Hospital Management System Database Documentation

Overview

This document provides a detailed description of the SQL scripts used to create and manage the `HospitalManagementSystem` database. This includes the creation of the database, tables, and constraints, as well as some Data Query Language (DQL) examples to retrieve specific information from the database.

Database Creation

The following SQL script creates the `HospitalManagementSystem` database and specifies the physical location and size properties for the database files.

```
USE master;
CREATE DATABASE HospitalManagementSystem
ON (
    NAME = HospitalManagementSystem_data,
    FILENAME = 'D:\Edu\Courses\DB\Apps\Hospital Management System.mdf',
    SIZE = 10MB,
    MAXSIZE = 100MB,
    FILEGROWTH = 10MB
)
LOG ON (
    NAME = HospitalManagementSystem_log,
    FILENAME = 'D:\Edu\Courses\DB\Apps\Hospital Management System.ldf',
    SIZE = 10MB,
    MAXSIZE = 100MB,
    FILEGROWTH = 10MB
);
```

Tables and Constraints

Patients Table

The `Patients` table stores information about patients.

```
CREATE TABLE Patients (
    PatientID BIGINT PRIMARY KEY,
    FirstName VARCHAR(20),
```

```
    LastName VARCHAR(20),
    DOB DATE,
    Gender CHAR,
    ContactNumber VARCHAR(14),
    Address VARCHAR(100),
    Email VARCHAR(50),
    EmergencyContact VARCHAR(14),
    MedicalHistory VARCHAR(200)
);
```

Doctors Table

The `Doctors` table stores information about doctors.

```
CREATE TABLE Doctors (
    DoctorID BIGINT PRIMARY KEY,
    FirstName VARCHAR(20),
    LastName VARCHAR(20),
    Specialization VARCHAR(30),
    ContactNumber VARCHAR(14),
    Email VARCHAR(50),
    DepartmentID BIGINT,
    HireDate DATE,
    CONSTRAINT Dr_Dp_FK FOREIGN KEY (DepartmentID)
        REFERENCES Departments (DepartmentID)
);
```

Departments Table

The `Departments` table stores information about hospital departments.

```
CREATE TABLE Departments (
    DepartmentID BIGINT PRIMARY KEY,
    DepartmentName VARCHAR(30),
    HeadOfDepartment VARCHAR(50)
);
```

Appointments Table

The `Appointments` table stores information about appointments.

```
CREATE TABLE Appointments (
    AppointmentID BIGINT PRIMARY KEY,
```

```
    PatientID BIGINT,
    DoctorID BIGINT,
    AppointmentDate DATE,
    AppointmentTime TIME,
    ReasonForVisit VARCHAR(15),
    Status VARCHAR(40),
    CONSTRAINT PT_App_FK FOREIGN KEY (PatientID) REFERENCES Patients
(PatientID),
    CONSTRAINT Dr_App_FK FOREIGN KEY (DoctorID) REFERENCES Doctors
(DoctorID)
);
ALTER TABLE Appointments
ALTER COLUMN ReasonForVisit VARCHAR(50);
```

Staff Table

The `Staff` table stores information about hospital staff.

```
CREATE TABLE Staff (
    StaffID BIGINT,
    FirstName VARCHAR(20),
    LastName VARCHAR(20),
    Position VARCHAR(20),
    DepartmentID BIGINT,
    ContactNumber VARCHAR(14),
    Email VARCHAR(30),
    HireDate DATE,
    CONSTRAINT St_Dp_FK FOREIGN KEY (DepartmentID)
        REFERENCES Departments (DepartmentID)
);
ALTER TABLE Staff
ALTER COLUMN StaffID BIGINT NOT NULL;
ALTER TABLE Staff
ADD CONSTRAINT ST_PK PRIMARY KEY (StaffID);
```

Billing Table

The `Billing` table stores information about billing.

```
CREATE TABLE Billing (
    BillID BIGINT PRIMARY KEY,
    PatientID BIGINT,
    AppointmentID BIGINT,
    Amount BIGINT,
    BillDate DATE,
```

```
    PaymentStatus VARCHAR(5) CHECK (PaymentStatus IN ('DONE', 'NOT')),
    CONSTRAINT Bi_Pt_FK FOREIGN KEY (PatientID) REFERENCES Patients
(PatientID),
    CONSTRAINT Bi_Ap_FK FOREIGN KEY (AppointmentID) REFERENCES Appointments
(AppointmentID)
);
ALTER TABLE Billing
ALTER COLUMN PaymentStatus VARCHAR(6);
ALTER TABLE Billing
ADD CONSTRAINT Ck_payment
CHECK (PaymentStatus IN ('Paid', 'Unpaid'));
```

MedicalRecords Table

The `MedicalRecords` table stores information about medical records.

```
CREATE TABLE MedicalRecords (
    RecordID BIGINT PRIMARY KEY,
    PatientID BIGINT,
    DoctorID BIGINT,
    VisitDate DATE,
    Diagnosis VARCHAR(100),
    Treatment VARCHAR(100),
    Prescription VARCHAR(500),
    CONSTRAINT PT_Mr_FK FOREIGN KEY (PatientID) REFERENCES Patients
(PatientID),
    CONSTRAINT Dr_Mr_FK FOREIGN KEY (DoctorID) REFERENCES Doctors (DoctorID)
);
```

Data Query Language (DQL) Examples

Retrieve all appointments for a specific doctor on a given date

```
SELECT * FROM Appointments
WHERE DoctorID = @DoctorID AND AppointmentDate = @AppointmentDate;
```

List all patients who have visited in the past month

```
SELECT * FROM Patients
WHERE PatientID IN (
    SELECT PatientID FROM Appointments
    WHERE AppointmentDate > DATEADD(month, -1, GETDATE())
```

```
)
ORDER BY PatientID;
```

Generate a report of total revenue generated by each department

```
SELECT DepartmentName, SUM(Amount) AS "Total Revenue"
FROM Billing b
JOIN Appointments a ON b.AppointmentID = a.AppointmentID
JOIN Doctors d ON a.DoctorID = d.DoctorID
JOIN Departments dp ON d.DepartmentID = dp.DepartmentID
GROUP BY DepartmentName;
```

List of diagnosis, treatment, and prescription for each patient

```
SELECT FirstName + ' ' + LastName AS "Patient Name", MedicalHistory,
Diagnosis, Treatment, Prescription
FROM Patients p
JOIN MedicalRecords m ON p.PatientID = m.PatientID
ORDER BY FirstName;
```

Retrieve patient billing information along with doctor and department details

```
SELECT p.FirstName + ' ' + p.LastName AS "Patient Name", b.Amount,
b.PaymentStatus, d.FirstName + ' ' + d.LastName AS "Doctor Name",
dp.DepartmentName
FROM Billing b
JOIN Patients p ON b.PatientID = p.PatientID
JOIN Appointments ap ON ap.PatientID = p.PatientID
JOIN Doctors d ON ap.DoctorID = d.DoctorID
JOIN Departments dp ON d.DepartmentID = dp.DepartmentID;
```

List all unique first names from both doctors and patients

```
SELECT FirstName
FROM Doctors
UNION
SELECT FirstName
```

```
FROM Patients
ORDER BY FirstName;
```

This documentation provides a comprehensive overview of the database creation and management scripts for the `HospitalManagementSystem`. The DQL examples demonstrate how to retrieve specific data from the database, facilitating various hospital management tasks.