Accenture Data Analyst Interview

YoE (1–4 Years)

CTC: 9–13 LPA

SQL Questions

1. Write a query to fetch the top 3 employees with the highest salaries from an Employees table.

Fetch the Top 3 Employees with the Highest Salaries

Input Table: Employees

emp_id	**emp_name**	**salary**
101	Alice	95000
102	Bob	87000
103	Charlie	91000
104	David	75000
105	Eve	99000

Query:

SELECT emp_id, emp_name, salary

```
FROM Employees

ORDER BY salary DESC

LIMIT 3;
```

2. Write a Query to Find Customers Who Have Placed More Than 5 Orders in the Last 30 Days

Input Table: Orders

order_id	customer_id	order_date
201	1	2024-02-15
202	2	2024-02-10
203	1	2024-02-18
204	3	2024-02-20
205	1	2024-02-25
206	2	2024-02-27
207	1	2024-03-02

Query:

```
SELECT customer_id

FROM Orders

WHERE order_date >= NOW() - INTERVAL 30 DAY

GROUP BY customer_id

HAVING COUNT(order_id) > 5;
```

3. Write a Query to Retrieve the First and Last Purchase Date for Each Customer

Input Table: Transactions

transaction_id	customer_id	purchase_date
301	1	2024-01-10
302	2	2024-01-12
303	1	2024-02-15
304	2	2024-02-20
305	1	2024-03-05

Query:

```
SELECT customer_id,
    MIN(purchase_date) AS first_purchase,
    MAX(purchase_date) AS last_purchase
FROM Transactions
GROUP BY customer_id;
```

---

4. Write a Query to Fetch All Orders Placed on Weekends

Input Table: Orders

order_id	customer_id	order_date
401	1	2024-02-17
402	2	2024-02-18
403	3	2024-02-19
404	1	2024-02-24

Query:

```
SELECT *

FROM Orders

WHERE DAYOFWEEK(order_date) IN (1, 7);  -- 1 for Sunday, 7 for Saturday
```

5. Write a Query to Calculate the Average Sales per Region

Input Table: Sales

sales_id	region	sales_amount
501	North	50000
502	South	70000
503	North	80000
504	West	90000

Query:

```
SELECT region,

    AVG(sales_amount) AS avg_sales

FROM Sales

GROUP BY region;
```

6. Write a query to Find the Second-Highest Salary Without Using LIMIT or TOP

Query:

```
SELECT MAX(salary) AS second_highest_salary

FROM Employees

WHERE salary < (SELECT MAX(salary) FROM Employees);
```

7. Write a Query to Find Customers Who Have Never Placed an Order

Input Tables:

Customers Table

customer_id	customer_name
1	John
2	Alice
3	Bob

Orders Table

order_id	customer_id
101	1
102	2

Query:

```
SELECT c.customer_id, c.customer_name

FROM Customers c

LEFT JOIN Orders o ON c.customer_id = o.customer_id

WHERE o.customer_id IS NULL;
```

8. Write a Query to Find the Most Frequently Sold Product in Each Category

Input Table: Products

product_id	category	product_name
601	A	Apple

product_id	category	product_name
602	B	Banana
603	A	Orange

Input Table: Sales

sale_id	product_id	quantity_sold
701	601	50
702	602	30
703	601	40
704	603	60

Query:

```
SELECT p.category, p.product_name, s.total_sold
FROM (
  SELECT product_id, SUM(quantity_sold) AS total_sold,
      RANK() OVER (PARTITION BY p.category ORDER BY SUM(quantity_sold) DESC) AS rnk
  FROM Sales s
  JOIN Products p ON s.product_id = p.product_id
  GROUP BY p.category, s.product_id
) s
JOIN Products p ON s.product_id = p.product_id
WHERE s.rnk = 1;
```

9.Write a SQL query to Calculate Profit Margin for Each Product and Rank Them Within Their Category

Input Table: Products

product_id	category	revenue	cost
801	A	10000	7000
802	A	15000	9000
803	B	20000	15000

Query:

```
SELECT product_id, category,
    (revenue - cost) / revenue AS profit_margin,
    RANK() OVER (PARTITION BY category ORDER BY (revenue - cost) / revenue DESC) AS rank
FROM Products;
```

10. Write a query to Fetch Orders Where the Total Order Value Exceeds the Average Order Value

Input Table: Orders

order_id	customer_id	total_value
901	1	5000
902	2	7000
903	3	6000
904	4	8000

Query:

```
SELECT *
FROM Orders
WHERE total_value > (SELECT AVG(total_value) FROM Orders);
```

Power BI Questions

1. How would you create a dynamic report that shows revenue by region and allows filtering by product and sales channel?

Creating a Dynamic Report for Revenue by Region with Filters for Product and Sales Channel

Steps:

1. Load the data into Power BI.
2. Create a **bar chart** or **matrix visual** to display revenue by region.
3. Add slicers for **Product** and **Sales Channel**:
    - Click on the **Slicer** visual in Power BI.
    - Drag **Product** and **Sales Channel** fields into separate slicers.
4. Ensure that the slicers are synced across all report pages.
5. Use **interactions** to control how visuals respond to filters.

2. Write a DAX measure to calculate month-over-month revenue growth.

DAX Measure for Month-over-Month Revenue Growth

```
MoM Growth =
VAR PrevMonthSales =
  CALCULATE(
    SUM(Sales[Revenue]),
    PREVIOUSMONTH(Sales[Date])
  )
VAR CurrentMonthSales = SUM(Sales[Revenue])
RETURN
  IF(PrevMonthSales = 0, BLANK(), (CurrentMonthSales - PrevMonthSales) /
PrevMonthSales)
```

- This calculates the percentage change in revenue compared to the previous month.

---

3. Create a DAX measure to calculate cumulative sales for a product category year-to-date.

DAX Measure for Cumulative Sales Year-to-Date (YTD) for a Product Category

```
Cumulative Sales YTD =
CALCULATE(
  SUM(Sales[Revenue]),
  FILTER(
    ALLSELECTED(Sales),
    Sales[Date] <= MAX(Sales[Date])
  )
)
```

- This calculates running total sales for the selected product category in the current year.

---

4. Explain how to create a slicer in Power BI to toggle between different time periods (Monthly, Quarterly, Yearly).

Creating a Slicer to Toggle Between Monthly, Quarterly, and Yearly Views

Steps:

1. Create a **calculated table** with time periods:

```
TimePeriod = DATATABLE(
   "Period", STRING,
   { "Monthly", "Quarterly", "Yearly" }
)
```

2. Add this table to a **slicer**.
3. Create a **switch-based measure** to adjust the date aggregation:

```
Revenue by Period =
SWITCH(
   SELECTEDVALUE(TimePeriod[Period]),
   "Monthly", SUM(Sales[Revenue]),
   "Quarterly", CALCULATE(SUM(Sales[Revenue]), DATESQTD(Sales[Date])),
   "Yearly", CALCULATE(SUM(Sales[Revenue]), DATESYTD(Sales[Date])),
   SUM(Sales[Revenue])  -- Default to monthly if none selected
)
```

4. Use this measure in the report visuals.

---

5. Write a DAX formula to rank sales representatives based on their total sales within each region.

DAX Formula to Rank Sales Representatives by Total Sales Within Each Region

```
Sales Rank =
RANKX(
   FILTER(ALL(Sales), Sales[Region] = MAX(Sales[Region])),
   SUM(Sales[Revenue]),
   ,
   DESC,
   DENSE
```

)

- This ranks sales reps based on total sales within their respective regions.

---

6. How would you implement Row-Level Security (RLS) in Power BI to restrict data access based on user roles?

Implementing Row-Level Security (RLS) in Power BI

Steps:

1. Go to **Model View** → **Manage Roles** → **Create New Role**.
2. Define **DAX Filter** for the role:

```
[Region] = USERPRINCIPALNAME()
```

   - This restricts users to see only their assigned region's data.

3. Assign roles to users in the **Power BI Service** under **Security** settings.

---

7. How would you design a dashboard that dynamically highlights top-performing regions and underperforming products?

Designing a Dashboard to Highlight Top-Performing Regions and Underperforming Products

Key Features:

- **KPI Cards:** Show overall revenue, profit, and sales trends.
- **Bar Chart (Top-Performing Regions)**:
  - Sort revenue in descending order.
  - Use conditional formatting to highlight regions above the target.
- **Table (Underperforming Products)**:
  - Apply a **DAX measure** to flag products with sales below a threshold:

```
Low Sales Flag = IF(SUM(Sales[Revenue]) < 10000, "Underperforming", "Good")
```

    - Apply **conditional formatting** to highlight low-performing products.
- **Slicers** for dynamic filtering by region, product, and sales channel.

---

8. Explain the steps to create a Data Flow in Power BI Service to consolidate data from multiple sources.

Creating a Data Flow in Power BI Service

Steps:

1. Open Power BI Service → Go to **Workspaces** → Click **New** → **Dataflow**.
2. Select **Add Data** → Choose a **data source** (SQL, Excel, API, etc.).
3. Use **Power Query Online** to clean and transform data.
4. Define relationships and apply aggregations if needed.
5. Save and Refresh the Dataflow → Use it as a dataset in Power BI reports.

---

9. How do you optimize a large dataset in Power BI to improve performance and refresh time?

Optimizing Large Datasets in Power BI for Performance & Refresh Time

Best Practices:

- **Use Import Mode** for smaller datasets, **DirectQuery** for real-time access.
- **Remove Unused Columns** in Power Query to reduce model size.
- **Use Aggregations** to pre-calculate summaries instead of detailed data.
- **Create Star Schema**: Avoid too many relationships and use **fact & dimension tables**.
- **Optimize DAX Measures**: Prefer SUMX() over FILTER() when working with row-level calculations.
- **Reduce Data Granularity**: Use **pre-aggregated tables** if possible.
- **Enable Query Folding**: Push transformations back to the database.

10. Demonstrate how you would use Power Query to:

Replace missing values in a dataset.

Unpivot columns to normalize the data.

Merge data from two files with different column names.

Power Query Tasks

(a) Replacing Missing Values

Steps:

1. Open **Power Query Editor**.
2. Select the column with missing values.
3. Click **Transform** → **Replace Values** → Enter a default value (0, Unknown, etc.).
4. Click **Close & Apply** to update the dataset.

M Code Alternative:

```
Table.ReplaceValue(Source, null, "Unknown", Replacer.ReplaceValue, {"ColumnName"})
```

(b) Unpivoting Columns to Normalize Data

Before Unpivoting:

Region	Jan Sales	Feb Sales	Mar Sales
North	5000	6000	7000
South	4000	5000	6000

After Unpivoting:

Region	Month	Sales
North	Jan Sales	5000

Region	Month	Sales
North	Feb Sales	6000
South	Jan Sales	4000

Steps in Power Query:

1. Select **Jan Sales, Feb Sales, Mar Sales** columns.
2. Click **Transform** → **Unpivot Columns**.
3. Rename the columns as **Month** and **Sales**.

(c) Merging Data from Two Files with Different Column Names

Example:

- **File 1:** Contains Customer_ID, Full_Name, Amount_Spent.
- **File 2:** Contains ID, Name, Total_Spent.

Steps in Power Query:

1. Load both files into Power Query.
2. Rename columns in **File 2** to match **File 1** (ID → Customer_ID, Name → Full_Name, Total_Spent → Amount_Spent).
3. Click **Home** → **Merge Queries**.
4. Choose Customer_ID as the common key.
5. Expand the merged table and select required columns.

Behavioral & Scenario-Based Questions

---

1. Describe a time when you automated a reporting process to improve efficiency.

Automating a Reporting Process for Efficiency

Situation: At my current role at Unisys, I was responsible for generating weekly reports on incident resolution times for two clients. The manual process involved exporting data from a data warehouse, cleaning it in Excel, and then updating Power BI dashboards.

Task: The process was time-consuming and prone to errors, so I needed to automate it to improve efficiency.

Action:

- I leveraged **Power Query** to directly connect to the data warehouse and clean the data within Power BI instead of using Excel.
- I used **scheduled refresh in Power BI Service** to automate data updates.
- I built **DAX measures** to calculate key metrics dynamically instead of using static Excel formulas.

Result: The automation reduced report preparation time from **4 hours per week to 30 minutes**, eliminated manual errors, and improved data refresh speed, allowing stakeholders to access real-time insights.

2. Share an experience where you discovered a key business insight through data analysis that influenced a decision.

Discovering a Key Business Insight Through Data Analysis

Situation: While analyzing customer support data for a client, I noticed an unusual increase in **chat support resolution times** over the last quarter.

Task: My goal was to identify the cause of the issue and suggest improvements.

Action:

- I performed a **trend analysis** in Power BI and found that chat response times spiked between **6 PM - 10 PM**.
- Using **data segmentation**, I discovered that a new offshore support team was handling chats during these hours.
- I compared chat transcripts before and after the shift change and identified that **agents lacked proper training**, leading to delays.

Result: I presented the findings to the operations team, who implemented targeted training. Within a month, chat resolution times improved by **25%**, reducing customer complaints.

3. How do you prioritize tasks when handling multiple projects with tight deadlines?

Prioritizing Tasks With Multiple Projects & Tight Deadlines

Situation: As a Business Intelligence Engineer, I often work on **multiple dashboards for different clients**, each with **urgent deadlines**.

Task: I needed to manage my time effectively while ensuring all deliverables met business expectations.

Action:

- I used the **Eisenhower Matrix** to categorize tasks into **urgent-important, important-not-urgent, etc.**
- I communicated with **stakeholders** to understand which insights were most critical for decision-making.
- I leveraged **Power BI templates and reusable DAX measures** to speed up development.
- I set up **early-stage reviews** to catch potential issues before the final submission.

Result: By focusing on high-impact tasks first and automating repetitive steps, I successfully delivered **all projects on time** without compromising quality.

4. Have you faced discrepancies in data from different sources? How did you resolve it?

Handling Discrepancies in Data From Different Sources

Situation: While building a **financial dashboard**, I noticed a mismatch between **SQL-based revenue data** and **Excel reports from the finance team**.

Task: My goal was to identify the root cause and ensure accurate reporting.

Action:

- I compared data from **multiple sources** to pinpoint where discrepancies originated.
- I checked **data refresh logs** in Power BI and found that SQL data was updating **daily**, while the Excel reports were **manual uploads** that were updated **weekly**.
- I worked with the finance team to standardize the **data refresh frequency**.
- I added a **data validation step** in Power Query to flag mismatches before the report refresh.

Result: After implementing these changes, we eliminated data mismatches and improved stakeholder trust in the dashboard.

---

5. How do you collaborate with cross-functional teams to ensure data insights align with business goals?

Collaborating With Cross-Functional Teams for Business Insights

Situation: I worked on a **customer retention dashboard** for a retail client, requiring inputs from **marketing, sales, and customer service teams**.

Task: My challenge was to ensure the dashboard provided insights that aligned with all stakeholders' needs.

Action:

- I conducted **stakeholder interviews** to understand key metrics each team needed.
- I created **mockup dashboards** in Power BI and iterated based on feedback.
- I used **row-level security (RLS)** to ensure teams saw only relevant data.
- I set up **monthly review meetings** to refine the dashboard based on evolving business needs.

Result: The final dashboard provided actionable insights, helping **increase customer retention by 12%** in three months through targeted marketing campaigns.