```
-- ============================================
-- STAR SCHEMA DIMENSION TABLES
-- ============================================

CREATE TABLE dim_location (
   location_id NUMBER PRIMARY KEY,
   location_type VARCHAR2(20),
   region VARCHAR2(50),
   latitude NUMBER,
   longitude NUMBER
);

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

CREATE TABLE dim_probe (
   probe_id NUMBER PRIMARY KEY,
   probe_name VARCHAR2(50),
   model VARCHAR2(50),
   install_year NUMBER
);

-- ============================================
-- FACT TABLE (Hourly Weather Data)
-- ============================================

CREATE TABLE fact_weather_data (
   weather_id NUMBER PRIMARY KEY,
   probe_id NUMBER,
   location_id NUMBER,
   time_id NUMBER,
   temperature NUMBER,
   air_pressure NUMBER,
   precipitation NUMBER,
```

```
    FOREIGN KEY (probe_id) REFERENCES dim_probe(probe_id),
    FOREIGN KEY (location_id) REFERENCES dim_location(location_id),
    FOREIGN KEY (time_id) REFERENCES dim_time(time_id)
);

-- ============================================
-- SAMPLE DATA INSERTION (Star Schema)
-- ============================================

INSERT INTO dim_location VALUES (1, 'Land', 'Northern Plains', 40.7, -97.3);
INSERT INTO dim_location VALUES (2, 'Ocean', 'Pacific Zone', -12.4, 145.6);
INSERT INTO dim_location VALUES (3, 'Land', 'Southern Ridge', 15.6, 75.3);
INSERT INTO dim_location VALUES (4, 'Ocean', 'Atlantic Zone', 22.3, -45.7);
INSERT INTO dim_location VALUES (5, 'Land', 'Eastern Range', 33.2, 78.1);

INSERT INTO dim_time VALUES (101, TO_DATE('2025-01-01','YYYY-MM-DD'),
10, 1, 1, 2025);
INSERT INTO dim_time VALUES (102, TO_DATE('2025-01-01','YYYY-MM-DD'),
15, 1, 1, 2025);
INSERT INTO dim_time VALUES (103, TO_DATE('2025-02-01','YYYY-MM-DD'),
11, 2, 1, 2025);
INSERT INTO dim_time VALUES (104, TO_DATE('2025-03-01','YYYY-MM-DD'),
13, 3, 1, 2025);
INSERT INTO dim_time VALUES (105, TO_DATE('2025-03-02','YYYY-MM-DD'),
16, 3, 1, 2025);

INSERT INTO dim_probe VALUES (201, 'WX-100', 'ThermoPro', 2015);
INSERT INTO dim_probe VALUES (202, 'WX-200', 'HydroSense', 2017);
INSERT INTO dim_probe VALUES (203, 'WX-300', 'AeroMax', 2018);
INSERT INTO dim_probe VALUES (204, 'WX-400', 'ClimateX', 2019);
INSERT INTO dim_probe VALUES (205, 'WX-500', 'StormTrack', 2020);

INSERT INTO fact_weather_data VALUES (1, 201, 1, 101, 22.5, 1012, 0.0);
INSERT INTO fact_weather_data VALUES (2, 202, 2, 102, 27.0, 1008, 5.2);
INSERT INTO fact_weather_data VALUES (3, 203, 3, 103, 30.1, 1005, 1.0);
INSERT INTO fact_weather_data VALUES (4, 204, 4, 104, 18.7, 1015, 3.3);
INSERT INTO fact_weather_data VALUES (5, 205, 5, 105, 25.3, 1009, 0.5);

-- ============================================
-- SNOWFLAKE SCHEMA EXTENSION
```

```
-- ============================================

CREATE TABLE dim_region_snowflake (
   region_id NUMBER PRIMARY KEY,
   region_name VARCHAR2(50)
);

ALTER TABLE dim_location ADD (region_id NUMBER);
ALTER TABLE dim_location ADD CONSTRAINT fk_region_snowflake FOREIGN
KEY (region_id) REFERENCES dim_region_snowflake(region_id);

INSERT INTO dim_region_snowflake VALUES (10, 'Northern Plains');
INSERT INTO dim_region_snowflake VALUES (20, 'Pacific Zone');
INSERT INTO dim_region_snowflake VALUES (30, 'Southern Ridge');
INSERT INTO dim_region_snowflake VALUES (40, 'Atlantic Zone');
INSERT INTO dim_region_snowflake VALUES (50, 'Eastern Range');

UPDATE dim_location SET region_id = 10 WHERE location_id = 1;
UPDATE dim_location SET region_id = 20 WHERE location_id = 2;
UPDATE dim_location SET region_id = 30 WHERE location_id = 3;
UPDATE dim_location SET region_id = 40 WHERE location_id = 4;
UPDATE dim_location SET region_id = 50 WHERE location_id = 5;

CREATE TABLE dim_model_snowflake (
   model_id NUMBER PRIMARY KEY,
   model_name VARCHAR2(50)
);

ALTER TABLE dim_probe ADD (model_id NUMBER);
ALTER TABLE dim_probe ADD CONSTRAINT fk_model_snowflake FOREIGN KEY
(model_id) REFERENCES dim_model_snowflake(model_id);

INSERT INTO dim_model_snowflake VALUES (1001, 'ThermoPro');
INSERT INTO dim_model_snowflake VALUES (1002, 'HydroSense');
INSERT INTO dim_model_snowflake VALUES (1003, 'AeroMax');
INSERT INTO dim_model_snowflake VALUES (1004, 'ClimateX');
INSERT INTO dim_model_snowflake VALUES (1005, 'StormTrack');

UPDATE dim_probe SET model_id = 1001 WHERE probe_id = 201;
UPDATE dim_probe SET model_id = 1002 WHERE probe_id = 202;
```

```
UPDATE dim_probe SET model_id = 1003 WHERE probe_id = 203;
UPDATE dim_probe SET model_id = 1004 WHERE probe_id = 204;
UPDATE dim_probe SET model_id = 1005 WHERE probe_id = 205;

-- OLAP operations will follow this structure using both star and snowflake
tables.
-- Full OLAP operations block to be added next.
-- ============================================
-- 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 specific probe
(WX-100)');
END;
/
-- Example 1: Star schema slice
SELECT f.weather_id, f.temperature, f.air_pressure, f.precipitation
FROM fact_weather_data f
JOIN dim_probe p ON f.probe_id = p.probe_id
WHERE p.probe_name = 'WX-100';

-- Example 2: Snowflake schema slice
SELECT f.weather_id, f.temperature, f.air_pressure, f.precipitation
FROM fact_weather_data f
JOIN dim_probe p ON f.probe_id = p.probe_id
JOIN dim_model_snowflake m ON p.model_id = m.model_id
WHERE m.model_name = 'ThermoPro';

-- Operation 2: DICE (Multidimensional Filtering)
BEGIN
   DBMS_OUTPUT.PUT_LINE('=== DICE OPERATION ===');
   DBMS_OUTPUT.PUT_LINE('Description: Filters data across multiple
dimensions, e.g., Pacific Zone in February');
END;
/
-- Example 1: Star schema dice
SELECT f.weather_id, f.temperature, f.precipitation, l.region, t.month
```

```
FROM fact_weather_data f
JOIN dim_location l ON f.location_id = l.location_id
JOIN dim_time t ON f.time_id = t.time_id
WHERE l.region = 'Pacific Zone' AND t.month = 2;

-- Example 2: Snowflake schema dice
SELECT f.weather_id, f.temperature, f.precipitation, rs.region_name, t.month
FROM fact_weather_data f
JOIN dim_location l ON f.location_id = l.location_id
JOIN dim_region_snowflake rs ON l.region_id = rs.region_id
JOIN dim_time t ON f.time_id = t.time_id
WHERE rs.region_name = 'Pacific Zone' AND t.month = 2;

-- Operation 3: DRILL-DOWN (Increasing Detail)
BEGIN
   DBMS_OUTPUT.PUT_LINE('=== DRILL-DOWN OPERATION ===');
   DBMS_OUTPUT.PUT_LINE('Description: Analyzes weather data from year →
month → day.');
END;
/
-- Example 1: Yearly to monthly breakdown
SELECT t.year, t.month, AVG(f.temperature) AS avg_temperature
FROM fact_weather_data f
JOIN dim_time t ON f.time_id = t.time_id
GROUP BY t.year, t.month
ORDER BY t.year, t.month;

-- Example 2: Monthly to daily breakdown for March
SELECT t.month, t.day, AVG(f.temperature) AS daily_temperature
FROM fact_weather_data f
JOIN dim_time t ON f.time_id = t.time_id
WHERE t.month = 3
GROUP BY t.month, t.day
ORDER BY 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;
/
-- Example 1: Roll-up to yearly temperature averages
SELECT t.year, AVG(f.temperature) AS avg_temperature
FROM fact_weather_data f
JOIN dim_time t ON f.time_id = t.time_id
GROUP BY t.year
ORDER BY t.year;

-- Example 2: Roll-up to region temperature averages
SELECT rs.region_name, AVG(f.temperature) AS avg_temperature
FROM fact_weather_data f
JOIN dim_location l ON f.location_id = l.location_id
JOIN dim_region_snowflake rs ON l.region_id = rs.region_id
GROUP BY rs.region_name
ORDER BY avg_temperature DESC;

-- Operation 5: PIVOT (Cross-tabulation)
BEGIN
   DBMS_OUTPUT.PUT_LINE('=== PIVOT OPERATION ===');
   DBMS_OUTPUT.PUT_LINE('Description: Creates cross-tab reports of weather
data by region and model.');
END;
/
-- Example 1: Pivot by region and probe model
SELECT *
FROM (
   SELECT
      rs.region_name,
      m.model_name,
      f.temperature
   FROM fact_weather_data f
   JOIN dim_location l ON f.location_id = l.location_id
   JOIN dim_region_snowflake rs ON l.region_id = rs.region_id
   JOIN dim_probe p ON f.probe_id = p.probe_id
   JOIN dim_model_snowflake m ON p.model_id = m.model_id
)
PIVOT (
   AVG(temperature)
```

```
    FOR model_name IN ('ThermoPro' AS 'ThermoPro', 'HydroSense' AS
'HydroSense', 'AeroMax' AS 'AeroMax', 'ClimateX' AS 'ClimateX', 'StormTrack'
AS 'StormTrack')
)
ORDER BY region_name;
```