```
-- ==================================================
-- STEP 1: CREATE STAR SCHEMA TABLES WITH PRIMARY KEYS
-- ==================================================

CREATE TABLE dim_doctor (
   doctor_id NUMBER PRIMARY KEY,
   doctor_name VARCHAR2(50),
   department VARCHAR2(50),
   specialization VARCHAR2(50)
);

CREATE TABLE dim_patient (
   patient_id NUMBER PRIMARY KEY,
   patient_name VARCHAR2(50),
   gender CHAR(1),
   city VARCHAR2(50)
);

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

CREATE TABLE fact_visits (
   visit_id NUMBER PRIMARY KEY,
   star_doctor_id NUMBER,
   star_patient_id NUMBER,
   snowflake_doctor_id NUMBER,
   snowflake_patient_id NUMBER,
   time_id NUMBER,
   charges NUMBER,
   FOREIGN KEY (star_doctor_id) REFERENCES dim_doctor(doctor_id),
   FOREIGN KEY (star_patient_id) REFERENCES dim_patient(patient_id),
   FOREIGN KEY (time_id) REFERENCES dim_time(time_id)
);

-- ========================================================
-- STEP 2: CREATE SNOWFLAKE DIMENSIONS (DEPT + LOCATION) FIRST
--        So they can be referenced by patient & doctor tables
-- ==========================================================

CREATE TABLE dim_department_snowflake (
   dept_id NUMBER PRIMARY KEY,
   dept_name VARCHAR2(50)
);

CREATE TABLE dim_location_snowflake (
   location_id NUMBER PRIMARY KEY,
   city VARCHAR2(50),
   state VARCHAR2(50)
);

-- ====================================================
-- STEP 3: CREATE SNOWFLAKE DIMENSION TABLES WITH FKs
-- ====================================================

CREATE TABLE dim_doctor_snowflake (
   doctor_id NUMBER PRIMARY KEY,
   doctor_name VARCHAR2(50),
   specialization VARCHAR2(50),
   dept_id NUMBER,
   FOREIGN KEY (dept_id) REFERENCES dim_department_snowflake(dept_id)
);

CREATE TABLE dim_patient_snowflake (
   patient_id NUMBER PRIMARY KEY,
   patient_name VARCHAR2(50),
   gender CHAR(1),
   location_id NUMBER,
   FOREIGN KEY (location_id) REFERENCES dim_location_snowflake(location_id)
);

-- Now alter fact_visits to connect with snowflake keys
ALTER TABLE fact_visits ADD (
   FOREIGN KEY (snowflake_doctor_id) REFERENCES
dim_doctor_snowflake(doctor_id),
   FOREIGN KEY (snowflake_patient_id) REFERENCES
dim_patient_snowflake(patient_id)
);

-- ==============================================
-- SAMPLE DATA INSERTION (Same as before)
-- ==============================================

-- Star doctor
INSERT INTO dim_doctor VALUES (101, 'Dr. Smith', 'Heart', 'Cardiology');
INSERT INTO dim_doctor VALUES (102, 'Dr. Jane', 'Neuro', 'Neurology');

-- Star patient
INSERT INTO dim_patient VALUES (201, 'Alice', 'F', 'New York');
INSERT INTO dim_patient VALUES (202, 'Bob', 'M', 'Los Angeles');

-- Time
INSERT INTO dim_time VALUES (1, 12, 3, 1, 2023);
INSERT INTO dim_time VALUES (2, 15, 3, 1, 2023);
INSERT INTO dim_time VALUES (3, 10, 4, 2, 2023);

-- Snowflake department
INSERT INTO dim_department_snowflake VALUES (1, 'Cardiology');
INSERT INTO dim_department_snowflake VALUES (2, 'Neurology');

-- Snowflake location
INSERT INTO dim_location_snowflake VALUES (1, 'New York', 'NY');
INSERT INTO dim_location_snowflake VALUES (2, 'Los Angeles', 'CA');

-- Snowflake doctor
INSERT INTO dim_doctor_snowflake VALUES (104, 'Dr. A', 'Cardiology', 1);
INSERT INTO dim_doctor_snowflake VALUES (105, 'Dr. B', 'Neurology', 2);

-- Snowflake patient
INSERT INTO dim_patient_snowflake VALUES (301, 'Alice', 'F', 1);
INSERT INTO dim_patient_snowflake VALUES (302, 'Bob', 'M', 2);

-- Fact table
INSERT INTO fact_visits VALUES (1, 101, 201, 104, 301, 1, 5000);
INSERT INTO fact_visits VALUES (2, 102, 202, 105, 302, 2, 7000);
INSERT INTO fact_visits VALUES (3, 101, 201, 104, 301, 3, 6000);

-- ==================================================
-- OLAP OPERATIONS WITH EXPLANATIONS
-- ==================================================

-- Operation 1: SLICE (Vertical filtering)
BEGIN
   DBMS_OUTPUT.PUT_LINE('=== SLICE OPERATION ===');
   DBMS_OUTPUT.PUT_LINE('Description: Extracts data for a single doctor (Dr.
Smith)');
END;
/
-- Star
SELECT f.visit_id, d.doctor_name, p.patient_name, f.charges
FROM fact_visits f
JOIN dim_doctor d ON f.star_doctor_id = d.doctor_id
JOIN dim_patient p ON f.star_patient_id = p.patient_id
WHERE d.doctor_id = 101;

-- Snowflake
SELECT f.visit_id, ds.doctor_name, ps.patient_name, f.charges
FROM fact_visits f
JOIN dim_doctor_snowflake ds ON f.snowflake_doctor_id = ds.doctor_id
JOIN dim_patient_snowflake ps ON f.snowflake_patient_id = ps.patient_id
WHERE ds.doctor_id = 104;

-- Operation 2: DICE (Multidimensional filtering)
BEGIN
   DBMS_OUTPUT.PUT_LINE('=== DICE OPERATION ===');
   DBMS_OUTPUT.PUT_LINE('Description: Filters data for Cardiology
department in New York');
END;
/
-- Star
SELECT f.visit_id, d.doctor_name, p.patient_name, f.charges
FROM fact_visits f
JOIN dim_doctor d ON f.star_doctor_id = d.doctor_id
JOIN dim_patient p ON f.star_patient_id = p.patient_id
WHERE d.department = 'Heart' AND p.city = 'New York';
```

```
-- Snowflake
SELECT f.visit_id, ds.doctor_name, ps.patient_name, ls.city, f.charges
FROM fact_visits f
JOIN dim_doctor_snowflake ds ON f.snowflake_doctor_id = ds.doctor_id
JOIN dim_patient_snowflake ps ON f.snowflake_patient_id = ps.patient_id
JOIN dim_location_snowflake ls ON ps.location_id = ls.location_id
JOIN dim_department_snowflake dept ON ds.dept_id = dept.dept_id
WHERE dept.dept_name = 'Cardiology' AND ls.city = 'New York';

-- Operation 3: DRILL-DOWN (Increasing detail)
BEGIN
   DBMS_OUTPUT.PUT_LINE('=== DRILL-DOWN OPERATION ===');
   DBMS_OUTPUT.PUT_LINE('Description: Analyzes revenue from year →
month → day');
END;
/
-- Year to month
SELECT t.year, t.month, SUM(f.charges) AS total_charges
FROM fact_visits f
JOIN dim_time t ON f.time_id = t.time_id
GROUP BY t.year, t.month
ORDER BY t.year, t.month;

-- Month to day (Q1)
SELECT t.month, t.day, SUM(f.charges) AS daily_charges
FROM fact_visits f
JOIN dim_time t ON f.time_id = t.time_id
WHERE t.quarter = 1
GROUP BY t.month, t.day
ORDER BY t.month, t.day;

-- Operation 4: ROLL-UP (Decreasing detail)
BEGIN
   DBMS_OUTPUT.PUT_LINE('=== ROLL-UP OPERATION ===');
   DBMS_OUTPUT.PUT_LINE('Description: Aggregates data from day → month
→ year');
END;
/
-- Yearly totals
```

```
SELECT t.year, SUM(f.charges) AS yearly_revenue
FROM fact_visits f
JOIN dim_time t ON f.time_id = t.time_id
GROUP BY t.year
ORDER BY t.year;

-- Roll-up to department
SELECT d.department, SUM(f.charges) AS department_revenue
FROM fact_visits f
JOIN dim_doctor d ON f.star_doctor_id = d.doctor_id
GROUP BY d.department
ORDER BY department_revenue DESC;

-- Operation 5: PIVOT (Cross-tabulation)
BEGIN
   DBMS_OUTPUT.PUT_LINE('=== PIVOT OPERATION ===');
   DBMS_OUTPUT.PUT_LINE('Description: Creates cross-tab reports of visit
data');
END;
/
-- Pivot by department and gender
SELECT *
FROM (
   SELECT d.department, p.gender, f.charges
   FROM fact_visits f
   JOIN dim_doctor d ON f.star_doctor_id = d.doctor_id
   JOIN dim_patient p ON f.star_patient_id = p.patient_id
)
PIVOT (
   SUM(charges) FOR gender IN ('M' AS Male, 'F' AS Female)
)
ORDER BY department;

-- Pivot by city and specialization
SELECT *
FROM (
   SELECT ls.city, ds.specialization, f.charges
   FROM fact_visits f
   JOIN dim_doctor_snowflake ds ON f.snowflake_doctor_id = ds.doctor_id
   JOIN dim_patient_snowflake ps ON f.snowflake_patient_id = ps.patient_id
```

```
    JOIN dim_location_snowflake ls ON ps.location_id = ls.location_id
)
PIVOT (
    SUM(charges) FOR specialization IN ('Cardiology' AS Cardio, 'Neurology' AS Neuro)
)
ORDER BY city;
```