

# Database Systems Laboratory Work

Magrupov Almas

ID:B031867

## Task 1.1

### 1. Superkeys:

{EmpID}, {SSN}, {Email}, {Phone}, {EmpID, Name}, {SSN, Phone}

### 2. Candidate keys:

{EmpID}, {SSN}, {Email}, {Phone}

### 3. Primary key:

**EmpID** (best choice since it's numeric)

### 4. Phone numbers:

In real life, two employees can share the same phone number (e.g., office phones), so the phone is not a reliable identifier, even though it appears unique in the sample data.

### 1) Minimum Attributes for the Primary Key

The minimal and optimal primary key for this table is the single attribute EmpID. No combination of other attributes is necessary, as EmpID alone is sufficient to guarantee that every row is unique.

### 2) Justification for the Chosen Primary Key (EmpID)

The EmpID attribute is the most appropriate choice for the primary key for several key reasons:

1. **Uniqueness:** Each employee is assigned a distinct EmpID, ensuring no two rows in the table can have the same value. This satisfies the core requirement of a primary key.
2. **Non-Null:** As a mandatory organizational identifier, EmpID is guaranteed to be present for every employee record; it cannot be null.
3. **Stability:** Unlike personal information (e.g., name, phone number, or email), an employee's ID number is unlikely to change during their tenure. This stability prevents the need for complex updates to foreign key references in related tables.
4. **Efficiency:** EmpID is typically a compact numeric field, which makes it highly efficient for database operations like indexing, sorting, and joining tables.

5. **Minimality:** The key consists of only one attribute. It is impossible to remove any attribute and still maintain uniqueness, confirming that it is a minimal set.

### 3) Existence of Other Candidate Keys

Based on the sample data provided, other attributes also possess the uniqueness required to be considered candidate keys:

1. **SSN:** The Social Security Number is designed to be unique for each individual.
2. **Email:** The data suggests each email address in the table is unique.
3. **Phone:** The provided phone numbers also appear to be unique in this specific dataset.

#### Important Considerations on Reliability:

While SSN, Email, and Phone are technically candidate keys based on uniqueness, they are generally poor practical choices for a primary key compared to a surrogate key like EmpID:

- 1) **SSN** is highly sensitive information, raising significant privacy and security concerns if used as a widespread identifier.
- 2) **Email addresses** can change if an employee changes departments or leaves the organization.
- 3) **Phone numbers** are not truly stable; they can be reassigned, shared, or changed (e.g., moving from a desk phone to a mobile number).

Therefore, while these attributes can serve as unique identifiers, EmpID remains the superior, more reliable, and more secure choice for the primary key.

#### Task 1.2

#### Tables & Keys

##### 1. **Student(StudentID, Name, Email, Major, AdvisorID)**

1.1.PK: StudentID

##### 2. **Professor(ProfID, Name, Department, Salary)**

2.1.PK: ProfID

##### 3. **Course(CourseID, Title, Credits, DepartmentCode)**

3.1.PK: CourseID

##### 4. **Department(DeptCode, DeptName, Budget, ChairID)**

4.1.PK: DeptCode

5. **Enrollment(StudentID, CourseID, Semester, Grade)**

5.1.PK: usually composite (StudentID, CourseID, Semester)

---

1. **Foreign Keys**

2. **Student → Professor**

2.1.AdvisorID → Professor.ProfID

(Each student has an advisor who is a professor.)

3. **Student → Department**

3.1.Major → Department.DeptCode

(A student's major must be one of the existing departments.)

4. **Course → Department**

4.1.DepartmentCode → Department.DeptCode

(Each course belongs to a department.)

5. **Department → Professor**

5.1.ChairID → Professor.ProfID

(Each department has a professor as its chair.)

6. **Enrollment → Student**

6.1.StudentID → Student.StudentID

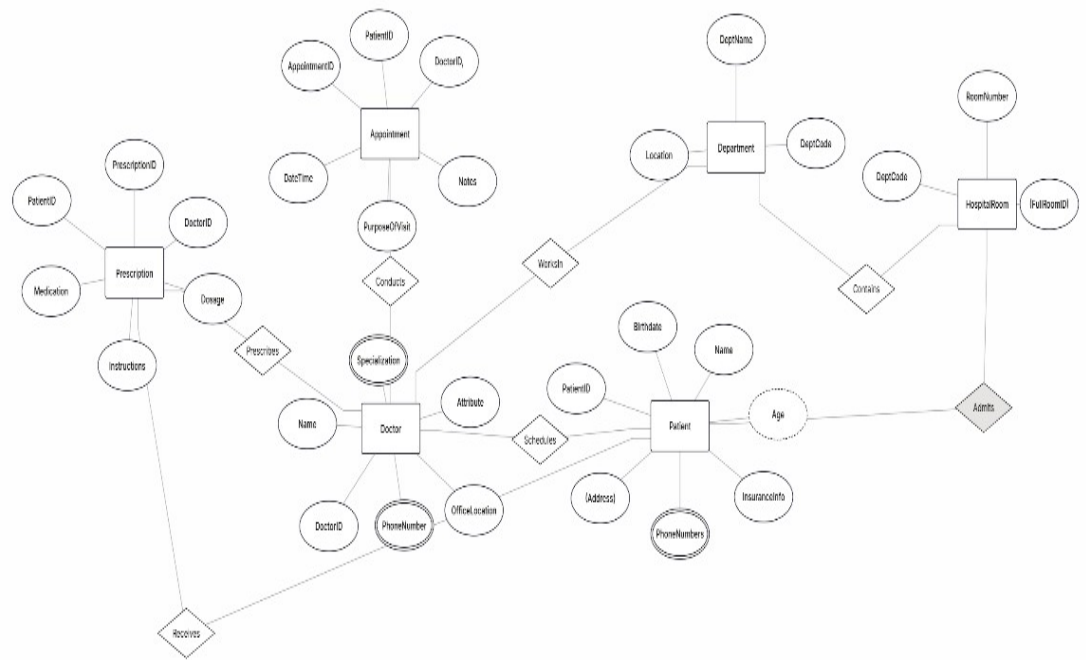
(Each enrollment record refers to a student.)

7. **Enrollment → Course**

7.1.CourseID → Course.CourseID

(Each enrollment record refers to a course.)

Task 2.1



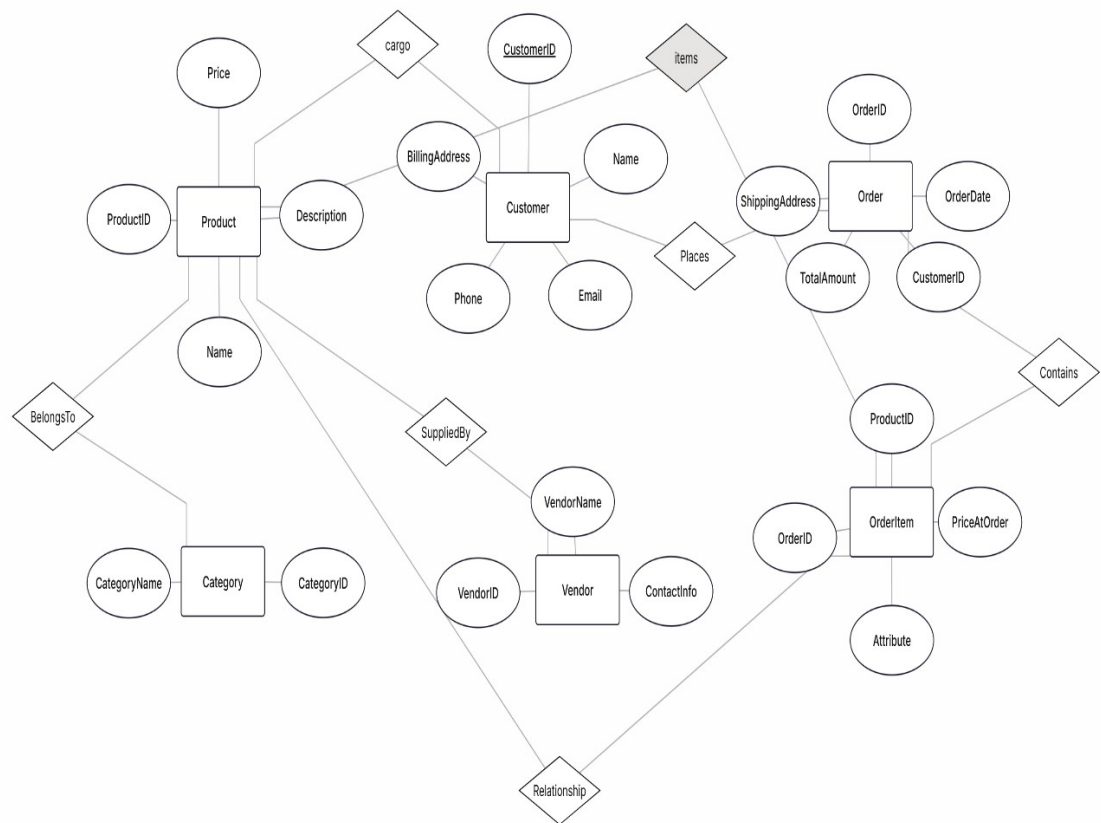
## Primary Keys (PK)

1. **Patient** → PatientID
  2. **Doctor** → DoctorID
  3. **Department** → DeptCode
  4. **Appointment** → AppointmentID (или составной: PatientID + DoctorID + DateTime)
  5. **Prescription** → PrescriptionID (или составной: PatientID + DoctorID + Medication)
  6. **HospitalRoom** → (RoomNumber, DeptCode) (составной PK)
-

## Foreign Keys (FK)

1. Appointment.PatientID → Patient.PatientID
2. Appointment.DoctorID → Doctor.DoctorID
3. Doctor.DeptCode → Department.DeptCode
4. Prescription.PatientID → Patient.PatientID
5. Prescription.DoctorID → Doctor.DoctorID
6. HospitalRoom.DeptCode → Department.DeptCode

## Task 2.2



## Task 4.1

### Functional Dependencies (FDs):

1. StudentID  $\rightarrow$  StudentName, StudentMajor
2. ProjectID  $\rightarrow$  ProjectTitle, ProjectType, StartDate, EndDate
3. SupervisorID  $\rightarrow$  SupervisorName, SupervisorDept
4. (StudentID, ProjectID)  $\rightarrow$  Role, HoursWorked

**Problems:**

- Redundancy (student, project, supervisor details repeated).
- Update anomaly (e.g., supervisor name change everywhere).
- Insert anomaly (cannot add student/supervisor without project).
- Delete anomaly (losing student data when project is removed).

**Decomposition to 3NF:**

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

Task 4.2

**Functional Dependencies (FDs):**

1. StudentID  $\rightarrow$  StudentMajor
2. CourseID  $\rightarrow$  CourseName, InstructorID
3. InstructorID  $\rightarrow$  InstructorName
4. TimeSlot, Room  $\rightarrow$  Building (rooms are unique across campus)
5. (StudentID, CourseID)  $\rightarrow$  (all other attributes specific to enrollment)

**Primary Key:** (StudentID, CourseID)

**BCNF Decomposition:**

1. **Student**(StudentID, StudentMajor)
2. **Course**(CourseID, CourseName, InstructorID)
3. **Instructor**(InstructorID, InstructorName)
4. **Room**(Room, Building)
5. **CourseSchedule**(StudentID, CourseID, TimeSlot, Room)

### Check BCNF:

- All FDs have determinants as candidate keys → now in BCNF.

### Possible Loss of Information:

- No loss of essential information, but queries require more joins after decomposition.

### Task 5.1

