DBS211 – Final Project – 15%

Group 04:

- Alejandro Mercado Monetti  
- Renan De Alencar Queiroz  
- Seulgi Lee

## Part 1 : Project Idea and Proposal

The first step is to choose a topic and get a basic idea of the requirements for the database. Each group will submit a word document, with cover page, that outlines their topic. The document should include the following sections:

#### Introduction

The Hospital Management System (HMS) is designed to streamline and automate various hospital operations, ensuring efficient management of patient information, doctor schedules, appointments, medical records, and billing. This system addresses the need for a comprehensive solution that integrates all critical hospital processes into a unified database.

#### Problem Statement

Hospitals face significant challenges in managing large volumes of data related to patients, appointments, doctor schedules, medical records, and billing. Traditional paper-based systems or disjointed digital systems can lead to inefficiencies, errors, and delays in patient care. There is a need for a robust system that can handle these operations efficiently and provide accurate, real-time information to support decision-making and improve patient outcomes.

*Solution*

The HMS provides a centralized database system to manage hospital operations. The system includes:

- Patient Management: Registration and management of patient information.

- Appointment Scheduling: Efficient scheduling and management of appointments with doctors.

- Doctor Management: Tracking doctor availability, specialties, and schedules.

- Billing: Handling billing information and payment statuses.

- Medical Records: Storing patient medical history and treatment records.

The database is designed with tables for patients, doctors, appointments, medical records, and billing, along with views for upcoming appointments, patient treatment history, and billing summaries.

*Requirements*

- Hardware and Software:

- Server to host the database and application.

- Computers or tablets for hospital staff to access the system.

- Database management system (e.g., MySQL, SQL Server).

- Functional Requirements:

- Ability to register and manage patient information.

- Functionality to schedule and manage appointments.

- Capability to track doctor schedules and availability.

- System to handle billing and track payment statuses.

- Storage of medical records with secure access.

- Non-Functional Requirements:

- Data security and privacy to protect patient information.

- High availability and reliability to ensure continuous operation.

- Scalability to handle increasing data volumes and users.

- User-friendly interface for easy access and management by hospital staff.

The HMS aims to improve operational efficiency, reduce errors, and enhance patient care by providing a reliable and integrated management system for hospitals.

## Part 2 : Database Design (10 Points)

This milestone will include the design component of the project. Groups will progress through the data modelling and normalization processes in order to finalize a database design. Working within their group and in consultation with your professor, students will practice the processes learned in class to design a relational database.

**ERD**: Groups will create a UML Entity Relationship Diagram of their database design following the database modelling process. This design should then be checked to be compliant 3rd normal form through the normalization techniques learned in class.

A diagram of a patient schedule

Description automatically generated with medium confidence

**Data Dictionary**: Groups will produce a data dictionary for **each** of their designed tables. This data dictionary will include the same information as the ERD in table format in addition to data types, sizes, notes, and example data. A sample table is included below.

TABLE: **Employees (SAMPLE)**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Column** | **Data Type** | **Size, Precision** | **Default** | **PK/FK** | **Required** | **Range** | **Sample Data** | **Notes** |
| EmployeeID | NUMBER | 4 |  | PK | Y | 1-9999 | 1234 | Autonumbered identity |
| firstName | String | 25 |  |  | Y |  | “Bob” |  |
| lastName | String | 25 |  |  | Y |  | “McKenzie” |  |
| phone | NUMBER | 11 |  |  | Y | 2000000000-9999999999 | 9055551212 | Assuming North American phone number |
| balanceOwing | NUMBER | 9,2 | 0.00 |  | Y | -10000 to 10000 | 345.65 | Monetary value |
| DOB | DATE |  |  |  | Y |  | 1972/05/16 | Date of Birth (YYYY/MM/DD) |

-----------------------------------------------------------------------------------------------------------------------------------------

**<Database for a hospital>**

TABLE**: Patients**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Column** | **Data Type** | **Size/Precision** | **Default** | **PK/FK** | **Required** | **Range** | **Sample Data** | **Notes** |
| **PatientID** | NUMBER | 5 |  | PK | Yes | 1-99999 | 12345 | Autonumbered identity |
| FirstName | VARCHAR2 | 25 |  |  | Yes |  | "Bob" | Patient's first name |
| LastName | VARCHAR2 | 25 |  |  | Yes |  | "McKenzie" | Patient's last name |
| DOB | DATE |  |  |  | Yes |  | 1972/05/16 | Date of Birth (YYYY/MM/DD) |
| Gender | CHAR | 1 |  |  | Yes | M/F | M | Gender |
| ContactNumber | NUMBER | 11 |  |  | Yes | 2000000000-9999999999 | 4164915050 | Contact number |
| Email | VARCHAR2 | 100 |  |  | No |  | slee588@myseneca.ca | Email address |
| Address | VARCHAR2 | 255 |  |  | No |  | 1750 Finch Ave E | Address |
| EmergencyContact | NUMBER | 11 |  |  | No | 2000000000-9999999999 | 4164915050 | Emergency contact |
| InsuranceDetails | VARCHAR2 | 255 |  |  | No |  | Morecare Insurance | Insurance details |

TABLE**: Doctors**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Column** | **Data Type** | **Size/Precision** | **Default** | **PK/FK** | **Required** | **Range** | **Sample Data** | **Notes** |
| **DoctorID** | NUMBER | 5 |  | PK | Yes | 1-99999 | 12345 | Autonumbered identity |
| FirstName | VARCHAR2 | 25 |  |  | Yes |  | "Alice" | Doctor's first name |
| LastName | VARCHAR2 | 25 |  |  | Yes |  | "Smith" | Doctor's last name |
| Specialty | VARCHAR2 | 100 |  |  | Yes |  | Cardiology | Specialty |
| ContactNumber | NUMBER | 11 |  |  | Yes | 2000000000-9999999999 | 4164915050 | Contact number |
| Email | VARCHAR2 | 100 |  |  | No |  | alice@clinic.com | Email address |
| OfficeNumber | VARCHAR2 | 10 |  |  | No |  | 101 | Office number |

TABLE**: Appointments**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Column** | **Data Type** | **Size/Precision** | **Default** | **PK/FK** | **Required** | **Range** | **Sample Data** | **Notes** |
| **AppointmentID** | NUMBER | 5 |  | PK | Yes | 1-99999 | 12345 | Autonumbered identity |
| **PatientID** | NUMBER | 5 |  | FK | Yes | 1-99999 | 12345 | Patient ID |
| **DoctorID** | NUMBER | 5 |  | FK | Yes | 1-99999 | 12345 | Doctor ID |
| AppointmentDate | DATE |  |  |  | Yes |  | 2024/08/05 | Appointment date (YYYY/MM/DD) |
| AppointmentTime | TIMESTAMP |  |  |  | Yes |  | 10:00:00 | Appointment time |
| ReasonForVisit | VARCHAR2 | 255 |  |  | Yes |  | Check-up | Reason for visit |
| Status | VARCHAR2 | 20 |  |  | Yes |  | Scheduled, Completed, Cancelled | Appointment status |

TABLE**: Doctors\_Schedules**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Column** | **Data Type** | **Size/Precision** | **Default** | **PK/FK** | **Required** | **Range** | **Sample Data** | **Notes** |
| **ScheduleID** | NUMBER | 5 |  | PK | Yes | 1-99999 | 12345 | Autonumbered identity |
| **DoctorID** | NUMBER | 5 |  | FK | Yes | 1-99999 | 12345 | Doctor ID |
| DayOfWeek | VARCHAR2 | 10 |  |  | Yes |  | Monday | Day of the week |
| StartTime | TIMESTAMP |  |  |  | Yes |  | 09:00:00 | Start time |
| EndTime | TIMESTAMP |  |  |  | Yes |  | 17:00:00 | End time |
| Availability | CHAR | 1 | 'N' |  | Yes | Y/N | Y | Availability |

TABLE**: Billing**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Column** | **Data Type** | **Size/Precision** | **Default** | **PK/FK** | **Required** | **Range** | **Sample Data** | **Notes** |
| **BillingID** | NUMBER | 5 |  | PK | Yes | 1-99999 | 12345 | Autonumbered identity |
| **PatientID** | NUMBER | 5 |  | FK | Yes | 1-99999 | 12345 | Patient ID |
| **AppointmentID** | NUMBER | 5 |  | FK | Yes | 1-99999 | 12345 | Appointment ID |
| TotalAmount | NUMBER | 10, 2 | 0.00 |  | Yes |  | 200.00 | Total amount |
| AmountPaid | NUMBER | 10, 2 | 0.00 |  | No |  | 100.00 | Amount paid |
| PaymentStatus | VARCHAR2 | 20 |  |  | Yes |  | Unpaid, Paid, Refunded | Payment status |
| BillingDate | DATE |  |  |  | Yes |  | 2024/08/05 | Billing date (YYYY/MM/DD) |

TABLE**: Records**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Column** | **Data Type** | **Size/Precision** | **Default** | **PK/FK** | **Required** | **Range** | **Sample Data** | **Notes** |
| **RecordID** | NUMBER | 5 |  | PK | Yes | 1-99999 | 12345 | Autonumbered identity |
| **PatientID** | NUMBER | 5 |  | FK | Yes | 1-99999 | 12345 | Patient ID |
| **DoctorID** | NUMBER | 5 |  | FK | Yes | 1-99999 | 12345 | Doctor ID |
| VisitDate | DATE |  |  |  | Yes |  | 2024/08/05 | Visit date (YYYY/MM/DD) |
| Diagnosis | VARCHAR2 | 255 |  |  | Yes |  | Hypertension | Diagnosis |
| Treatment | VARCHAR2 | 255 |  |  | No |  | Medication | Treatment |
| Notes | VARCHAR2 | 4000 |  |  | No |  | Follow-up in 2 weeks | Doctor's notes |

**Database IMPLEMENTATION**

In your SQL file, write the SQL statements to:

1. Create a database named DBS211\_P2\_YourGroupNo.
2. Create the tables of your relational database model. Specify the PK, FK, and other necessary constraints.
3. Show a screenshot of the database relationships of all the tables you have created.

**<database relationship>**

Table : **Appointments**

A diagram of a data flow

Description automatically generated with medium confidence

Table : **Billing**

A diagram of a computer

Description automatically generated with medium confidence

Table : **Doctors**

A diagram of a medical data flow

Description automatically generated with medium confidence

Table : **Doctors\_Schedules**

A screenshot of a computer

Description automatically generated

Table : **Patients**

A diagram of a patient

Description automatically generated

Table : **Records**

A diagram of a data flow

Description automatically generated

## Part 3 Database Scripting and Creation along with connectivity in visual studio 2022

## (10 Points)

Due – End of Week 13

This will be the final submission for the project and include everything previously completed, with corrections based on feedback, and include the scripts to create the database, populate the database with data, and produce some views to act as data reports for extracting information from the database for the business purposes.

### Deliverables

Groups will produce each of the following:

* **Creation Script**: A *single* SQL file providing the scripting to create all the tables, including all constraints and features as designed in the ERD and data dictionary.

A screenshot of a computer code

Description automatically generated

A screenshot of a computer

Description automatically generated

A computer code with text

Description automatically generated with medium confidence

A screenshot of a computer code

Description automatically generated

* A screenshot of a computer

  Description automatically generated  
  A screenshot of a computer

  Description automatically generated
* **Sample Data Script:** A single SQL file providing the scripting to insert sample data into all tables. The amount of data should be enough to allow adequate testing of an application build upon the designed database. Rule of Thumb: 20-30 rows per data tables and their associated bridge tables, and an appropriate number of rows in lookup tables.

A screenshot of a computer

Description automatically generated

* **Business Reports:** A single SQL file providing the scripting to create at least 4 VIEWS, that provide a report on the data that support the business, or organization, in making informed business decisions. Each report should have a paragraph writeup – in comment form, that explain the purpose of the report and how the business will benefit from having the report.

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

A white background with black text

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

A close up of a list of different types of text

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

A white paper with black text

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

A white paper with black text

Description automatically generated

NOTE: All scripts should execute in their entirety without error for both the scenario where no existing tables or database objects exist, or where the scripts are being run on a database where the objects already exist, and the idea is that they are being overwritten.

**<Visual Studio connectivity>**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

### A screenshot of a computer screen Description automatically generated

A screen shot of a computer

Description automatically generated

### (Final) Submission

Groups will submit 4 documents

1. Main word document (with updates and changes made to reflect feedback and changes in the design made throughout the process)
2. Power Point Presentation – To Describe contribution of group Members / Problem statement / Project Description.
3. SQL Script File.
4. Business Reports – ER Diagram of each table, Normalization (3NF), screenshots of database connectivity and retrieve information from visual studio .
5. Create own Scenarios – Execute in SQL