*Ngo Viet Hoang*

*REPORT*

**ER Mapping and Normalisation**

1. **Mapping**

GUEST (GuestID, FirstName, LastName, Address, MobilePhone, DateOfBirth, DriverlicenseNo)

BOOKING (BookingID, StartDate, EndDate, NumOfPeople, TotalCharge, Status, *GuestID, RoomID*)

Foreign Key (GuestID) References GUEST

Foreign Key (RoomID) References BOOKING

ROOM (RoomID, MaxNumOfAccom, NumOfBed*, MotelID, BookingID, TypeID*)

Foreign Key (MotelID) References Motel

Foreign Key (RoomID) References Room

ROOMTYPE (TypeID, TypeDescription, PricePerNight)

MOTEL (MotelID, MotelName, Address, Telephone)

DIGITALMEDIA (MediaID, MediaName, WebAddress, Phone, ContactPerson)

ADVERTISEMENT (AdvertisementID, AdvertisedDate, Cost, *MotelID, MediaID*)

Foreign Key (MotelID) References Motel

Foreign Key (MediaID) References DIGITALMEDIA

EMPLOYEE (StaffID, FirstName, LastName, Address, Phone, Gender, StaffType)

CONTRACTOR (*StaffID*, TradeLicenseNo, SkillDescription)

Foreign Key (StaffID) References EMPLOYEE

ADMIN (*StaffID*, AnualSalary)

Foreign Key (StaffID) References EMPLOYEE

MANAGER (*StaffID*, Qualification, YearsOfExperience)

Foreign Key (StaffID) References EMPLOYEE

REPAIRJOB (JobID, *StaffID*, *RoomID*, JobCompletionDate, LabourCost, MaterialCost, JobDescription)

Foreign Key (StaffID) References EMPLOYEE

Foreign Key (RoomID) References ROOM

1. **Normalisation**

GuestID -> FirstName, LastName, Address, MobilePhone, DateOfBirth, DriverlicenseNo

* The relation satisfies 1NF because it does not contain repeating group. It has a primary key, GuestID. For each GuestID, there will be ONLY one last name and first name. Guest may have multiple phone numbers and addresses but only one of each is recorded. Every guest only has one date of birth, and the same applies to.
* The relation satisfies 2NF because it satisfied 1NF and does not have partial dependencies since candidate key is not composite. Driver license can be a candidate key; however, candidate key should be minimal and GuestID as candidate key is already sufficient.
* The relation satisfies 3NF because it satisfied 1NF and 2NF. Last name and first name cannot be used to determine any other attributes since different people can have similar lastname/firstname. The same can be said to other attributes, including driver license.

TypeID -> TypeDescription, PricePerNight

TypeDescription-> TypeID, PricePerNight

PricePerNight -> TypeID, TypeDescription

* The relation satisfies 1NF because it does not contain repeating group. It has a primary key, TypeID. For each TypeID, there will be ONLY one description and price.
* The relation satisfies 2NF because it satisfied 1NF and does not have partial dependencies since candidate key is not composite (TypeID is the only candidate key).
* The relation satisfies 3NF because it satisfied 1NF and 2NF. It does not contain transitive dependency. Type description and price per night are both functionally determined by Type ID.

**Implementation Report**

**I do need to spend quite a lot of time in creating logical database model (EER) in MySQL workbench. It was my first time translating conceptual model to logical model hence I encountered many issues, especially when trying forward engineering a logical database. Here, I encounter most of the issues relating to attributes and cardinalities. In a sense, the feature “forward engineering” does expose me to the imperfection and errors of model I created. Although this feature pretty much writing everything for me, however, I am able to understand the basic structure of a query thanks to this feature. In the end, I was able to successfully forward engineering my logical database. In addition, query number 6 took me the longest to get the desired outcome. Since It requires combination of almost everything I have learnt, it was great to pull together all the knowledge about COUNT, GROUPBY, MAX, HAVING and some of the related rules. I was exposed to what is working and what is not, and what is needed, therefore strengthen my understanding about query and syntax.**