

# Project Title: Telecom Billing & Usage Analytics

## 1. Project Overview

The **Telecom Billing & Usage Analytics Project** simulates a real-world telecom database, enabling analysis of customer usage patterns, billing accuracy, and revenue insights.

**Purpose:** - Track customer usage (calls and data) - Generate and reconcile bills - Identify high-value customers and usage patterns - Detect anomalies like missing usage or billing discrepancies - Compute revenue metrics and rankings

---

## 2. Database Design

### 2.1 Database

- **Database Name:** telecom\_project
- **SQL Command to Create:**

```
CREATE DATABASE IF NOT EXISTS telecom_project;  
USE telecom_project;
```

### 2.2 Tables Overview

#### 2.2.1 Customers Table

Column	Data Type	Description
customer_id	INT	Primary Key, Unique customer ID
customer_name	VARCHAR(100)	Customer's full name
region	VARCHAR(50)	Customer region
activation_date	DATE	Date of account activation

**Sample Data:**

```
INSERT INTO customers (customer_id, customer_name, region, activation_date)  
VALUES  
(101, 'Amit Sharma', 'West', '2023-01-10');
```

### 2.2.2 Usage Details Table

Column	Data Type	Description
usage_id	INT	Primary Key
customer_id	INT	Foreign Key referencing customers
usage_date	DATE	Date of usage
call_minutes	INT	Minutes of call used
data_mb	INT	Data used in MB

### 2.2.3 Billing Table

Column	Data Type	Description
bill_id	INT	Primary Key
customer_id	INT	Foreign Key referencing customers
bill_month	VARCHAR(7)	Month of billing (YYYY-MM)
call_charge	DECIMAL(10,2)	Call charge
data_charge	DECIMAL(10,2)	Data charge
total_charge	DECIMAL(10,2)	Total billed amount

---

## 3. Key SQL Operations & Analytics

### 3.1 Usage Analysis

- Total call minutes & data per customer for June

```
SELECT c.customer_name, SUM(u.call_minutes) AS total_minutes,  
       SUM(u.data_mb) AS total_data  
FROM customers c  
JOIN usage_details u ON c.customer_id = u.customer_id  
WHERE MONTH(u.usage_date) = 6  
GROUP BY c.customer_name;
```

- Identify customers with no usage

```
SELECT c.customer_name  
FROM customers c  
LEFT JOIN usage_details u ON c.customer_id = u.customer_id  
WHERE u.usage_id IS NULL;
```

## 3.2 Billing & Reconciliation

### • Bill vs usage reconciliation

```
WITH usage_cost AS (  
    SELECT customer_id, DATE_FORMAT(usage_date, '%Y-%m') AS month,  
           SUM(call_minutes)*5 AS call_charge_calc,  
           SUM(data_mb)*0.3 AS data_charge_calc  
    FROM usage_details  
    GROUP BY customer_id, month  
)  
SELECT b.customer_id, b.bill_month, b.total_charge AS billed_amount,  
       (u.call_charge_calc + u.data_charge_calc) AS calculated_amount,  
       b.total_charge - (u.call_charge_calc + u.data_charge_calc) AS  
difference  
FROM billing b  
JOIN usage_cost u ON b.customer_id = u.customer_id AND b.bill_month =  
u.month;
```

### • Detect missing bills

```
SELECT c.customer_id, c.customer_name  
FROM customers c  
LEFT JOIN billing b ON c.customer_id = b.customer_id  
WHERE b.bill_id IS NULL;
```

## 3.3 Advanced Analytics

### • Usage classification

```
SELECT customer_id, data_mb,  
       CASE WHEN data_mb > 1000 THEN 'High'  
            WHEN data_mb > 500 THEN 'Medium'  
            ELSE 'Low' END AS usage_level  
FROM usage_details;
```

### • Customers above average usage

```
SELECT customer_id  
FROM usage_details  
GROUP BY customer_id  
HAVING SUM(data_mb) > (  
    SELECT AVG(total_data) FROM (  
        SELECT SUM(data_mb) AS total_data FROM usage_details GROUP BY  
customer_id
```

```
    ) AS t  
);
```

- **Month with highest revenue**

```
SELECT bill_month, SUM(total_charge) AS monthly_revenue  
FROM billing  
GROUP BY bill_month  
ORDER BY monthly_revenue DESC  
LIMIT 1;
```

---

## 4. Key Features

1. End-to-end **data capture** for customers, usage, and billing.
2. **Reconciliation** of billed vs calculated charges.
3. **Usage classification** and ranking.
4. Detect **anomalies** like missing bills.
5. Compute **monthly revenue** and running totals.

---

## 5. Potential Enhancements

- Integrate **prepaid/postpaid billing logic**.
- Add **dynamic tariffs** by region or plan.
- Include **data visualization** via Power BI or Tableau.
- Automate **alerts for high usage** or unpaid bills.

---

## 6. Conclusion

This project demonstrates practical SQL-based telecom billing analytics, enabling business insights for customer behavior, revenue analysis, and operational efficiency.