#1..

1️⃣ Decoding Transaction Dynamics

1.Total Transactions & Value by State (Top Performing States)
```
SELECT
    State,
    SUM(Transacion_count) AS Total_Transactions,
    SUM(Transacion_amount) AS Total_Value
FROM Aggregated_Transaction_data
GROUP BY State
ORDER BY Total_Value DESC
LIMIT 10;
```

2.Yearly Growth Trend Across India
```
SELECT
    Year,
    SUM(Transacion_count) AS Total_Transactions,
    SUM(Transacion_amount) AS Total_Value
FROM Aggregated_Transaction_data
GROUP BY Year
ORDER BY Year;
```

3.Quarterly Performance by State (Seasonal Insights)
```
SELECT
    State,
    Year,
    Quater,
    SUM(Transacion_count) AS Total_Transactions,
    SUM(Transacion_amount) AS Total_Value
FROM Aggregated_Transaction_data
GROUP BY State, Year, Quater
ORDER BY Year, Quater;
```

4.Transaction Share by Transaction Type

```
SELECT
    Transacion_type,
    SUM(Transacion_count) AS Total_Transactions,
    SUM(Transacion_amount) AS Total_Value,
    ROUND(SUM(Transacion_amount) * 100 /
          (SELECT SUM(Transacion_amount) FROM Aggregated_Transaction_data), 2) AS
Percentage_Share
FROM Aggregated_Transaction_data
GROUP BY Transacion_type
ORDER BY Total_Value DESC;
```

5.Identify Low-Performing States (Bottom 10)
```
SELECT
    State,
```

```sql
    SUM(Transacion_count) AS Total_Transactions,
    SUM(Transacion_amount) AS Total_Value
FROM Aggregated_Transaction_data
GROUP BY State
ORDER BY Total_Value ASC
LIMIT 10;
```


#2.....

2️⃣ Device Dominance & User Engagement

1️⃣ Top 5 Performing Brands by Total Registered Users
```sql
SELECT
    a.Brand_type,
    SUM(m.RegisteredUsers) AS Total_Registered_Users
FROM Aggregated_User_data AS a
JOIN Map_User_data AS m
    ON a.State = m.State AND a.Year = m.Year AND a.Quater = m.Quater
GROUP BY a.Brand_type
ORDER BY Total_Registered_Users DESC
LIMIT 5;
```

2️⃣ App Engagement Efficiency (App Opens per User by Brand)
```sql
SELECT
    a.Brand_type,
    ROUND(SUM(m.AppOpens) / NULLIF(SUM(m.RegisteredUsers), 0), 2) AS
AppOpens_per_User
FROM Aggregated_User_data AS a
JOIN Map_User_data AS m
    ON a.State = m.State AND a.Year = m.Year AND a.Quater = m.Quater
GROUP BY a.Brand_type
ORDER BY AppOpens_per_User DESC
LIMIT 5;
```

3️⃣ Top 5 States with Highest App Usage

```sql
SELECT
    m.State,
    SUM(m.AppOpens) AS Total_App_Opens
FROM Aggregated_User_data AS a
JOIN Map_User_data AS m
    ON a.State = m.State AND a.Year = m.Year AND a.Quater = m.Quater
GROUP BY m.State
ORDER BY Total_App_Opens DESC
```

```sql
LIMIT 5;
```

4 Underperforming States (Lowest Engagement per User)
```sql
SELECT
    m.State,
    ROUND(SUM(m.AppOpens) / NULLIF(SUM(m.RegisteredUsers), 0), 2) AS
AppOpens_per_User
FROM Aggregated_User_data AS a
JOIN Map_User_data AS m
    ON a.State = m.State AND a.Year = m.Year AND a.Quater = m.Quater
GROUP BY m.State
ORDER BY AppOpens_per_User ASC
LIMIT 5;
```

5 Yearly Trend of App Usage Across India
```sql
SELECT
    m.Year,
    SUM(m.RegisteredUsers) AS Total_Registered,
    SUM(m.AppOpens) AS Total_App_Opens,
    ROUND(SUM(m.AppOpens) / NULLIF(SUM(m.RegisteredUsers), 0), 2) AS
Avg_AppOpens_per_User
FROM Aggregated_User_data AS a
JOIN Map_User_data AS m
    ON a.State = m.State AND a.Year = m.Year AND a.Quater = m.Quater
GROUP BY m.Year
ORDER BY m.Year;
```


#3.......
3 Insurance Penetration & Growth Potential Analysis

1 Total Insurance Transactions & Value by State (Top Performing States)
```sql
SELECT
    State,
    SUM(Insurance_count) AS Total_Transactions,
    SUM(Insurance_amount) AS Total_Value
FROM Aggregated_Insurance_data
GROUP BY State
ORDER BY Total_Value DESC
LIMIT 10;
```

2 Yearly Growth Trend of Insurance Transactions

```sql
SELECT
    Year,
    SUM(Insurance_count) AS Total_Transactions,
    SUM(Insurance_amount) AS Total_Value
FROM Aggregated_Insurance_data
GROUP BY Year
ORDER BY Year;
```

3️⃣Quarter-wise Distribution of Insurance Value
```sql
SELECT
    Year,
    Quater,
    SUM(Insurance_amount) AS Total_Value
FROM Aggregated_Insurance_data
GROUP BY Year, Quater
ORDER BY Year, Quater;
```

4️⃣Top 5 High-Value States in the Latest Year
```sql
SELECT
    State,
    SUM(Insurance_count) AS Total_Transactions,
    SUM(Insurance_amount) AS Total_Value
FROM Aggregated_Insurance_data
WHERE Year = (SELECT MAX(Year) FROM Aggregated_Insurance_data)
GROUP BY State
ORDER BY Total_Value DESC
LIMIT 5;
```

5️⃣YoY Growth Rate of Insurance Value

```sql
SELECT
    a.State,
    a.Year,
    SUM(a.Insurance_amount) AS Total_Value,
    ROUND(
        (SUM(a.Insurance_amount) - LAG(SUM(a.Insurance_amount)) OVER (PARTITION BY
a.State ORDER BY a.Year))
        / NULLIF(LAG(SUM(a.Insurance_amount)) OVER (PARTITION BY a.State ORDER BY
a.Year), 0) * 100, 2
    ) AS YoY_Growth_Percentage
FROM Aggregated_Insurance_data a
GROUP BY a.State, a.Year
ORDER BY a.State, a.Year;
```


#4......

🌏 Business Case 4 – User Engagement and Growth Strategy
1️⃣Top 10 States by Total App Engagement
SELECT
    State,
    SUM(AppOpens) AS Total_App_Opens,
    SUM(RegisteredUsers) AS Total_Users
FROM Map_User_data
GROUP BY State
ORDER BY Total_App_Opens DESC
LIMIT 10;

2️⃣Yearly Growth of User Base and App Engagement
SELECT
    Year,
    SUM(RegisteredUsers) AS Total_Users,
    SUM(AppOpens) AS Total_App_Opens
FROM Map_User_data
GROUP BY Year
ORDER BY Year;

3️⃣Quarterly Engagement Pattern
SELECT
    Year,
    Quater,
    SUM(AppOpens) AS Total_App_Opens
FROM Map_User_data
GROUP BY Year, Quater
ORDER BY Year, Quater;

4️⃣Engagement Efficiency (App Opens per User by State)
SELECT
    State,
    ROUND(SUM(AppOpens) / NULLIF(SUM(RegisteredUsers), 0), 2) AS AppOpens_per_User
FROM Map_User_data
GROUP BY State
ORDER BY AppOpens_per_User DESC
LIMIT 10;

5️⃣YoY Growth in App Engagement (By State)

SELECT
    State,

```
    Year,
    SUM(AppOpens) AS Total_App_Opens,
    ROUND(
        (SUM(AppOpens) - LAG(SUM(AppOpens)) OVER (PARTITION BY State ORDER BY
Year))
        / NULLIF(LAG(SUM(AppOpens)) OVER (PARTITION BY State ORDER BY Year), 0) *
100, 2
    ) AS YoY_Growth_Percentage
FROM Map_User_data
GROUP BY State, Year
ORDER BY State, Year;
```


#5......

        "5 Transaction Analysis Across States & Districts"

1 Top 10 States by Total Transaction Value
```
SELECT
    State,
    SUM(Transaction_count) AS Total_Transactions,
    SUM(Transaction_amount) AS Total_Value
FROM Map_Transaction_data
GROUP BY State
ORDER BY Total_Value DESC
LIMIT 10;
```

2 Yearly Transaction Growth Trend

```
SELECT
    Year,
    SUM(Transaction_count) AS Total_Transactions,
    SUM(Transaction_amount) AS Total_Value
FROM Map_Transaction_data
GROUP BY Year
ORDER BY Year;
```

◇ 3 Quarter-wise Performance by Year
```
SELECT
    Year,
    Quater,
    SUM(Transaction_count) AS Total_Transactions,
```

```
    SUM(Transaction_amount) AS Total_Value
FROM Map_Transaction_data
GROUP BY Year, Quater
ORDER BY Year, Quater;
```
◇ 4️⃣ Top 5 States in the Latest Year
```
SELECT
    State,
    SUM(Transaction_count) AS Total_Transactions,
    SUM(Transaction_amount) AS Total_Value
FROM Map_Transaction_data
WHERE Year = (SELECT MAX(Year) FROM Map_Transaction_data)
GROUP BY State
ORDER BY Total_Value DESC
LIMIT 5;
```

◇ 5️⃣ Year-over-Year (YoY) Growth in Transaction Value by State
```
SELECT
    State,
    Year,
    SUM(Transaction_amount) AS Total_Value,
    ROUND(
        (SUM(Transaction_amount) -
         LAG(SUM(Transaction_amount)) OVER (PARTITION BY State ORDER BY Year)) /
         NULLIF(LAG(SUM(Transaction_amount)) OVER (PARTITION BY State ORDER BY
Year), 0) * 100,
    2) AS YoY_Growth_Percentage
FROM Map_Transaction_data
GROUP BY State, Year
ORDER BY State, Year;
```