Laboratory 1. Databases

Task 1. Relation A.

I. Superkeys:
   1. EmpID
   2. SSN
   3. Email
   4. EmpID, Name
   5. Email, Phone
   6. EmpID, SSN, Department

II. Candidate keys:
   1. EmpID
   2. SSN
   3. Email

III. I would choose the candidate key EmpID as the primary key, because SSN and email address may change in the future while EmpID is stable. In addition, EmpID is shorter and easier to use compared to SSN and email.

IV. Yes, two empoyees can have the same phone number, because it may be shared office number of their department.

Relation B.

1. StudentID, CourseCode, Section, Semester, Year
2. StudentID is important because student can take several courses.
   CourseID is important because several students may choose the same course.
   Section is important because some courses can have several sections.
   Semester is important because student can take the same course in two semesters.
   Year is important because student can take the same course next year.
3. No, because to be unique row these only 5 attributes are necessary. Grade is not important because it depends on student's performance in a particular course. Credits are also not important because they are fixed for a given course and don't provide additional uniqueness.

Task 1.2.

Student(AdvisorID) -> Professor(ProfID)

Student(Major) -> Department(DeptCode),

Professor(Department) -> Department(DeptCode

Course(DepartmentCode) -> Department(DeptCode)

Department(ChairID) -> Professor(ProfID)

Enrollment(StudentID) -> Student(StudentID)

Enrollment(CourseID) -> Course(CourseID)

4.

Task 1

1. Functional Dependencies:
   StudentID -> StudentName, StudentMajor
   ProjectID -> ProjectTitle, ProjectType, SupervisorID
   SupervisorID -> SupervisorName, SupervisorDept
   (StudentID, ProjectID) -> Role, HoursWorked, StartDate, EndDate
2. Redundancy
   If student works on multiple projects there will be repetitions of student information.(StudentName, StudentMajor).
   For students on the same projects there will be repetitions on project information(ProjectTitle, ProjectType, SupervisorID)
   Repetition of supervisor details(SupervisorName, SupervisorDept)

Anomalies

Update anomaly: If SupervisorName changes, we must update it in all rows for all projects under that supervisor.

Insert anomaly: Cannot insert a new student without also assigning them to a project.

Delete anomaly: If the last student on a project is deleted the project and supervisor details are lost.

3. 1NF

No multi-valued attributes are shown (Role, HoursWorked are single values). However, if one student can have multiple roles per project, then this violates 1NF. To fix we can make several rows for student.

4. 2NF

The primary key (StudentID, ProjectID)

Partial Dependencies:

- StudentID -> StudentName, StudentMajor
- ProjectID -> ProjectTitle, ProjectType, SupervisorID

2NF Decomposition:

1. Student(StudentID, StudentName, StudentMajor)
2. Project(ProjectID, ProjectTitle, ProjectType, SupervisorID)
3. Supervisor(SupervisorID, SupervisorName, SupervisorDept)

4. StudentProject(StudentID, ProjectID, Role, HoursWorked, StartDate, EndDate)

5. 3NF

Transitive Dependencies:

In Project, SupervisorID -> SupervisorName, SupervisorDept

Fix by decomposition:

We already separated Supervisor so this transitive dependency is solved.

3NF Decomposition

1. Student(StudentID, StudentName, StudentMajor)
2. Project(ProjectID, ProjectTitle, ProjectType, SupervisorID)
3. Supervisor(SupervisorID, SupervisorName, SupervisorDept)
4. StudentProject(StudentID, ProjectID, Role, HoursWorked, StartDate, EndDate)

Task 4

1. Primary Key

One course is taught by one instructor, in one room, at one time, so these are important attributes

Primary key = (StudentID, CourseID, TimeSlot, Room)

2. Functional Dependencies:

1. StudentID -> StudentMajor
   (each student has exactly one major)
2. CourseID -> CourseName
   (each course has a name)
3. InstructorID -> InstructorName
4. Room -> Building
5. (CourseID, TimeSlot, Room) ->InstructorID
   (course section is defined by these and taught by one instructor)
6. (CourseID, TimeSlot, Room) -> InstructorName
7. (CourseID, TimeSlot, Room) -> Building
8. (StudentID, CourseID, TimeSlot, Room) -> all attributes

3. BCNF

- StudentID -> StudentMajor, StudentID is not a superkey.
- CourseID -> CourseName, CourseID is not a superkey.
- InstructorID -> InstructorName, InstructorID is not a superkey.
- Room -> Building, Room is not a superkey.

- (CourseID, TimeSlot, Room) -> InstructorID, candidate key

the table is not in BCNF.

Decomposition BCNF

Students(StudentID, StudentMajor)

1. Courses(CourseID, CourseName)
2. Instructors(InstructorID, InstructorName)
3. Rooms(Room, Building)
4. Sections(CourseID, TimeSlot, Room, InstructorID)
5. Enrollment(StudentID, CourseID, TimeSlot, Room)

No information is lost, but the information is now splitted to several smaller tables.