**BI Publisher Concepts**

* Data Models (SQL, View Objects, Web Services)
* Layout Templates (RTF, Excel, ETEXT)
* Output formats (PDF, Excel, RTF, HTML, CSV, etc)
* Bursting
* Scheduling
* Integration with HCM Extracts

🡪components of BI report: Data model, Report(Layout-RTF file), Output format

🡪Data model – it is an object that contains a set of instructions for BI publisher to retrieve data for a report.

Workstructure reports like department validation report, job validation report, grades validation report, location validation report.

Worker related report – personal information report, assignment validation report, salary information report.

🡪first we can create department validation report: build sql query by using tables like per\_departments, click on create and select data model, create report, generate rtf template, view report, save, we can get formatted output like html, pdf, rtf, excel, pptx by clicking on export button), to edit the report click on edit report then you can download the rtf template and edit the columns and then delete the template, add new layout and upload it by saving. [RTF TEMPLATE]

🡪add parameters - :P\_dept\_name in sql query, nvl(:P\_dept\_name,PD.name), list of values for the parameters,

🡪[ETEXT TEMPLATE] & [EXCEL TEMPLATE]

🡪BI reports use duty roles & data roles

Sample BI reports:

Workstructure reports like department validation report, job validation report, grades validation report, location validation report.

Worker related report – personal information report, assignment validation report, salary information report.

Grade change report, Rehire employee details report

🡪common tables in core hr,

Per\_all\_assignments\_m - contains all the history of assignment details

Per\_all\_people\_f – main table. contains personal information like person id,

Per\_persons\_name\_f - names

Per\_all\_assignments\_f – view of the per\_all\_assignments\_m table where we do not have history of the assignment just the latest data. For example: current grade which he is in.

Per\_email\_addresses

Per\_persons – stores person date of birth details

Per\_person\_types

Per\_phones

Per\_religions

Per\_users

🡪work structure tables:

Hr\_locations, hr\_all\_organization\_units, HR\_organization\_units\_F\_TL

Per\_jobs, per\_grades, hr\_all\_positions

🡪absence: anc\_per\_abs\_entries, anc\_per\_abs\_plan\_entries, anc\_abs\_balances

* Independent parameter: go to data model – click on add parameters- provide select query Name: P\_COUNTRY, data type, Menu, create LOV, assign LOV to parameter.
* Dependent parameter: create independent parameter and create LOV for the dependent parameter with SQL like select state\_name from states where country\_code = :P\_country,

Create dependent parameter – assign LOV. Make sure to **check the box** for **Refresh other parameters on change** in P\_COUNTRY so that P\_STATE refreshes when P\_COUNTRY changes.

🡪The choice between these two depends on the business requirement. If we want to show **no data or filtered data** when nothing is selected, we use **Null Values Passed(WHERE (:P\_DEPT IS NULL OR department\_id = :P\_DEPT)**. If we want to show **all data by default**, we use **All Values Passed(WHERE department\_id IN (:P\_DEPT)**.

* We can store large number of values in single-select parameter but we have limitations for multi select parameter 999.
* Action Link:  **Action Links** are typically used in **OTBI reports**, not directly in BI Publisher (using RTF template). They allow users to click on a value in a report and navigate to: Another OTBI analysis or dashboard
* Views(custom SQL query) and Secured views: by default BIP report will show all the data irrespective of the role. So we must use **secured views** for security like If you use PER\_PERSON\_SECURED\_LIST\_V in your BI Publisher report, it will **respect the user's data access** (e.g., HR can see all employees, but a manager sees only their team).
* Custom ESS job in BIP: enterprise scheduler job which is used to automate and schedule task such as running report or executing different processes.

Create bip report, then Settings & actions- setup and maintenance – tasks – search for manage enterprise scheduler job definition and job sets for human capital management

* Click **Create New Job Definition**.
* Fill in:
  + **Name**: XX\_EMPLOYEE\_DETAILS\_JOB
  + **Display Name**: Employee Details Report
  + **Application**: Choose appropriate module (e.g., Human Capital Management)
  + **Job Type**: BIPJobType
  + **Report Path**: /Custom/HCM/Reports/EmployeeDetails/EmployeeDetailsReport.xdo
  + **Output Type**: PDF, Excel, etc. (create parameters if any)

**Under tools** – schedule process – schedule new process – provide report name – submit.

Click on advanced button while submitting the job – we can give frequency.

* How to give view only permissions to the user: select report- more - permissions – set folder permissions like view/edit/schedule etc.
* To restrict bi report when sql return nothing then we use event trigger option to restrict bi report to not send empty output.

Under data model – under data sets section – find section called event triggers.

Language PL/SQL:

BEGIN

  SELECT COUNT(\*) INTO :X\_COUNT FROM your\_table WHERE your\_conditions;

  IF :X\_COUNT = 0 THEN

    RAISE\_APPLICATION\_ERROR(-20001, 'No data found for the selected criteria.');

  END IF;

END;

* RTF template (formatted reports, documents etc – o/p pdf, rtf, excel etc) – use only 63 columns

Excel (tabular reports & analysis, o/p – excel, pdf)

Etext(Flat files, o/p:TXT,CSV)

* By using archive and unarchive options, we can migrate bi reports btw instances.

Query optimization – select \* should not be given, avoid sub queries, small alias names to the tables, not to use multiple datsets, avoid using groupby instead use windows function.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

🡪Bursting concept, its usecases in real time, parameters like email/FTP etc.

bursting allows you to split a single report into multiple outputs based on a key (e.g., employee ID) and deliver each output to different destinations like email, printer, or FTP..

Key components:

* **Split By**: Element used to divide the report (e.g., EMPLOYEE\_ID)
* **Deliver By**: Element used to determine delivery method (e.g., EMAIL)
* **Delivery Query**: SQL query that defines delivery details (email, format, template)

🡪

1. Create a **Data Model** with required data sets.
2. Define **Bursting Definition** in the Data Model.
3. Create a **Delivery Query** with columns like KEY, TEMPLATE, LOCALE, OUTPUT\_FORMAT, DELIVERY\_METHOD, EMAIL.
4. Attach the bursting definition to the report.
5. Schedule the report using **"Bursting Report Job"**.

🡪whomever employee did not update the phone number for those employees only we have to send notification.

To achieve this, I developed a query

Person id, person number, person email, person name and in the where condition is where phone number is null.

Then we can go with bursting concept.

Here we have 2 concepts, split by(on which basis we are splitting the information) & deliver by(who wants to receive the notification)

We have multiple parameters and key value is person\_id

Output is pdf, template rtf, parameter 1-to, cc, bcc,subject, body, attachment is true/false.

After this we will schedule a report automatically whomever not updated their phone number they will receive notification.

SELECT

    ppf.PERSON\_ID,

    ppf.FULL\_NAME,

    ppf.EMAIL\_ADDRESS,

    'Phone number missing. Please update your contact details.' AS MESSAGE

FROM PER\_ALL\_PEOPLE\_F ppf

LEFT JOIN PER\_PHONES ph ON ppf.PERSON\_ID = ph.PERSON\_ID AND ph.PHONE\_TYPE = 'M'

WHERE ph.PHONE\_NUMBER IS NULL

AND SYSDATE BETWEEN ppf.EFFECTIVE\_START\_DATE AND ppf.EFFECTIVE\_END\_DATE

AND ppf.CURRENT\_EMPLOYEE\_FLAG = 'Y'

Bursting query:

SELECT DISTINCT

    ppf.PERSON\_ID AS "KEY",                                -- Split by employee

    'MissingPhoneTemplate' AS TEMPLATE,                    -- Template name

    'en-US' AS LOCALE,                                     -- Language

    'PDF' AS OUTPUT\_FORMAT,                                -- Format

    'EMAIL' AS DEL\_CHANNEL,                                -- Delivery method

    ppf.EMAIL\_ADDRESS AS PARAMETER1,                       -- To: Employee email

    'hr.team@yourcompany.com' AS PARAMETER2,               -- Cc (optional)

    'support@yourcompany.com' AS PARAMETER3,               -- Bcc (optional)

    'Update Your Mobile Number' AS PARAMETER4,             -- Email Subject

    'Hi ' || ppf.FULL\_NAME || CHR(13) ||                   -- Email Body

    'Our records show your mobile number is missing.' || CHR(13) ||

    'Please update it in Fusion HCM.' AS PARAMETER5,

    'true' AS PARAMETER6,                                  -- Attach report

    'donotreply@yourcompany.com' AS PARAMETER7             -- From address

FROM PER\_ALL\_PEOPLE\_F ppf

LEFT JOIN PER\_PHONES ph

    ON ppf.PERSON\_ID = ph.PERSON\_ID

    AND ph.PHONE\_TYPE = 'M'

WHERE ph.PHONE\_NUMBER IS NULL

  AND SYSDATE BETWEEN ppf.EFFECTIVE\_START\_DATE AND ppf.EFFECTIVE\_END\_DATE

  AND ppf.CURRENT\_EMPLOYEE\_FLAG = 'Y'

  AND ppf.EMAIL\_ADDRESS IS NOT NULL

🡪If any name changes like first/last name got changed in single day then those details needs to send to employee at the end of the day.

Tables: per\_all\_people\_f && per\_person\_names\_f

SELECT

papf.person\_id,

papf.person\_number,

papf.effective\_start\_date AS papf\_eff\_start,

papf.effective\_end\_date AS papf\_eff\_end,

ppnf.first\_name,

ppnf.last\_name,

ppnf.effective\_start\_date AS ppnf\_eff\_start,

ppnf.effective\_end\_date AS ppnf\_eff\_end,

ppnf.last\_update\_date

FROM per\_all\_people\_f papf , per\_person\_names\_f ppnf

WHERE papf.person\_id = ppnf.person\_id

AND ppnf.name\_type= 'GLOBAL'

AND trunc(sysdate) BETWEEN papf.effective\_start\_date AND papf.effective\_end\_date

AND trunc(sysdate) BETWEEN ppnf.effective\_start\_date AND ppnf.effective\_end\_date

AND trunc(ppnf.last\_update\_date) between trunc(sysdate -1) and trunc(sysdate)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

🡪common BIP reports scenario-based questions, SQL queries, joins

Grade change report:

"In our project, we created a BI Publisher report to track employee grade changes using PER\_ALL\_ASSIGNMENTS\_M and PER\_GRADES. We filtered records based on LAST\_UPDATE\_DATE and included assignment details. The report was scheduled monthly and used bursting to notify HR and managers about grade updates."

SELECT

    papf.PERSON\_NUMBER,

    papf.FULL\_NAME,

    paam.ASSIGNMENT\_ID,

    paam.EFFECTIVE\_START\_DATE,

    paam.EFFECTIVE\_END\_DATE,

    paam.GRADE\_ID,

    pg.NAME AS GRADE\_NAME,

    paam.LAST\_UPDATE\_DATE

FROM PER\_ALL\_ASSIGNMENTS\_M paam

JOIN PER\_ALL\_PEOPLE\_F papf ON paam.PERSON\_ID = papf.PERSON\_ID

JOIN PER\_GRADES pg ON paam.GRADE\_ID = pg.GRADE\_ID

WHERE paam.LAST\_UPDATE\_DATE BETWEEN SYSDATE - 30 AND SYSDATE

AND paam.ASSIGNMENT\_TYPE = 'E'

AND paam.GRADE\_ID IS NOT NULL

ORDER BY paam.LAST\_UPDATE\_DATE DESC

Rehire employee details report:

"We created a BI Publisher report using PER\_PERIODS\_OF\_SERVICE to identify rehired employees. We filtered records where REHIRE\_REASON is not null and joined with assignment and person tables to show rehire date, termination date, and current status. This helped HR track rehires for onboarding and compliance."

SELECT

    papf.PERSON\_NUMBER,

    papf.FULL\_NAME,

    ppos.START\_DATE AS REHIRE\_DATE,

    ppos.END\_DATE AS TERMINATION\_DATE,

    paam.ASSIGNMENT\_STATUS\_TYPE,

    paam.EFFECTIVE\_START\_DATE,

    paam.EFFECTIVE\_END\_DATE

FROM PER\_PERIODS\_OF\_SERVICE ppos

JOIN PER\_ALL\_PEOPLE\_F papf ON ppos.PERSON\_ID = papf.PERSON\_ID

JOIN PER\_ALL\_ASSIGNMENTS\_M paam ON papf.PERSON\_ID = paam.PERSON\_ID

WHERE ppos.REHIRE\_REASON IS NOT NULL

AND SYSDATE BETWEEN papf.EFFECTIVE\_START\_DATE AND papf.EFFECTIVE\_END\_DATE

AND papf.CURRENT\_EMPLOYEE\_FLAG = 'Y'

ORDER BY ppos.START\_DATE DESC

Mandatory conditions:

Paam.assignment type = E

Legistlation code = US

Empl\_type = E

Primary\_flay = Y (latest record)

Max condtion in ppos

Select max of paam. Effective start date from per\_all\_assignments\_f

🡪otbi – transactional data, useful for insights as we have drag and drop interface… not ideal for complex requirements as we do not use join

Blank ethnicity report

"We created an OTBI report using the 'Workforce Management – Person Real Time' subject area to identify employees with blank ethnicity. We applied a filter on the Ethnicity field to check for null or blank values and scheduled the report for HR to follow up with those employees."

Users only see data they are authorized to view (e.g., HR Specialist for India sees only India data)

blank gender report

"We created an OTBI report using the 'Workforce Management – Person Real Time' subject area to identify employees with blank gender values. We applied a filter on the Gender field to check for null or blank entries and scheduled the report for HR review."

🡪 Action link: Create target dashboard(employees filtered by department)-then department-level report (include dept name/dept ID) - add action link from source dpt column under column properties. Browse and select target dashboard

🡪how do we migrate report from one pod to another

🡪data security

🡪how to schedule a report in OTBI: using agent

🡪absence and talent management [common tables and functional configurations]

OTBI Concepts

Subject Areas

Real-time reporting

Filters, prompts, views

Drill-down and dashboard creation

Security via duty roles

🡪 Fully supported with Action Links and navigation

🡪 Supports agent-based scheduling

🡪 Used for interactive dashboards, KPIs, drill-downs

🡪 Automatically applies row-level security based on user roles

HCM extracts process - to implement changes only data extraction. First we have to create extract definitions(name, type, check changes only), then we have to create data group(logical grouping of related data)and select user entity – threading db item – enable root data group – we need to create records under which we add attributes. One key attribute as primary key i.e, person\_id.

Now we create BI publisher report to get the result – we have standard datamodel globalreportdatamodel – copy it – get etext template - ensure tag names are correct – create report attach data model - upload template – now map this report to the extract – go to extract delivery options provide report path name – under execution tree we have to select compile all formula.

Data exchange – view user entity details – user entity query.

**Tell me about yourself:**

**Good afternoon. First of all thank you for giving me this opportunity to be interviewed for this position today. Let me introduce myself**

**My name is gayathri. I have completed by bsc computer science in 2019. I have over all 6 years of experience and the relevant experience is around 3 years as oracle fusion hcm technical consultant currently associated with cognizant for about 3 years now. I have experience working in both implementation/support project. Currently im working for ASSA ABLOY client where I handle activities like preparing bi publisher, otbi reports mainly focusing on core hr and absence management. I also have good knowledge on HDL data loading and extracts.**

**I would describe myself as someone who is committed to continuous learning and delivering value. I’m really looking forward to applying my skills here and contribute effectively to the team.**

**Thank you!**

**Project experience –** I am currently working for ASSA ABLOY client, a global leader in access solutions. I am part of oracle hcm cloud implementation project team where we are mainly implementing core hr and absence module. my responsibilities include involving requirement gathering for reports from business users. Helping in report development using BIP and OTBI for various management reports. Participated in migrating reports across environments like dev, test \* prod Supporting data migration activities using HDL to load employee , job and other data into the system.

Functional design document,

technical design document

functional testing/ unit testing

email type -Work/Home

address table