



Data Analyst

Interview Experience (0-2 years)

SQL

Q 1. Write a query to calculate the total revenue generated by each region.

Table: Sales

```
CREATE TABLE Sales (
    Sale_ID INT PRIMARY KEY,
    Region VARCHAR(50),
    Product VARCHAR(50),
    Revenue DECIMAL(10,2)
);
```

```
INSERT INTO Sales VALUES
(1, 'East', 'Laptop', 1500.00),
(2, 'West', 'Tablet', 800.00),
(3, 'East', 'Mobile', 1200.00),
(4, 'North', 'Monitor', 700.00),
(5, 'West', 'Laptop', 1000.00),
```

```
(6, 'South', 'Headphones', 300.00),  
(7, 'North', 'Keyboard', 400.00);
```

Query:

```
SELECT  
    Region,  
    SUM(Revenue) AS Total_Revenue  
FROM Sales  
GROUP BY Region;
```

Q 2 . Display the names of employees who have a salary above the average salary in their department.

Table: Employees

```
CREATE TABLE Employees (  
    Emp_ID INT PRIMARY KEY,  
    Emp_Name VARCHAR(50),  
    Department VARCHAR(50),  
    Salary DECIMAL(10,2)  
)
```

```
INSERT INTO Employees VALUES  
(1, 'Amit', 'HR', 50000),  
(2, 'Riya', 'HR', 60000),  
(3, 'Vikram', 'Finance', 55000),  
(4, 'Sneha', 'Finance', 75000),  
(5, 'Raj', 'IT', 65000),  
(6, 'Priya', 'IT', 85000);
```

 **Query:**

```
SELECT
    Emp_Name,
    Department,
    Salary
FROM Employees e
WHERE Salary > (
    SELECT AVG(Salary)
    FROM Employees
    WHERE Department = e.Department
);
```

Q 3 . Identify the second highest salary in each department.

Table: Employees_Salary

```
CREATE TABLE Employees_Salary (
    Emp_ID INT PRIMARY KEY,
    Department_ID INT,
    Salary DECIMAL(10,2)
);
```

```
INSERT INTO Employees_Salary VALUES
(1, 101, 50000),
(2, 101, 70000),
(3, 101, 60000),
(4, 102, 45000),
(5, 102, 55000),
```

```
(6, 102, 65000),  
(7, 103, 40000);
```

Query:

```
SELECT Department_ID, Salary AS Second_Highest_Salary  
FROM (   
    SELECT  
        Department_ID,  
        Salary,  
        DENSE_RANK() OVER (PARTITION BY Department_ID ORDER BY Salary DESC) AS rnk  
    FROM Employees_Salary  
 ) ranked  
WHERE rnk = 2;
```

Q  . Write a SQL query to find employees who have not had any recent sales in the last 3 months.

Tables:

- Employees_Sales
- Sales_Transactions

```
CREATE TABLE Employees_Sales (   
    Emp_ID INT PRIMARY KEY,  
    Emp_Name VARCHAR(50)  
);
```

```
CREATE TABLE Sales_Transactions (   
    Sale_ID INT PRIMARY KEY,  
    Emp_ID INT,
```

```
Sale_Date DATE,  
Amount DECIMAL(10,2)  
);
```

```
INSERT INTO Employees_Sales VALUES  
(1, 'Amit'),  
(2, 'Riya'),  
(3, 'Vikram'),  
(4, 'Sneha');
```

```
INSERT INTO Sales_Transactions VALUES  
(101, 1, '2025-09-10', 500),  
(102, 2, '2025-10-15', 1000),  
(103, 3, '2025-04-12', 800),  
(104, 1, '2025-05-01', 700);
```

 **Query:**

```
SELECT e.Emp_Name  
FROM Employees_Sales e  
LEFT JOIN Sales_Transactions s  
ON e.Emp_ID = s.Emp_ID  
GROUP BY e.Emp_Name  
HAVING MAX(s.Sale_Date) < DATEADD(MONTH, -3, GETDATE())  
OR MAX(s.Sale_Date) IS NULL;
```

Python

Q 1 . Write a Python function to filter out customers who have made more than 5 purchases in the last 6 months.

Dataset Example:

```
import pandas as pd
from datetime import datetime, timedelta

# Sample data
data = {
    'Customer_ID': [101, 102, 101, 103, 101, 102, 101, 104, 102, 101],
    'Purchase_Date': [
        '2025-10-01', '2025-09-25', '2025-08-15', '2025-07-20', '2025-06-10',
        '2025-05-15', '2025-04-20', '2025-03-10', '2025-02-05', '2025-01-15'
    ]
}
```

```
df = pd.DataFrame(data)
df['Purchase_Date'] = pd.to_datetime(df['Purchase_Date'])
print(df)
```

✓ Function:

```
def filter_frequent_customers(df):
    today = datetime.today()
    six_months_ago = today - timedelta(days=180)

    recent_df = df[df['Purchase_Date'] >= six_months_ago]
```

```
customer_counts =  
recent_df.groupby('Customer_ID')['Purchase_Date'].count().reset_index()  
  
frequent_customers = customer_counts[customer_counts['Purchase_Date'] > 5]  
  
return frequent_customers  
  
# Call function  
  
filtered_customers = filter_frequent_customers(df)  
  
print(filtered_customers)
```

Q 2 . Create a program that reads a CSV file, extracts unique values in a column, and saves the results in a new file.

Assume we have a CSV file: customers.csv

Customer_ID,Region

101,East

102,West

103,East

104,North

105,West

 **Python Script:**

```
import pandas as pd
```

```
# Read CSV file
```

```
df = pd.read_csv('customers.csv')
```

```
# Extract unique values from 'Region' column
```

```
unique_regions = df['Region'].unique()

# Convert to DataFrame and save to new CSV
unique_df = pd.DataFrame(unique_regions, columns=['Unique_Regions'])

unique_df.to_csv('unique_regions.csv', index=False)

print("Unique regions saved to 'unique_regions.csv'")
```

 **Output file (unique_regions.csv):**

Unique_Regions

East

West

North

Q 3 . Develop a Python script to visualize monthly sales trends for a dataset using Matplotlib or Seaborn.

Sample Dataset:

```
import pandas as pd
```

```
data = {
    'Month': ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep'],
    'Sales': [1200, 1500, 1800, 1700, 1600, 2000, 2100, 1900, 2200]
}
```

```
df = pd.DataFrame(data)
```

 **Visualization Script:**

```
import matplotlib.pyplot as plt
```

```
import seaborn as sns

plt.figure(figsize=(8,5))
sns.lineplot(data=df, x='Month', y='Sales', marker='o', color='blue')

plt.title('Monthly Sales Trend', fontsize=14)
plt.xlabel('Month')
plt.ylabel('Sales Amount')
plt.grid(True)
plt.show()
```

Output:

A clean line chart showing sales increasing over months — perfect for showcasing time-based sales performance.

Power BI

Q 1 . Explain how you would create a dynamic date filter in Power BI for last month's data.

Steps:

1. Go to **Modeling** → **New Table** and create a **Date Table**:
2. `DateTable = CALENDAR(DATE(2023,1,1), TODAY())`
3. In your main data table (say Sales), ensure you have a **Date** column.
4. Create a **measure** to filter **last month's sales** dynamically:
5. `LastMonthSales =`
`CALCULATE(`
`SUM(Sales[SalesAmount]),`

```

DATESINPERIOD(
    'DateTable'[Date],
    EOMONTH(TODAY(), -1),
    1,
    MONTH
)
)

```

6. Add this measure to a **card** or **visual**, and it will always update automatically for the **previous month** (no manual filter needed).

 *Alternative:* You can also create a relative date filter directly in the visual filter pane → “Relative Date” → “is in the last 1 month”.

Q 2 . Describe the steps for setting up role-based access in Power BI using Row-Level Security (RLS).

 **Steps:**

1. **Go to Modeling** → **Manage Roles**.
2. Create a **new role** (e.g., “RegionalManager”).
3. Define a **DAX filter** for that role, for example:
4. `[Region] = "East"`
5. **Save and validate** the role.
6. In **Power BI Desktop**, test using → *Modeling* → *View as Roles* to verify access.
7. **Publish the report** to Power BI Service.
8. In **Power BI Service** → **Dataset** → **Security**, assign **users or groups** to that role.

 *Result:* Each user sees only their region’s data (based on assigned roles).

Q 3 . What is the difference between a calculated column and a measure in Power BI?

Feature	Calculated Column	Measure
Evaluation Context	Row-by-row	Aggregate (filter) context
Storage	Stored in model → increases file size	Calculated on-the-fly → memory efficient
Use Case	Needed for filtering, slicing, or grouping	Used for KPIs, totals, averages
Example	Profit = Sales[Revenue] - Sales[Cost]	Total Profit = SUM(Sales[Revenue]) - Sales[Cost])

 **Tip:** Use **measures** wherever possible — they're more efficient and dynamic.

Q  . How would you approach building a KPI dashboard that tracks multiple metrics over time?

 **Step-by-Step Approach:**

1. **Identify key KPIs** (e.g., Revenue, Profit Margin, Customer Count, Sales Growth).
2. **Connect Data Source** (SQL, Excel, Data Lake, etc.) → Load into Power BI.
3. **Create a Date Table** for time intelligence (YOY, MOM trends).
4. **Build DAX Measures** for each KPI:
 5. Total Revenue = `SUM(Sales[Revenue])`
 6. MOM Growth =
`DIVIDE([Total Revenue] - CALCULATE([Total Revenue], DATEADD('DateTable'[Date], -1, MONTH)), CALCULATE([Total Revenue], DATEADD('DateTable'[Date], -1, MONTH)))`
7. **Visualize KPIs** using:
 - o **Cards** (for headline numbers)

- **Line charts** (for trends)
 - **Bar charts** (for category comparison)
 - **KPIs visual** (for goal vs actual)
8. **Add slicers** (for date, region, product).
 9. **Apply conditional formatting** for color-based performance indicators (e.g., green for growth, red for decline).
 10. **Publish and share** securely with RLS if required.

 **Tip:** Always keep KPI visuals consistent in size and color scheme for better readability.

KPMG case study (guesstimate round)

Estimate the number of coffee cups sold in Delhi daily.

Let's break it down step by step using a structured **top-down approach** 

Step 1 — Define the Scope

We're estimating **daily coffee consumption in Delhi** (not pan India, not packaged coffee, just total cups sold daily in cafes, offices, and households).

Step 2 — Start with Population

Approximate **population of Delhi = 2 crore (20 million)** people.

Step 3 — Segment the Population

Let's divide into **working** and **non-working** populations:

- Working population → 60% → 12 million
 - Non-working population → 40% → 8 million
-

Step 4 — Estimate Coffee Drinkers

Assume **40%** of working and **10%** of non-working people drink coffee daily.

- Working coffee drinkers = 40% of 12M = 4.8M
 - Non-working coffee drinkers = 10% of 8M = 0.8M
 - **Total coffee drinkers = 5.6 million**
-

Step 5 — Average Cups per Day

- Working professionals → 2 cups/day (morning + office)
- Non-working people → 1 cup/day

So total cups = $(4.8M \times 2) + (0.8M \times 1)$

= 9.6M + 0.8M

= **10.4 million cups/day**

Step 6 — Cross-Check by Type of Consumption

Breakdown (approximate):

- **Cafes / Coffee Shops:** 10% → 1.04M cups
- **Offices (Vending Machines):** 40% → 4.16M cups
- **Households (Instant Coffee):** 50% → 5.2M cups

All add up to ~10.4M cups/day 

Step 7 — Final Answer

 **Delhi sells approximately 10–11 million cups of coffee daily.**

Bonus: How to Impress the Interviewer

If the interviewer pushes for reasoning:

- **Sensitivity check:** “If the coffee-drinking population is 30% instead of 40%, sales drop to ~8 million cups/day.”

- **Comparative validation:** “Given Delhi’s mix of cafes, IT parks, and offices, the number seems realistic compared to metros like Mumbai or Bangalore.”
 - **Add context:** “The trend is increasing due to the café culture and work-from-home setups.”
-

Pratik Jugant Mohapatra