Database lab1  
Task 1.1  
Superkeys:EmpID,SSN,Email,Phone

Candidate keys:EmpID,SSN,Email,Phone  
  
Task 1.1

A  
1.Superkeys: EmpID;SSN;Email;EmpID,Name;SSN,Name;EmpID,SSN,Email

2. Candidate keys:EmpID,SSN,Email,Phone  
3.EmpID because it is the best choise due to can not be used by multiusers

4.Actually according to our data the answer is no.But I think that actually one number can be used by one or more users in one time, it is not unreal

B

1.Minimum attributes for the primary key:  
StudentID, CourseCode, Section, Semester, Year

2.Explanation of why each attribute is necessary:

StudentID: Identifies the student.

CourseCode: Identifies the course.

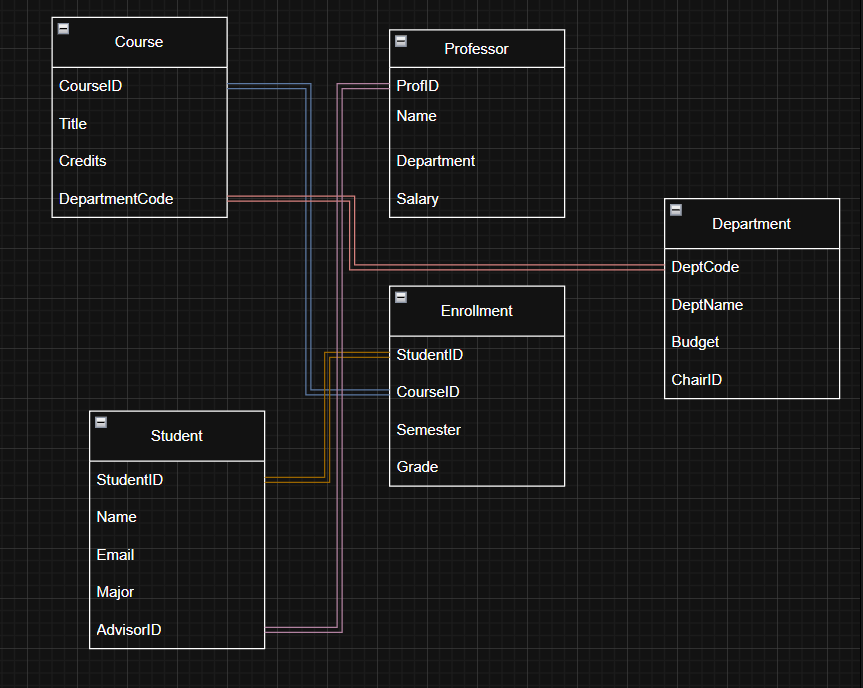
Section: Identifies the specific section of the course.

Semester: Identifies the semester the course is taken in.

Year: Identifies the academic year the course is taken in.

3.Additional candidate keys:  
StudentID,CourseCode,Semester,Year

Task 1.2



Task 2.1

1.

Strong entities: Patient.PatientID,Doctor.DoctorID,Department.DeptCode,Room.RoomNumber

Weak entities: Appointment.PatientID, Appointment.DoctorID, Prescription.PatientID, Prescription.DoctorID

2.

Patient Entity:

PatientID: Simple

Name: Simple BirthDate:

Simple Street: Composite

City: Composite

State: Composite

Zip: Composite

Phone: Multi-valued

InsuranceInfo: Simple

Doctor Entity:

DoctorID: Simple

Name: Simple

Specialization: Multi-valued

Phone: Multi-valued

OfficeLocation: Simple

Department Entity:

DeptCode: Simple

DeptName: Simple

Location: Simple

ChairID: Simple

Appointment Entity:

AppointmentID: Simple

PatientID: Simple

DoctorID: Simple

DateTime: Simple

PurposeOfVisit: Simple

Notes: Simple

Prescription Entity:

PrescriptionID: Simple

PatientID: Simple

DoctorID: Simple

Medication: Simple

Dosage: Simple

Instructions: Simple

Room Entity:

RoomNumber: Simple

DeptCode: Simple   
Capacity: Simple

Equipment: Simple

3.

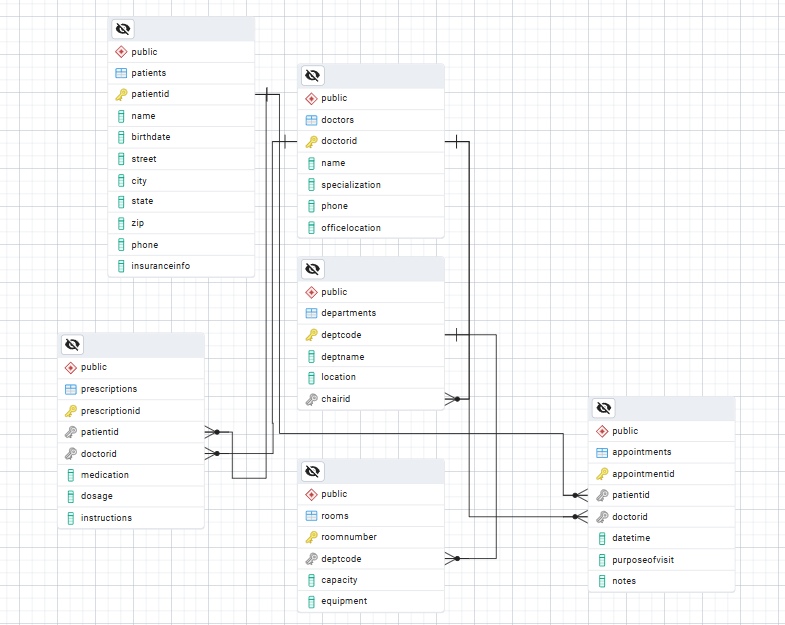
Patient -> Appointment: One to many

Doctor -> Appointment: One to many

Patient -> Prescription: One to many  
Doctor -> Prescription: One to many

Department -> Doctor: One to one

4.



5.

Patient: PatientID

Doctor:DoctorID

Department:DeptCode

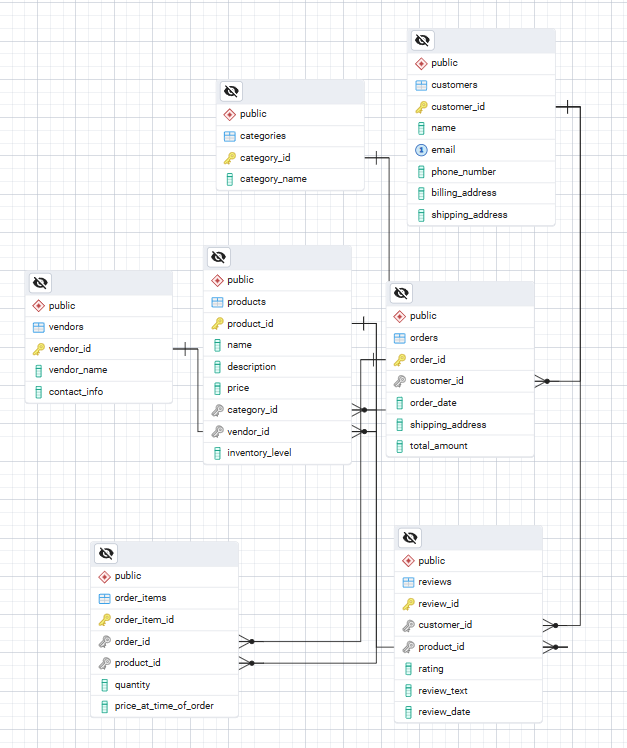
Appointment:AppointmentID

Prescription:PrescriptionID’

Room:RoomNumber

Task 2.2

1.



2.order\_items is weak entity, because this entity can not exist without another entities

3. Customers can review multiple Products, and each Product can have multiple Customers reviewing it.

This relationship is many-to-many because:

A single customer can write multiple reviews for different products.

A single product can have reviews from multiple customers.

Task 4.1

1.

StudentID -> StudentName, StudentMajor

ProjectID -> ProjectTitle, ProjectType

SupervisorID -> SupervisorName, SupervisorDept

StudentID, ProjectID -> Role, HoursWorked, StartDate, EndDate

StudentID, ProjectID -> SupervisorID, SupervisorName, SupervisorDept

2.

- If a student participates in multiple projects, their information like name and major will be repeated for each project.

3.

The StudentProject table is already in 1NF. There are no violations, and no changes are needed for this normal form.

4.

The `StudentProject` table is decomposed into 4 smaller tables in 2NF, which removes partial dependencies and ensures that each non-prime attribute depends on the full composite primary key.

Task 4.2

1.

primary key is combination of attributes such as CourseID, InstructorID, TimeSlot

2.

CourseID -> CourseName

InstructorID -> InstructorName

Room -> Building

CourseID, InstructorID, TimeSlot, Room -> StudentID

StudentID, CourseID, TimeSlot, Room -> StudentMajor

3.

Not BCNF

4.

Courses Table:

course\_id, course\_name

Instructors Table:

instructor\_id, instructor\_name

Rooms Table:

room\_id, room\_number, building

StudentEnrollments Table:

student\_id, course\_id, time\_slot, room\_id, student\_major

CourseSchedule Table:

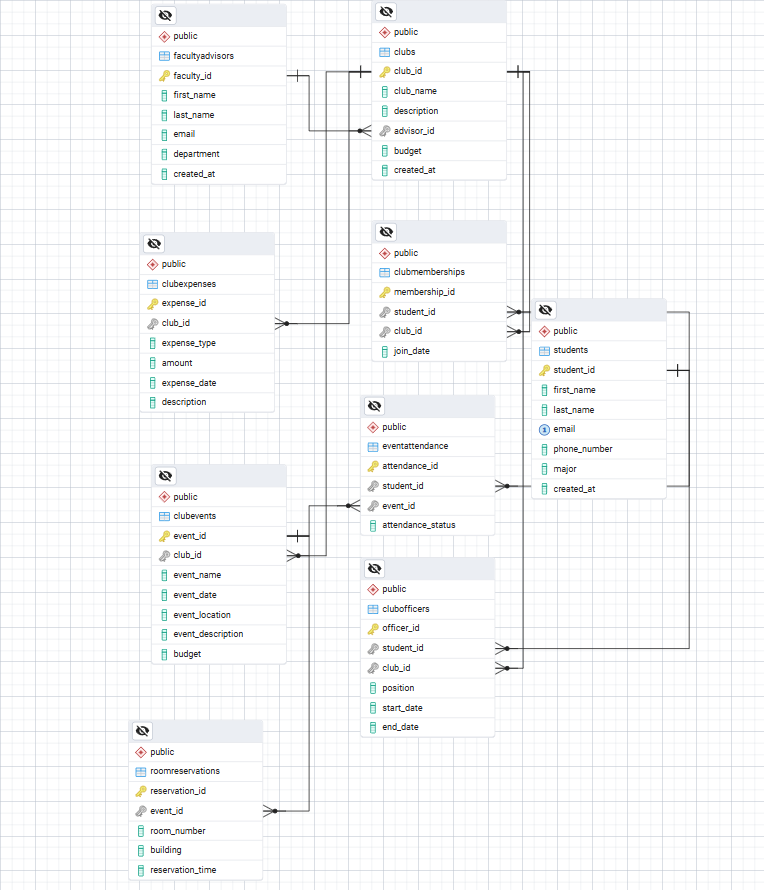
course\_id, instructor\_id, time\_slot, room\_id

5.

It lose its convenience in use

Task 5.1

1.



3.

I chose the many-to-many relationship table because it aligns better with normalization principles and offers more flexibility for future requirements