# Part 1: Key Identification Exercises

## Task 1.1: Superkey and Candidate Key Analysis

Relation A:

1.at least 6 superkeys:EmpID,SSN,email,{EmpID, SSN},{EmpID, Email},{SSN, Email},{EmpID, Phone}

2.empID,email,SSN

3. primary key: empID , because this is n artificial identifier(independent from external data) and also email, ssn are mutable and confidential.

4. yes, it can be so that's why phone doesn't fit as key.

Relation B:

1. StudentID,CourseCode, Section, Semester, Year

2. StudentID- defines student, CourseCode-defines a course, Section-spesific a group of course, Semester- unify the record

3. StudentID, CourseCode, Section, Semester, Year

## Task 1.2: Foreign Key Design

*Given Tables:*

```
Student(StudentID, Name, Email, Major, AdvisorID)
Professor(ProfID, Name, Department, Salary)
Course(CourseID, Title, Credits, DepartmentCode)
Department(DeptCode, DeptName, Budget, ChairID)
Enrollment(StudentID, CourseID, Semester, Grade)
```

all foreign keys: Student.AdvisorID--> Proffesor.ProfID,Course.CourseID-->Enroolment.CourseID,Student.StudentID-->Enrollment.StudentID,Course.DepartmentCode-->Department.DeptCode

# Part 2: ER Diagram Construction

## Task 2.1: Hospital Management System

1. patients(patientID,names,birthday,addresses,phone numbers,insurance)

Doctor(DoctorID, Name, Specialization, Phone, Location)

Department(DeptCode, DeptName, Location)

Appointment(AppointmentID, Date, Time, Purpose, Notes)

Prescription(PrescriptionID, Medication, Dosage, Instructions)

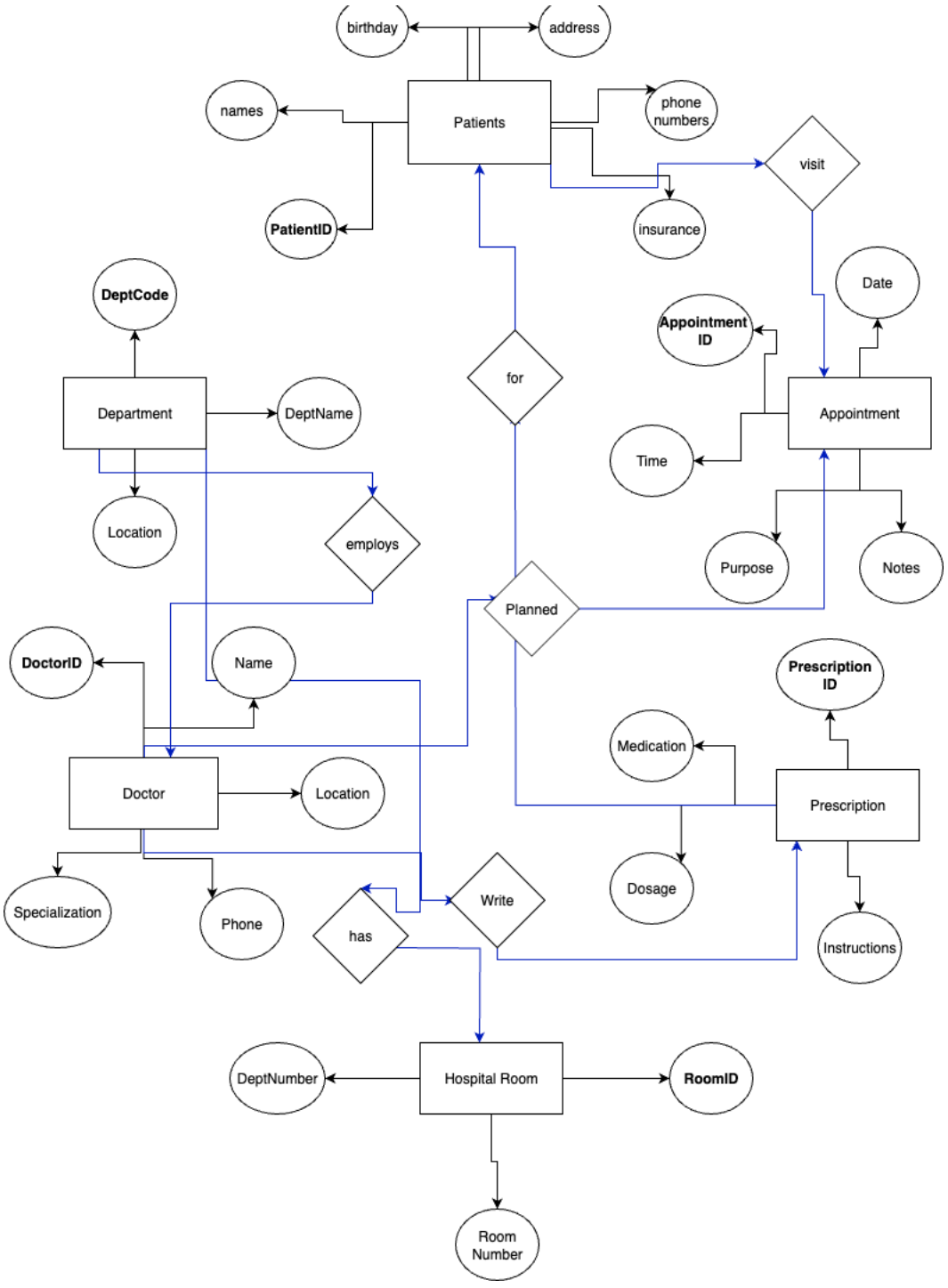
HospitalRoom(RoomID, RoomNumber, DeptCode)

2. Patient-->Doctors from Appointments(M:N)

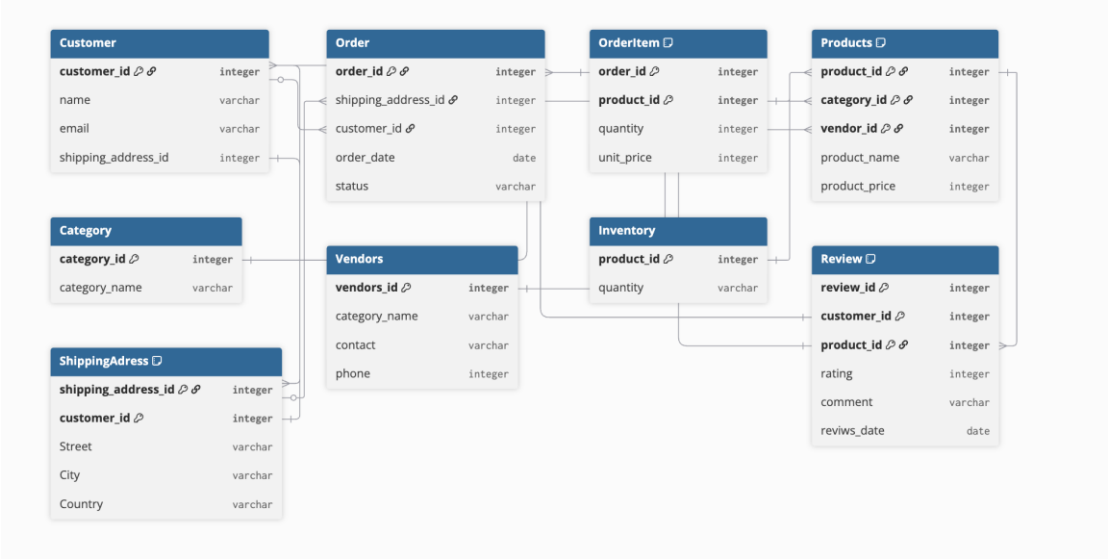
Doctor --> Patient from Prescriptions(M:N)

Departments-->Doctors (1:M)

Department-->Hospital Room(1:N)




## Task 2.2: E-commerce Platform




2) at least one weak entity is Orders items , because it cant exist without Orders and Products.Its only unique by combining both.

3) Orders--> Products(M:N) atributes: they already have from e **Order Items** (quantity and price at time of order)

## Part 4: Normalization Workshop

### Task 4.1: Denormalized Table Analysis

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

1) StudentID-->(StudentName,StudentMajor)
ProjectID-->(ProjectTitle,ProjectType)
SuperVisorID-->(SupervisorName,SupervisorDept)
StudentIDxProjectID-->(Role,HoursWorked,StartDate,EndDate)

2) Redundacy: SupervisorName is repeated many times for each project
Update anomaly: If supersvisor change his project to new one , but some of table its not updated. It must be updated everywhere
Insert anomaly: if we want to insert new student, but he hasn't have project , we can not add a new student without also assigning a project
Delete anomaly:deleting the last project also removes about a student.

3) table already has 1NF

4) Apply 2NF
primary key:StudentID, ProjectID
partial dependences:StudentID-->(StudentName,StudentMajor)
ProjectID-->(ProjectTitle,ProjectType)
Students(StudentID,StudentName, StudentMajor)
Projects(ProjectID, ProjectTitle,
ProjectType, SupervisorID, SupervisorName, SupervisorDept)
Work(Role, HoursWorked, StartDate, EndDate)

5) Apply 3NF
transitive dependences: SupervisorID-->SupervisorName, SupervisorDept
Supervisors(SupervisorID, SupervisorName, SupervisorDept)

### Task 4.2: Advanced Normalization

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

Each student has exactly one major
Each course has a fixed name
Each instructor has exactly one name
Each time slot in a room determines the building (rooms are unique across campus)
Each course section is taught by one instructor at one time in one room
A student can be enrolled in multiple course sections

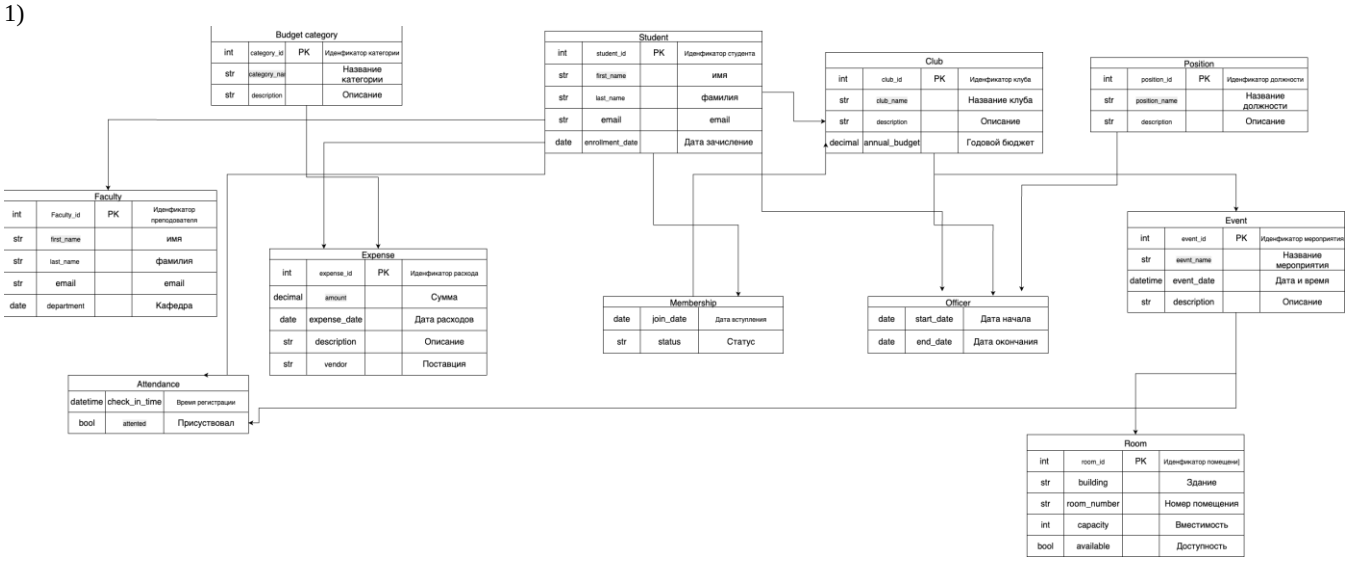
1) Primary key: StudentID,CourseID,Timeslot
2) Functional Dependences:
StudentID--> StudentMajor

CourseID--> CourseName
InrtuctorID-->InstructorName
Room-->Building
(CourseID,TimeSlot)-->(InstructorID,Room)
3)
There are some issues with BCNF, because StudentID-->StudentName(its not the key of entire table)
4) Decompose BCNF
Students(StudentID,StudentID, StudentMajor)
Course(CourseID, CourseName)
Instructor(InstructorID, InstructorName)
Rooms(Room,Building)
Schedule(CourseID,TimeSlot,Room,InsructorID)
Enrollment(StudentID,CourseID,TimeSlot)
5) no loss information and dependencies can be restored with JOIN

## Part 5: Design Challenge

### Task 5.1: Real-World Application

1)




2)
**STUDENT** (student_id PK, first_name, last_name, email, enrollment_date)
**FACULTY** (faculty_id PK, first_name, last_name, email, department)
**CLUB** (club_id PK, club_name, description, faculty_advisor FK-->FACULTY, annual_budget)
**ROOM** (room_id PK, building, room_number, capacity, available)
**EVENT** (event_id PK, club_id FK-->CLUB, event_name, event_date, description, room_id FK-->ROOM)
**POSITION** (position_id PK, position_name, description)
**BUDGET_CATEGORY** (category_id PK, category_name, description)
**EXPENSE** (expense_id PK, club_id FK-->CLUB, category_id FK-->BUDGET_CATEGORY, amount, expense_date, description, vendor)
**MEMBERSHIP** (student_id FK→STUDENT + club_id FK→CLUB PK, join_date, status)
**OFFICER** (student_id FK-->STUDENT + club_id FK-->CLUB + position_id F-->POSITION PK, start_date, end_date)
**ATTENDANCE** (student_id FK-->STUDENT + event_id FK-->EVENT PK, check_in_time, attended)
3)
**Option A (Chosen):** Model OFFICER as a separate associative entity with a composite key (student_id, club_id, position_id) and attributes like start_date. This explicitly defines that a student holds a specific position in a specific club.

it more occurately and clearly models the real-world scenario. Being an officer (e.g., President) is a distinct role from simply being a member. A student must first be a member to be an officer, but the two concepts are separate. This design:

**Prevents Nulls:** Avoids having many null position_id values in the MEMBERSHIP table for regular members.
**Is Explicit:** Creates a clear, dedicated structure for tracking leadership roles, their effective dates, and the specific position held.

4)

Event Planning: "For all events scheduled in the 'Main Hall' building next week, list the event name, club hosting it, expected time, and the room number. Also, show the name of the club's president."

Membership Engagement: "Find the top 3 most active clubs based on the average number of members attending their events in the last semester."