERRAMIENTAS (SCHEDULE PROCESS) > PROCECAS PROGRAMADOS (SCHEDULE NEW PROCESSES) > PROGRAMAR NUEVO PROCESO > XXAR > BEACAR (SEARCH) > ACEPTAR > ACEPTAR (OK) > OPCIOMAS DE PROCESO (PROCESS OPTIONS)> ALREADY LANG SELECTED > ACCEPTOR > DO SAME AS ABOVE REFERENCE.

SO BASED ON THE SELECTED LANGUAGES THE REPORTS WILL BE GENERATED.

The Oracle logo is displayed in a large, light red, sans-serif font. The word "ORACLE" is followed by a registered trademark symbol (®).

29. BURSTING - DELIVERY CHANNEL - EMAIL AND FTP AND SFTP AND FILE

THERE ARE SOME SPECIAL SCENARIOS WHERE AN ORGANIZATION NEEDS TO SEND CORRESPONDENCE TO AN EMPLOYEE/EX-EMPLOYEE.

WHILE ONE MAY USE BI PUBLISHER REPORTS FOR THE SAME USING OTBI REPORTS HAS SOME ADVANTAGE INCLUDING BUT NOT LIMITED TO BELOW:

1. ACCESS CONTROLLED
2. SECURED (OTBI REPORTS HAVE SAME ACCESS AS A USER HAS)
3. OTBI REPORT CREATION IS EASY TO CREATE (MOST OF THE TIMES)
4. USER-FRIENDLY

SOME OF THE SPECIFIC CASES WHERE OTBI BURSTING HELPS ARE:

- EXPERIENCE LETTER TO EX-EMPLOYEES
- NEW JOINING INTIMATION LETTER TO POTENTIAL NEW HIRES
- RELIVING LETTER TO EX-EMPLOYEES
- SALARY CERTIFICATE TO EX-EMPLOYEES
- TAX CERTIFICATE.... AND MUCH MORE

XXTEST FOLDER > **DATAMODEL** FOLDER > **XXINVOICE_REPORT_MULTILANG** > EDIT > DATA > VIEW > EXPORT > SAVE > TO SEE THE XML OUTPUT >

OR DATA > VIEW > TABLE VIEW FOR CHECKING WITH ONE INVOICES ONLY

WHAT IF WE HAVE MORE THAN ONE INVOICE?

IF THE OUTPUT IS HAVING 1 INVOICE: I WANT TO SEND SINGLE INVOICE TO CUSTOMER.

IF THE OUTPUT IS HAVING 2 INVOICES: WE NEED TO SEND OUTPUT TO CUSTOMER BY TRX NUMBER. CUSTOMER MAY CHOOSE THE FOLLOWING METHODS:

DELIVARY CHANNEL

EMAIL

FAX

FTP

FILE

SFTP

PRINT

WEBDAV

IF YOU WANT TO SEND OUTPUT BY INVOICE NUMBER SPLIT BY TRX_MUNBER OR CUSTOMER_TRX_ID

IF YOU WANT TO SEND THE OUTPUT BY CUSTOMER SPLIT BY CUSTOMER_ID

10 INVOICES	DIVIDED INVOICES	NUMBER OF PDF GENERATED BY CUSTOMER	NUMBER OF PDF GENERATED BY TRX_NUMBER
	5 - CUSTOMER 1	1 FILE 5 INVOICES	5 FILES GENERATED
	2 - CUSTOMER 2	1 FILE 2 INVOICES	2 FILES GENERATED
	3 - CUSTOMER 3	1 FILE 3 INVOICES	3 FILES GENERATED

BURSTING > + NEW >

NAME: **INVOICE_BURSTING**

TYPE: SQL_QUERY

DATA SOURCE: FUSIONDB

SPLIT BY AND DELIVERED BY: **CUSTOMER_TRX_ID**

SQL_QUERY:

WHEN DELIVERY CHANNEL IS EMAIL

```

SELECT TRX_NUMBER          KEY, --IT MUST BE SAME AS SPLIT BY COLUMN
'XXNAME'                  TEMPLATE, -- LAYOUT NAME GET THIS FROM CATALOG
'XDO_USER_REPORT_LOCALE' LOCALE, --GET THIS FROM XDO USER REPORT LOCALE OR
HARDCODE IT TO EN-US.
'PDF'                     OUTPUT_FORMAT, --ODF/XML/ETEXT/CSV/EXCEL/HTML/RTF
'EMAIL'                   DEL_CHANNEL, --EMAIL/FAX/FTP/FILE/SFTP/PRINT/WEBDAV
'AR INVOICE ' || TRX_NUMBER OUTPUT_NAME, --OUTFILE NAME
--IF YOU WANT TO DERIVE DYNAMIC NAME COLUMN NAME MUST COME FROM SELECT

'nikita23@gmail.com'      PARAMETER1, --EMAIL ADDRESS
--                        PARAMETER2, --CC
'NK1234@GMAIL.COM'       PARAMETER3, --FROM
'XX INVOICE REPORT'      PARAMETER4, --SUBJECT
'DEAR CUSTOMER,
PLEASE FIND ATTACHED INVOICE FOR THE CURRENT MONTH
THANKS AND REGARDS.
XXABC'                   PARAMETERS5, --MESSAGE BODY
'TRUE'                   PARAMETER6, --ATTACHED VALUE ('TRUE' OR 'FALSE')
'NOREPLY@GMAIL.COM'      PARAMETER7, --REPLY-TO
--                        PARAMETER8  --BCC

FROM
(
SELECT      B.TRX_NUMBER
FROM        RA_CUSTOMER_TRX_LINE_ALL A
            RA_CUSTOMER_TRX_ALL B
WHERE A.CUSTOMER_TRX_ID = B.CUSTOMER_TRX_ID,
A.CUSTOMER_TRX_ID IN (1005, 1)

--OR--

SELECT TRX_NUMBER FROM RA_CUSTOMER_TRX_ALL
)

```

SAVE > CATALOG > **XINVOICE_MULTILAYOUT** > EDIT > REPORT PROPERTIES > GENERAL > ENABLE BURSTING > SAVE > CATALOG > OUTPUT > CHECK USE BURSTING AND SAVE DATA FOR REPUBLISHING > SUBMIT > **BURSTING_TEST_001** > OK > RETURN > **XINVOICE_MULTILAYOUT** > HISTORY > CLICK ON REPORT > WE CAN SEE THAT HERE ARE TWO INVOICES GENERATED AT OUTPUT

AR_INVOICES4

AR_INVOICES5

WE CAN DOWNLOAD AND SAVE FILES LIKE LOG FILE AND XML FILE.

WE CAN CHECK WHETHER WE GOT 2 INVOICES IN OUR MAIL OR NOT

WHEN DELIVERY CHANNEL IS FILE TO GET OUTPUT TO LOCAL DESKTOP.

ONLY DEL CHANNEL AND PARAMETER HAS TO BE CHANGED.

```

SELECT TRX_NUMBER      KEY, --IT MUST BE SAME AS SPLIT BY COLUMN
'XXNAME'              TEMPLATE, -- LAYOUT NAME GET THIS FROM CATALOG
'XDO_USER_REPORT_LOCALE' LOCALE, --GET THIS FROM XDO USER REPORT LOCALE OR
HARDCODE IT TO EN-US.
'PDF'                 OUTPUT_FORMAT, --ODF/XML/ETEXT/CSV/EXCEL/HTML/RTF
'FILE'               DEL_CHANNEL, --EMAIL/FAX/FTP/FILE/SFTP/PRINT/WEBDAV
'AR INVOICE_' || TRX_NUMBER OUTPUT_NAME, --OUTFILE NAME
--IF YOU WANT TO DERIVE DYNAMIC NAME COLUMN NAME MUST COME FROM SELECT.
'XXSERVER_NAME'      PARAMETER1, --SERVERNAME
'USERNAME'           PARAMETER2, --USERNAME
'PASSWORD'           PARAMETER3, --PASSWORD
'U01/WW/AR/INVOICE/INFILE' PARAMETER4, --REMOTE DIRECTORY
'XX INVOICE REPORT'   PARAMETER5, --REMOTE FILE NAME
'XXTEST_REMOTE_FILE' PARAMETER6, --SECURE (SET THIS VALUE TO 'TRUE' TO SET
THIS FTP)
--PARAMETER 7 TO 10 IS NOT USED.

```

```

FROM
(
SELECT      B.TRX_NUMBER
FROM        RA_CUSTOMER_TRX_LINE_ALL A
            RA_CUSTOMER_TRX_ALL B
WHERE A.CUSTOMER_TRX_ID = B.CUSTOMER_TRX_ID,
A.CUSTOMER_TRX_ID IN (1002, 1)
)

```

WHEN DELIVERY CHANNEL IS FTP/ SFTP TO GET OUTPUT TO REMOTE SERVER.

ONLY DEL CHANNEL AND PARAMETER HAS TO BE CHANGED.

```

SELECT TRX_NUMBER      KEY, --IT MUST BE SAME AS SPLIT BY COLUMN
'XXNAME'              TEMPLATE, -- LAYOUT NAME GET THIS FROM CATALOG
'XDO_USER_REPORT_LOCALE' LOCALE, --GET THIS FROM XDO USER REPORT LOCALE OR
HARDCODE IT TO EN-US.
'PDF'                 OUTPUT_FORMAT, --ODF/XML/ETEXT/CSV/EXCEL/HTML/RTF
'FTP'               DEL_CHANNEL, --EMAIL/FAX/FTP/FILE/SFTP/PRINT/WEBDAV
'AR INVOICE_' || TRX_NUMBER OUTPUT_NAME, --OUTFILE NAME
--IF YOU WANT TO DERIVE DYNAMIC NAME COLUMN NAME MUST COME FROM SELECT.

```

```

'C: /'                PARAMETER1, --DIRECTORY
'XXTEST'              PARAMETER2, --FILE NAME

```

```

FROM
(
SELECT      B.TRX_NUMBER
FROM        RA_CUSTOMER_TRX_LINE_ALL A
            RA_CUSTOMER_TRX_ALL B
WHERE A.CUSTOMER_TRX_ID = B.CUSTOMER_TRX_ID,
A.CUSTOMER_TRX_ID IN (1002, 1)
)

```

30. BURSTING REPORT TESTING FIXING THE ISSUES IN BURSTING QUERY

WE SHOULD ONLY GET 2 MAILS IN OUR INBOX BUT WE GET MORE THAN ONE MAIL WHILE SELECTING EMAIL AS A DELIVERY CHANNEL.

CATALOG > **XXTEST** FOLDER > **XINVOICE_MULTILAYOUT** > EDIT > SELECT DATA MODEL > **XXINVOICE_REPORT_MULTILANG** > BURSTING > **INVOICE_BURSTING** > THE SQL QUERY SHOULD BE >

```
SELECT TRX_NUMBER      KEY,
'XXNAME'              TEMPLATE,
'XDO_USER_REPORT_LOCALE' LOCALE,
'PDF'                 OUTPUT_FORMAT,
'EMAIL'              DEL_CHANNEL,
'AR INVOICE_' || TRX_NUMBER OUTPUT_NAME,
'nikita2309@gmail.com' PARAMETER1,
                        NULL      PARAMETER2,
'NK1234@GMAIL.COM'     PARAMETER3,
'XX INVOICE REPORT'    PARAMETER4,
'DEAR CUSTOMER,
PLEASE FIND ATTACHED INVOICE FOR THE CURRENT MONTH
THANKS AND REGARDS.
XXABC'                 PARAMETER5,
'TRUE'                 PARAMETER6,
'NOREPLY@GMAIL.COM'    PARAMETER7,

FROM
(
SELECT DISTINCT B.TRX_NUMBER
FROM
RA_CUSTOMER_TRX_LINE_ALL A
RA_CUSTOMER_TRX_ALL B
WHERE A.CUSTOMER_TRX_ID = B.CUSTOMER_TRX_ID,
A.CUSTOMER_TRX_ID IN (1005, 1)
)
```

CATALOG > **XXTEST** FOLDER > **XINVOICE_MULTILAYOUT** > MORE > SCHEDULE > SUBMIT > **BURSTING_TEST_002** > OK > CATALOG > **XXTEST** FOLDER > **XINVOICE_MULTILAYOUT** > MORE > HISTORY > CLICK ON REPORT **BURSTING_TEST_002** > WE CAN SEE THAT HERE ARE TWO INVOICES GENERATED AT OUTPUT > CLICK ON AR_INVOICES4 > OBSERVE THE TABLE.

NOW GO TO YOUR GMAIL ACCOUNT AND SEE YOU WILL HAVE TWO INVOICES REPORT

31. AR INVOICE REPORT DEVELOPMENT FROM SCRATCH - SQL QUERY DEVELOPMENT

TAB 1: OPEN THE LINK OF THE ORACLE FUSION CLOUD > REPORTS AND ANALYTICS.

TAB 2: THE NEXT TAB WILL BE OPENED FOR CREATING THE DATA MODEL.

TAB 3: AGAIN OPEN THE NEXT TAB TO SEE THE BASE TABLES FOR THE ACTUAL QUERIES.

TAB 1: GO TO RECIEVABLES > BILLINGS.

ON THE RIGHT SIDE WE SEE THE SEARCH TRANSACTIONS. PUT ANY TRANSACTION NUMBER (**TRX_NUMBER**) EG: 1000001.

WE WILL BE REFERING THIS PAGE ONE BY ONE TO EXTRACT EACH DATA FROM DIFFERENT TABLES AND JOINING IN THE ACTUAL TABLE.

TAB 3: BASE TABLES WERE CHECKED, FOR TABLES HAVING INFORMATION ABOUT AR INVOICES, EXECUTE SINGLE STATEMENTS ONLY.

DATA MODEL > **TEST**.

--HEADER INFORMATION

```
SELECT * FROM RA_CUSTOMER_TRX_ALL
WHERE TRX_NUMBER = '1000001'
```

--BUSINESS UNIT INFORMATION

```
SELECT * FROM HR_ORGANISATION_UNIT
```

-- TRANSACTION SOURCES

```
SELECT * FROM RA_BATCH_SOURCES_ALL
```

-- TRANSACTION TYPE INFORMATION

```
SELECT * FROM RA_CUST_TRX_TYPES_ALL
```

-- SALES REP INFORMATION

```
SELECT * FROM RA_SALESREPS
```

-- PAYMENT TERMS

```
SELECT * FROM RA_CUST_TRX_TYPES_ALL
```

-- CUSTOMERS OUTSTANDING (HOURS AND WORKING HRS PAY)

```
SELECT * FROM AR_PAYMENT_SCHEDULES_ALL
```

-- CUSTOMERS IS NOTHING BUT PARTY

HZ_PARTIES

EVERY CUSTOMER WILL HAVE MORE THAN ONE ACCOUNTS

HZ_CUST_ACCOUNTS

HZ_CUST_ACC_SITES_ALL

EVERY CUSTOMER WILL HAVE 1 OR MORE BILL TO SITES

EVERY CUSTOMER WILL HAVE 1 OR MORE SHIP TO SITES

HZ_CUST_SITE_USES_ALL

--EVERY CUSTOMER ACCOUNT

```
SELECT * FROM HZ_CUST_ACCOUNTS
```

--TRX_LINES

```
SELECT * FROM RA_CUSTOMER_TRX_LINES_ALL
```

SAVE > **XXTESTFILE** > DATA > VIEW > TABLE VIEW

TAB 2: EXAMINE THE BASE TABLE QUERIES AND JOINING THEM HERE

DATA MODEL: **XXARINVREP**

```
SELECT      RCT.TRX_NUMBER,
            HOU.NAME BUSINESS_UNIT,
            ABSA.NAME TRANSACTION_SOURCE,
            RCTT.NAME TRANSACTION_TYPE,
            RS.NAME SALES_REP,
            RT.NAME PAYMENT_TERM,
            APS.DUE_DATE DUE,
            HCA_BILL_TO.ACCOUNT_NUMBER,
            HCA_BILL_TO.STATUS,
            HCAUA_BILLTO.SITE_USE_CODE,
            HCAUA_BILLTO.LOCATION BILL_TO_SITE,
            HZ_BILL_TO.PARTY_NAME BILL_NAME,
```

```

        RCTLA.CUSTOMER_TRX_LINE_ID,
        RCTLA.LINE_NUMBER,
        RCTLA.LINE_TYPE,
        RCTLA.EXTENDED_AMOUNT,
        RCTLA.LINK_TO_CUST_TRX_LINE_ID,
        RCTLA.TAX_RATE,
        RCTLA.VAT_TAX_ID,
        RCTLA.ORG_ID,
        RCTLA.TAXABLE_AMOUNT,
        RCTLA.WAREHOUSE_ID,
        RCTLA.TAX_LINE_ID,
        RCTLA.INVENTORY_ITEM_ID,
        RCTLA.DESCRPTION,
        RCTLA.QUANTITY_INVOICED,
        RCTLA.UNIT_SELLIN_PRICE,
        RCTLA.UOM_CODE
FROM RA_CUSTOMER_TRX_ALL RCT,
     HR_ORGANISATION_UNITS HOU,
     RA_BATCH_SOURCES_ALL ABSA,
     RA_CUST_TRX_TYPES_ALL RCTT,
     RA_SALESREPS RS,
     RA_TERMS TI RT,
     AR_PAYMENT_SCHEDULES_ALL APS,
     HZ_CUST_ACCOUNTS HCA_BILL_TO,
     HZ_CUST_ACCT_SITES_ALL HCAS_BILL_TO
     HZ_CUST_SITE_USES_ALL HCAUA_BILLTO
     HZ_PARTIES HZ_BILL_TO
     RA_CUSTOMERS_TRX_LINES_ALL RCTLA
WHERE RCT.TRX_NUMBER = '1000001'
AND   HOU.ORGANIZATION_ID = RCT.ORG_ID
AND   RCT.BATCH_SOURCE_SEQ_ID = ABSA.BATCH_SOURCE_SEQ_ID
AND   RCTT.CUST_TRX_TYPE_SEQ_ID = RCT.CUST_TRX_TYPE_SEQ_ID
AND   RS.RESOURCE_SALESREP_ID = RCT.PRIMARY_RESOURCE_SALESREP_ID
AND   RT.TERM_ID = RCT.TERM_ID
AND   RT.LANGUAGE = 'US'
AND   RCT.CUSTOMER_TRX_ID = APS.CUSTOMER_TRX_ID
AND   RCT.BILL_TO_CUSTOMER_ID = HCA_BILL_TO.CUST_ACCOUNT_ID
AND   HZ_BILL_TO.PARTY_ID = HCA_BILL_TO.PARTY_ID
AND   HCA.BILL_TO_CUS_ACCOUNT_ID = HCAS_BILL_TO.CUST_ACCOUNT_ID
AND   HCAUA_BILLTO.CUST_ACCT_SITE_ID = HCAS_BILL_TO.CUST_ACCT_SITE_ID
AND   HCAUA_BILLTO.SITE_USE_CODE = 'BILL_TO'
AND   RCT.CUSTOMER_TRX_ID = RCTLA.CUSTOMER_TRX_ID
AND   RCT.TRX_NUMBER BETWEEN :P_TRX_NUM_FRM AND :P_TRX_NUM_TO
AND   HZ_BILL_TO.PARTY_NAME  =:P_CUSTOMER_NAME) OR (:P_CUSTOMER_NAME IS
NULL))

```

```

SAVE AS > XXTEST_UDEMY FOLDER > XXAR_INVOICE_REPORT > OK.
DATA > VIEW > TABLE VIEW.
SELECT MORE THAN ONE BIND PARAMETERS
SELECT PARAMETER TYPE AS TEXT FOR NOW
SAVE.

```

32. WORKING WITH EXCEL TEMPLATES - PART 1

1. CREATE A DATAMODEL AND SVE TO CATALOG.

```
DATA MODEL> WORK_EXCEL_RPT
```

```
SELECT A.PO_HEADER_ID,A.SEGMENT1,A.TERMS_ID,FROM PO_HEADERS_ALL A, PO_LINES_ALL B  
WHERE A.PO_HEADER_ID = B.PO_HEADER_ID  
AND ROWNUM<=20
```

```
DATA>VIEW>SAMPLE DATA>OK>SAVE AS>MY FOLDER/TEST/WORK_EXCEL_REPORT_DM>OK
```

2. CREATE EXCEL TEMPLATE AND SAVE AS .XLS AND ATTACH DATA MODEL TO LAYOUT
POINTS TO REMEMBER:

FOR ELEMENTS OR COLUMNS USE SYNTAX: XDO_?ELEMENT_NAME?>

EX: XDO_?PO_HEADER_ID?>

FOR GROUP USE SYNTAX: XDO_GROUP_?GROUP_NAME?>

EX: XDO_GROUP_?G_1?>

NEW >EXCEL SHEET>SAVE AS WORK_EXCEL_RPT_LAYOUT

The Oracle logo is displayed in a large, light red, sans-serif font. The word "ORACLE" is followed by a registered trademark symbol (®).

HOW TO CREATE ANALYSIS- FIRST REPORT

SUBJECT AREA: COLUMNS THAT ARE PROVIDED BY ORACLE WHICH YOU WANT TO PUT IN A REPORT.

REPORT	SQL QUERY/DATA MODEL	SELECT COLUMN	LAYOUT	RUN THE REPORT	OUTPUT
ANALYSIS/OTBI REPORT	SUBJECT AREA (PREBUILD QUERY)	SELECT COLUMN	LAYOUT	RUN THE REPORT	OUTPUT

PROCEDURE :

NAVIGATION>MENU>TOOLS>REPORT AND ANALYTICS>BROWSE CATALOG>NEW> ANALYSIS (OTBI REPORT)>SUBJECT AREA> WE WILL SELECT ACCORDINGLY

ADVANTAGES :

1. ORACLE ALREADY PROVIDED THE STANDARD PREBUILD SUBJECT AREA ONLY WE NEED TO CREATE ANALYSIS, AS PER BUSINESS BY SELECTION REQUIRED COLUMNS. WE HAVE IT FOR THE DIFFERENT USERS LIKE:
 - SCM—
 - CRM—
 - HCM—
 - FINANCIAL—
2. FUNCTIONAL AND TECHNICAL GUYS CAN WORK.
3. WE CAN CREATE DASHBOARD IN WHICH WE CAN EMBED MORE THAN ONE ANALYSIS.

DISADVANTAGES :

1. WE CANNOT CREATE CUSTOM/NEW SUBJECT AREA.

ORACLE®

WORKING WITH COMPOUND LAYOUT (TITLE, LOGO, CALCULATED FIELD, PARAMETERS)

AGENDA

1. CREATE SUPPLIER WISE INVOICE LISTING REPORT.

NAVIGATION>MENU>TOOLS>REPORT AND ANALYTICS>BROWSE CATALOG>NEW> ANALYSIS (OTBI REPORT)>SUBJECT AREA>PAYABLE INVOICES - TRANSACTIONS REAL TIME.>SELECT BELOW FIELDS AND SAVE IT AS > **SUPPLIER_WISE_INVOICE_LISTING_REPORT** > OK

CRITERIA >

- BUSINESS UNIT > **BUSINESS UNIT NAME**
- SUPPLIER > **SUPPLIER**
- SUPPLIER > **SUPPLIER NUMBER**
- SUPPLIER SITE > **SITE**
- INVOICE DETAILS > GENERAL INFORMATION > **INVOICE ID**
- INVOICE DETAILS > GENERAL INFORMATION > **INVOICE NUMBER**
- INVOICE AMOUNT > **INVOICE AMOUNT**
- INVOICE AMOUNT > **TOTAL TAX CHANGED**
- INVOICE DETAILS > GENERAL INFORMATION > **INVOICE CREATION DATE**
- INVOICE AMOUNT > **DISCOUNTABLE AMOUNT**
- INVOICE AMOUNT > **INVOICE CURRENCY**
- INVOICE LINES > INVOICE LINE AMOUNT > **INCLUDED TAX AMOUNT**
- INVOICE LINES > INVOICE LINE AMOUNT > **INVOICE LINE AMOUNT**
- --TAX AMOUNT LINE
- --**ORIGINAL AMOUNT=THIS IS INVOICE LINE AMOUNT-TAX AMOUNT**

ADDING FILTERS SIMILAR TO WHERE CLAUSE: BY GOING TO THE SETTING> FILTER

- BUSINESS UNIT > **BUSINESS UNIT NAME** > FILTER > VALUE = US1> SEARCH > US1 > MATCH CASE>SEARCH > SELECT > **US1 BUSINESS UNIT**
- SUPPLIER > **SUPPLIER** > FILTER > VALUE = ABC BANK, ABC CONSULTING > SEARCH > US1 > MATCH CASE>SEARCH > SELECT > **US1 BUSINESS UNIT**

BEFORE WORKING WITH PAGE CONTROLS WE WILL **REMOVE SUPPLIER FILTER IN CRITERIA** TO ACCESS ALL SUPPLIERS IN COMPOUND LAYOUT. AS DROPDOWN LIST WILL NOT APPEAR IF IT IS NOT REMOVED

OK> **RESULTS** >

COMPONENT LAYOUT: TITLE AND TABLE.

SELECTION STEPS: COMMON TO ALL

TOOL BAR> SHOW HOW THE RESULT LOOK IN THE DASHBOARD > DEFAULT.

2. CHANGING THE COMPOUND LAYOUT NAME.

IN TOOL BAR > RENAME COMPONENT LAYOUT > **SUPPLIER_INVOICE_DETAILS**

3. WORKING WITH TITLE AND LOGO.

COMPOUND LAYOUT >

- TITLE > FORMAT CONTAINER > CHANGING THE BACKGROUND COLOR AND SPACE OF TITLE.
- TITLE > EDIT VIEW > WRITE TITLE > UNCHECK DISPLAY SAVED NAME > OK
- TITLE > EDIT VIEW > CHECK LOGO > SELECT LOGO AND SUBTITLE > OK
SAVE.

WE CAN ALSO REMOVE TITLE BUT IF WE WANT AGAIN THEN WE CAN DRAG AND DROP FROM VIEW> TITLE

4. WORKING WITH TABLE LAYOUT

- ALTERNATIVE HIGHLIGHTING OF DATA
TABLE > EDIT VIEW > CHECK ENABLE ALTERNATE STYLING > OK
YOU CAN DRAG THE COLUMN UPWARDS IN TABLE MENU FOR DROP DOWN SELECTION.
- CHANGING SPACES LPAD

TABLE > FORMAT CONTAINER > ADDITIONAL FORMATTING OPTIONS > LEFT PAD 30 > OK > SAVE > TOOL BAR> SHOW HOW THE RESULT LOOK IN THE DASHBOARD > DEFAULT.

- CONTENT PAGING := PAGE CONTROLS (FOR NOT HAVING SIDE SCROLL BAR)
VIEW PROPERTIES > STYLE > SELECT CONTENT PAGING >PAGE CONTROLS: HIDE>OK.

FOR SHOWING LINES PER PAGE:

VIEW PROPERTIES > STYLE > SELECT CONTENT PAGING > PAGE CONTROLS: TOP > ROWS PER PAGE: 25 >OK.

SAVE > CHECK WITH DASHBOARD > TOOL BAR> SHOW HOW THE RESULT LOOK IN THE DASHBOARD > DEFAULT.

5. WORKING WITH SIMPLE CALCULATED FIELD.

ORIGINAL AMOUNT = (THIS IS INVOICE LINE AMOUNT - TAX AMOUNT)

TABLE > EDIT VIEW.

TOOL > NEW CALCULATED MEASURE > FOLDER HEADING: ORIGINAL_AMOUNT > CUSTOM HEADING: ORIGINAL_AMOUNT > SUBJECT AREA > INVOICE LINES > INVOICE LINE AMOUNT (DRAG ON RIGHT) > SUBTRACT SIGH > INCLUDED TAX AMOUNT

".INVOICE AMOUNTS"."INVOICE AMOUNT" - ".INVOICE AMOUNTS"."TOTAL TAX CHARGED/AMOUNT"

OK > SAVE > NEW COLUMN WILL BE SEEN IN CRITERIA.

IF WE WANT TO EDIT FORMULA > ORIGINAL_AMOUNT > EDIT FORMULA.

6. HIDING THE COLUMNS IN OTBI REPORT LAYOUT

7. TABLE > EDIT VIEW> COLUMNS AND MEASURES > SETTINGS PER COLUMN > HIDDEN.