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

**<SQL>**

-- CREATE TABLE

-- Patients

CREATE TABLE Patients (

PatientID NUMBER(5) PRIMARY KEY,

FirstName VARCHAR2(25) NOT NULL,

LastName VARCHAR2(25) NOT NULL,

DOB DATE NOT NULL,

Gender CHAR(1) CHECK (Gender IN ('M', 'F')) NOT NULL,

ContactNumber NUMBER(11) NOT NULL,

Email VARCHAR2(100),

Address VARCHAR2(255),

EmergencyContact NUMBER(11),

InsuranceDetails VARCHAR2(255)

);

-- Doctors

CREATE TABLE Doctors (

DoctorID NUMBER(5) PRIMARY KEY,

FirstName VARCHAR2(25) NOT NULL,

LastName VARCHAR2(25) NOT NULL,

Specialty VARCHAR2(100) NOT NULL,

ContactNumber NUMBER(11) NOT NULL,

Email VARCHAR2(100),

OfficeNumber VARCHAR2(10)

);

-- Appointments

CREATE TABLE Appointments (

AppointmentID NUMBER(5) PRIMARY KEY,

PatientID NUMBER(5) REFERENCES Patients(PatientID),

DoctorID NUMBER(5) REFERENCES Doctors(DoctorID),

AppointmentDate DATE NOT NULL,

AppointmentTime TIMESTAMP NOT NULL,

ReasonForVisit VARCHAR2(255) NOT NULL,

Status VARCHAR2(20) CHECK (Status IN ('Scheduled', 'Completed', 'Cancelled')) NOT NULL

);

-- Doctors\_Schedules

CREATE TABLE Doctors\_Schedules (

ScheduleID NUMBER(5) PRIMARY KEY,

DoctorID NUMBER(5) REFERENCES Doctors(DoctorID),

DayOfWeek VARCHAR2(10) NOT NULL,

StartTime TIMESTAMP NOT NULL,

EndTime TIMESTAMP NOT NULL,

Availability CHAR(1) DEFAULT 'N' CHECK (Availability IN ('Y', 'N')) NOT NULL

);

-- Billing

CREATE TABLE Billing (

BillingID NUMBER(5) PRIMARY KEY,

PatientID NUMBER(5) REFERENCES Patients(PatientID),

AppointmentID NUMBER(5) REFERENCES Appointments(AppointmentID),

TotalAmount NUMBER(10, 2) DEFAULT 0.00 NOT NULL,

AmountPaid NUMBER(10, 2) DEFAULT 0.00,

PaymentStatus VARCHAR2(20) CHECK (PaymentStatus IN ('Unpaid', 'Paid', 'Refunded')) NOT NULL,

BillingDate DATE NOT NULL

);

-- Records

CREATE TABLE Records (

RecordID NUMBER(5) PRIMARY KEY,

PatientID NUMBER(5) REFERENCES Patients(PatientID),

DoctorID NUMBER(5) REFERENCES Doctors(DoctorID),

VisitDate DATE NOT NULL,

Diagnosis VARCHAR2(255) NOT NULL,

Treatment VARCHAR2(255),

Notes VARCHAR2(4000)

);

-- INSERT SAMPLE DATA

-- Patients

INSERT INTO Patients (PatientID, FirstName, LastName, DOB, Gender, ContactNumber, Email, Address, EmergencyContact, InsuranceDetails)

VALUES (12345, 'Bob', 'McKenzie', DATE '1972-05-16', 'M', 4164915050, 'slee588@myseneca.ca', '1750 Finch Ave E', 4164915050, 'Morecare Insurance');

-- Doctors

INSERT INTO Doctors (DoctorID, FirstName, LastName, Specialty, ContactNumber, Email, OfficeNumber)

VALUES (12345, 'Alice', 'Smith', 'Cardiology', 4164915050, 'alice@clinic.com', '101');

-- Appointments

INSERT INTO Appointments (AppointmentID, PatientID, DoctorID, AppointmentDate, AppointmentTime, ReasonForVisit, Status)

VALUES (12345, 12345, 12345, DATE '2024-08-05', TIMESTAMP '2024-08-05 10:00:00', 'Check-up', 'Scheduled');

-- Doctors\_Schedules

INSERT INTO Doctors\_Schedules (ScheduleID, DoctorID, DayOfWeek, StartTime, EndTime, Availability)

VALUES (12345, 12345, 'Monday', TO\_TIMESTAMP('09:00:00', 'HH24:MI:SS'), TO\_TIMESTAMP('17:00:00', 'HH24:MI:SS'), 'Y');

-- Billing

INSERT INTO Billing (BillingID, PatientID, AppointmentID, TotalAmount, AmountPaid, PaymentStatus, BillingDate)

VALUES (12345, 12345, 12345, 200.00, 100.00, 'Unpaid', DATE '2024-08-05');

-- Records

INSERT INTO Records (RecordID, PatientID, DoctorID, VisitDate, Diagnosis, Treatment, Notes)

VALUES (12345, 12345, 12345, DATE '2024-08-05', 'Hypertension', 'Medication', 'Follow-up in 2 weeks');

-- SELECT TABLE

SELECT \* FROM Patients;

SELECT \* FROM Doctors;

SELECT \* FROM Appointments;

SELECT \* FROM Doctors\_Schedules;

SELECT \* FROM Billing;

SELECT \* FROM Records;