```sql
-- =======================
-- 1. CREATE SEQUENCES FOR SURROGATE KEYS
-- =======================

CREATE SEQUENCE seq_course_section START WITH 1;
CREATE SEQUENCE seq_professor START WITH 1;
CREATE SEQUENCE seq_student START WITH 1;
CREATE SEQUENCE seq_period START WITH 1;
CREATE SEQUENCE seq_department START WITH 1;
CREATE SEQUENCE seq_room START WITH 1;
CREATE SEQUENCE seq_fact_id START WITH 1;


-- =======================
-- 2. STAR SCHEMA DIMENSION TABLES
-- =======================

CREATE TABLE dim_course_section (
    course_section_sk INT PRIMARY KEY,
    course_id INT,
    section_id INT,
    course_name VARCHAR2(100),
    units INT,
    room_id VARCHAR2(20),
    room_capacity INT
);

CREATE TABLE dim_professor (
    professor_sk INT PRIMARY KEY,
    professor_id INT,
    professor_name VARCHAR2(100),
    title VARCHAR2(50),
    department_id INT,
    department_name VARCHAR2(100)
);

CREATE TABLE dim_student (
    student_sk INT PRIMARY KEY,
    student_id INT,
    student_major VARCHAR2(100)
);
```

```sql
CREATE TABLE dim_period (
    period_sk INT PRIMARY KEY,
    semester_id INT,
    year INT
);


-- ======================
-- 3. SNOWFLAKE DIMENSIONS
-- ======================

CREATE TABLE dim_department_snowflake (
    department_sk INT PRIMARY KEY,
    department_id INT,
    department_name VARCHAR2(100)
);

CREATE TABLE dim_room_snowflake (
    room_sk INT PRIMARY KEY,
    room_id VARCHAR2(20),
    room_capacity INT
);


-- ======================
-- 4. FACT TABLE
-- ======================

CREATE TABLE fact_course_grades (
    fact_id INT PRIMARY KEY,
    course_section_sk INT,
    professor_sk INT,
    student_sk INT,
    period_sk INT,
    department_sk INT,
    room_sk INT,
    grade VARCHAR2(2),
    FOREIGN KEY (course_section_sk) REFERENCES
dim_course_section(course_section_sk),
    FOREIGN KEY (professor_sk) REFERENCES dim_professor(professor_sk),
    FOREIGN KEY (student_sk) REFERENCES dim_student(student_sk),
```

```sql
    FOREIGN KEY (period_sk) REFERENCES dim_period(period_sk),
    FOREIGN KEY (department_sk) REFERENCES
dim_department_snowflake(department_sk),
    FOREIGN KEY (room_sk) REFERENCES dim_room_snowflake(room_sk)
);


-- =======================
-- 5. INSERT DATA INTO DIMENSIONS
-- =======================

-- Course Section
INSERT INTO dim_course_section VALUES (seq_course_section.NEXTVAL, 101,
1, 'DBMS', 4, 'R101', 60);
INSERT INTO dim_course_section VALUES (seq_course_section.NEXTVAL, 102,
1, 'OS', 3, 'R102', 55);
INSERT INTO dim_course_section VALUES (seq_course_section.NEXTVAL, 103,
1, 'CN', 4, 'R103', 50);
INSERT INTO dim_course_section VALUES (seq_course_section.NEXTVAL, 104,
1, 'AI', 3, 'R104', 65);
INSERT INTO dim_course_section VALUES (seq_course_section.NEXTVAL, 105,
1, 'ML', 3, 'R105', 60);

-- Professor
INSERT INTO dim_professor VALUES (seq_professor.NEXTVAL, 201, 'Dr. Mehta',
'Associate Prof', 301, 'Computer Science');
INSERT INTO dim_professor VALUES (seq_professor.NEXTVAL, 202, 'Dr. Rao',
'Assistant Prof', 302, 'IT');
INSERT INTO dim_professor VALUES (seq_professor.NEXTVAL, 203, 'Dr. Nair',
'Professor', 303, 'Electronics');
INSERT INTO dim_professor VALUES (seq_professor.NEXTVAL, 204, 'Dr. Kapoor',
'Professor', 301, 'Computer Science');
INSERT INTO dim_professor VALUES (seq_professor.NEXTVAL, 205, 'Dr. Sharma',
'Lecturer', 304, 'AI & DS');

-- Student
INSERT INTO dim_student VALUES (seq_student.NEXTVAL, 1001, 'Computer
Science');
INSERT INTO dim_student VALUES (seq_student.NEXTVAL, 1002, 'IT');
INSERT INTO dim_student VALUES (seq_student.NEXTVAL, 1003, 'Electronics');
INSERT INTO dim_student VALUES (seq_student.NEXTVAL, 1004, 'AI & DS');
```

```sql
INSERT INTO dim_student VALUES (seq_student.NEXTVAL, 1005, 'Computer
Science');

-- Period
INSERT INTO dim_period VALUES (seq_period.NEXTVAL, 1, 2023);
INSERT INTO dim_period VALUES (seq_period.NEXTVAL, 2, 2023);
INSERT INTO dim_period VALUES (seq_period.NEXTVAL, 3, 2024);
INSERT INTO dim_period VALUES (seq_period.NEXTVAL, 4, 2024);
INSERT INTO dim_period VALUES (seq_period.NEXTVAL, 5, 2025);

-- Department Snowflake
INSERT INTO dim_department_snowflake VALUES (seq_department.NEXTVAL,
301, 'Computer Science');
INSERT INTO dim_department_snowflake VALUES (seq_department.NEXTVAL,
302, 'IT');
INSERT INTO dim_department_snowflake VALUES (seq_department.NEXTVAL,
303, 'Electronics');
INSERT INTO dim_department_snowflake VALUES (seq_department.NEXTVAL,
304, 'AI & DS');
INSERT INTO dim_department_snowflake VALUES (seq_department.NEXTVAL,
305, 'Data Science');

-- Room Snowflake
INSERT INTO dim_room_snowflake VALUES (seq_room.NEXTVAL, 'R101', 60);
INSERT INTO dim_room_snowflake VALUES (seq_room.NEXTVAL, 'R102', 55);
INSERT INTO dim_room_snowflake VALUES (seq_room.NEXTVAL, 'R103', 50);
INSERT INTO dim_room_snowflake VALUES (seq_room.NEXTVAL, 'R104', 65);
INSERT INTO dim_room_snowflake VALUES (seq_room.NEXTVAL, 'R105', 60);

-- =========================
-- 6. INSERT INTO FACT TABLE (Sample Data using existing SKs)
-- =========================

-- Assuming you fetch SKs using SELECTs or know their order:
INSERT INTO fact_course_grades VALUES (seq_fact_id.NEXTVAL, 1, 1, 1, 1, 1, 1,
'A');
INSERT INTO fact_course_grades VALUES (seq_fact_id.NEXTVAL, 2, 2, 2, 2, 2, 2,
'B');
INSERT INTO fact_course_grades VALUES (seq_fact_id.NEXTVAL, 3, 3, 3, 3, 3, 3,
'C');
```

```sql
INSERT INTO fact_course_grades VALUES (seq_fact_id.NEXTVAL, 4, 4, 4, 4, 4, 4,
'A');
INSERT INTO fact_course_grades VALUES (seq_fact_id.NEXTVAL, 5, 5, 5, 5, 5, 5,
'B');


-- =======================
-- 7. OLAP OPERATIONS
-- =======================

-- SLICE: View grades from year 2023
SELECT f.*, p.year
FROM fact_course_grades f
JOIN dim_period p ON f.period_sk = p.period_sk
WHERE p.year = 2023;

-- DICE: View records where year is 2023 and grade is 'A'
SELECT f.*, p.year
FROM fact_course_grades f
JOIN dim_period p ON f.period_sk = p.period_sk
WHERE p.year = 2023 AND f.grade = 'A';

-- DRILL-DOWN: From year to semester-level view
SELECT p.year, p.semester_id, COUNT(*) AS grade_count
FROM fact_course_grades f
JOIN dim_period p ON f.period_sk = p.period_sk
GROUP BY p.year, p.semester_id
ORDER BY p.year, p.semester_id;

-- ROLL-UP: Total grades per year
SELECT p.year, COUNT(*) AS total_grades
FROM fact_course_grades f
JOIN dim_period p ON f.period_sk = p.period_sk
GROUP BY p.year
ORDER BY p.year;

-- PIVOT: Grades distribution per year (manually pivoted)
SELECT
  p.year,
  SUM(CASE WHEN f.grade = 'A' THEN 1 ELSE 0 END) AS A_Grades,
  SUM(CASE WHEN f.grade = 'B' THEN 1 ELSE 0 END) AS B_Grades,
```

```
  SUM(CASE WHEN f.grade = 'C' THEN 1 ELSE 0 END) AS C_Grades
FROM fact_course_grades f
JOIN dim_period p ON f.period_sk = p.period_sk
GROUP BY p.year
ORDER BY p.year;
```