Laboratory Work 1.

Part 1.

Task 1.1: Superkey and Candidate Key Analysis

Relation A:

1) Superkeys:

1. {EmpID},
2. {SSN},
3. {Email},
4. {Phone},
5. {EmpID,SSN},
6. {SSN, Email}
7. {EmpID, Name}

2)Candidate Keys:

1. {EmpID},
2. {SSN},
3.{Email}
4.{Phone}

3) I will choose EmpID as Primary key. Because it is not a surrogate key and does not rely on private information like in SSN, Email, Phone. It is integer number, so that is easy index and use it as foreign key.

4)Each phone number is unique as it is shown on the table. Although there is no rule that two employees cannot use one phone number. So that it is possible to have the same phone number.

Task 1.1: Relation B – Course registration

1) The minimum attributes needed for primary key: {StudentID, CourseCode, Section, Semester, Year}
2) StudentID: determine the student uniquely
   CourseCode: determine the course
   Section: find many sections from one course
   Semester and Year: are needed to perform rule “A student can take the same course in different semesters”
3) If attributes Grade or Credits would not be unique like others, then they are only candidate keys.

Task 1.2: Foreign Key Design

- Student.AdvisorID → Professor.ProfID
- Department.ChairID → Professor.ProfID
- Enrollment.StudentID → Student.StudentID
- Enrollment.CourseID → Course.CourseID

- Course.DepartmentCode → Department.DeptCode

Part 2: ER Diagram Construction

Task 2.1 Hospital Management System

**1. Entities:**

**Patient (Strong):** PatientID, Name, Birthdate, (Address - composite), InsuranceInfo.
**Doctor (Strong):** DoctorID, Name, Phone (multi-valued), OfficeLocation.
**Department (Strong):** DeptCode, Name, Location.
**Appointment (Weak):** Depens on Patient and Doctor . May not have own unique ID
**Prescription (Weak):** Depens on Patient and Doctor
**Room (Strong):** But it's primary key should be {RoomNumber, DeptCode}, Because "room 101 in Cardiology is different from room 101 in Neurology".

**2. Attributes :**

**Patient:**
Simple: PatientID, Name, Birthdate, InsuranceInfo
Composite: Address -> {Street, City, State, Zip}
Multi-valued: Phone

**Doctor:**
Simple: DoctorID, Name, OfficeLocation
Multi-valued: Phone, Specialization (Doctor may have many specializations)

**Appointment:**
Simple: DateTime, Purpose, Notes

**Room:**
Simple: RoomNumber

**Department:**
Simple: DeptCode, Name, Location

**Prescription:**
Simple: MedicationName, Dosage, Instructions, DatePrescribed

**3. Relationships**

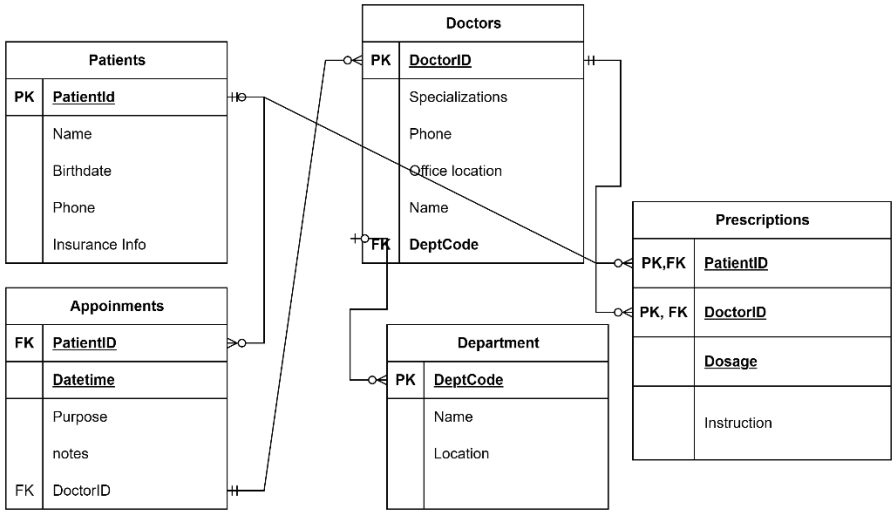
Patient **makes** Appointment: (1 to N) One patient may have many appoinments, one appointment belons to one patient.

Doctor **has** Appointment: (1 to N) One doctor may have many appoinments, one appointment belons to one doctor.

Doctor **belongs to** Department: (N to 1) One doctor belongs to ine depertment, one department have many doctors.

Doctor **writes** Prescription: (1 to N)

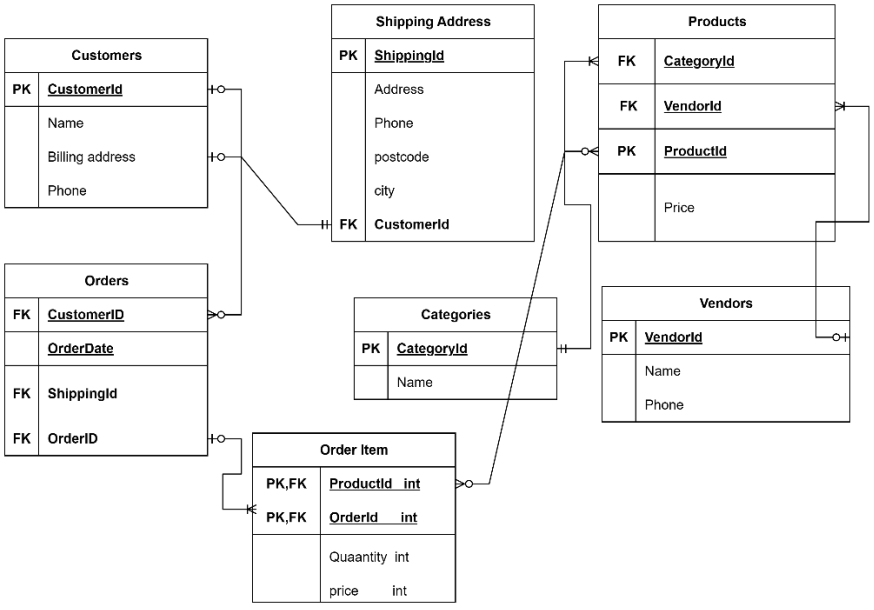
Patient **receives** Prescription: (1 to N)




Draw.io

Task 2.2: E-commerce Platform

1) ER diagram




2) Weak Entity: Order Item. It is depend on Orders entity to be primaty key.It does not make any sense, it is just a part of process.
3) Products and Order(Many to Many)

Task 4.1: Denormalized Table Analysis

1) **Functional dependencies:**
   StudentID -> StudentName, StudentMajor
   ProjectID -> ProjectTitle, ProjectType
   SupervisorID -> SupervisorName, SupervisorDept
   (StudentID, ProjectId) - > Role, HoursWorked, StartDate,EndDate

2) **Partial dependency :**

StudentID → StudentName, StudentMajor

ProjectID → ProjectTitle, ProjectType

SupervisorID → SupervisorName, SupervisorDept

Task 4.2

**Functional Dependencies:**

StudentID → StudentMajor

CourseID → CourseName

InstructorID → InstructorName

Room → Building

(CourseID, TimeSlot, Room) → InstructorID

5.1 Design challenge

Entities:

Students(StudentID, Name, Major, Year)

Clubs(ClubID, ClubName, Description, Budget)

Memberships(StudentID, ClubID, JoinDate, RoleID)

Officers(RoleID, RoleName)

Faculty(FacultyID, FacultyName, Department)

Advisors(ClubID, FacultyID)

Events(EventID, ClubID, EventName, EventDate, RoomID)

Attendance(EventID, StudentID, Status)

Rooms(RoomID, RoomName, Building, Capacity)

Expenses(ExpenseID, ClubID, Amount, ExpenseDate, Description)

3) One option would be to keep the officer role directly as an attribute (Role) inside the Membership table.

But creating a separate Officers table is better because it allows standardized officer roles (President, Treasurer, etc.). If a role changes, it only needs to be updated in one place.

4)Example Queries

1. "Find all students who are officers in the Computer Science Club."
2. "List all events scheduled for next week with their assigned rooms."
3. "Show the total expenses for each club in the current semester."