Functional Dependencies and Normalisation

-Shaan Chopra(2015090)

-Aayushi Malik(2015003)

The functional dependencies in our relational model are:

Mess
MessID -> HostelID

Hostel
HostelID -> Hname
Phone_number -> HostelID
(Because phone_number is a multivalued attribute.)

Room
RoomID -> {HostelID, MessID, Floor, Type, Description, Status}
Floor -> MessID
MessID -> {HostelID, Floor}

Room Type
Type -> Room price
Room price -> Type

Application
AppID -> {RoomID, HostelID}

Student
StudentID -> {Sname, Sex, AppID, email, address, course_name}
email -> StudentID
Phone_number -> StudentID
(Because phone_number is a multivalued attribute.)

Decomposition:

- Inititally, our entity Room contained attributes type as well as price. However, since there were functional dependencies type -> price and price -> type, to attain 3NF, we decomposed Room into Room and Room Type. Now, Room contains attributes (RoomID, MessID, HostelID, Floor, Type, Description, Status) and Room Type has attributes (Type, Price).

- Since there are functional dependencies AppID -> RoomID and RoomID -> HostelID in the system, we can decompose Application into (AppID, RoomID) and (HostelID) to remove transitivity and achieve 3NF. As HostelID is a single attribute, we can discard it. So, we remove the attribute HostelID from Application.
- There are functional dependencies MessID -> RoomID and RoomID -> HostelID in the system so we must remove the dependency MessID -> HostelID to ensure there is no transitivity and 3NF form is achieved. So, since the entity Mess has only two attributes MessID and HostelID, we can remove the entity from our system. (Entity Room already contains an attribute MessID)
- Since there is a functional dependency MessID -> Floor in entity Room and Floor is a non-prime attribute and MessID is not the primary key, we can decompose Room into (RoomID, HostelID, Floor, Type, Description, Status) and (MessID, Floor). Now, we can recreate the entity Mess with attributes (MessID, Floor).
- Because Phone_number is a multivalued attribute and there is a functional dependency Phone_number -> HostelID, we have decomposed Hostel into Hostel(HostelID, Hname) and HostelNumber(HostelID, Phone_Number)
- Because Phone_number is a multivalued attribute and there is a functional dependency Phone_number -> StudentID, we have decomposed Student into Student(StudentID, Sname, Sex, Email, Address Course_name) and StudentNumber(StudentID, Phone_Number)

Another change in our system: we have removed the attribute AppID from Student and added attribute StudentID to Application instead, as a Student may not have applied for a room in the hostel at a particular instant in which case she/he will not have any AppID and we will have a null value. So, attribute StudentID has been added to Application instead.

Revised Relational Schema:

Room(RoomID, HostelID, Floor, Type, Description, Status)
Room Type(Type, Price)
Mess(MessID, Floor)
Student(StudentID, Sname, Sex, Email, Address, Course_Name)
StudentNumber(StudentID, Phone_number)
Hostel(HostelID, Hname)
HostelNumber(HostelID, Phone_number)
Application(AppID, StudentID, RoomID)