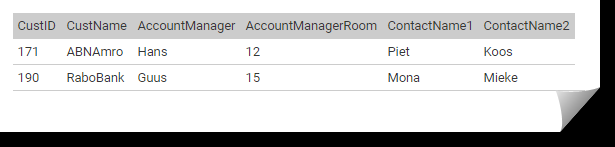
**ASSIGNMENT:**

***1.Find the functional dependencies for the below and normalize it till BCNF:***

****

**Solution:** The table is already in 1 NF form because each attribute has atomic values.

The table is already in 2 NF form because there is no partial dependency.

The dependencies are as follows:

Custid 🡪 CustName

Custid 🡪 AccountManager

Custid 🡪 ContactName1

Custid 🡪 ContactName2

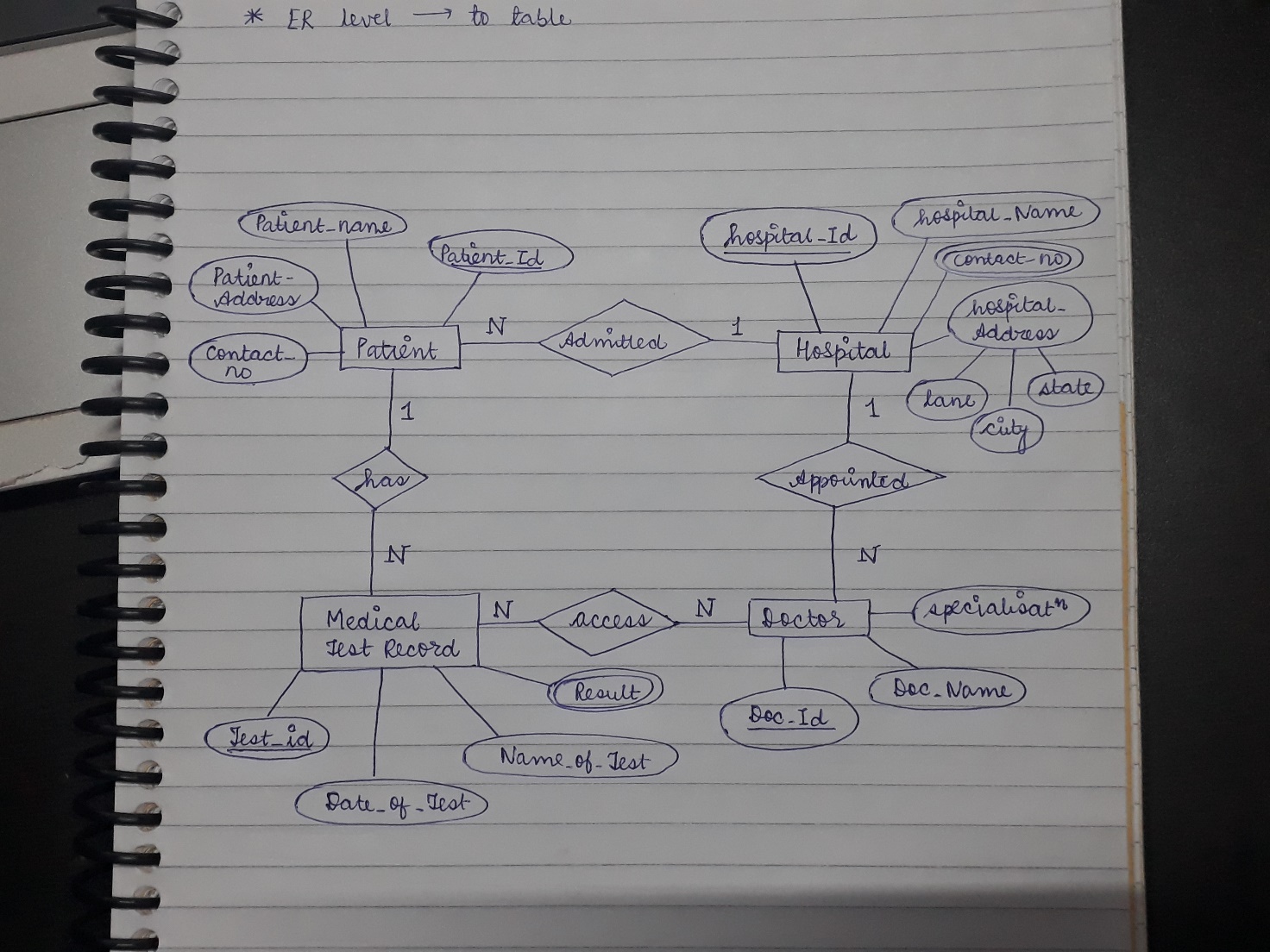
AccountManager 🡪 AccountManagerRoom ------------------------------------------------ (1)

So, there is no partial but transitive dependency as in (1), hence we split the table into two, making it in 3NF and BCNF.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Custid (primary key) | CustName | AccountManager  (foreign key) | ContactName1 | ContactName2 |

|  |  |
| --- | --- |
| AccountManager (Primary Key) | AccountManagerRoom |

***2. Draw an ER diagram for a hospital management system.***

****

***3.Consider a relation Student (StudentID, ModuleID, ModuleName, StudentName, StudentAddress, TutorId, TutorName). Each student is given a StudentID and each module given a ModuleID. A student can register more modules and a module can be registered by more students. TutorID is the ID of the student's personal tutor, it is not related to the modules that the student is taking. Each student has only one tutor, but a tutor can have many tutees. Different students can have the same name. Different students can be living at the same address.***

***Find all the functional dependencies holding in this relation and normalize the table to 3NF.***

Solution: The dependencies are as follows:

StudentID 🡪 ModuleId

ModuleId 🡪 StudentID

StudentID🡪 StudentName

StudentID🡪 StudentAddress

StudentID 🡪 TutorId

TutorId 🡪TutorName

ModuleId 🡪 ModuleName

Also, given that a student can opt for multiple modules and a tutor is assigned to a student not to the module type.

Therefore, StudentID and ModuleId will be the candidate key.

Hence five tables will be formed to normalise the relation to 3 NF

R1(StudentID, ModuleId)

R2(StudentID, TutorId): functional dependency of tutor on the student

R3 (StudentID, StudentName, StudentAddress): separate relation due to partial dependency of attributes on StudentID

R4(TutorId, TutorName): transitive relationship from StudentID 🡪 TutorId 🡪 TutorName

R5(ModuleId, ModuleName): functional dependency of ModuleName on ModuleId

\*underlined attributes are the primary key for the respective relations