Creating a complete ER (Entity-Relationship) diagram for a hospital database involves several entities, relationships, and attributes. Here's a simplified example to get you started. Please note that real-world hospital databases can be more complex and may require a thorough analysis based on specific requirements.

Entities:

1. *Patient:*
  - Attributes: PatientID (PK), Name, DateOfBirth, Gender, Address, ContactNumber

2. *Doctor:*
  - Attributes: DoctorID (PK), Name, Specialization, ContactNumber

3. *Nurse:*
  - Attributes: NurseID (PK), Name, Department, ContactNumber

4. *Department:*
  - Attributes: DepartmentID (PK), Name, Description

5. *Appointment:*
  - Attributes: AppointmentID (PK), Date, Time

6. *Prescription:*
  - Attributes: PrescriptionID (PK), Date, Medication, Dosage

Relationships:

- A patient can have multiple appointments (One-to-Many relationship between Patient and Appointment).
- A doctor can have multiple appointments (One-to-Many relationship between Doctor and Appointment).
- A nurse can be assigned to multiple patients (One-to-Many relationship between Nurse and Patient).
- A doctor can write multiple prescriptions (One-to-Many relationship between Doctor and Prescription).
- An appointment is associated with one doctor (Many-to-One relationship between Appointment and Doctor).
- An appointment is associated with one patient (Many-to-One relationship between Appointment and Patient).

- A prescription is associated with one doctor (Many-to-One relationship between Prescription and Doctor).
- A prescription is associated with one patient (Many-to-One relationship between Prescription and Patient).

These relationships can be visually represented in an ER diagram using appropriate symbols and lines connecting entities. Keep in mind that this is a simplified example, and in a real-world scenario, you may need to consider additional entities, attributes, and relationships based on specific requirements and constraints.

Question two :

Designing an ER (Entity-Relationship) diagram for a university database involves identifying entities, their attributes, and the relationships between them. Below is a simplified example of an ER diagram for a university database:

Entities:
1. *Student:*
- Attributes: StudentID (Primary Key), Name, Address, DateOfBirth

2. *Course:*
- Attributes: CourseID (Primary Key), Title, Credits

3. *Department:*
- Attributes: DepartmentID (Primary Key), Name

4. *Professor:*
- Attributes: ProfessorID (Primary Key), Name, Office, DepartmentID (Foreign Key)

5. *Enrollment:*
- Attributes: EnrollmentID (Primary Key), StudentID (Foreign Key), CourseID (Foreign Key), Semester, Grade

Relationships:
1. *Teaches:*
- Between Professor and Course

- Cardinality: One Professor teaches many Courses (1:N)

2. *Enrolls:*
   - Between Student and Enrollment
   - Cardinality: One Student enrolls in many Courses (1:N)

3. *WorksIn:*
   - Between Professor and Department
   - Cardinality: One Professor works in one Department (1:1)

4. *Manages:*
   - Between Department and Professor
   - Cardinality: One Department is managed by one Professor (1:1)

5. *BelongsTo:*
   - Between Student and Department
   - Cardinality: One Student belongs to one Department (1:1)

This is a basic representation and may not cover all the complexities of a real-world university database. Depending on the specific requirements, you may need to extend or modify the entities and relationships. Additionally, attributes such as timestamps, constraints, and other relevant details should be considered in a complete database design.