PART 1

Task 1.1

Relation A: Employee

1) Superkeys (at least 6):

1.1 EmpID

1.2 SSN

1.3 Phone

1.4 EmpID + SSN

1.5 EmpID + Name

1.6 SSN + Name + Department + Email

1.7 EmpID + SSN + Email + Phone + Name + Department + Salary\

2) Candidate keys:

1.1 EmpID

1.2 SSN

1.3 Phone (only if country code and all necessary parts are added, not like in samples. For example, +7 123 123 11 22)

1.4 Email (if it proveded by the company itlesf, not employees filling a form with their personal email as several relatives may enter the same email address, because of Digital Literacy)

1.0 In any other cases information to unify a person will not be enough, because it we consider email, phone, name, department and salary all together, then there can be a case when two people have the same data. For example, two relatives one of whom is not good with technology and doesn't have email, so both entered the same email, then they don't have personal phone number so they entered the same home phone number or payphone (таксофон) number.

3) Primary keys can be SSN and EmpID.

SNN, because Social Security Number is unique and can be given only by US government.

EmpID, because the employer company have a database of its employees and giving the same EmpID to different people would be pointless, so it is also unique.

Phone number can also be primary key if it is entered fully, but it may be home phone address, so there are some extreme cases when it can be doubled.

4) Based on the data in Phone column we see that it is a short form of a phone number so potentially there can be identical phone numbers for different people depending on countrty, region/state.

Relation B: Course Registration

1) StudentID, CourseCode, Section, Semester, Year

2) If we consider a table called Registrations, then we must know who is registering, what course and according to the business logic what section he chose, the semester and year as the DB is saving a historical event under hood.

3) No

TASK 1.2

1) Student: Major (depends on departments, cause they choose what courses specific majors must study) and AdvisorID (there should be another table of advisors the ID refers to OR Professors can be advisors at least)
2) Professor: Department (Obvious reason)
3) Course: DepartmentCode (Obvious reason)
4) Department: ChairID
5) Enrollment: StudentID, CourseID

PART 2

TASK 2.1

1)

Strong: Patients, Doctors, Departments, Appointments, Prescriptions.

Weak:, Hospital Rooms.

2 and 5)

Patients:

1. **PatientID** – Simple (Primary)
2. Name – Simple
3. Birthdate – Simple
4. Address – Composite (City, Street, building, room and so on.)
5. Phone Number**s** – Multi-valued
6. Insurance Info – Composite

Doctors:

1. **DoctorID** – Simple (Primary)

2. Name – Simple
3. Specialization – Multi-valued
4. Phone Number**s** – Multi-valued
5. DepartmentID – Simple (FK)
6. Office location – Simple (FK)

Departments:

1. **DepartmentCode** – Simple (Primary)
2. Name – simple
3. Location – Composite (FK)

Prescriptions:

1. **PrescriptionID** – Simple (Primary)
2. DoctorID – Simple (FK)
3. PatientID – Simple (FK)
4. Dosage – Simple, but usually depending on the weight doctor can calculate it, so in this case it can also be derived)
5. Intruction – Simple

Appointments:

1. **AppointmentID** – Simple (Primary)
2. DoctorID – Simple (FK)
3. PatientID -Simple (FK)
4. Date&Time – Simple
5. Purpose – Simple
6. Notes – Simple

Hospital Rooms:

1. DepatrmentID – Simple (FK)
2. Room Number – Simple

3)

1.1 Patients + Prescriptions (1:N)

1.2 Patients + Appointments (1:N)

1.3 Patient + Hospital Rooms (1:N)

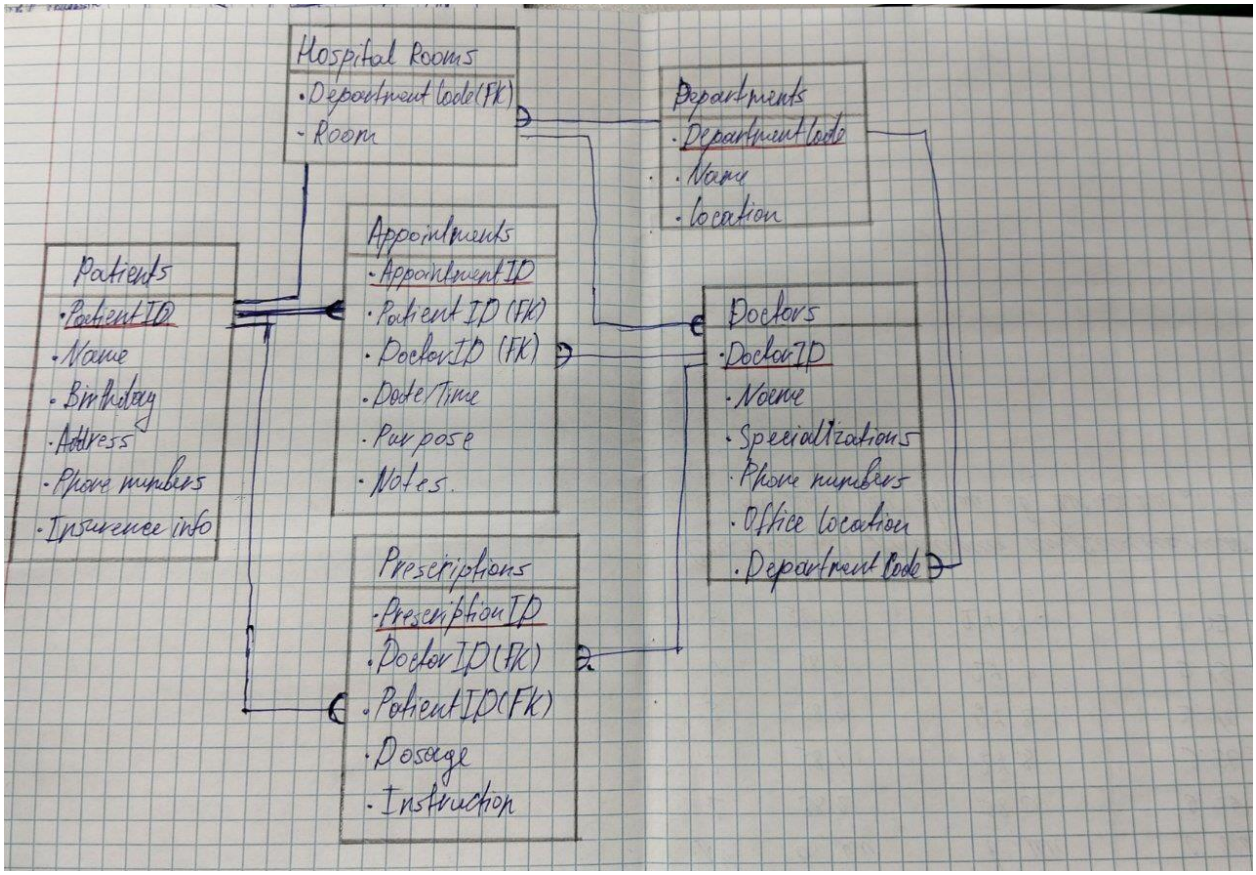
2.1 Doctors + Prescriptions (1:N)

2.2. Doctors + Appointments (1:N)

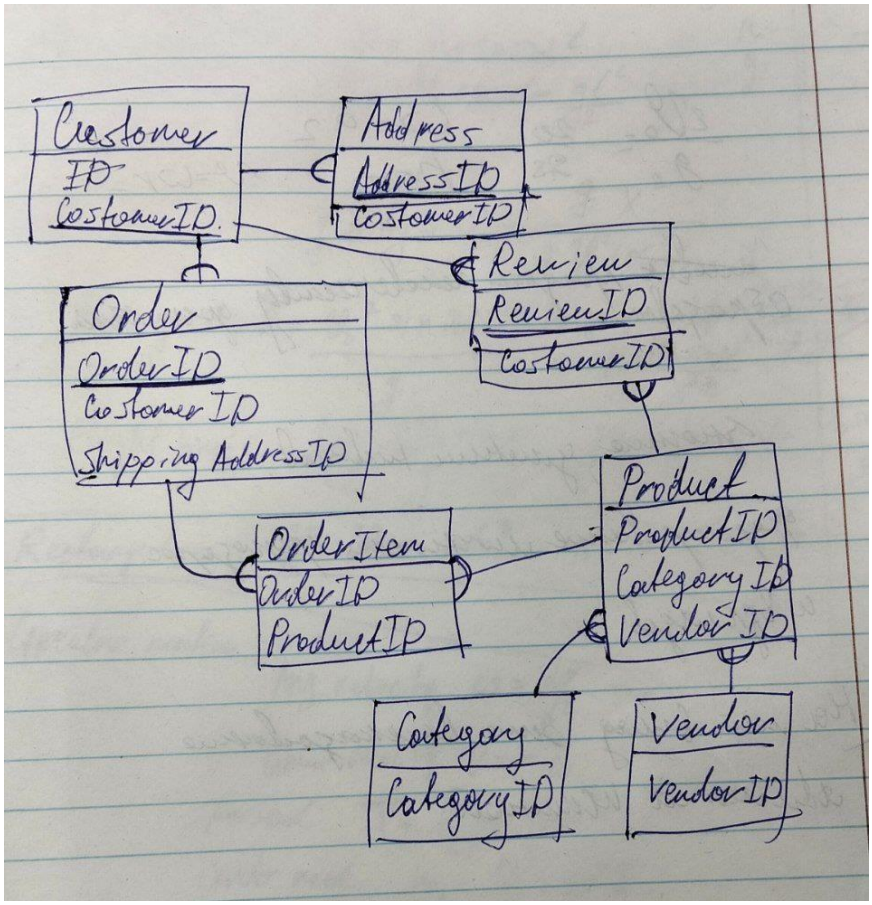
2.3 Doctors + Departments (1:N)

3.1 Department + Hospital Rooms (1:N)

2.1 Diagram



TASK 2.2



2. OrderItem because it depends on OrderID and ProductID. It cannot be independent.

3. Product to Order (M:N). It can be solved by OrderItem using quantity attribute.