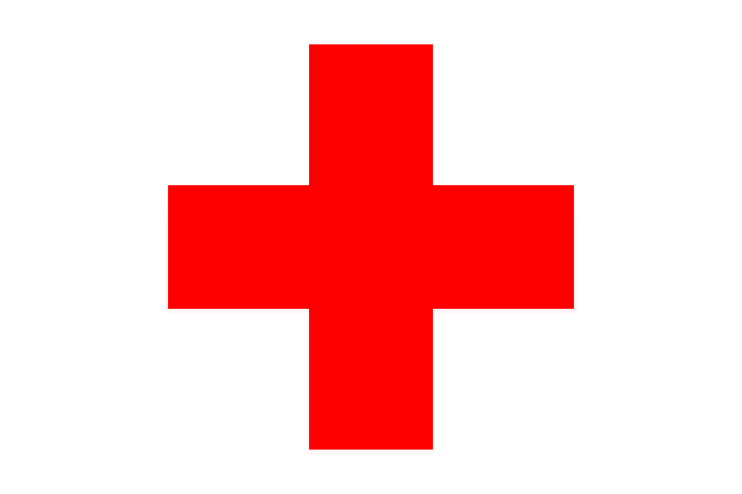
Hospital Database Management System



A project on database management systems by:

1. **DIPTARK BOSE**

SECTION: A BRANCH: CSE ROLL NO: 41

REG. NO: 150905198

1. **ANKUR GOHIT**

SECTION: A BRANCH: CSE ROLL NO: 48

REG. NO: 150905250

Hospitals are places of critical interest for people of the nearby region. It is therefore mandatory to ensure a smooth functioning of such an enterprise. Apart from patients, hospitals are largely staffed by professional physicians, surgeons, and nurses. In order to manage the hospital and allocate its resources properly, an efficient database management system has to be developed. We attempt to create such a database using our present skills in SQL. The database would encompass:

1. Patients: The database stores information regarding the patient, his ailment and other details.
2. Doctors: Database would contain personal info about the doctor, his area of specialization, department etc.
3. Nurses: Contains data about nurses, and the rooms they are monitoring.
4. Rooms: Hospital has many rooms, which are assigned to patients according to their needs. Each rooms has its own ID and type.
5. Departments: The hospital is a multi-specialty establishment, and offers treatments pertaining to different ailments offered by different departments such as Cardiology, Oncology.
6. Appointment: Contains details about appointments given to patients.
7. Blocks: Stores information about blocks in which rooms are located.

**Problem Statement**

It is vital to maintain an efficient database to handle the information of a hospital. Our application provides a way to record this information and to access it in a fast and elegant manner. The application contains the following information about the hospital:

**Departments**

This section contains the different Department IDs with their names. New departments can be added as and when inaugurated.

**Physicians**

This section contains the Employee ID, Name, Position, Department names of the Physicians working in the Hospital. The application can be used to display the information about the various physicians and also to add information of new physicians joining the Hospital, or delete the information of physicians who leave the hospital.

**Patients**

This section contains the Patient ID, name of the patient, Disease they are suffering from, Room No. allotted (if they are admitted), address and phone no. of the patient. The application can be used to display any of the details about a particular patient and also to add or delete patient information in the system.

**Nurse**

This section contains the Employee IDs of nurses and the room governed by each of them. Each nurse is in charge of one room. The details of the nurses can be browsed or edited as per requirement using the application.

**Appointments**

This section contains details about appointments given to patients for consultation with various physicians. It contains the Appointment ID, Patient ID of the patient and the Employee ID of the physician whom the patient wishes to consult. The application can be used to browse information about existing appointments and to add new appointments to the system.

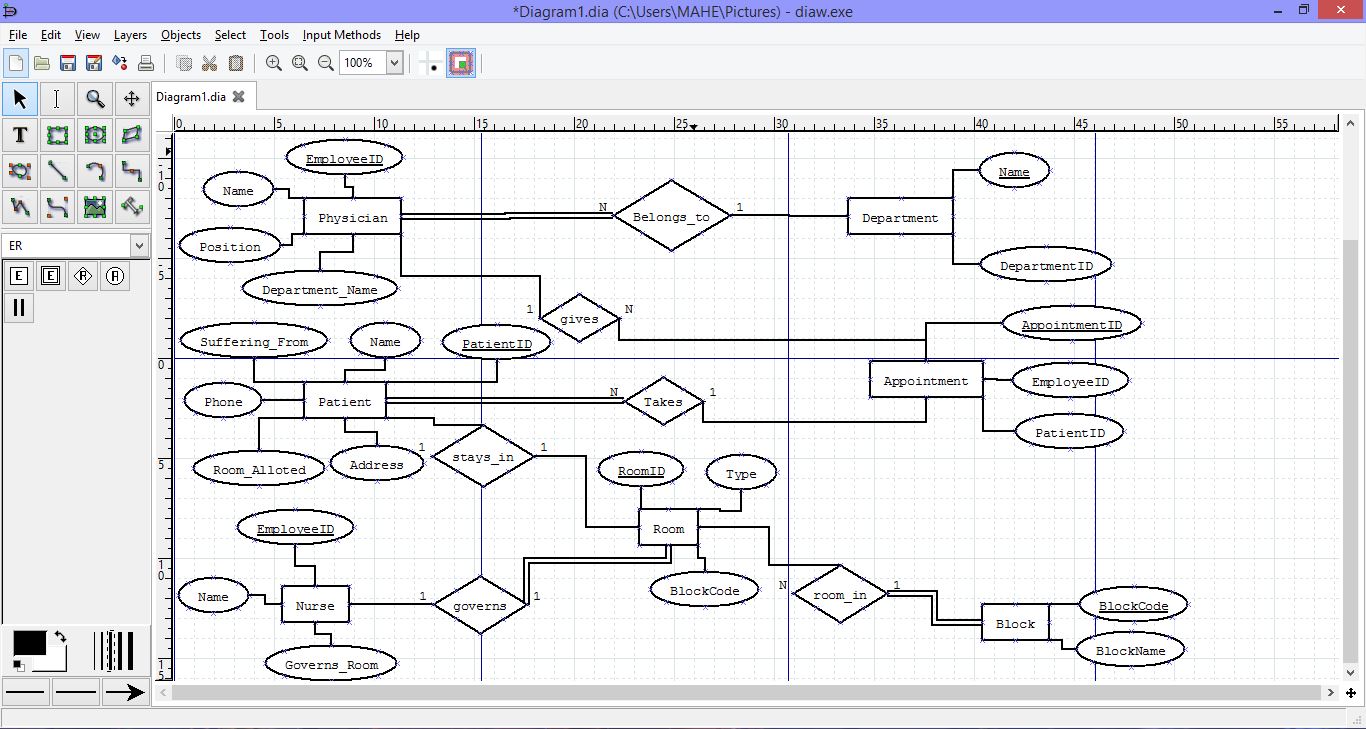
**Rooms**

This section contains details about the various rooms available for admitting patients in the hospital. It contains the Room No., Room Type and Block No. of the block in which the room is located.

**Blocks**

This section contains details about the Blocks in which patient rooms are located in the Hospital. Information like the Block No. and Block Name is available in this section.

**ER Diagram**



**Relational Tables**

**Departments**

|  |  |
| --- | --- |
| **DepartmentID** | **Name** |
| 1 | General Medicine |
| 2 | Surgery |
| 3 | Psychiatry |

**Rooms**

|  |  |  |
| --- | --- | --- |
| **RoomID** | **Type** | **BlockCode** |
| 101 | Single | 1 |
| 102 | Single | 1 |
| 103 | Single | 1 |
| 201 | Single | 2 |
| 202 | Single | 2 |

**Physicians**

|  |  |  |  |
| --- | --- | --- | --- |
| **EmployeeID** | **Name** | **Position** | **Department** |
| 1 | John Dorian | Staff Internist | General Medicine |
| 2 | Elliot Reid | Attending Physician | General Medicine |
| 3 | Christopher Turk | Surgical Attending Physician | Surgery |
| 4 | Percival Cox | Senior Attending Physician | Surgery |
| 5 | Bob Kelso | Head Chief of Medicine | General Medicine |
| 6 | Todd Quinlan | Surgical Attending Physician | Surgery |
| 7 | John Wen | Surgical Attending Physician | Surgery |
| 8 | Keith Dudemeister | MD Resident | General Medicine |
| 9 | Molly Clock | Attending Psychiatrist | Psychiatry |

**Patients**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **PatientID** | **Name** | **Suffering\_from** | **Room** | **Address** | **Phone** |
| 101 | John Smith | Tuberculosis | 101 | 42 Foobar Lane | 555-0256 |
| 102 | Grace Ritchie | Shoulder Injury | 201 | 37 Snafu Drive | 555-0512 |
| 103 | Random J. Patient | Common Cold | NULL | 101 Omgbbq Street | 555-1204 |
| 104 | Dennis Doe | Depression | NULL | 1100 Foobaz Avenue | 555-2048 |
| 105 | James Bond | Constipation | NULL | 134 Privet Drive | 555-0234 |

**Nurses**

|  |  |  |
| --- | --- | --- |
| **EmployeeID** | **Name** | **Governs\_room** |
| 101 | Carla Espinosa | 101 |
| 102 | Laverne Roberts | 202 |
| 103 | Paul Flowers | 103 |
| 104 | Nishant Bhat | 102 |

**Appointments**

|  |  |  |
| --- | --- | --- |
| **AppointmentID** | **PatientID** | **EmployeeID** |
| 1 | 101 | 2 |
| 2 | 104 | 9 |
| 3 | 103 | 8 |
| 4 | 102 | 6 |
| 5 | 105 | 4 |

**Blocks**

|  |  |
| --- | --- |
| **BlockCode** | **BlockName** |
| 1 | Sigmund Freud |
| 2 | Ron Paul |

**DCL Commands for create table**

CREATE TABLE Department (

DepartmentID VARCHAR(10) NOT NULL,

Name VARCHAR(30) PRIMARY KEY NOT NULL

);

CREATE TABLE Room (

RoomID INTEGER PRIMARY KEY NOT NULL,

Type VARCHAR(20) NOT NULL,

BlockCode INTEGER,

FOREIGN KEY(BlockCode) REFERENCES Block

);

CREATE TABLE Physician (

EmployeeID VARCHAR(20) PRIMARY KEY NOT NULL,

Name VARCHAR(30) NOT NULL,

Position VARCHAR(30) NOT NULL,

Department\_Name VARCHAR(30) NOT NULL,

FOREIGN KEY(Department\_Name) REFERENCES Department

);

CREATE TABLE Patient (

PatientID INTEGER PRIMARY KEY NOT NULL,

Name VARCHAR(30) NOT NULL,

Suffering\_From VARCHAR(40),

Room\_alloted INTEGER,

Address VARCHAR(40) NOT NULL,

Phone NUMERIC(10,0) NOT NULL,

FOREIGN KEY(Room\_alloted) REFERENCES Room,

);

CREATE TABLE Nurse (

EmployeeID INTEGER PRIMARY KEY NOT NULL,

Name VARCHAR(30) NOT NULL,

Governs\_Room INTEGER,

FOREIGN KEY(Governs\_Room) REFERENCES Room

);

CREATE TABLE Appointment(

AppointmentID INTEGER PRIMARY KEY NOT NULL,

PatientID INTEGER,

EmployeeID VARCHAR(20),

FOREIGN KEY(PatientID) REFERENCES Patient,

FOREIGN KEY(EmployeeID) REFERENCES Physician

);

CREATE TABLE Block(

BlockCode INTEGER PRIMARY KEY NOT NULL,

BlockName VARCHAR(20) NOT NULL,

);

**Queries Used**

1. select \* from patient

2. select \* from physician

3. select \* from nurse

4. select \* from department

5. select \* from appointment

6. select \* from room

7. select \* from block

8. SELECT \* from Patient where PatientID=?

9. SELECT \* from Physician where EmployeeID=?

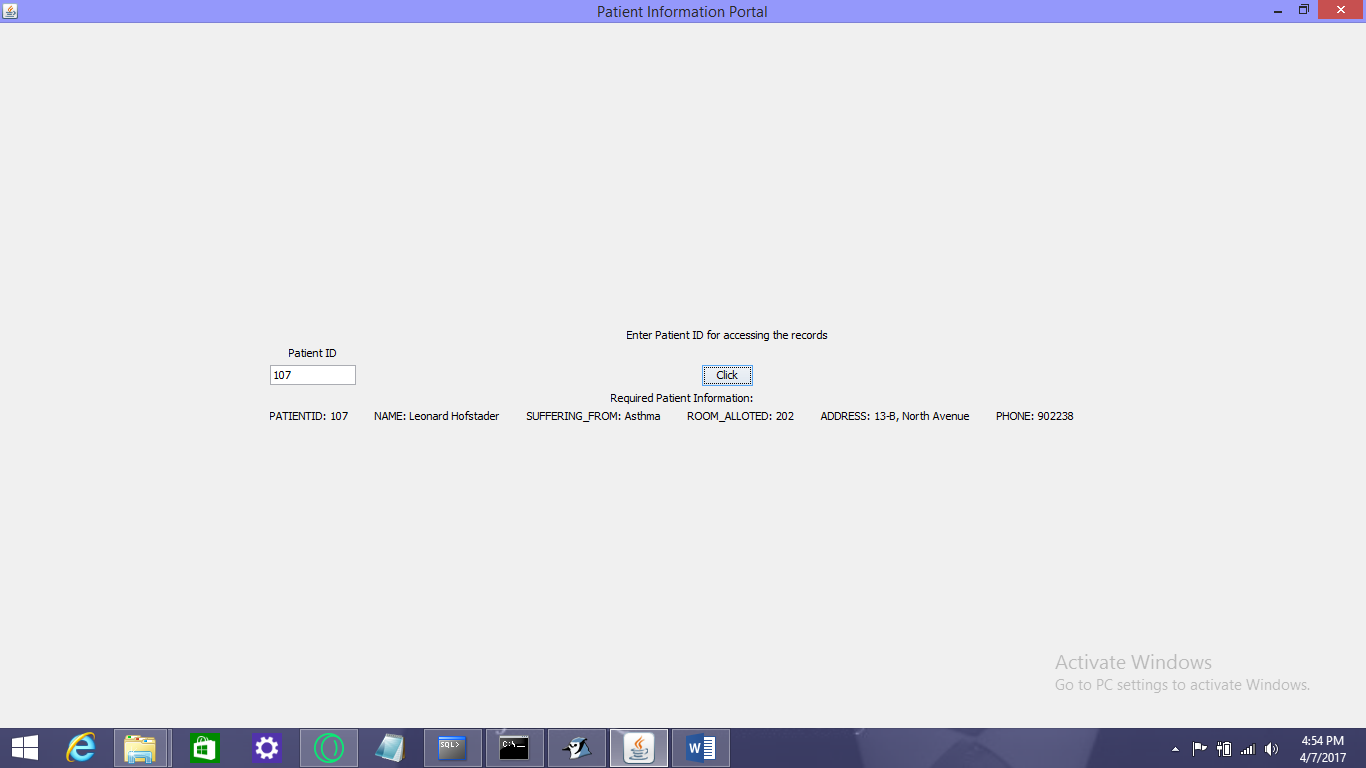
10. SELECT \* from Appointment where AppointmentID=?

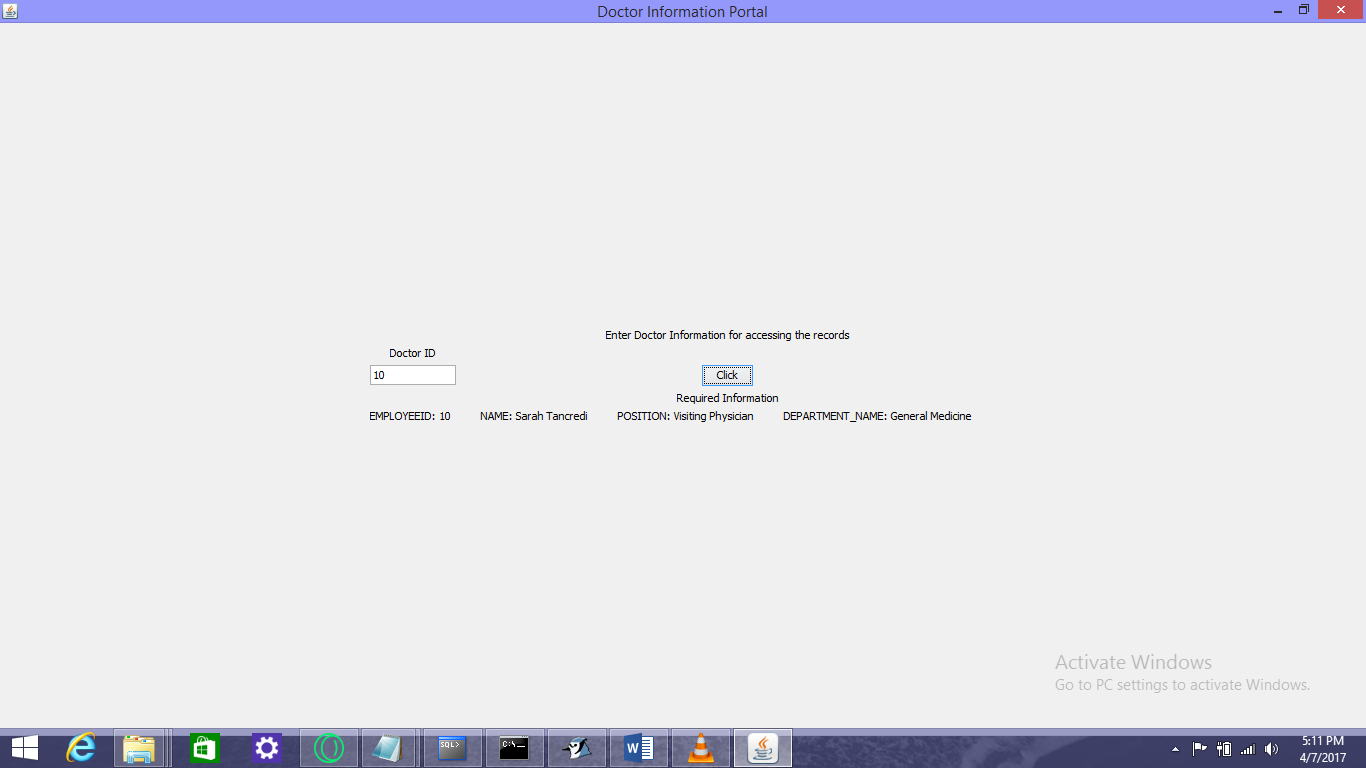
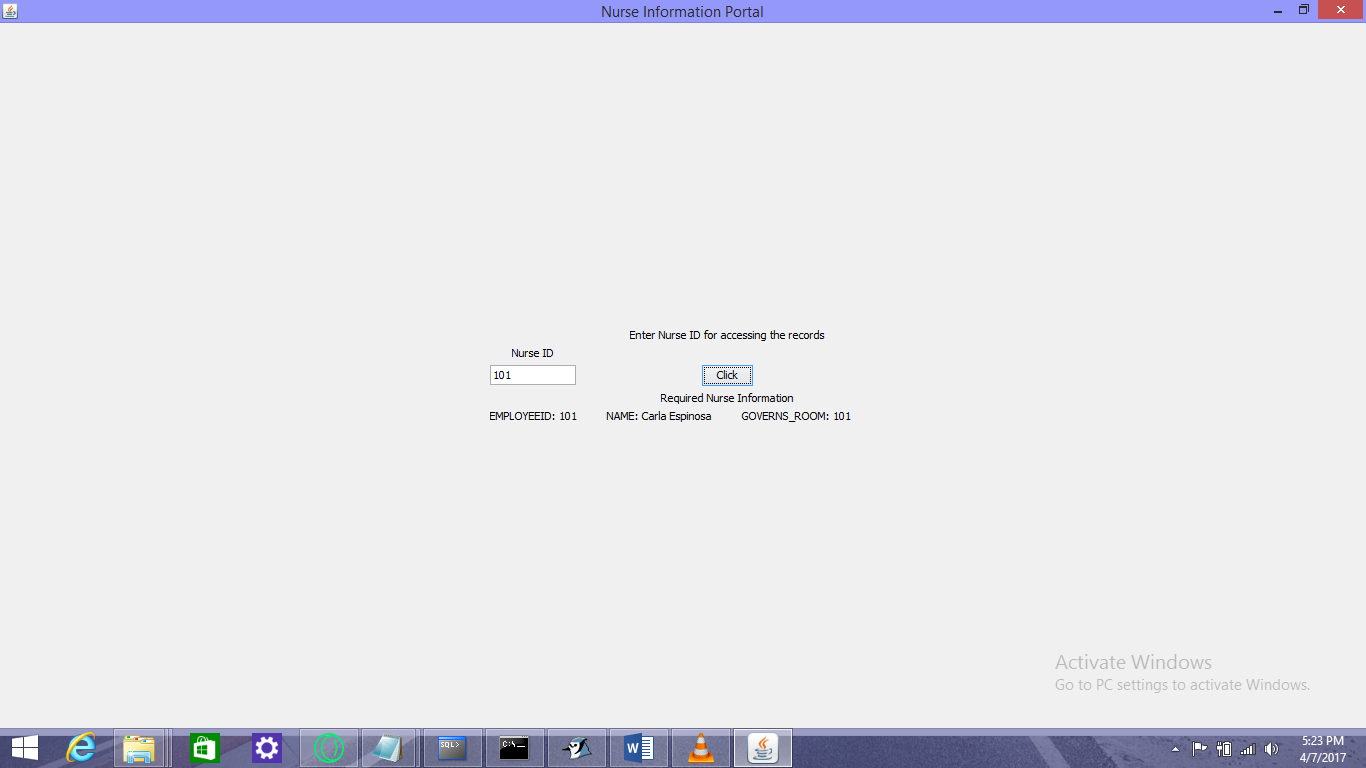
11. SELECT \* from nurse where employeeid=?

where ? denotes user input

Guest Screen

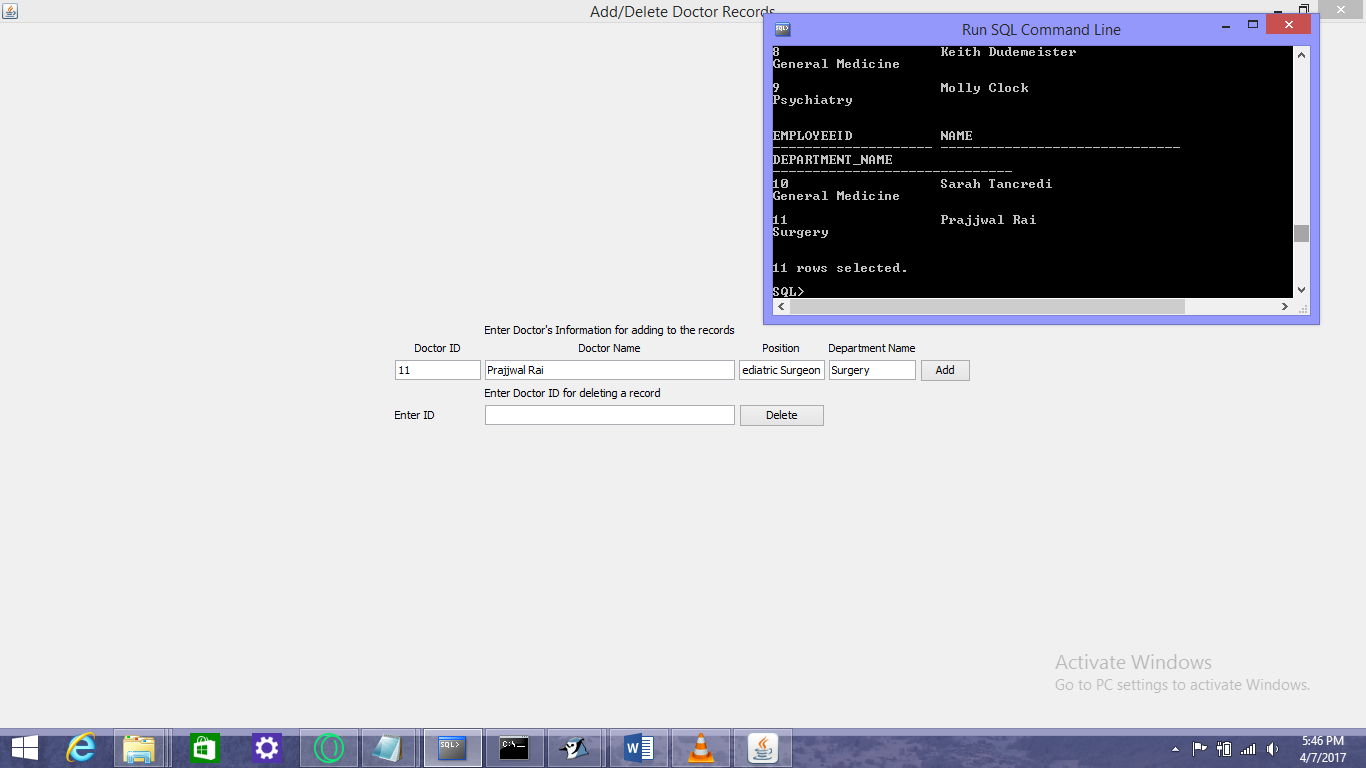
1. Searching for Patient information via unique Patient ID:

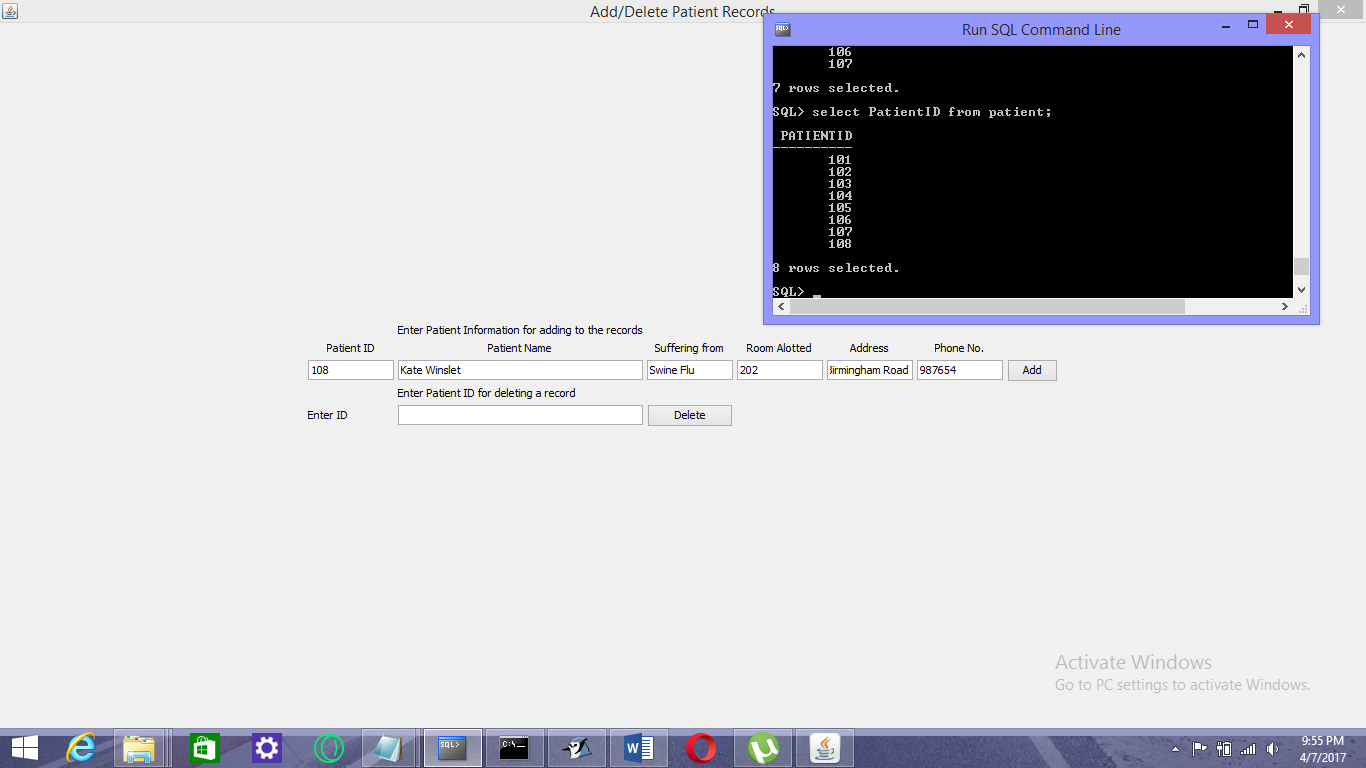


2. Searching for Doctor Information via unique Employee IDSearching for Nurse Information via unique Employee ID: 

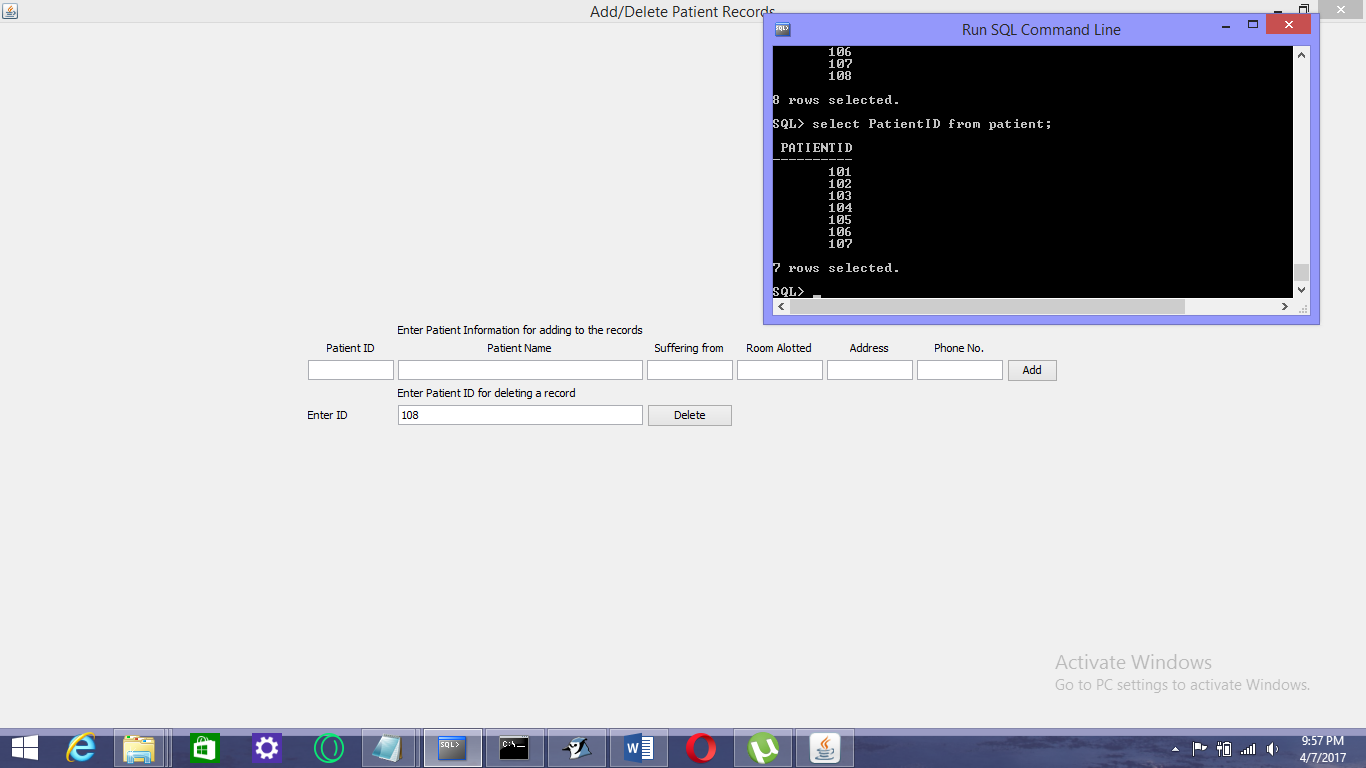
Administrator Screen

1. Adding Doctor Record

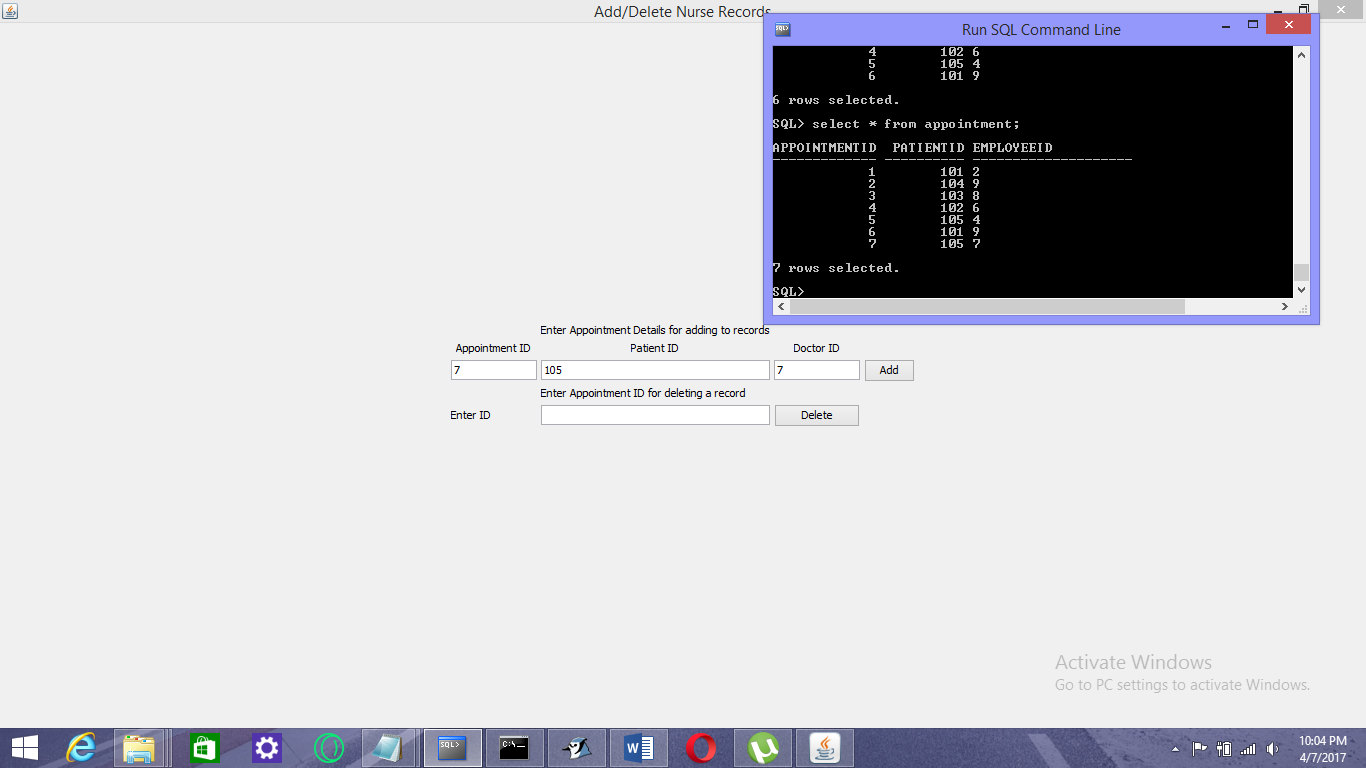


2. Adding Patient Record

1. Deleting Patient Record



1. Adding an Appointment



**Pseudocode**

Connection Snippet:

try {

Class.forName("oracle.jdbc.driver.OracleDriver");

con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","system","dip");

st = (Statement) con.createStatement();

ResultSet resultSet = st.executeQuery("SELECT \* from Patients");

ResultSetMetaData rsmd = resultSet.getMetaData();

int columnsNumber = rsmd.getColumnCount();

while (resultSet.next()) {

for (int i = 1; i <= columnsNumber; i++) {

if (i > 1) answer=answer+("\n ");

String columnValue = resultSet.getString(i);

//System.out.print(rsmd.getColumnName(i)+ ": " +columnValue);

answer=answer+(rsmd.getColumnName(i)+ ": " +columnValue+" ");

}

answer=answer+("\n\n");

}

} catch (Exception ex) {

}

**References**

1. G. Reese, “Database Programming with JDBC and Java”, O’REILLY, Second edition, 2000.
2. Silberchatz, Korth, Sudarshan, “Database System Concepts”, McGrawHill, 6th Edition, 2011.
3. <http://www.javatpoint.com/java-jdbc>
4. <https://docs.oracle.com/javase/tutorial/jdbc/basics/connecting.html>