

-- === 1. Create Sequences for Surrogate Keys ===

-- Create sequences to generate surrogate keys

```
CREATE SEQUENCE seq_time_id START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE seq_product_id START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE seq_branch_id START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE seq_location_id START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE seq_sub_category_id START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE seq_sale_id START WITH 1 INCREMENT BY 1;
```

-- === 2. Create Star Schema Tables (Dimensions) ===

-- Time Dimension Table with surrogate key

```
CREATE TABLE dim_time (
    time_id INT PRIMARY KEY,
    year INT,
    quarter INT,
    month INT,
    day INT
);
```

-- Product Dimension Table with surrogate key

```
CREATE TABLE dim_product (
    product_id INT PRIMARY KEY,
    product_name VARCHAR2(50),
    category VARCHAR2(50),
    sub_category_id INT
);
```

-- Branch Dimension Table with surrogate key

```
CREATE TABLE dim_branch (
    branch_id INT PRIMARY KEY,
    branch_name VARCHAR2(50),
    branch_location VARCHAR2(50)
);
```

-- Location Dimension Table with surrogate key

```
CREATE TABLE dim_location (
    location_id INT PRIMARY KEY,
    location_name VARCHAR2(50),
```

```
    city VARCHAR2(50),  
    region VARCHAR2(50)  
);
```

-- Sub Category Dimension Table for Snowflake Schema

```
CREATE TABLE dim_sub_category (  
    sub_category_id INT PRIMARY KEY,  
    sub_category_name VARCHAR2(50)  
);
```

-- === 3. Create Fact Table ===

```
CREATE TABLE fact_sales (  
    sale_id INT PRIMARY KEY,  
    time_id INT,  
    product_id INT,  
    branch_id INT,  
    location_id INT,  
    sub_category_id INT,  
    sold INT,  
    units_sold INT,  
    FOREIGN KEY (time_id) REFERENCES dim_time(time_id),  
    FOREIGN KEY (product_id) REFERENCES dim_product(product_id),  
    FOREIGN KEY (branch_id) REFERENCES dim_branch(branch_id),  
    FOREIGN KEY (location_id) REFERENCES dim_location(location_id),  
    FOREIGN KEY (sub_category_id) REFERENCES  
dim_sub_category(sub_category_id)  
);
```

-- === 4. Insert Data into Star Schema Using Surrogate Keys ===

-- Insert into dim_time using sequence for time_id

```
INSERT INTO dim_time (time_id, year, quarter, month, day)  
VALUES (seq_time_id.NEXTVAL, 2025, 1, 1, 1);
```

```
INSERT INTO dim_time (time_id, year, quarter, month, day)  
VALUES (seq_time_id.NEXTVAL, 2025, 1, 2, 5);
```

```
INSERT INTO dim_time (time_id, year, quarter, month, day)  
VALUES (seq_time_id.NEXTVAL, 2025, 2, 3, 10);
```

```
INSERT INTO dim_time (time_id, year, quarter, month, day)
VALUES (seq_time_id.NEXTVAL, 2025, 3, 4, 15);
```

```
INSERT INTO dim_time (time_id, year, quarter, month, day)
VALUES (seq_time_id.NEXTVAL, 2025, 4, 5, 20);
```

```
-- Insert into dim_product using sequence for product_id
INSERT INTO dim_product (product_id, product_name, category,
sub_category_id)
VALUES (seq_product_id.NEXTVAL, 'Laptop', 'Electronics', 1);
```

```
INSERT INTO dim_product (product_id, product_name, category,
sub_category_id)
VALUES (seq_product_id.NEXTVAL, 'Smartphone', 'Electronics', 2);
```

```
INSERT INTO dim_product (product_id, product_name, category,
sub_category_id)
VALUES (seq_product_id.NEXTVAL, 'Tablet', 'Electronics', 1);
```

```
INSERT INTO dim_product (product_id, product_name, category,
sub_category_id)
VALUES (seq_product_id.NEXTVAL, 'Headphones', 'Electronics', 3);
```

```
INSERT INTO dim_product (product_id, product_name, category,
sub_category_id)
VALUES (seq_product_id.NEXTVAL, 'Smartwatch', 'Electronics', 4);
```

```
-- Insert into dim_branch using sequence for branch_id
INSERT INTO dim_branch (branch_id, branch_name, branch_location)
VALUES (seq_branch_id.NEXTVAL, 'Branch A', 'City X');
```

```
INSERT INTO dim_branch (branch_id, branch_name, branch_location)
VALUES (seq_branch_id.NEXTVAL, 'Branch B', 'City Y');
```

```
INSERT INTO dim_branch (branch_id, branch_name, branch_location)
VALUES (seq_branch_id.NEXTVAL, 'Branch C', 'City Z');
```

```
INSERT INTO dim_branch (branch_id, branch_name, branch_location)
VALUES (seq_branch_id.NEXTVAL, 'Branch D', 'City X');
```

```
INSERT INTO dim_branch (branch_id, branch_name, branch_location)
VALUES (seq_branch_id.NEXTVAL, 'Branch E', 'City Y');
```

```
-- Insert into dim_location using sequence for location_id
INSERT INTO dim_location (location_id, location_name, city, region)
VALUES (seq_location_id.NEXTVAL, 'Location A', 'City X', 'Region 1');
```

```
INSERT INTO dim_location (location_id, location_name, city, region)
VALUES (seq_location_id.NEXTVAL, 'Location B', 'City Y', 'Region 2');
```

```
INSERT INTO dim_location (location_id, location_name, city, region)
VALUES (seq_location_id.NEXTVAL, 'Location C', 'City Z', 'Region 3');
```

```
INSERT INTO dim_location (location_id, location_name, city, region)
VALUES (seq_location_id.NEXTVAL, 'Location D', 'City X', 'Region 1');
```

```
INSERT INTO dim_location (location_id, location_name, city, region)
VALUES (seq_location_id.NEXTVAL, 'Location E', 'City Y', 'Region 2');
```

```
-- Insert into dim_sub_category using sequence for sub_category_id
INSERT INTO dim_sub_category (sub_category_id, sub_category_name)
VALUES (seq_sub_category_id.NEXTVAL, 'Computers');
```

```
INSERT INTO dim_sub_category (sub_category_id, sub_category_name)
VALUES (seq_sub_category_id.NEXTVAL, 'Mobile');
```

```
INSERT INTO dim_sub_category (sub_category_id, sub_category_name)
VALUES (seq_sub_category_id.NEXTVAL, 'Audio');
```

```
INSERT INTO dim_sub_category (sub_category_id, sub_category_name)
VALUES (seq_sub_category_id.NEXTVAL, 'Wearables');
```

```
-- Insert into fact_sales using sequence for sale_id
INSERT INTO fact_sales (sale_id, time_id, product_id, branch_id, location_id,
sub_category_id, sold, units_sold)
VALUES (seq_sale_id.NEXTVAL, 1, 1, 1, 1, 1, 1000, 50);
```

```
INSERT INTO fact_sales (sale_id, time_id, product_id, branch_id, location_id,
sub_category_id, sold, units_sold)
VALUES (seq_sale_id.NEXTVAL, 2, 2, 2, 2, 2, 1500, 30);
```

```
INSERT INTO fact_sales (sale_id, time_id, product_id, branch_id, location_id,
sub_category_id, sold, units_sold)
VALUES (seq_sale_id.NEXTVAL, 3, 3, 3, 3, 3, 1200, 20);
```

```
INSERT INTO fact_sales (sale_id, time_id, product_id, branch_id, location_id,
sub_category_id, sold, units_sold)
VALUES (seq_sale_id.NEXTVAL, 4, 4, 4, 4, 4, 800, 60);
```

```
INSERT INTO fact_sales (sale_id, time_id, product_id, branch_id, location_id,
sub_category_id, sold, units_sold)
VALUES (seq_sale_id.NEXTVAL, 5, 5, 5, 5, 4, 1300, 40);
```

```
-- === SLICE OPERATION ===
```

```
BEGIN
```

```
  DBMS_OUTPUT.PUT_LINE('=== SLICE OPERATION ===');
```

```
  DBMS_OUTPUT.PUT_LINE('Description: Slice by Product ID (Laptop)');
```

```
  DBMS_OUTPUT.PUT_LINE('Extracting data for Product ID = 1 (Laptop)');
```

```
END;
```

```
/
```

```
SELECT f.sale_id, p.product_name, b.branch_name, l.location_name, t.year,
t.month, t.day, f.sold, f.units_sold
```

```
FROM fact_sales f
```

```
JOIN dim_product p ON f.product_id = p.product_id
```

```
JOIN dim_branch b ON f.branch_id = b.branch_id
```

```
JOIN dim_location l ON f.location_id = l.location_id
```

```
JOIN dim_time t ON f.time_id = t.time_id
```

```
WHERE p.product_id = 1;
```

```
-- === DICE OPERATION ===
```

```
BEGIN
```

```
  DBMS_OUTPUT.PUT_LINE('=== DICE OPERATION ===');
```

```
  DBMS_OUTPUT.PUT_LINE('Description: Filter data for Sold > 1000 and
Location = Location A');
```

```
END;
```

```
/
```

```
SELECT f.sale_id, p.product_name, b.branch_name, l.location_name, t.year,
t.month, t.day, f.sold, f.units_sold
```

```
FROM fact_sales f
```

```
JOIN dim_product p ON f.product_id = p.product_id
```

```
JOIN dim_branch b ON f.branch_id = b.branch_id
JOIN dim_location l ON f.location_id = l.location_id
JOIN dim_time t ON f.time_id = t.time_id
WHERE f.sold > 1000 AND l.location_name = 'Location A';
```

```
-- === DRILL-DOWN OPERATION ===
```

```
BEGIN
```

```
    DBMS_OUTPUT.PUT_LINE('=== DRILL-DOWN OPERATION ===');
```

```
    DBMS_OUTPUT.PUT_LINE('Description: Drill down from Year → Month for  
Sales in 2025');
```

```
    DBMS_OUTPUT.PUT_LINE('Drilling down from Year to Month for Sales in  
2025');
```

```
END;
```

```
/
```

```
SELECT t.year, t.month, SUM(f.sold) AS total_sold
FROM fact_sales f
JOIN dim_time t ON f.time_id = t.time_id
WHERE t.year = 2025
GROUP BY t.year, t.month
ORDER BY t.year, t.month;
```

```
-- === ROLL-UP OPERATION ===
```

```
BEGIN
```

```
    DBMS_OUTPUT.PUT_LINE('=== ROLL-UP OPERATION ===');
```

```
    DBMS_OUTPUT.PUT_LINE('Description: Roll-up from Day → Month for Sales');
```

```
    DBMS_OUTPUT.PUT_LINE('Rolling up from Day to Month');
```

```
END;
```

```
/
```

```
SELECT t.year, t.month, SUM(f.sold) AS total_sold
FROM fact_sales f
JOIN dim_time t ON f.time_id = t.time_id
GROUP BY t.year, t.month
ORDER BY t.year, t.month;
```

```
-- === PIVOT OPERATION ===
```

```
BEGIN
```

```
    DBMS_OUTPUT.PUT_LINE('=== PIVOT OPERATION ===');
```

```
    DBMS_OUTPUT.PUT_LINE('Description: Pivot by Month to Show Total Sold for  
Each Product');
```

```
END;
```

```
/
SELECT *
FROM (
  SELECT p.product_name, t.month, f.sold
  FROM fact_sales f
  JOIN dim_product p ON f.product_id = p.product_id
  JOIN dim_time t ON f.time_id = t.time_id
)
PIVOT (
  SUM(sold)
  FOR month IN (1 AS "January", 2 AS "February", 3 AS "March", 4 AS "April", 5
AS "May")
);
```