Day 4 – Intermediate SQL (Part 2)

1. Hands-on: Multi-Table Joins & Churn Dataset

Concept Explanation

Churn datasets usually contain:

- **Customer profile data** (CUSTOMERS table)
- **Subscription/billing records** (SUBSCRIPTIONS table)
- **Usage metrics** (USAGE table)
- **Cancellation records** (CHURN table)

Multi-table joins help answer:

- Which customers churned?
- What usage pattern predicts churn?
- What was their subscription value?

Common join pattern:
`CUSTOMERS → SUBSCRIPTIONS → USAGE → CHURN`

Query: Join customers with subscription details

```
SELECT c.CUSTOMER_ID, c.CUSTOMER_NAME, s.PLAN_ID, s.START_DATE
FROM CUSTOMERS c
JOIN SUBSCRIPTIONS s
    ON c.CUSTOMER_ID = s.CUSTOMER_ID;
```

Explanation: Combines customer data with subscription plan information.

Query: Add usage metrics into the join

```
SELECT c.CUSTOMER_ID, c.CUSTOMER_NAME,
       s.PLAN_ID, u.MONTHLY_USAGE
FROM CUSTOMERS c
JOIN SUBSCRIPTIONS s
    ON c.CUSTOMER_ID = s.CUSTOMER_ID
LEFT JOIN USAGE u
    ON c.CUSTOMER_ID = u.CUSTOMER_ID;
```

Explanation: Helps identify how much customers used the service.

Query: Identify churned customers with full context

```
SELECT c.CUSTOMER_ID, c.CUSTOMER_NAME,
       s.PLAN_ID, u.MONTHLY_USAGE, ch.CHURN_DATE
FROM CUSTOMERS c
LEFT JOIN SUBSCRIPTIONS s ON c.CUSTOMER_ID = s.CUSTOMER_ID
LEFT JOIN USAGE u          ON c.CUSTOMER_ID = u.CUSTOMER_ID
LEFT JOIN CHURN ch         ON c.CUSTOMER_ID = ch.CUSTOMER_ID;
```

Explanation: LEFT JOIN shows **all customers**, marking churners with CHURN_DATE.

Query: Churn rate calculation (simple)

```
SELECT
    COUNT(CASE WHEN CHURN_DATE IS NOT NULL THEN 1 END) AS CHURNED,
    COUNT(*) AS TOTAL_CUSTOMERS,
    COUNT(CASE WHEN CHURN_DATE IS NOT NULL THEN 1 END) / COUNT(*) AS
CHURN_RATE
FROM CHURN;
```

Explanation: Calculates the churn percentage.

---

2. Schema Design & Table Relationships

Concept Explanation

Schema design defines **how data tables are structured and connected**.
Key relational concepts:

- **Primary Key (PK)** → Unique identifier (e.g., CUSTOMER_ID)
- **Foreign Key (FK)** → Links tables together (e.g., CUSTOMER_ID in SUBSCRIPTIONS)
- **One-to-Many** → Customer has many orders
- **Many-to-Many** → Products in multiple orders
- **Normalization** → Remove duplication
- **Denormalization** → Combine for analytics

Snowflake does **not enforce PK/FK constraints**, but design still matters.

Query: Create a schema for an analytics project

```
CREATE OR REPLACE SCHEMA CUSTOMER_ANALYTICS;
```

Explanation: Creates a workspace for tables related to customer analytics and churn.

Query: Define tables with logical PK/FK

```
CREATE OR REPLACE TABLE CUSTOMERS (
    CUSTOMER_ID INTEGER,
    CUSTOMER_NAME STRING,
    SIGNUP_DATE DATE,
    PRIMARY KEY (CUSTOMER_ID)
);

CREATE OR REPLACE TABLE SUBSCRIPTIONS (
    SUBSCRIPTION_ID INTEGER,
    CUSTOMER_ID INTEGER,
    PLAN_ID STRING,
    START_DATE DATE,
    END_DATE DATE,
    FOREIGN KEY (CUSTOMER_ID) REFERENCES CUSTOMERS(CUSTOMER_ID)
);
```

Explanation: Snowflake won't enforce these constraints, but declaring them improves clarity, lineage, and documentation.

---

3. Basics of Data Modeling for Analytics

Concept Explanation

Analytics data models are often built using **Star Schema** or **Snowflake Schema**.

⭐ Star Schema

- **Fact table** → Contains numeric measurements (FACT_USAGE, FACT_BILLING)
- **Dimension tables** → Describe the "who/what/when" context (DIM_CUSTOMER, DIM_PLAN, DIM_DATE)

Benefits:

- Fast reporting
- Simple joins
- Optimized for BI tools (Tableau, Power BI, Looker)

Typical Churn Model:

- **DIM_CUSTOMER**
- **DIM_PLAN**
- **FACT_SUBSCRIPTIONS**
- **FACT_USAGE**
- **FACT_CHURN**

Query: Example Fact Table

```
CREATE OR REPLACE TABLE FACT_USAGE AS
SELECT
    u.USAGE_ID,
    u.CUSTOMER_ID,
    u.MONTH,
    u.MONTHLY_USAGE,
    u.CREATED_AT
FROM USAGE u;
```

Explanation: Fact table contains numeric/measurable data.

Query: Example Dimension Table

```
CREATE OR REPLACE TABLE DIM_CUSTOMER AS
SELECT
    CUSTOMER_ID,
    CUSTOMER_NAME,
    SIGNUP_DATE,
    REGION
FROM CUSTOMERS;
```

Explanation: Dimension tables contain descriptive attributes.

Query: Sample Join to Create a Wide Analytic View

```
SELECT
    d.CUSTOMER_ID,
    d.CUSTOMER_NAME,
    f.MONTHLY_USAGE,
    s.PLAN_ID,
    ch.CHURN_DATE
FROM DIM_CUSTOMER d
LEFT JOIN FACT_USAGE f ON d.CUSTOMER_ID = f.CUSTOMER_ID
LEFT JOIN SUBSCRIPTIONS s ON d.CUSTOMER_ID = s.CUSTOMER_ID
LEFT JOIN CHURN ch ON d.CUSTOMER_ID = ch.CUSTOMER_ID;
```

Explanation: Combines fact + dimension tables into a single analytics-ready dataset.

---

4. Hands-on: Build a Simple Data Model in Snowflake

Concept Explanation

Building a data model involves:

1. **Identifying entities** → Customers, Plans, Usage, Churn
2. **Separating facts & dimensions**
3. **Creating tables & views**
4. **Designing join paths for analytics**

Query: Build an analytic view for churn prediction

```
CREATE OR REPLACE VIEW CUSTOMER_CHURN_ANALYTICS AS
SELECT
    c.CUSTOMER_ID,
    c.CUSTOMER_NAME,
    s.PLAN_ID,
    u.MONTHLY_USAGE,
    ch.CHURN_DATE,
    CASE WHEN ch.CHURN_DATE IS NOT NULL THEN 1 ELSE 0 END AS IS_CHURNED
FROM CUSTOMERS c
LEFT JOIN SUBSCRIPTIONS s ON c.CUSTOMER_ID = s.CUSTOMER_ID
LEFT JOIN USAGE u ON c.CUSTOMER_ID = u.CUSTOMER_ID
LEFT JOIN CHURN ch ON c.CUSTOMER_ID = ch.CUSTOMER_ID;
```

Explanation: Creates a **single flattened view** for machine learning, reporting, or BI.

Query: Validate the model

```
SELECT PLAN_ID,
       COUNT(*) AS TOTAL_CUSTOMERS,
       SUM(IS_CHURNED) AS CHURNED,
       SUM(IS_CHURNED) / COUNT(*) AS CHURN_RATE
FROM CUSTOMER_CHURN_ANALYTICS
GROUP BY PLAN_ID;
```

Explanation: Quick check of churn rate per subscription plan.