ACCOLITE UNIVERSITY, JULY 2020

RDBMS Concepts

Assignment Submission

By: Ayush Malik

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

CustID	CustName	AccountManager	AccountManagerRoom	ContactName1	ContactName2
171	ABNAmro	Hans	12	Piet	Koos
190	RaboBank	Guus	15	Mona	Mieke

SOLUTION:

Functional Dependencies in the single table before normalization:

- CustID - > CustID, CustName, AccountManager, AccountManagerRoom, ContactName1, ContactName2
- AccountManager -> AccountManager, AccountManagerRoom

2 Tables will be formed after normalization to BCNF:

1. CUSTOMER

CustID	CustName	AccountManager	ContactName1	ContactName2

Functional Dependency

- CustID - > CustID, CustName, AccountManager, ContactName1, ContactName2

2. ACCOUNT_MANAGER

AccountManager	AccountManagerRoom

Functional Dependency

- AccountManager - > AccountManager, AccountManagerRoom

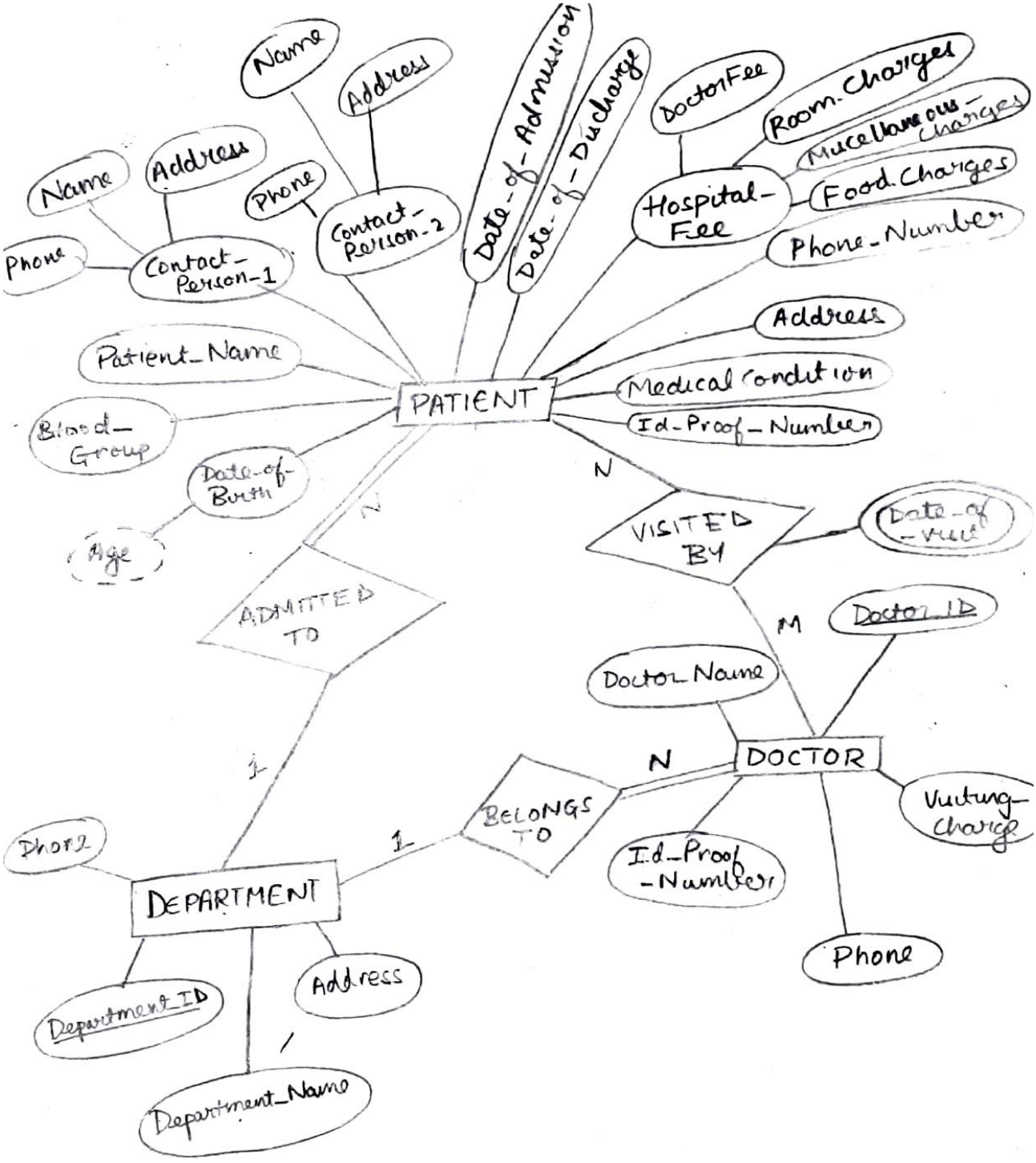
2. Draw an ER diagram for a hospital management system.

SOLUTION:

The following problem description and assumptions have been taken into consideration while designing the ER Model of the system.

PROBLEM DESCRIPTION AND ASSUMPTIONS:

- The hospital management system handles 3 types of records: Patient, Doctor and Department.
- Every time a patient is admitted to the hospital, a new record is created. Even if the same patient gets admitted multiple times, a different record for him/her will be created each time. Once admitted, the patient can be visited by different doctors any number of times before he/she is discharged.
- A patient must be admitted to only one department of the hospital.
- A doctor must belong to only one department of the hospital.
- A doctor can treat a patient from any department.
- A patient can be treated/visited by any number of doctors from any department.
- A patient's record consists of the following: patient id, name, identity proof number, contact person one, contact person two, date of admission, date of discharge, hospital charges, date of birth, department id, medical condition, phone number, blood group, the doctors he/she has been treated by along with the corresponding dates of visit/treatment, hospital fee.
    - Patient's id uniquely identifies each patient.
    - For both contact persons of the patient, the following information will be stored: name, phone number and address.
    - It is assumed that the visiting dates of the doctors will be between the date of admission and date of discharge of the patient (both inclusive).
    - Hospital Fee is further divided into the following types of fee: Doctor Fee, Room Charges, Medicine, Food and Miscellaneous.
    - It is assumed that the date of admission is always before the date of discharge or the same as the date of discharge.
    - A patient can be visited by the same doctor multiple times.
- A doctor's record consists of the following: doctor's id, name, identity proof number, department id, contact number and visiting charges.
    - A doctor's id uniquely identifies each doctor.
    - Visiting timings of the doctors have not been taken into consideration.
- A department's record consists of the following: department id, department name, phone and address.
    - A department's id uniquely identifies each department.
    - Visiting timings of the department/ hospital have not been taken into consideration.
- It is assumed that no photos are to be stored in the database.
- It is assumed that the addresses are stored as a simple string.


Name
Address
Phone
Contact_Person_2
Name
Address
Phone
Contact_Person_1
Date_of_Admission
Date_of_Discharge
DoctorFee
Room_Charges
Miscellaneous_charges
Food_Charges
Hospital_Fee
Phone_Number
Address
Medical condition
Id_Proof_Number
PATIENT
Patient_Name
Blood_Group
Date_of_Birth
Age
N
N
ADMITTED TO
1
VISITED BY
Date_of_visit
M
Doctor_ID
Doctor_Name
N
DOCTOR
Visiting_charge
BELONGS TO
1
Id_Proof_Number
Phone
Phone
DEPARTMENT
Department_ID
Address
Department_Name

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:

Functional Dependencies before normalization:

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

Note: ModuleID and ModuleName are multi-valued before normalization.

4 Tables will be formed after normalization to 3NF:

1. STUDENT

StudentID	StudentName	StudentAddress	TutorID

Functional Dependency
- StudentID - > StudentID, StudentName, StudentAddress, TutorID

2. TUTOR

TutorID	TutorName

Functional Dependency
- TutorID - > TutorID, TutorName

3. MODULE

ModuleID	ModuleName

Functional Dependency
- ModuleID - > ModuleID, ModuleName

4. STUDENT_MODULE

StudentID	ModuleID

Functional Dependency
- StudentID, ModuleID - > StudentID, ModuleID

Note: Here, (StudentID, ModuleID) together form the primary key.