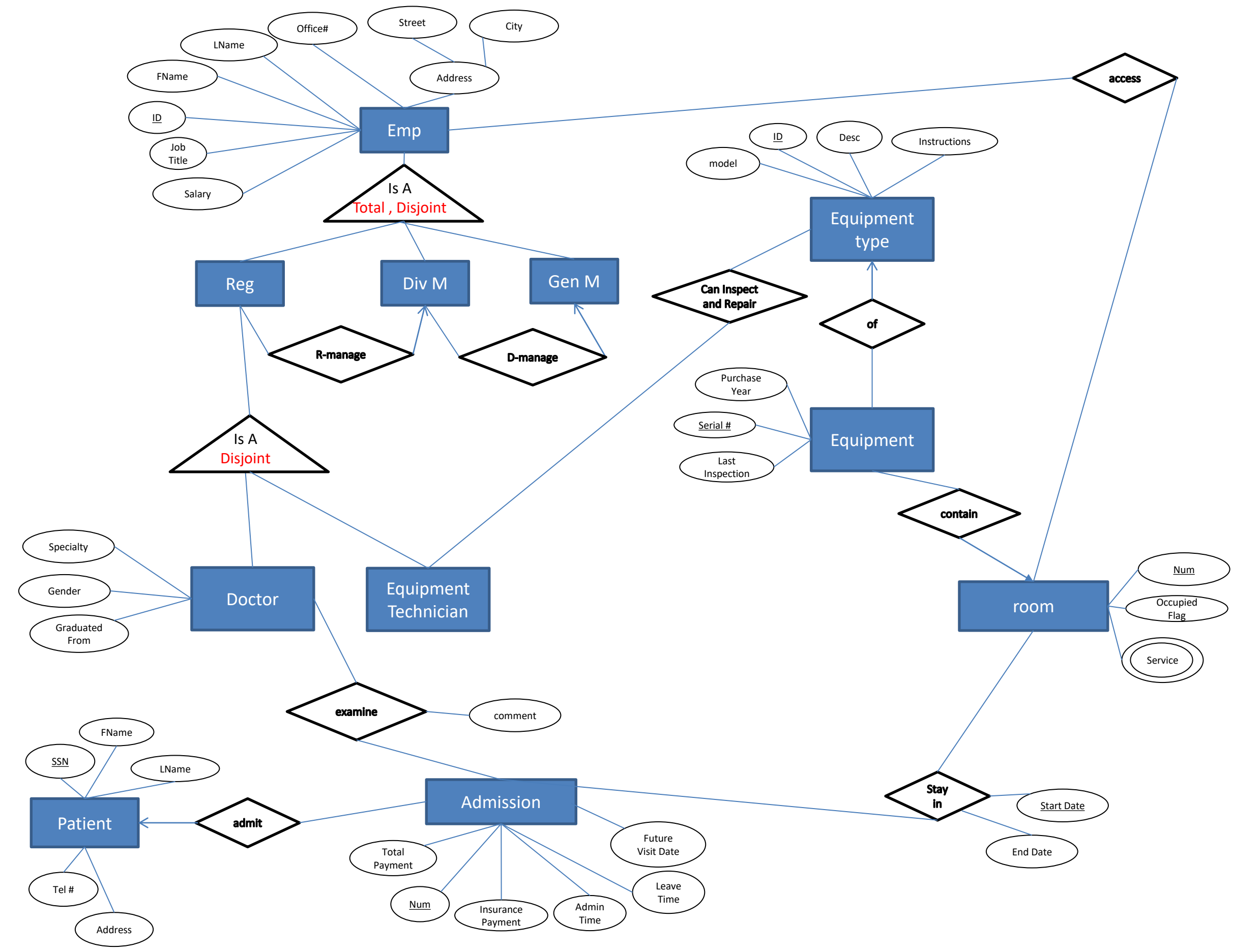
Database Systems I
CS3431

Solution Phase 1 Project


Office#
Street
City
LName
FName
Address
ID
Job Title
Salary
Emp
access
Is A
Total , Disjoint
ID
Desc
Instructions
model
Equipment type
Reg
Div M
Gen M
Can Inspect and Repair
of
R-manage
D-manage
Purchase Year
Serial #
Last Inspection
Equipment
Is A
Disjoint
contain
Specialty
Gender
Graduated From
Doctor
Equipment Technician
Num
Occupied Flag
room
Service
examine
comment
FName
SSN
LName
Stay in
Start Date
Patient
admit
Admission
Future Visit Date
Total Payment
Leave Time
End Date
Tel #
Num
Insurance Payment
Admin Time
Address

The Relational Model

Employee(ID, FName, LName, Salary, jobTitle, OfficeNum, empRank, supervisorID, AddressStreet, AddressCity)
//That is one design, other designs for the ISA relationship are possible…
Doctor (EmployeeID, gender, specialty, GraduatedFrom)
EquipmentTechnician(EmployeeID)
CanRepairEquipment(EmployeeID, EquipmentType)

EquipmentType (ID , Desc , model, instructions)
Equipment (Serial#, TypeID, PurchaseYear, LastInspection , roomNum)

Room(Num, occupied flag)
RoomService (roomNum , service)
RoomAccess (roomNum , EmpID)

Patient (SSN, FirstName, LastName, Address, TelNum)
Admission (Num, AdmissionDate, LeaveDate, TotalPayment, InsurancePayment, Patient_SSN, FutureVisit)
Examine (Doctor ID, AdmissionNum, comment)
StayIn(AdmissionNum , RoomNum, startDate, endDate)

Notes:

1- There are other possible designs that can be also correct and capture the application requirements.

2- For example, the "Emp" entity set can be assumed as the regular employees and in this case, the "ISA" relationship will have only two subclasses "Division Managers" and "General Managers".

3. As we mentioned in class, there are several way to convert "ISA" relationship to the relational model. The one given above is the most compact and efficient one. The "Emp_rank" field is used to capture whether the employee is regular (rank = 0), division manager (rank = 1), or general manager (rank = 2).

4- The room services can be modeled as a multi-valued attribute, and then in the relational model it will become a relation by itself (that is the approach we considered above). Another way, is to model the services as a separate entity set in the ERD that has a "1-M" relationship with the "room" entity set.

5-In the "StayIn", we assumed a patient may go the same room multiple times in the same visit, e.g., go to the emergency room twice. That is why we added the "start date" as part of the key.
(This assumption is not mandatory, you may assume that a room may be used at most once for a given visit)

6- For the future visits, if you assume a single admission can have multiple future visits assigned, then you should make this attribute as multi-valued attribute.