Task 1.1: Superkey and Candidate Key Analysis

Relation A: Employee Employee(EmpID, SSN, Email, Phone, Name, Department, Salary)

1. Superkeys

- (EmpID)
- (SSN)
- (Email)
- (SSN, Email)
- (EmpID, SSN)
- (EmpID, Email)

2 Candidate keys

- (EmpID)
- (SSN)
- (Email)

3 Choosing Primary key

- I will choose Empl ID as a primary key. Since it is short and convenient, when working with other databases, one short number will be much easier to work with them than an email or insurance account.

4 2 Phones

- Yes, the phone is not unique as it could be a regular office phone, so it is not a key.

Relation B: Course Registration Registration(StudentID, CourseCode, Section, Semester, Year, Grade, Credits)

1. Candidate keys

- (StudentID, CourseCode, Section, Semester, Year)

2 Explaining

- StudentID identifies the student
- CourseCode identifies the course
- Section identifies the section of the course
- Semester and Year specifies the time the course was taken

3 Primary key

- There are no additional candidate keys for the entire table.

Task 1.2: Foreign Key Design

1. Foreign key relationships

- **Student.AdvisorID** is a foreign key referencing **Professor.ProfID**.
- **Course.DepartmentCode** is a foreign key referencing **Department.DeptCode**.
- **Department.ChairID** is a foreign key referencing **Professor.ProfID**.
- **Enrollment.StudentID** is a foreign key referencing **Student.StudentID**.
- **Enrollment.CourseID** is a foreign key referencing **Course.CourseID**.

Task 4.1: Denormalized Table Analysis

1. Dependencies

- StudentID → StudentName, StudentMajor
- ProjectID → ProjectTitle, ProjectType
- SupervisorID → SupervisorName, SupervisorDept

2. Applying 1NF

- All attributes are simple (no multiple values or nested attributes).

3. Applying 2NF

- Primary key: StudentID, ProjectID

- Partial dependencies:
- StudentID → StudentName, StudentMajor
- ProjectID → ProjectTitle, ProjectType
- SupervisorID → SupervisorName, SupervisorDept

- Decomposition**:**
- Student(StudentID, StudentName, StudentMajor)
- Project(ProjectID, ProjectTitle, ProjectType, SupervisorID)
- Supervisor(SupervisorID, SupervisorName, SupervisorDept)
- StudentProject(StudentID, ProjectID, Role, HoursWorked, StartDate, EndDate)

Task 4.2: Advanced Normalization

1. Primary key
- StudentID
- CourseID

2. Functional dependencies
- StudentID → StudentMajor
- CourseID → CourseName
- InstructorID → InstructorName
- Room → Building

3. BCNF checking
- The left set must be a key
- StudentID → StudentMajor (StudentID isn’t key).
- CourseID → CourseName (CourseID isn’t key).
- InstructorID → InstructorName (InstructorID isn’t key).
- Room → Building (Room isn’t key).
- ANSWER: It is not BNCF

4. Decomposition to BCNF
- Student(StudentID, StudentMajor)
- Course(CourseID, CourseName)
- Instructor(InstructorID, InstructorName)
- Room(Room, Building)
- Section(CourseID, TimeSlot, InstructorID, Room)
- Enrollment(StudentID, CourseID, TimeSlot)

5. Loss of information
- The information is now broken down into parts, so it can't be retrieved from a single table - you'll have to use a JOIN. This isn't data loss, it's just a change in how it's stored.

Task 5.1: Real-World Application

Examples**:**

- List all events along with the names of students who attended them.
- Find all clubs that have more than 10 members.
- **Find the names of all students who never attended any club event.**