

**CS 557 – Introduction to Database Systems
Fall 2015**

**Term-Project – Hospital Management System
Submission Deadline – Monday, December 15**

Total grade – 100 points

1 Project Description

The APOLLO HOSPITAL is a multi-facility hospital in the New York City.

The hospital has multiple departments and units such as Pathology, Emergency, Dental, Obstetrics and Gynecology, Cardiology, Gastroenterology, ICUs(Intensive Care Unit), Ear nose and throat (ENT), Orthopaedic, Neurology, Cardiology, Diagnostic imaging and so on. There is a Patient Welcome Unit where the visiting patients can obtain a card (that is, an entry card of that patient) for check up from a concerned doctor. After making an entry in the card, they visit the concerned doctor in their room. Depending on the situation, the doctor can either prescribe medicine or admit the patients in the respective departments. The patient may choose either private or general room according to their needs. But before getting an admission in the hospital, the patient has to fulfil certain formalities for instance clearing certain bills for : room charges, meal charges and others. After completion of the treatment, the doctor releases the patient. Prior to the release, the patient has to complete certain formalities again, such as full payments of any unpaid bills, insurance paperworks and some other issues (if any).

There are primarily two classes of doctors in the hospital, viz., regular doctors and call-on doctors. Regular doctors are those doctors who come to the hospital daily. Call-on doctors are those doctors who are requested by the hospital to attend some patients if the concerned doctor is unavailable.

The aim of this mini-project is to design and develop a database for the aforementioned hospital to maintain the records of various departments, rooms, and doctors in the hospital. The database must also maintain records of the outdoor patients, patients admitted in the hospital, the medical records of the patients by the doctors, the patients who have been operated and the patients who are already released from the hospital.

2 Database Design

The database should include the following entities.

➤ **Patient** : It keeps track of all the details about both the admitted patients and the outdoor patients. A unique ID is generated for each patient after registration. Patient ID, patient's name, patient's address, admission date, doctor's name, treatment details, room number, room type etc. are to be recorded. Also particular patient details can be viewed in the table using a separate form with an attribute patient id, doctor's name and admission date.

➤ **Admission** : This records the basic patient related information, which is registered when the patient visits the hospital for the first time. Each patient is allocated with an unique patient identification number. It should also record details of all the formalities to be fulfilled by the patient.

➤ **Outdoor–Patient** : This manages activities related to a patient who visits the Hospital Resident Doctor or Call-on Doctor for medical consultations, diagnosis and treatment.

➤ **Staff** : It keeps track of all the details about doctors and other staff members of the hospital. Staff member's name, designation, staff ID, address, qualification, cell no, e-mail are recorded.

➤ **Doctor** : An entity for the doctors is also required to record the details such as patients attended by a doctor, tests conducted, doctor ID, department and so on.

➤ **Billing** : This keeps track of the bills of both the admitted patients as well as outdoor patients who come to the hospital. Prior to the release from the hospital, every patient needs to complete certain formalities of the hospital such as payment of bills (if any), test charges, operation charges (if any), doctors' charges, etc. These charges are required to be recorded in the billing details.

➤ **Department** : This keeps track of department details for each patient. Patient id, department name, doctor and other related attributed are recorded here.

☛ You may add attributes other than those mentioned above which you think is required to be recorded. Apart from these, you may also include any other entity/table which you think is necessary/relevant to your approach.

3 SQL Queries

1. Create the tables relevant to your approach of design. Be precise while defining the domains, primary keys and the foreign keys in each relation.
2. Display the details of a patient named John Brown and also display the name of doctor(s) who diagnosed him and also his total bill amount.
3. Display the total number of patients diagnosed by Kate Smith on January 1st 2009.
4. Display the year which has the maximum number of patients.

5. Display the details of a patient and the supervising doctor who was admitted more than once during the year 2010.
6. Display details of the department/units the doctors of which have billed the maximum amounts.

4 Objectives

1. Construct the ER diagram and design the relations relevant to your approach. Please specify each attribute to avoid any kind of ambiguity in the design. Also, describe briefly your idea behind the database design. Be specific and clear about the primary and foreign keys. You may consider some assumptions during the design process, but make sure to mention them clearly in the project report.
2. For each relation that have been defined in the above step, write down an appropriate set of functional dependencies. The set should abide by the project specifications. Based on the set of functional dependencies defined, normalize your relations to BCNF. You should show the steps of normalization.
3. Write down the SQL statements for the given SQL queries.

5 Deliverables

Please upload in **D2L** a project report **PDF** containing all the materials instructed in the last section to present. You must execute the SQL statements for the above mentioned queries in the following website,

http://sqlzoo.net/wiki/Main_Page.

For each execution of the SQL statements, take a **screenshot** and include in the report PDF.

 *Failure to include appropriate screenshots would result in point deduction.*