Part 1

1.1 (a) Relation A: Employee(EmpID, SSN, Email, Phone, Name, Department, Salary)

1. Superkeys (at least 6):

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

2. Candidate keys:

{EmpID}, {SSN}, {Email}

3. Primary key:

EmpID, because it is a system-generated identifier that never changes and does not contain sensitive personal data like SSN or information that may change such as Email or Phone.

4. Phone numbers:

In the sample data, each employee has a unique phone number. However, in real-world scenarios phone numbers are not guaranteed to be unique (e.g., shared office phones), so Phone should not be treated as a candidate key.

1.1 (b) Relation B: Registration(StudentID, CourseCode, Section, Semester, Year, Grade, Credits)

1. Minimal attributes for the primary key:

{StudentID, CourseCode, Section, Semester, Year}

2. Why each attribute is necessary:

- StudentID: identifies the student.
- CourseCode: identifies which course is being taken.
- Section: distinguishes between different sections of the same course.
- Semester: prevents multiple registrations for the same course in the same semester.
- Year: ensures that semesters across different years can be distinguished.

3. Additional candidate keys:

There are no other natural candidate keys. A surrogate key such as RegistrationID could be added if desired.

1.2 Foreign Keys

- Student(AdvisorID) → Professor(ProfID)
- Professor(Department) → Department(DeptCode)
- Course(DepartmentCode) → Department(DeptCode)
- Department(ChairID) → Professor(ProfID)
- Enrollment(StudentID) → Student(StudentID)
- Enrollment(CourseID) → Course(CourseID)

Part 2

2.1 Hospital ERD

Entities: Patients, Doctors, Departments, Appointments, Prescriptions, HospitalRooms

Attributes:

- Patients: PatientID (simple), Name (composite), Birthdate (simple), Address (composite), PhoneNumbers (multi-valued), InsuranceInfo (simple)
- Doctors: DoctorID (simple), Name (composite), Specialization (multi-valued), PhoneNumbers (multi-valued), OfficeLocation (simple)
- Departments: DepartmentCode (simple), Name (composite), Location (simple)
- Appointments: PatientID (derived), DoctorID (derived), DateTime (simple), PurposeOfVisit (simple), Notes (simple)
- Prescriptions: PatientID (derived), DoctorID (derived), Medications (multi-valued), Dosage (simple), Instructions (simple)

- HospitalRooms: DepartmentCode (derived), Room (simple)

Relationships:
- Patients–Appointments: 1:N
- Doctors–Appointments: 1:N
- Departments–HospitalRooms: 1:N
- Doctors–Prescriptions: 1:N
- Patients–Prescriptions: 1:N

2.2 E-commerce ERD

Weak entity: OrderItem is a weak entity because it requires both Order and Product to uniquely identify each instance.

Identifying relationships:
- Order ↔ OrderItem (1:N)
- Product ↔ OrderItem (1:N)

Part 4

4.1 Normalization

Step 1: Functional Dependencies (FDs)

For Employee(EmpID, SSN, Email, Phone, Name, Department, Salary):
- EmpID → SSN, Email, Phone, Name, Department, Salary
- SSN → EmpID, Email, Phone, Name, Department, Salary
- Email → EmpID, SSN, Phone, Name, Department, Salary

For Registration(StudentID, CourseCode, Section, Semester, Year, Grade, Credits):
- (StudentID, CourseCode, Section, Semester, Year) → Grade
- (CourseCode, Section, Semester, Year) → Credits

Step 2: 1NF → 2NF → 3NF
- Both relations are already in 1NF (all attributes are atomic).
- Employee: All non-key attributes depend fully on the candidate keys. No partial dependencies, so it is in 2NF. There are no transitive dependencies, so it is in 3NF.
- Registration: The dependency (CourseCode, Section, Semester, Year) → Credits causes redundancy, since Credits depend only on the course offering and not on the student. To reach 3NF, we decompose into:
- Registration(StudentID, CourseCode, Section, Semester, Year, Grade)
- CourseOffering(CourseCode, Section, Semester, Year, Credits)

Thus both Employee and Registration relations are in 3NF after decomposition.