

-- === 1. Create Star Schema Tables (Dimensions) ===

-- Time Dimension Table

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

-- Product Dimension Table

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

-- Branch Dimension Table

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

-- Location Dimension Table

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

-- === 2. Create Fact Table ===

```
CREATE TABLE fact_sales (  
    sale_id INT PRIMARY KEY,  
    time_id INT,  
    product_id INT,  
    branch_id INT,  
    location_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)  
);
```

-- === 3. Insert Data into Star Schema ===

-- Insert into dim_time

```
INSERT INTO dim_time VALUES (1, 2025, 1, 1, 1);  
INSERT INTO dim_time VALUES (2, 2025, 1, 2, 5);  
INSERT INTO dim_time VALUES (3, 2025, 2, 3, 10);  
INSERT INTO dim_time VALUES (4, 2025, 3, 4, 15);  
INSERT INTO dim_time VALUES (5, 2025, 4, 5, 20);
```

-- Insert into dim_product

```
INSERT INTO dim_product VALUES (1, 'Laptop', 'Electronics');  
INSERT INTO dim_product VALUES (2, 'Smartphone', 'Electronics');  
INSERT INTO dim_product VALUES (3, 'Tablet', 'Electronics');  
INSERT INTO dim_product VALUES (4, 'Headphones', 'Electronics');  
INSERT INTO dim_product VALUES (5, 'Smartwatch', 'Electronics');
```

-- Insert into dim_branch

```
INSERT INTO dim_branch VALUES (1, 'Branch A', 'City X');  
INSERT INTO dim_branch VALUES (2, 'Branch B', 'City Y');  
INSERT INTO dim_branch VALUES (3, 'Branch C', 'City Z');  
INSERT INTO dim_branch VALUES (4, 'Branch D', 'City X');  
INSERT INTO dim_branch VALUES (5, 'Branch E', 'City Y');
```

-- Insert into dim_location

```
INSERT INTO dim_location VALUES (1, 'Location A', 'City X', 'Region 1');  
INSERT INTO dim_location VALUES (2, 'Location B', 'City Y', 'Region 2');  
INSERT INTO dim_location VALUES (3, 'Location C', 'City Z', 'Region 3');  
INSERT INTO dim_location VALUES (4, 'Location D', 'City X', 'Region 1');  
INSERT INTO dim_location VALUES (5, 'Location E', 'City Y', 'Region 2');
```

-- Insert into fact_sales

```
INSERT INTO fact_sales VALUES (1, 1, 1, 1, 1, 1000, 50);  
INSERT INTO fact_sales VALUES (2, 2, 2, 2, 2, 1500, 30);  
INSERT INTO fact_sales VALUES (3, 3, 3, 3, 3, 1200, 20);
```

```
INSERT INTO fact_sales VALUES (4, 4, 4, 4, 4, 800, 60);
INSERT INTO fact_sales VALUES (5, 5, 5, 5, 5, 1300, 40);
```

```
-- === 4. Create Snowflake Schema Tables ===
```

```
-- Snowflake for Product (add sub_category dimension)
```

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

```
-- Modify Product table to include sub_category_id
```

```
ALTER TABLE dim_product ADD (sub_category_id INT);
```

```
UPDATE dim_product SET sub_category_id = 1 WHERE product_id IN (1, 3);
```

```
UPDATE dim_product SET sub_category_id = 2 WHERE product_id IN (2, 5);
```

```
UPDATE dim_product SET sub_category_id = 3 WHERE product_id = 4;
```

```
-- Alter Fact Table to include references to Snowflake Schema
```

```
ALTER TABLE fact_sales ADD (sub_category_id INT);
```

```
UPDATE fact_sales SET sub_category_id = (SELECT sub_category_id FROM
dim_product WHERE dim_product.product_id = fact_sales.product_id);
```

```
-- === 5. Insert Data into Snowflake Schema ===
```

```
-- Insert data into dim_sub_category
```

```
INSERT INTO dim_sub_category VALUES (1, 'Computers');
```

```
INSERT INTO dim_sub_category VALUES (2, 'Mobile');
```

```
INSERT INTO dim_sub_category VALUES (3, 'Audio');
```

```
INSERT INTO dim_sub_category VALUES (4, 'Wearables');
```

```
-- === 6. Perform OLAP Operations with Print Statements ===
```

```
-- === 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")
);

```