PBI- HR DATA ASSESSMENT QUESTIONS

Q1.) Data Import and Transformation: Can you show me how to import the employee data

from the Excel files and transform it to remove any unnecessary columns or rows?

Answer- Here are steps I took to import and transform the data.

1. First of all i imported all files by using Get Data in home ribbon
2. Then I cleaned the “employee\_survey\_data” table. By replacing “NA” value with 0 from all three columns, I chose 0 because all three columns values are numeric.
3. Then I checked the “general\_data” table, there is no missing or null value there.
4. Then I cleaned the “in\_time” table. I removed column 2, column 11, column19, column 47 and many others similar to it,because they had only one row with data info, and all rows with null. Furthermore, i changed the data type of all columns from text to date/time and then i used fill down method to replace null. (null became ‘error’ after changing the data type , i again changed ‘error’ to null before using fill down method)
5. Next table was “manager\_survey\_data”, so I didn't find any null/missing values. I changed their data type from text to whole number.
6. In next column “out\_time”, i removed first row of all columns, then i removed columns with ‘null’ in all rows, then i changed the data type of all columns from text to data/time, due to this all null values converted into ‘error’, so i replaced ’error’ back to null. And replaced null with fill down method.
7. In ‘data dictionary’ table, in first column and in second column i used fill down, and in 3rd column i replaced null with NA.

Q2.) Basic Visualization: Create a simple bar chart to visualize the distribution of employees by

Department.

Answer- I have created a stacked bar chart, kindly check in my power BI file. I am providing the approach below.

1. In the Visualizations pane, select the "Stacked Bar Chart" icon to create a bar chart.
2. Now drag the department field Y-axis
3. And employee\_id field to X-axis.
4. Power Bi will create a visualization.

Q3.) Filtering Data: How can you create a slicer to allow users to filter employees based on

their job role in Power BI?

Answer- I created a slicer, kindly check in my power BI file. I am providing the approach below.

1. Choose slicer from the visualization pane.
2. Drag the "Job Role" field to the Values area of the slicer.
3. Slicer is ready, you can customize it in the format pane.

Q4.) Joining Data: Explain what kind of join you would use to combine the employee data with

the in-time and out-time data, and why?

Answer- I would use a left join to combine together employee data with in-time and out-time details. This way, every employee is included in the final set, whether or not they have recorded time. It ensures we keep all employee info intact, even if some don't have associated time records. This approach is about completeness and flexibility for cases with missing time entries.

Q5.) Calculated Columns: Create a calculated column to determine the age group of employees

(e.g., under 30, 30-40, 40-50, over 50).

Answer- I have created this calculated column using the DAX formula, i am providing below. Kindly check in my Power BI report.

Age Group =

SWITCH(

TRUE(),

'general\_data'[Age] < 30, "Under 30",

'general\_data'[Age] >= 30 && 'general\_data'[Age] <= 40, "30-40",

'general\_data'[Age] > 40 && 'general\_data'[Age] <= 50, "40-50",

'general\_data'[Age] > 50, "Over 50"

)

Q6.) Measures in DAX: Calculate the average monthly income for employees and display it in a

card visualization.

Answer- I did this by two different approaches, 1st using DAX,

1. I went to modeling, selected ‘New Measure ’.
2. Then I created AvgMonthlyIncome using this DAX formula.
3. AvgMonthlyIncome = AVERAGE(general\_data[MonthlyIncome]).
4. Select Card from visualization, And frag it into card’s data field.

2nd approach.

There is a column in table ‘general\_data’ named ‘MonthlyIncome’ so you can directly drag it into the card’s data field.

Q7.) Time Intelligence: How can you use DAX to calculate the year-over-year growth in monthly

income for employees?

Answer- I am providing a DAX formula by which year-over-year growth in monthly income for employees can be achieved. I Made a measure ‘YoY\_Growth\_MonthlyIncome’ in my report. You can check in my power BI file. But this measure can’t be visualized because the dataset doesn't have columns which represent the proper date dimension, but this DAX is correct and free from syntax error.

YoY\_Growth\_MonthlyIncome =

VAR CurrentMonthIncome = SUM('general\_data'[MonthlyIncome])

VAR PreviousYearIncome = CALCULATE(SUM('general\_data'[MonthlyIncome]),

SAMEPERIODLASTYEAR('general\_data'[TotalWorkingYears]))

RETURN

IF(ISBLANK(PreviousYearIncome), BLANK(), DIVIDE(CurrentMonthIncome - PreviousYearIncome, PreviousYearIncome, 0))

Q8.) Hierarchies: Create a hierarchy for the date and time columns to allow for easy drill-down

analysis.

Answer - Here is an approach to achieve this. There is no column related to date and time in the given dataset. So I am sharing the step by step approach.

1. Assuming you have a date column named 'Date' in your 'general\_data' table.
2. Locate the 'Date' column in your 'general\_data' table.
3. Right-click on the 'Date' column.
4. Choose "New hierarchy" from the context menu.
5. Drag and drop the hierarchy levels in the desired order. Typically, you would want to arrange them from years to months and then days.
6. Right-click on the new hierarchy.
7. Choose "Rename" to give it a meaningful name, such as 'Date Hierarchy.'
8. Go to the "Report" view in Power BI Desktop.
9. Drag the 'Date Hierarchy' into your visualizations.
10. This might be a line chart, bar chart, or any other suitable visualization.

Q.9) Advanced DAX Calculation: Calculate the attrition rate for each department and visualize it using a heatmap.

Answer- Here are the steps, By which I have achieved this.

1. First of all i checked data types of all required columns, and changed them accordingly.
2. The attrition column is in Yes and No, So, I created a new column named ‘Attrition\_int’ which indicates Yes as 1 and No as 0.
3. Then I created a new Measure using DAX, using assume and divide attrition by number of employees or number of row count, to get the attrition rate.
4. Then I created a Heat map, using Table Heatmap.
5. Kindly check it on my Power BI report.

Q10.) Advanced Join: Combine the employee data with a different dataset using a left join and explain the potential pitfalls.

Answer- I cleaned both tables by removing nulls and blanks, I merged ‘Employee\_survey\_data’ to ‘general\_data’ by the common column as a key ‘EmployeeId’. And created a new column named ‘merge employee data with general data’. Kindly check in my Power Bi report.

Explaining Potential pitfalls.

1. Pitfall: If there are multiple surveys for a single employee in employee\_survey\_data, it may lead to redundancy.

Solution: Review and aggregate data as needed to avoid overcounting.

1. Pitfall: Aggregating data after the left join may be impacted by survey-related columns.

Solution/Remedy: Use appropriate aggregation functions, and understand the nature of the merged data.

1. Pitfall: Left joins with large datasets can still impact performance.

Solution: Optimize performance by selecting the right columns for the join and filtering data where possible.

1. Pitfall: Ensure that the employee\_id is unique in both datasets to avoid unexpected results.

Solution: Check for duplicates and resolve any data integrity issues.

Q11.) Complex Filtering: Create a dynamic filter that allows users to filter employees based on both department and job role simultaneously.

Answer- Select slicer from visualization Pane, First drag Department column into the data field of slicer and then Drag and drop the Job Role field. Your dynamic filter is ready and you can customize it according to the needs from the format pane.

Q12.) Advanced Time Intelligence: Calculate the moving average of monthly income over a rolling 3-month period using DAX

Answer- To calculate the moving average of monthly income over a rolling 3-month period using DAX in Power BI, I used the AVERAGEX function.

As i mentioned above, ‘Date’ column is missing in dataset. So, i can’t perform this in my Power BI report but i am giving DAX to create this with explanation.

MovingAverage3Months =

CALCULATE(

AVERAGEX(

DATESINPERIOD(

'general\_data'[Date],

LASTDATE('general\_data'[Date]),

-3,

MONTH

),

'general\_data'[MonthlyIncome]

)

)

Explanation of the DAX measure:

1. DATESINPERIOD: Generates a table of dates for the last 3 months from the current date.
2. AVERAGEX: Iterates over the dates in the generated table and calculates the average of the 'MonthlyIncome' for each date.
3. CALCULATE: Modifies the context in which the calculation is made.

Q13.) Conditional Formatting: Apply conditional formatting to a table to highlight employees with the highest and lowest monthly incomes.

Answer- Here are steps that I have applied.Kindly check on my PowerBI Report.

1. Firstly I chose Table from the visualization pane.
2. Dragged the MonthlyIncome EmployeeID column into the data field.
3. Then I select the MonthlyIncome in the data field and right click on it.
4. Where I found the Conditional Formatting option and I went for Background color.
5. Then I selected the gradient option and chose light pink color for low income and Red color for high income and clicked on OK.
6. It will apply on the table.

Q14.) Parameter Tables: How can you use parameter tables to allow users to set their own thresholds for performance ratings and visualize the results?

Answer- Here is step by step approach by which i have done this complicated task, and i have added this in my Power BI report, kindly check that.

Steps:

1. Created Parameter Table:

I navigated to the "Home" tab and clicked on "Enter Data" to create a new table. I generated a table with two columns, namely 'RatingThreshold' and 'ThresholdValue.' Then, I entered the performance rating thresholds and corresponding threshold values in this table.

1. Established Relationship:

Moving to the "Model" view, I created a relationship between the 'PerformanceRating' column in my main data table and the 'RatingThreshold' column in the parameter table.

1. Crafted Measure:

In the "Model" view, I created a new measure using the following DAX formula to categorize performance ratings based on user-defined thresholds:

PerformanceCategory =

VAR MaxPerformanceRating = CALCULATE(MAX('Manager\_survey\_data'[PerformanceRating]))

VAR ThresholdValues = VALUES('ParameterTable'[ThresholdValue])

RETURN

SWITCH (

TRUE (),

MaxPerformanceRating = MAXX(ThresholdValues, 'ParameterTable'[ThresholdValue]), "High",

"Low" )

1. Built Visualizations:

I have visualized the performance category by card.

1. Added Slicers:

I added slicers based on the columns in the parameter table, enabling users to dynamically adjust the thresholds according to their preferences.

Q16.) Aggregations: Explain how to optimize performance by creating aggregations for large

datasets.

Answer- Optimizing performance for large datasets in Power BI involves creating aggregations to speed up queries and improve report responsiveness. Here are steps in simple terms:

1. Understand the Data Size: Before optimizing, understand the size and complexity of your dataset. Larger datasets may require special considerations for performance.
2. Identify Key Metrics: Identify the key metrics or measures that are frequently used in your reports. These are the values you want to aggregate.
3. Choose Aggregation Levels: Determine the aggregation levels or granularities that make sense for your data. For example, you might aggregate data at the day, week, or month level.
4. Create Aggregated Tables:
   1. In Power BI Desktop, go to the "Model" view.
   2. Duplicate your main data table and create aggregated tables using functions like GROUP BY or SUMMARIZE. Aggregate the data based on the identified levels.
5. Create Relationships: Establish relationships between your original (detailed) table and the newly created aggregated tables.
6. Build Measures on Aggregated Data: Create measures using the aggregated tables. These measures should reflect the same metrics as in your detailed data but leverage the aggregations for faster performance.
7. Optimize Visuals: Modify your reports to use the measures from the aggregated tables. This ensures that your visuals are now drawing on the pre-aggregated data.
8. Test and Refine: Test your reports to ensure they still provide accurate results. Refine the aggregations or relationships if needed.
9. Apply to Large Datasets: Once satisfied with the optimizations, publish the report to the Power BI service and apply the same principles to larger datasets if necessary.
10. Monitor Performance: Regularly monitor report performance and make adjustments as your data evolves. This might include refreshing aggregations or adding new ones based on changing reporting needs.

Q17.) What-If Analysis: Use What-If parameters to show how attrition rates change when you adjust different factors (e.g., salary increase)

Answer- It was really a very challenging task but I did it, kindly check it on my Power BI report.

Here i am sharing my step by step approach

Created What-If Parameter:

* I initiated the What-If analysis by entering the "Model" view in Power BI Desktop.
* Navigating to the "Model" ribbon, I clicked on the "New Parameter" button in the "External tools" group.
* Assigned the parameter a name, "SalaryIncrease," and chose the type as "Numeric range."
* Defined a suitable range (0% to 50%) and an increment of 1%.
* Clicked "OK" to create the numeric range parameter.

Created Measure for Adjusted Monthly Income:

* After establishing the What-If parameter, I proceeded to create a measure for adjusted monthly income.
* Used DAX to formulate the measure: 'AdjustedMonthlyIncome = 'general\_data'[MonthlyIncome] \* (1 + 'What-If Parameters'[SalaryIncrease]).
* This measure dynamically calculated the adjusted monthly income based on the chosen salary increase.
* Calculated Attrition Rate:
  + To gauge attrition rates, I created a measure named 'AttritionRate.'
  + The exact calculation relied on the definition of attrition in the model.
  + For example, I formulated it as COUNTROWS(FILTER('general\_data', 'general\_data'[AttritionFlag] = 1)) / COUNTROWS('general\_data').
  + This measure computed the attrition rate based on the number of rows with an 'AttritionFlag' of 1.
* Created Measure for Adjusted Attrition Rate:
  + Building on this foundation, I created another measure named 'AdjustedAttritionRate.'
  + This measure factored in the adjusted monthly income, calculating the adjusted attrition rate using CALCULATE([AttritionRate], 'general\_data'[MonthlyIncome] = 'general\_data'[AdjustedMonthlyIncome]).
* Built Visualizations:
  + With the measures in place, I moved to the report view.
  + Utilized the 'AdjustedAttritionRate' measure to construct visualizations showcasing the adjusted attrition rates.
* Added Slicers for User Interaction:
  + To enhance user interaction, I added slicers based on the 'What-If Parameters.'
  + This allowed users to dynamically adjust the salary increase and observe its impact on the adjusted attrition rate.

Q18.) Cross-Filtering: Demonstrate the use of cross-filtering between visuals to provide an interactive experience for users.

Answer- There is already a relationship created between employee\_survey\_data and general\_data because of previous tasks. Kindly check my Power BI report.  
So, To achieve this, I chose Table visualization and dragged the EnvironmentalSatisfaction column to the data field. Then I chose a stacked bar graph and dragged gender to Y-axis and EmployeeCount to X-axis from table general\_data. So, they are synced and interactive to each other.

Q19.) KPIs: Create Key Performance Indicators (KPIs) for employee performance using DAX calculations.

Answer- I am sharing my approach below, Kindly check it om my Power BI report.

* Employee Job Involvement KPI for manager\_survey\_data:
  + First, I created a DAX measure named JobInvolvementKPI for the 'manager\_survey\_data' table.
  + Using the SWITCH function, I categorized job involvement into qualitative levels.
  + Conditions were set to categorize Job Involvement levels 1 through 4 as "Low," "Moderate," "High," and "Very High" respectively.
  + If none of the conditions matched, I specified BLANK() to handle any unexpected values.
* Employee Performance Rating KPI for manager\_survey\_data:
  + Next, I created another DAX measure named PerformanceRatingKPI for the 'manager\_survey\_data' table.
  + Similar to the previous measure, I used the SWITCH function to categorize performance ratings into qualitative levels.
  + Conditions were set to categorize Performance Ratings 3 and 4 as "Satisfactory" and "Outstanding" respectively.
  + BLANK() was used as the default value for any unforeseen circumstances.
* Percent Salary Hike KPI for general\_data:
  + Moving on to the 'general\_data' table, I crafted a DAX measure called SalaryHikeKPI.
  + The SWITCH function was employed to categorize percent salary hikes into predefined ranges.
  + Ranges were set for below 10%, 10%-19%, 20%-29%, and 30% and above.
  + BLANK() was used as the default value for any values outside these ranges.
* Total Working Years KPI for general\_data:
  + Lastly, I created the WorkingYearsKPI DAX measure for the 'general\_data' table.
  + The SWITCH function was utilized to categorize total working years into different ranges.
  + Ranges were set for less than 5 years, 5-9 years, 10-14 years, and 15 years and above.
  + BLANK() was specified as the default value for any unexpected values.

Q20.) Dynamic Reporting: Show how to make a report dynamic by using bookmarks and buttons to switch between different views of the data.  
  
Answer- I have created a dynamic report for HR data, Kindly check it on the last page of my Power BI report named ‘DashBoard’. Thank you.