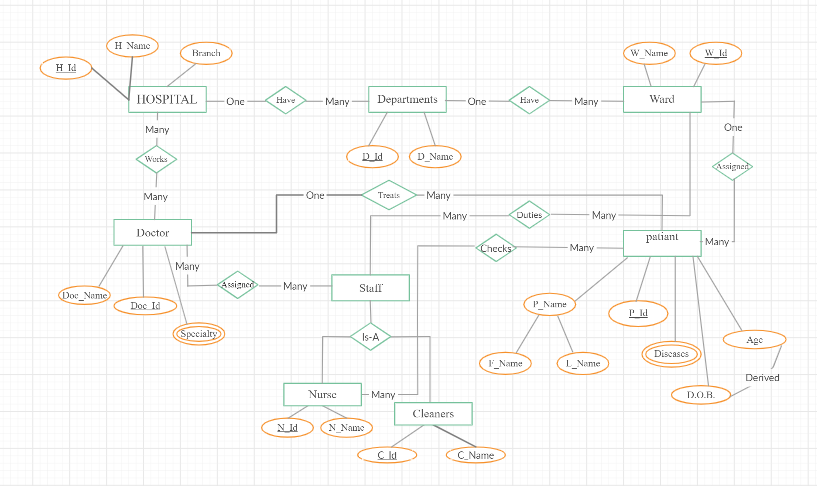
2. ER Diagram for Hospital management system -



----------------------------------------------------------------------

3.

Student (StudentID, ModuleID, ModuleName, StudentName, StudentAddress, TutorId, TutorName)

Functional dependencies holding in this relation:

StudentID -> StudentName, StudentAddress

StudentID -> TutorId

TutorId -> TutorName

ModuleID -> ModuleName

ModuleID, StudentID -> ModuleName, StudentName, StudentAddress, TutorId, TutorName

Here Primary Key is - ModuleID, StudentID

* The above relation is already in 1NF since all the attributes are atomic.
* For 2NF, There should not be any partial dependencies. As we can see,

StudentID+ = {StudentID, StudentName, StudentAddress, TutorId, TutorName}

TutorId+= {TutorId, TutorName}

ModuleID+ = {ModuleID, ModuleName}

ModuleID, StudentID+ = {ModuleID, StudentID, ModuleName, StudentName, StudentAddress, TutorId, TutorName}

As we can see, when we find closures for all the left hand side attributes, there are partial dependencies, we would like to split the above FDs into Student, Module, Student\_Module tables to remove it. Final tables will be:

1. Student{StudentID, StudentName, StudentAddress, TutorId, TutorName}

StudentID -> StudentName, StudentAddress

StudentID -> TutorId

TutorId -> TutorName

1. Module{ModuleID, ModuleName}

ModuleID -> ModuleName

1. Student\_Module{ModuleID, StudentID}

* For 3NF, there should not be any transitive dependencies. Above the Student table contains transitive dependency. So, we would handle it by splitting the tables into Student and Tutor. Final tables will be:

i. Student{StudentID, StudentName, StudentAddress, TutorId}

StudentID -> StudentName, StudentAddress

StudentID -> TutorId

ii. Tutor{TutorId, TutorName}

TutorId -> TutorName

After normalization(Upto 3NF), the tables will be:

Student (StudentID, StudentName, StudentAddress, TutorId)

Module (ModuleID, ModuleName)

Student\_Module (StudentID, ModuleID)

Tutor (TutorId, TutorName)

------------------------------------------------------------------------

1.

In the original table, FDs are(Assuming ContactName is related to Customer, Since it was not written anywhere about it.):

CustID -> CustName

CustID -> AccountManager

AccountManagerRoom -> AccountManager

CustID -> ContactName1

CustID -> ContactName2

The above table PK is {CustID, AccountManagerRoom}. All the attributes are atomic, so it is in 1NF. While finding the partial dependencies, we would split the table into Account{AccountManagerRoom, AccountManager} and Customer{CustID, CustName, AccountManager} then the FDs will be in 2NF. Since there are no transitive dependencies then it is in 3NF. For BCNF, attributes should be dependent only on the key. Since there are no such FDs. It is in BCNF.

After BCNF, the final tables will be:

Table 1:

Customer(CustID, CustName, AccountManager, ContactName1, ContactName2)

CustID -> CustName, AccountManager, ContactName1, ContactName2

Table 2:(Assuming in every room there will be one manager.)

Account(AccountManagerRoom, AccountManager)

AccountManagerRoom -> AccountManager

Table 3:

(CustID, ContactName1, ContactName2)

CustID -> ContactName1

CustID -> ContactName2

------------------------------------------------------------------------