NSBM
NATIONAL SCHOOL OF BUSINESS MANAGEMENT

IN PARTNERSHIP WITH PLYMOUTH UNIVERSITY

Databases – ISAD253SL

Coursework
2016 – 2017

Term: Term 1

Submission Deadline: To be confirmed

Coursework Type: Group assignment

Element of Assessment: C1 & P1

Module Leader: Ms. Dileeka Alwis

E-mail: dileeka@nsbm.lk

ISAD253SL - Databases

Coursework 2016/2017

Case Study: Hospital Management System

City Central hospital is a multi-specialty hospital that includes a number of departments, rooms, doctors, nurses, attendants, and other staff working in the hospital. Patients having different kinds of ailments come to the hospital and get check-up done from the concerned doctors. If required they are admitted in the hospital and discharged after treatments. In the hospital, there are many departments like Orthopedic, Pathology, Emergency, Dental, Gynecology, Anesthetics, I.C.U., Blood Bank, Operation Theater, Laboratory, M.R.I., Neurology, Cardiology, Cancer Department, Corpse, etc. Details about various departments in the hospital should be recorded within the database of the hospital management system such as department name, department location, facilities available in that department etc. Department name should be unique for each department.

There is an OPD where patients come and get a card (that is, entry card of the patient) for check up from the concerned doctor. Information about patients are collected when any patient arrives in the hospital for a check-up. The patient number which acts as a unique code begins with a prefix 'PT' should be generated when patient arrives. Other details like name, age, gender, address, data of birth, phone number, entry date, doctor referred to, diagnosis, and department is also gathered and should save in the database. Gender should be marked as 'M' or 'F' only. Doctor's details and department referred must exist within the database prior registering a patient. After collecting the necessary details, the entry card is generated and the patient is sent to the concerned doctor's room for check-up their ailments. According to the ailments, the doctor either prescribes medicine or admits the patient in the concerned department.

The details about the patients who get treatment from the doctor referred to should also store in the database. Details like patient number, doctor number, date of check-up, diagnosis, treatment, etc. should store. Status should be another field which is used to indicate whether the patient is admitted, referred for operation or prescribed medicine to a regular patient to the hospital. If the patient is admitted, or referred for an operation, or even a regular patient to the hospital, the further details should store separately within the database with reference to basic patient information.

When admitting, the patient may choose either a private or a general room according to his/her need.
But before getting admission in the hospital, the patient has to fulfil certain formalities of the hospital like room charges, etc. When a patient is admitted, information like patient number, advance payment, mode

of payment, room number, department, date of admission, initial condition, diagnosis, treatment, number of the doctor under whom treatment is done, gradient details and contact information, etc should be tracked. Patient number, department, doctor number, room number must be valid.

After the treatment is completed, the doctor discharges the patient. Before discharging from the hospital, the patient again has to complete certain formalities of the hospital like balance charges, test charges, operation charges (if any), blood charges, doctors' charges, etc. Whenever a patient gets discharged from the hospital an entry should record in the database which includes details like patient number, treatment given, treatment advice, payment made, mode of payment, date of discharge, etc.

Details of regular patients should also store in the database which includes the information like date of visit, diagnosis, treatment, medicine recommended, status of treatment, etc. There can be multiple entries of one patient as patient might be visiting hospital repeatedly for check-ups and there will be entry for patient's each visit. If patient is operated in the hospital, his/her details should also store in the database like patient number, date of admission, date of operation, number of the doctor who conducted the operation, number of the operation theatre in which operation was carried out, type of operation, patient's condition before and after operation, treatment advice, etc.

There are two types of the doctors in the hospital, namely, regular doctors and call on doctors. Regular doctors are those doctors who work in the hospital and come to the hospital daily. Calls on doctors are those doctors who are called by the hospital if the concerned doctor is not available or when additional doctors are required. Information about all the doctors working for the hospital and the departments they are associated with should be recorded in the database. Each doctor should be given a unique doctor number which is an identity number starting with 'DR' or 'DC' prefixes only. The corresponding department should exist in the database before registering a doctor to that particular department. The system database should store personal details of all the doctors like name, qualification, address, phone number, etc. The system should record special information about regular doctors like pre-defined salary, date of joining, etc while recording special information about calls on doctors like fees per call, payment due, etc.

The details of all rooms in the hospital should also record in the database like room number, room type (general or private), status (whether occupied or not), if occupied, then patient number, patient name, charges per day, etc. Room number should be unique and room type can only be 'G' or 'P' and status can only be 'Y' or 'N'.

Basic details about medicines (drugs), special treatments and operations should also save in the database separately which can be prescribed to patients. And also the database should track all the details about customer payments whenever they prescribe medicine, undergo operations, admit in the hospital and discharge from the hospital etc. Details of all the other hospital staff including nurses, attendants, laboratory staff, etc. should save in the database with their basic information like personal details, qualification and experience information, department which they are attached with etc.

1.1. Tasks

The aim of this case study is to design and develop a database for the hospital to maintain the records of various departments, rooms, doctors and other staff in the hospital, medicines used in the hospital, special treatments and operations conducted by the hospital etc. It also maintains records of the regular patients, patients admitted in the hospital, the check-up of patients done by the doctors, the patients that have been operated, and patients discharged from the hospital etc.

Analyze the above scenario and implement the resulting tables in Microsoft SQL Server including as much data integrity and data validation as possible. You are strongly recommended to consider the use of constraints and triggers.

Design and implement an appropriate database application to facilitate the followings.

1. Allow the administrative staff of the hospital to enter and maintain details about all the hospital activities regarding daily operations. Your application should include appropriate **Data Validation** mechanisms (constraints), **Triggers** and **User Defined Functions**.

2. Allow authorized parties of the hospital to view details about the daily activities in order to generate meaningful management reports to make strategic, long term and short term managerial decisions about the hospital activities. Your application should include appropriate **Views** and **Stored Procedures** to retrieve operational data.

As a stretch goal, you may wish to consider incorporating user logins and subsequent direction to a page according to the user's role. However, this is 'icing on the cake' and you should not attempt it until the rest of your application has a high level of functionality.

1.2. Groups

This is a group assignment. Project teams should ideally comprise 5 students, although exceptionally other sizes will be permitted by negotiation with the module leader. No allowance for group size will be made in the assessment of the work.

The module leader must be notified of group membership on the Second day of lectures. Any student not in a group by this deadline may be assigned to a group by the module leader.

1.3. Deliverables

Deliverable 1

You should submit **a document (both softcopy and hard copy)** containing:

Section 1:

- A basic introduction to the scenario with the important facts you identified and considered to your solution. (Do not copy and paste the given scenario as it is)
- An Entity Relationship (ER) or Extended Entity Relationship (EER) Diagram showing all of the entities, their attributes, relationships, cardinality rations and the participation constraints. (Should include a sensibly resized diagram which clearly show all the elements)
- A list of any additional assumptions you have made which affect your solution.
- Relational Mapping (Have to clearly indicate the steps of relational mapping with all table attributes, primary keys and foreign keys)
- Data Normalisation (Have to clearly indicate all the steps of up data normalisation up to Third normalization form).
- Data Dictionary of each normalised table. (Should contain all the details about each table field)

Section 2:

- Microsoft SQL Server **Create Table** statements with related **Constraints** for each table to validate data. (Should provide sensibly resized screen shots all the table creation statements which clearly show all the SQL statements)
- Database Diagram of your solution. (Should include a sensibly resized diagram which clearly show all the elements)
- A set of relevant and sensibly sized screen shots showing all the tables in your application with some meaningful sample records. (Should insert more than 10 meaningful sample records to each table in your database).

Section 3:

- Microsoft SQL Server **Create Trigger** statements for the triggers that you have created. (Should create at least two triggers for your database and should provide sensibly resized screen shots SQL statements)
- Microsoft SQL Server **Create Function** statements for the user defined functions that you have created. (Should create at least two user defined functions for your database and should provide sensibly resized screen shots SQL statements)
- Microsoft SQL Server **Create View** statements for the database views that you have created. (Should create at least two database views for your database and should provide sensibly resized screen shots SQL statements)
- Microsoft SQL Server **Create Procedure** statements for the stored procedures that you have created. (Should create at least two stored procedures for your database and should provide sensibly resized screen shots SQL statements)

Section 4:

- A critical appraisal of your solution highlighting worthy features, together with any shortcomings and how they might be resolved.
- Comments on further implementation of your application.
- **Work Load Matrix** to represent the individual contribution given by each group member by clearly mentioning who did what.
- **Peer Review Form** to evaluate group members by your own as a group.

Note that you should include all the SQL queries you have created within your database including Data Definition Language (DDL) and Data Manipulation language (DML) for tables, triggers, views, stored procedures and user defined functions. You should provide sensibly resized screen shots to show all the SQL statements within the database that you have created.

Deliverable 2

A complete backup of your database (.bak) with a softcopy of Deliverable 1 (.docx or .pdf) and the Presentation slide show (.pptx) written to a CD.

Note that any part of your submission in an incorrect file format cannot be marked.
Coursework may be submitted at any time ahead of the deadline time.
Please note the University regulations concerning late submission of coursework.
Please note that the late submissions of project deliverables will not be assessed.

Deliverable 3

Your group will be required to perform a 10-minute presentation of your solution followed by a 15-minute question and answering (viva) session based on your individual performance according to the work load matrix.

Note that the database objects and applications may be accessed in your absence as part of the assessment process.

You should ensure that your tables are populated with an adequate amount of sensible test data in advance of this session and it is particularly important that date and time-dependent data is applicable to the date and time of your solution.

Unless it is unavoidable, all group members should be present at this session.

Submission deadline: To be confirmed

All groups should submit Deliverable 1 and Deliverable 2 on or before the deadline to the program office before 4.00 pm.

The presentation and viva will be taken according to a schedule prepared by NSBM. Students will be informed about the schedule prior to the submission.

1.4. Assessment

The assignment assesses the Learning Outcomes of the module, viz. to

For ISAD253SL

1. Write effective SQL statements for defining, manipulating and controlling data.
2. design and implement a multi-user database application