Task 1

1. There are no actors in this database that have been in no movies.
   Maybe. The entity sets Actors and Movies are in a many-many relationship, an actor can be connected to 0, 1 or many movies, vice versa. So maybe there are no actors in this database that have been in no movies.

2. There are some actors who have acted in more than ten movies.
   Maybe. As said above, the entity sets Actors and Movies are in a many-many relationship, an actor can be connected to many movies, so maybe some actors who have acted in more than ten movies.

3. Some actors have done a lead role in multiple movies.
   True. The entity sets Movies and Actors are in a many-one relationship, an actor can be lead roles in 0, 1, or many movies. So some actors have done a lead role in multiple movies.

4. A movie can have only a maximum of two lead actors.
   False. The entity sets Movies and Actors are in a many-one relationship, a movie can be connected to at most 1 lead actor, so the statement is false.

5. Every director has been an actor in some movie.
   False. The entity sets Movies and Directors are in a many-one relationship, a director can be connected to 0, 1 or many movies, so the statement is false.

6. No producer has ever been an actor.
   False. The entity sets Producers and Actors are in one-one relationship, a producer can be 0 or 1 actor, so the statement is false.

7. A producer cannot be an actor in some other movie.
   True. The entity sets Producers and Actors are in one-one relationship, a producer can be at most 1 actor, so the producer who is an actor in a movie can not be an actor in some other movies.

8. There are movies with more than a dozen actors.
   Maybe. The entity sets Actors and Movies are in a many-many relationship, a movie can have 0, 1 or many actors, so maybe there are movies with more than a dozen actors.

9. Some producers have been a director as well.
   Maybe. The entity sets Producers and Actors are in a one-one relationship, and the entity sets Actors and Directors are in a one-one relationship. We can see that a

producer can be an actor or not, and an actor can be a director or not, so maybe some producers have been a director as well.

10. Most movies have one director and one producer

Maybe. The entity sets Movies and Directors are in a many-one relationship, a movie can have 1 director; The entity sets Movies and Producers are in a many-many relationship, means that a movie can have 1 producer. So maybe most movies have one director and one producer.

11. Some movies have one director but several producers.

True. The entity sets Movies and Directors are in a many-one relationship, a movie can have 1 director; The entity sets Movies and Producers are in a many-many relationship, means that a movie can have 0, 1, or many producer, so statement is true.

12. There are some actors who have done a lead role, directed a movie, and produced a movie.

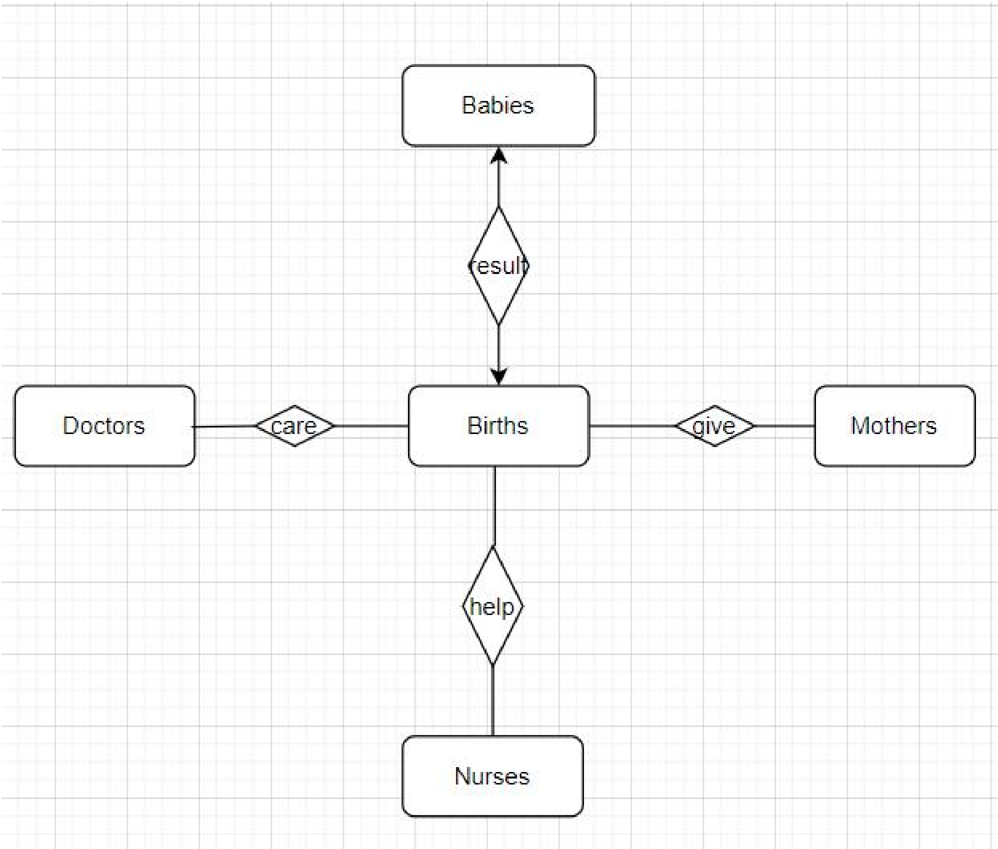
Maybe. As showed in the diagram, the entity sets Actors and Directors are in a one-one relationship, an actor can be a director; the entity sets Movies and Directors are in a many-one relationship, a director can direct a movie; the entity sets Movies and Actors are in a many-one relationship, an actor can be a lead role in a movie. Hence, maybe there are some actors who have done a lead role, directed a movie, and produced a movie.

13. No movie has a director who also acted in that movie.

Maybe. As showed in the diagram, the entity sets Movies and Directors are in a many-one relationship,the entity sets Actors and Directors are in a one-one relationship, the entity sets Movies and Actors are in a many-one relationship. We can see that a movie can have 1 dierctor, a director can be a an actor or not, an actor can be in 0 movie. Hence, maybe no movie has a director who also acted in that movie.

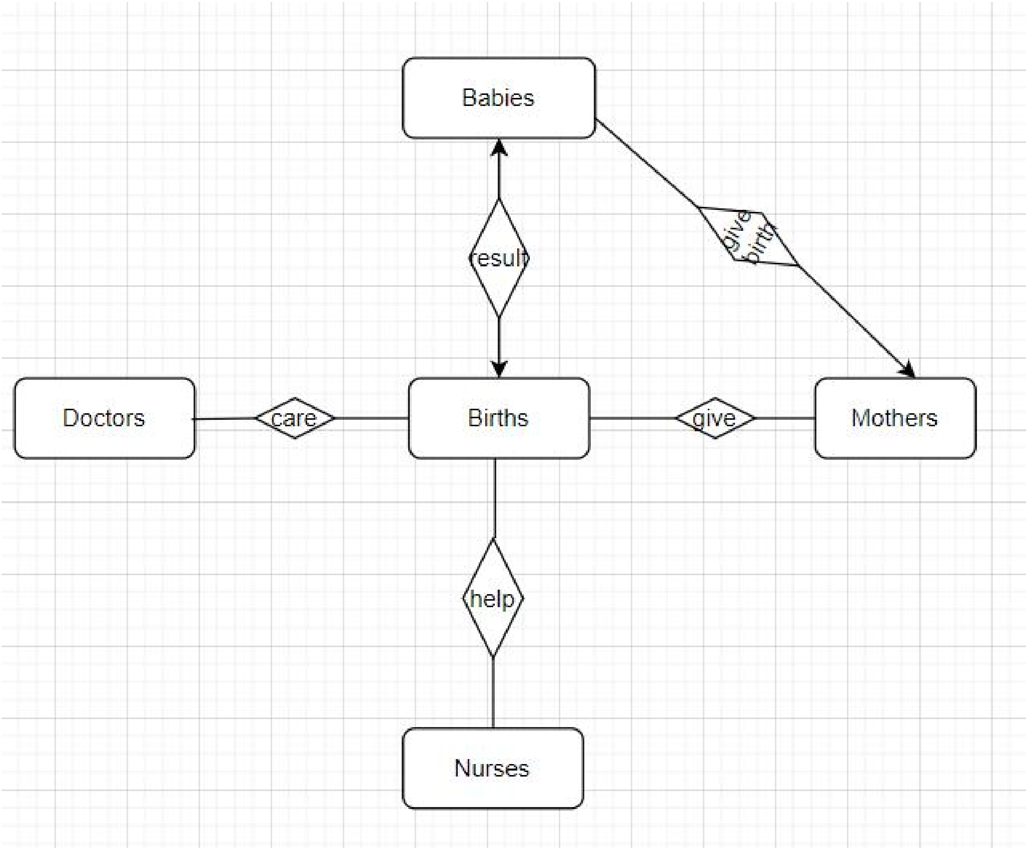
Task 2

1. Every baby is the result of a unique birth, and every birth is of a unique baby.



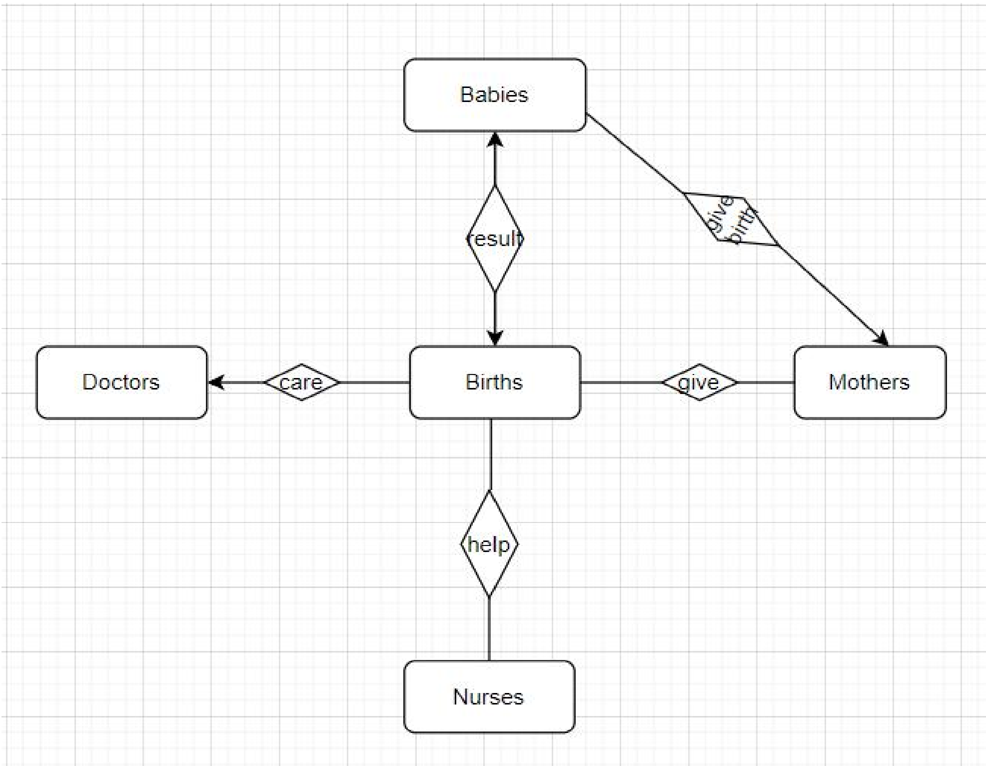
Firstly, there is a relationship between Mothers and Births and a relationship between Babies and Births, redundance exsits, reasonable situation is only one relaitonship and two entity sets; Secondly, mothers can give many births, and every birth may have more than one baby, but the added relationship between Babies and Births states every birth only has one baby, it’s not reasonable in the real world.

2. In addition to (1), every baby has a unique mother.

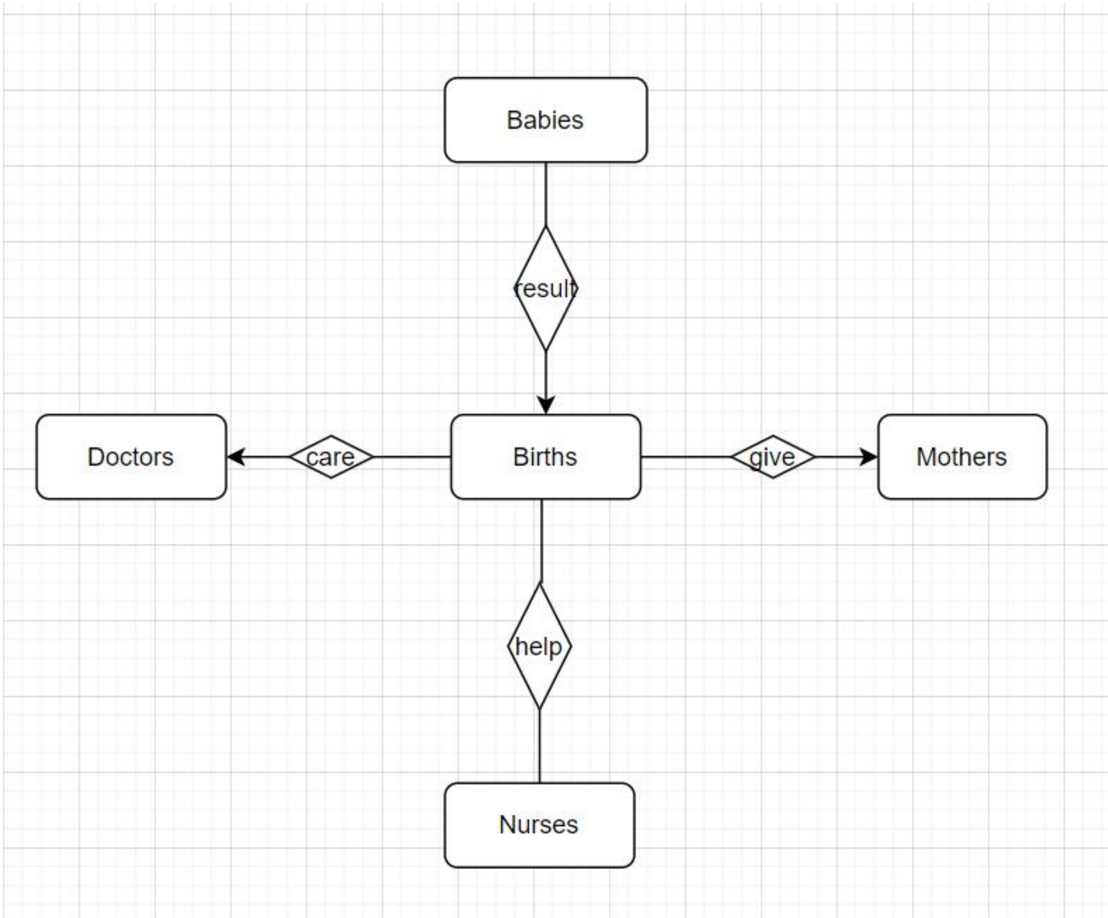


As stated in condition 1, redundance already exsits, but new relationship built between Babies and Mothers increases the redundance, because every baby is the result of unique birth, and of course every birth is given by unique mother, so there is no need to add a condition that every baby has a unique mother.

3. In addition to (1) and (2), for every birth there is a unique doctor.

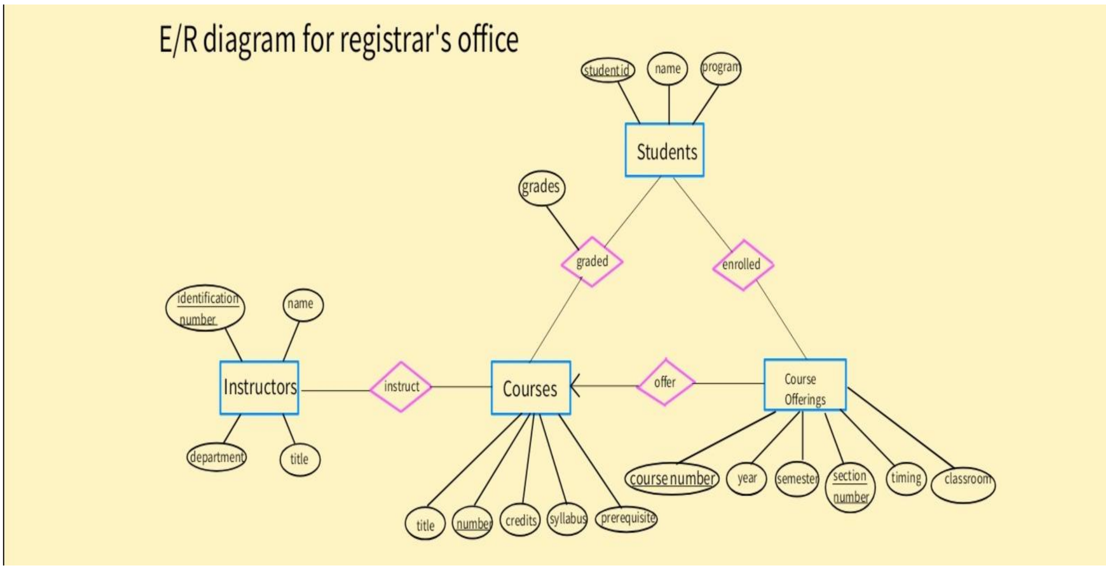


Every birth there is a unique doctor, in practice, every birth may have more than 1 doctors if emergency happens. New added condition increases the redundancy if there are more than 1 doctor.



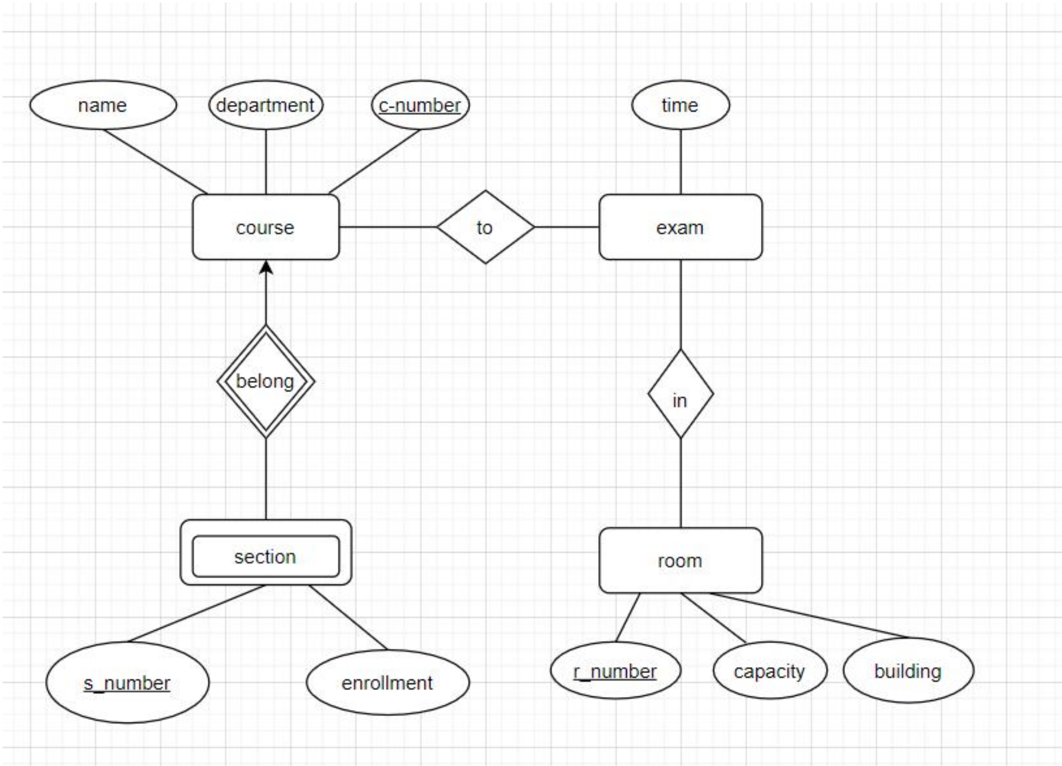
The relationship between Babies and Births changed from one-one from many-one, in order to represent the fact that every baby still has a unique mother, the relationship between Births and Mothers is change to many-one. There is no need to add a relationship between Babies and Mothers, hence it is deleted.

Task 3



The ER diagram for registrar’s office is above, there are four entities with their attributes, each has a key attribute noted with underline. Entities “Instructors” and “Courses” are in a many-many relationship “instruct”; The attribute “ instructors” in Entity set “Course Offerings” is deleted in the diagram as there would be redundance because an entity set “Instructors” already exsits. Entities Courses and Course offerings are in a one-many relationship “offer”; The enrollment of students in courses are modeled as a many-many relationship “enrolled” between entity sets “Students” and “Course Offerings”. The grades awarded to students in each course are modeled as a many-many relationship “graded” with an attributes “grades” between the entity sets “Students” and “Courses”.

Task 4



For entity set “room”, if entity sets “exam” and “room” are in a many-many or many-one relationship, and room is the “many” end, which means that a course exam take place in more than 1 room, “room” entity set should be included in the diagram, if not, the entity set “room” should not be included in it; For entity set “course”, if entity sets “course” and “exam” are in a many-many or many-one relationship, and “course” in the “many” end, which means that more than one courses exam will take place at the same time or the same room, the entity set “course” should be included, if not, the entity set “course” should not be included in the diagram; for entity set “ section”, if the entity set “courses” is included in the diagram, the “section” entity set should be in it as a weak entity set on “course”, if not, the “section” entity set should not be in the diagram.

Task5

Write the following queries in the relational algebra using relational schema:
Student(id, name)
EnrolledIn(id, code)
Subject(code, lecturer)

1. What are the names of students enrolled in 2dv513?

A new relation R built Using natural join,
R = Student ⋈ EnrolledIn
R(id, name, code)
R -- σ code='2dv513' -- π name
R(name) : π name(σ code = "2dv513"(R))

2. What are the names of students in both 1dv513 and 2dv513?

As above, a new relation R built using natural join method,
R = Student ⋈ EnrolledIn
R(id, name, code)
R -- σ code='2dv513' -- π name R1(name) : π name(σ code = "2dv513"(R))
R -- σ code='1dv513' -- π name R2(name) : π name(σ code = "1dv513"(R))
The names of students in both 1dv513 and 2dv513 is the intersection of R1(name) and R2(name), that is,
R(name) : R1(name) ∩ R2(name)

3. Who teaches 2dv610?

Using exsiting relation Subject(code, lecturer)
Subject -- σ code='2dv610' -- π lecturer
R(lecturer) : π lecturer(σ code = "2dv513"(Subject))

4. Who teaches 1dv513 and 2dv513?

Using exsiting relation Subject(code, lecturer)
Subject -- σ code='1dv513' -- π lecturer
R1(lecturer) : π lecturer(σ code = "1dv513"(Subject))
Subject -- σ code='2dv513' -- π lecturer
R2(lecturer) : π lecturer(σ code = "2dv513"(Subject))
The lecturer who teaches both 1dv513 and 2dv513 are intersection of R1(lecturer) and R2(lecturer), that is,
R(lecturer) : R1(lecturer) ∩ R2(lecturer)

5. What are the names of students who are taking a subject not taught by Ilir?

The names of students who are taking a subject not taught by llir is the difference between the set of all the names of students who are taking a subject and the set of the names of students who are taking a subject taught by llir.
First, build a new relation R using natural join
R = Student $\bowtie$ Subject
R(id, name, code, lecturer)
The names of students who are taking a subject:

R1(name) : π name(R)

The names of students who are taking a subject taught by llir:

R2(name) : π name(σ lecturer = “llir”(R))

R(name) = R1(name) - R2(name)

Task 6

Interviews(manager, applicant, day, time, room).

1. Find functional dependencies.

From text “Each manager uses the same room for all interviews during a day” we can find a FD:
manager day --> room
From the whole text, we can find FDs:
applicant, day --> manager, time, room
day, time, room --> manage, applicant
manager, day, time --> applicant, room

2. Find the keys of the relation.

{Applicant, day}: from the two attributes, we can determine the other 3 attributes(manager, time, room), as we can conclude from the descripition that an applicant interviewed by one manager one time at one day in one room.

{Day, time, room} : from the two attributes, we can determine the other 2 attributes(manager, applicant), as stated above one manager interview one applicant one time at one day in one room.

{manager, day, time}: from the 3 attributes, we can determine the other 2 attributes(applicant, room), as stated above, one manager interview one applicant at a time during a day.

3. Show that the relation is in 3NF but not in BCNF

Relation Interviews(manager, applicant, day, time, room) is in 3NF because manager day --> room is a nontrival FD, room is a member of the key(day, time, room);
Relation Interviews(manager, applicant, day, time, room) is not in BCNF: {Applicant, day}, {Day, time, room},{manager, day, time} are three keys for Interviews; The 4 FDs manager day --> room, applicant day --> manager time room, day time room --> manager applicant, manager day time --> applicant room, we can see that left side of the FDs are not a superkey.

4. Decompose the relation in relations that are in BCNF

A FD manager day --> room violates BCNF
Compute { manager day }+
{ manager day }+ = { manager day room }
Interviews - ({manager day}+ - {manager day}) = { manager day applicant time }
It follows that

Interviews1(manager, day, room)

Interviews2(manager, day, applicant, time)

Both are in BCNF

5. Draw an E/R diagram that describes the system. Try to incorporate all dependencies.