Part 1.

Task 1.1. Relation A:

1. List at least 6 different superkeys: EmpID, SSN, Email, Phone, Name, Salary.
2. Identify all candidate keys: EmpID, Email, Phone, Name, Salary.
3. Which candidate key would you choose as primary key and why? EmpID. Because EmpID is a stable, simple and secure identifier. Unlike SSN or Email, it does not change, is not confidential and is effective for linking to other tables.
4. Can two employees have the same phone number? Justify your answer based on the data shown. No, two employees cannot have one phone number, because the phone number is given to one person and if the same number is given to another, then the first person's number becomes non-working and it is impossible to reach him on this number (you reached the second person).

Relation B:

1. Determine the minimum attributes needed for the primary key: StudentID, CourseCode, Section, Semester, Year. (5)
2. StudentID identifies the student; without it, we cannot distinguish registrations of different students. CourseCode specifies which course the student is taking. Section distinguishes multiple offerings of the same course in the same semester. Semester and Year distinguish the same course/section offered in different terms. Without these, a student retaking the same course in another semester would not be uniquely identified. If any of these attributes is omitted, the key will no longer be unique.
3. There are no additional potential keys.

Task 1.2

1. Student(StudentID) > Enrollment(StudentID), Course(CourseID) > Enrollment(CourseID), Course(DepartmentCode) > Department(DeptCode), Professor(Department) > Department(DeptCode)

Part 2.

Task 2.1.

1. Patient - Strong. Doctor - Strong. Department - Strong. Appointment - Weak ( existence depends on both Patient and Doctor). Prescription - Weak (existence depends on both Doctor and Patient). Hospital Room - Weak (room number is only unique within a Department).
2. **Patient:** PatientID (Simple), Name (Simple), Birthdate (Simple), InsuranceInfo (Simple), Address (Composite), PhoneNumbers (Multi-valued). **Doctor:** DoctorID (Simple), Name (Simple), OfficeLocation (Simple), PhoneNumbers (Multi-valued), Specialization (Multi-valued). **Department**: DeptCode (Simple,), DeptName (Simple), Location (Simple). **Appointment (Weak):** AppointmentID (Partial Key), DateTime (Simple), Purpose (Simple), Notes (Simple). **Prescription (Weak):** PrescriptionID (Partial Key), Medication (Simple), Dosage (Simple), Instructions (Simple). **Hospital Room (Weak):** RoomNumber (Partial Key), RoomType (Simple).
3. **Patient – Appointment – Doctor**: Many-to-Many (M:N) between Patients and Doctors, resolved by Appointment (which contains details). **Doctor – Prescription – Patient:** Many-to-Many (M:N), resolved by Prescription. **Department – Doctor:** 1:N (one department employs many doctors). **Department – Patient: O**ptional (patients may be assigned to a department if admitted) → 1:N. **Department – HospitalRoom**: 1:N (one department has many rooms).

Part 4.

Task 4.1

1. Identify functional dependencies: StudentID → StudentName, StudentMajor. ProjectID → ProjectTitle, ProjectType, SupervisorID, SupervisorName, SupervisorDept. SupervisorID → SupervisorName, SupervisorDept. (StudentID, ProjectID) → Role, HoursWorked, StartDate, EndDate.