**Task 1: Implement the Schema**

**SQL Statements to Create Tables**

CREATE TABLE Students (

    Snum INTEGER PRIMARY KEY,

    Ssn INTEGER UNIQUE NOT NULL,

    Name TEXT NOT NULL,

    CurrentAddress TEXT,

    PermanentAddress TEXT,

    CurrentPhone TEXT,

    PermanentPhone TEXT,

    DegreeProgram1 TEXT,

    DegreeProgram2 TEXT,

    MajorDept TEXT,

    MinorDept TEXT

);

CREATE TABLE Departments (

    Dcode TEXT PRIMARY KEY,

    Dname TEXT UNIQUE NOT NULL,

    Dcollege TEXT,

    Doffice TEXT,

    Dphone TEXT

);

CREATE TABLE Courses (

    Cnum INTEGER PRIMARY KEY,

    Cname TEXT UNIQUE NOT NULL,

    Cdesc TEXT,

    Credit INTEGER,

    Level TEXT,

    Cdept TEXT,

    FOREIGN KEY (Cdept) REFERENCES Departments(Dcode)

);

CREATE TABLE Sections (

    Sec\_course INTEGER,

    Sec\_num INTEGER,

    Semester TEXT,

    Year INTEGER,

    PRIMARY KEY (Sec\_course, Sec\_num, Semester, Year),

    FOREIGN KEY (Sec\_course) REFERENCES Courses(Cnum)

);

CREATE TABLE Grades (

    Snum INTEGER,

    Sec\_course INTEGER,

    Sec\_num INTEGER,

    Semester TEXT,

    Year INTEGER,

    Grade TEXT,

    PRIMARY KEY (Snum, Sec\_course, Sec\_num, Semester, Year),

    FOREIGN KEY (Snum) REFERENCES Students(Snum),

    FOREIGN KEY (Sec\_course, Sec\_num, Semester, Year) REFERENCES Sections(Sec\_course, Sec\_num, Semester, Year)

);

CREATE TABLE StudentPrograms (

    Snum INTEGER,

    Prog TEXT,

    MajorDept TEXT,

    MinorDept TEXT,

    PRIMARY KEY (Snum, Prog, MajorDept, MinorDept),

    FOREIGN KEY (Snum) REFERENCES Students(Snum)

);

**Populate Tables with Sample Data**

-- Students

INSERT INTO Students (Snum, Ssn, Name, CurrentAddress, PermanentAddress, CurrentPhone, PermanentPhone, DegreeProgram1, DegreeProgram2, MajorDept, MinorDept)

VALUES (1, 111111111, 'John Doe', '123 Elm St', '456 Oak St', '555-1234', '555-5678', 'BSc CS', 'MSc CS', 'CS', 'Math');

INSERT INTO Students (Snum, Ssn, Name, CurrentAddress, PermanentAddress, CurrentPhone, PermanentPhone, DegreeProgram1, DegreeProgram2, MajorDept, MinorDept)

VALUES (2, 222222222, 'Jane Smith', '789 Pine St', '101 Maple St', '555-8765', '555-4321', 'BSc EE', 'MSc EE', 'EE', 'Physics');

INSERT INTO Students (Snum, Ssn, Name, CurrentAddress, PermanentAddress, CurrentPhone, PermanentPhone, DegreeProgram1, DegreeProgram2, MajorDept, MinorDept)

VALUES (3, 333333333, 'Alice Johnson', '234 Cedar St', '567 Birch St', '555-3456', '555-6543', 'BSc ME', 'MSc ME', 'ME', 'Chemistry');

-- Departments

INSERT INTO Departments (Dcode, Dname, Dcollege, Doffice, Dphone)

VALUES ('CS', 'Computer Science', 'Engineering', 'Room 101', '555-1111');

INSERT INTO Departments (Dcode, Dname, Dcollege, Doffice, Dphone)

VALUES ('EE', 'Electrical Engineering', 'Engineering', 'Room 102', '555-2222');

INSERT INTO Departments (Dcode, Dname, Dcollege, Doffice, Dphone)

VALUES ('ME', 'Mechanical Engineering', 'Engineering', 'Room 103', '555-3333');

-- Courses

INSERT INTO Courses (Cnum, Cname, Cdesc, Credit, Level, Cdept)

VALUES (101, 'Intro to CS', 'Introduction to Computer Science', 3, 'Undergraduate', 'CS');

INSERT INTO Courses (Cnum, Cname, Cdesc, Credit, Level, Cdept)

VALUES (102, 'Data Structures', 'Data Structures and Algorithms', 4, 'Undergraduate', 'CS');

INSERT INTO Courses (Cnum, Cname, Cdesc, Credit, Level, Cdept)

VALUES (201, 'Digital Circuits', 'Digital Logic and Circuits', 3, 'Undergraduate', 'EE');

-- Sections

INSERT INTO Sections (Sec\_course, Sec\_num, Semester, Year)

VALUES (101, 1, 'Fall', 2023);

INSERT INTO Sections (Sec\_course, Sec\_num, Semester, Year)

VALUES (102, 1, 'Spring', 2024);

INSERT INTO Sections (Sec\_course, Sec\_num, Semester, Year)

VALUES (201, 1, 'Fall', 2023);

-- Grades

INSERT INTO Grades (Snum, Sec\_course, Sec\_num, Semester, Year, Grade)

VALUES (1, 101, 1, 'Fall', 2023, 'A');

INSERT INTO Grades (Snum, Sec\_course, Sec\_num, Semester, Year, Grade)

VALUES (2, 102, 1, 'Spring', 2024, 'B');

INSERT INTO Grades (Snum, Sec\_course, Sec\_num, Semester, Year, Grade)

VALUES (3, 201, 1, 'Fall', 2023, 'A');

-- StudentPrograms

INSERT INTO StudentPrograms (Snum, Prog, MajorDept, MinorDept)

VALUES (1, 'BSc', 'CS', 'Math');

INSERT INTO StudentPrograms (Snum, Prog, MajorDept, MinorDept)

VALUES (2, 'BSc', 'EE', 'Physics');

INSERT INTO StudentPrograms (Snum, Prog, MajorDept, MinorDept)

VALUES (3, 'BSc', 'ME', 'Chemistry');

**Task 2: Data Redundancy and Anomalies**

**Insertion Anomaly**

-- Inserting a student with the same Ssn but different Snum, demonstrating redundancy

INSERT INTO Students (Snum, Ssn, Name, CurrentAddress, PermanentAddress, CurrentPhone, PermanentPhone, DegreeProgram1, DegreeProgram2, MajorDept, MinorDept)

VALUES (4, 111111111, 'John Doe Duplicate', '999 Willow St', '111 Spruce St', '555-9999', '555-8888', 'BSc CS', 'MSc CS', 'CS', 'Math');

**Explanation**: Inserting a new student with the same Ssn but a different Snum leads to redundancy. This anomaly is undesirable because it creates inconsistencies in the database, with multiple records for the same student but different student numbers.

**Modification Anomaly**

-- Changing the department name for a course, which requires multiple updates

UPDATE Departments SET Dname = 'Comp Sci' WHERE Dcode = 'CS';

**Explanation**: Modifying the department name requires updating multiple records that reference this department. If any record is missed, it leads to inconsistency. This anomaly is undesirable because it makes the database harder to maintain and increases the risk of errors.

**Deletion Anomaly**

-- Deleting a department that still has courses associated with it

DELETE FROM Departments WHERE Dcode = 'CS';

**Explanation**: Deleting a department removes all associated information, including courses, which leads to loss of valuable data. This anomaly is undesirable because it results in unintended data loss and disrupts the database's integrity.

**Task 3: Normalizing Relations to the Highest Normal Form**

**Students Relation**

**1NF**: Each field contains only atomic values.

**2NF**: No partial dependencies (already in 2NF since the primary key is a single field).

**3NF**: No transitive dependencies (already in 3NF as all attributes are directly dependent on the primary key).

**StudentPrograms Relation**

**1NF**: Each field contains only atomic values.

**2NF**: No partial dependencies (already in 2NF since the primary key is a composite key).

**3NF**: No transitive dependencies (already in 3NF as all attributes are directly dependent on the primary key).

**Departments Relation**

**1NF**: Each field contains only atomic values.

**2NF**: No partial dependencies (already in 2NF since the primary key is a composite key).

**3NF**: No transitive dependencies (already in 3NF as all attributes are directly dependent on the primary key).

**Courses Relation**

**1NF**: Each field contains only atomic values.

**2NF**: No partial dependencies (already in 2NF since the primary key is a composite key).

**3NF**: No transitive dependencies (already in 3NF as all attributes are directly dependent on the primary key).

**Sections Relation**

**1NF**: Each field contains only atomic values.

**2NF**: No partial dependencies (already in 2NF since the primary key is a composite key).

**3NF**: No transitive dependencies (already in 3NF as all attributes are directly dependent on the primary key).

**Grades Relation**

**1NF**: Each field contains only atomic values.

**2NF**: No partial dependencies (already in 2NF since the primary key is a composite key).

**3NF**: No transitive dependencies (already in 3NF as all attributes are directly dependent on the primary key).

**Question 2**

There are no violations of the 1NF to 4NF in the given schema. Therefore, there's no need to modify or create additional tables to demonstrate these violations.

**5NF or DKNF**

If you identify a table that exhibits a violation of 5NF or DKNF and normalize it, you may be awarded extra credit. This typically involves more complex decomposition based on join dependencies that are beyond the scope of the initial requirements.