**Lab 1**

**Task 1: 1.1**

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

1.List all elements:

1. Employee(EmpID)
2. Employee(SSN)
3. Employee(EmpId, SSN)
4. Employee(EmpID, Email)
5. Employee(EmpID, Phone)
6. Employee(SSN, Name, Department)
7. Employee(SSN, Name, Salary)
8. Etc...

2.Candidate Keys:

1. EmpID, SSN, Phone

3.Primary Key: EmpID

* Explanation: Because it best represents the tuples. Tuples should be unique and should have unique ID. EmpId is unchangeable and will represent a tuple even if there will be changes in attributes. Phone is also unique, but changeable, user will someday change them. SSN is unique, but it’s hard to access when we want to make changes in our code. So EmpID is both unchangeable and easy to access.

4.Yes, and it can’t be candidate key. But if every phone number is unique , then it can be candidate key

Relation B:

Registration(StudentID, CourseCode, Section, Semester, Year, Grade, Credits)

1.Candidate Keys:

* PK(StudentID, CourseCode, Section).
* StudentID can be duplicated because one student can chouse multiple courses. CourseCode and Section could be chosen by many student.

**Task 1: 1.2**

Sudent(StudentID, Name, Email, Major, AdvisorID) - AdvisorID is FK from Professor(ProfID)

Professor(ProfID, Name, Department, Salary) - Department is FK from Department(DeptCode)

Course(CourseID, Title, Credits, DepartmentCode) – DepartmentCode is FK from Department(DeptCode)

Department(DeptCode, DeptName, Budget, ChairID) – ChairID is FK from Professor(ProfID)

Enrollment(StudentID, CourseID, Semester, Grade) – StudentID is FK from Student(StudentID) and CourseID is FK from Course(CourseID)

**Task 4: 4.1**

StudentProject(StudentID, StudentName, StudentMajor, ProjectID, ProjectTitle, ProjectType, SupervisorID, SupervisorName, SupervisorDept, Role, HoursWorked, StartDate, EndDate)

1.Functional dependencies:

* (StudentID) -> StudentName, StudentMajor
* (ProjectID) -> ProjectTitle, ProjectType, SupervisorID
* (SupervisorID) -> SupervisorName, SupervisorDept, SupervisorID
* (StudentID, ProjecctID) -> Role, HoursWorked. StartDate, EndDate

2.Redundancy:

* StudentID will repeat due to the projects. For Example :

Update: will be hard because needed to find specific tuple among big database or to change project of every student, you have to change all tuples one by one

Insert: It will be hard because you can’t add a tuple if, for example, there are no stents yet

Delete: It will be hard because it will delete the tuple entirely with the information in it

* ProjectID will be repeated because of multiple students working with project

3. 1NF:

* StudentName could contain two cells like StudentName(“name, surename”).

Fixed: StudentName(“name”, “surename”)

* SupervisorName could also contain two information SupervisorName(“name, surename”). Fixed: Supervisor(“name”, ”surename”)

4. 2NF:

* PK – (StudentID, ProjectID)
* (StudentID) -> StudentName, StudentMajor

(ProjectID) -> ProjectTitle, ProjectType, SupervisorID

* 2NF decomposition:

Student(StudentID, StudentName, StudentSurename, StudentMajor)

Project(ProjectID, ProjectTitle, ProjectType, SupervisorID, SupervisorName, SupervisorSurename, SupervisorDept)

Overall(StudentID, ProjectID, Role, HoursWorked, StartDate, EndDate)

5. 3NF:

* Transitive dependencies:

Project(ProjectID, ProjectTitle, ProjectType, SupervisorID, SupervisorName, SupervisorSurename, SupervisorDept)

ProjectID, ProjectTitle, ProjectType -> SupervisorID

SupervisorID -> SupervisorID, SupervisorName, SupervisorSurename, SupervisorDept

* Student(StudentID, StudentName, StudentSurename, StudentMajor)

Project(ProjectID, ProjectTitle, ProjectType, SupervisorID,)

Supervisor(SupervisorID, SupervisorName, SupervisorSurename, SupervisorDept)

Student\_project(StudentID, ProjectID, Role, HoursWorked, StartDate, EndDate)

**Task 4: 4.2**

CourseSchedule(StudentID, StudentMajor, CourseID, CourseName. InstructorID, InstructorName, TimeSlot, Room, Building)

1. PK(StudentID, CourseID)
2. StudentID -> StudentMajor

CourseID -> CourseName, InstructorID, Room, TimeSlot

InstructorID -> InstructorName

Room -> Building

1. Student(StudentID, StudentMajor)

Course(CourseID, CourseName, InstructorID, Room, TimeSlot)

Instructor(InstructorID, InstructorName)

RoomTL(Room, Building)

Enrollment(StudentID, CourseID)

**Task 5**

Students(StudentID, StudentName)

Clubs(ClubID, ClubName, Information, budget, AdvisorID)

Events(EventID, ClubID, room, Attendance)

Advisor(AdvisorID, AdvisorName)

EventAtt(EventID, CountOfMembers)

Postions(postionID, positionName)

ClubMembership(StudentID, ClubID, positionID)

Selection of Members name by their ID

Selection of Budget of the club

To see who is the advisor of the club