1.1

a.

1)

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

2)

{EmpId}, {SSN}, {Phone}

3)

{EmpId}. Because this column is the system counter that is not likely to change in any case unlike other columns such ssn, phone number or name.

4)

No, based in the table data each employee has its onw unique phone number. As we see each Jonh, Mary and Bob have different phone numbers (555-0101,555-0102, 555-0103)

b.

1)

{StudentID, CourseCode, Section, Semester, Year}

2)

StudentID is essentioal since we have to identify the student who registers the course. That CourseCode is needed to identyfy the type of course that is being selected. We need Section part to make sure that there is no the sections of the same course are unique. Semester is also essential to make sure that two same courses are not taken in the same semester. Then year is also important to distinguish the semesters, since we can register the same course but it have to be different years.

3)

there is no such.

1.2

Student and Enrollment tables ON {StudentID}

Course and Enrollment tables ON {CourseID}

2.1

1)

Patients(S), Doctors(S), Departments(S), Appointments(W), Prescriptions(W), HospitalRooms(S)

2)

Patients{PatientsID - simple, name - composite, birthdate - simple, address - composite , phonenumbers - multi-valued , insurance information - simple}

Doctors{DoctorID - simple, name - composite, specialization - multi-valued, phone numbers - multi-valued, office location - simple}

Departments{DepartmentCode - simple, name - complosite, location - simple}

Appointments{PatientsID - derives, DoctorID - derived, DateTime - multi-valued, PurposeOfVisit - simple, notes - multi-valued}

Prescriptions{PatientsID - derives, DoctorID - derives, medications - multi-valued, dosage - simple, instructions - simple}

HospitalRooms{DepartmentCode - derived, room - simple}

3)

Patients - Appointments: 1:N
Doctors - Appointments: 1:N
Departments - Hospital Rooms: 1:N
Doctors - Prescriptions: 1:N
Patients - Prescriptions: 1:N

2.2

2)

OrderItem is a weak entity because it requires the identification of other entities Order and Product to uniquely define it.

3)

Products and Orders

4.1
1)
StudentID → StudentName
StudentID → StudentMajor
ProjectID → ProjectTitle
ProjectID → ProjectType
SupervisorID → SupervisorName
SupervisorID → SupervisorDept
2)
While listing the projects of one student, StudentID, StudentName, StudentMajor repeats itself many times. Therefore it is the redundancy.
Update problem: if we want to update some info like name of the student, because of the data redundancy we have to update each row where this name appears.
Insert problem: If we want to insert new student to the table we have to also fill the projects columns, even if student has not any projects, therefore there is a insert problem
3)
There are some NF1 vialations: attributes there are not atomic and there are rows that are not unique;
Ways to fix it: Ensure atomic values by breaking down multi-valued attributes into separate tables and remove any repeating groups, such as supervisors, into separate rows or tables.
4)
Primary key is a StudentID and ProjectID.
There is a partial dependancy there. For example StudentName and StudentMajor depend only on StudentID, SupervisorName and SupervisorDept depend only on SupervisorID.
Decomposition of this table to NF2. Separate out the attributes that depend only on part of the composite key into new tables. StudentTable{StudentID, StudentName, StudentMajor}, SupervisorTable{SupervisorID, SupervisorName, SupervisorDept}, ProjectTable{ProjectID, ProjectTitle, ProjectType}.
5)
SupervisorDept is transitively dependent on StudentID through SupervisorID. In other words, SupervisorDept is dependent on SupervisorName, which is dependent on SupervisorID
Move SupervisorDept to the SupervisorTable, ensuring that each supervisor's department is stored in one place, and SupervisorID can be used to access it.
3NF Decomposition: Student Table{StudentID,StudentName,StudentMajor}, Supervisor Table{SupervisorID, SupervisorName, SupervisorDept}, Project Table{ProjectID, ProjectTitle, ProjectType}, StudentProject Table{StudentID,ProjectID, Role, HoursWorked, StartDate, EndDate}

4.2
1)
Composite primary key for the CourseSchedule table could be the combination of StudentID, CourseID, and TimeSlot
2)
StudentID → StudentMajor
CourseID → CourseName
InstructorID → InstructorName
TimeSlot → Building
Room, TimeSlot → Building
Room → Building
3)
StudentID → StudentMajor: This violates BCNF because StudentID is not a superkey,CourseID → CourseName: This also violates BCNF, because CourseID alone does not uniquely identify the entire row
4)
turning the CourseSchedule table into Student{StudentID, StudentMajor}, Course{CourseID , CourseName}, Instructor{InstructorID, InstructorName}, TimeSlotTable{TimeSlot, Building}, CourseSchedule{StudentID, CourseID, InstructorID, TimeSlot, Room}
5)

Loss of Redundancy: The decomposition removes redundancy, as each table now stores only unique pieces of data related to students, courses, instructors, and time slots.