CSL3050: DBMS

Assignment 07

By Shashank Shekhar Asthana (B21CS093)

This assignment consists of creating a database named Hospital and then working on that database to extract the queries using JDBC.

First, we have created the database, including all the tables in that database, which are displayed below.

TABLE HOSPITAL

HOSP_NAME	COUNTRY	ADDRESS
Al-Hilal Hospital	Kingdom of Bahrain	Al-Hilal Hospital, Muharraq, Bahrain
American Mission Hospital	Kingdom of Bahrain	Sheikh Essa Road, Manama, Bahrain
Bahrain Defence Force Hospital	Kingdom of Bahrain	Waly Alahed Avenue, West Riffa, Bahrain
Dar Al Shifa Hospital	Kuwait	Beirut Street, Hawally, Kuwait
Emirates Hospital	United Arab Emirates	Jumeirah Beach Rd, Dubai, UAE
King Faisal Specialist Hospital & Research Centre	Kingdom of Saudi Arabia	Al Mathar Ash Shamali, Riyadh 11564, Saudi Ar...
King Hamad University Hospital	Kingdom of Bahrain	Sheikh Isa bin Salman Bridge, Al Sayh, Busaitee...
Mayo Clinic	United States	4500 San Pablo Road, US
Salamaniya Hospital	Kindgom of Bahrain	Salmaniya Medical Complex, Manama, Bahrain
St Thomas' Hospital London	United Kingdom	Westminster Bridge Rd, London
NULL	NULL	NULL

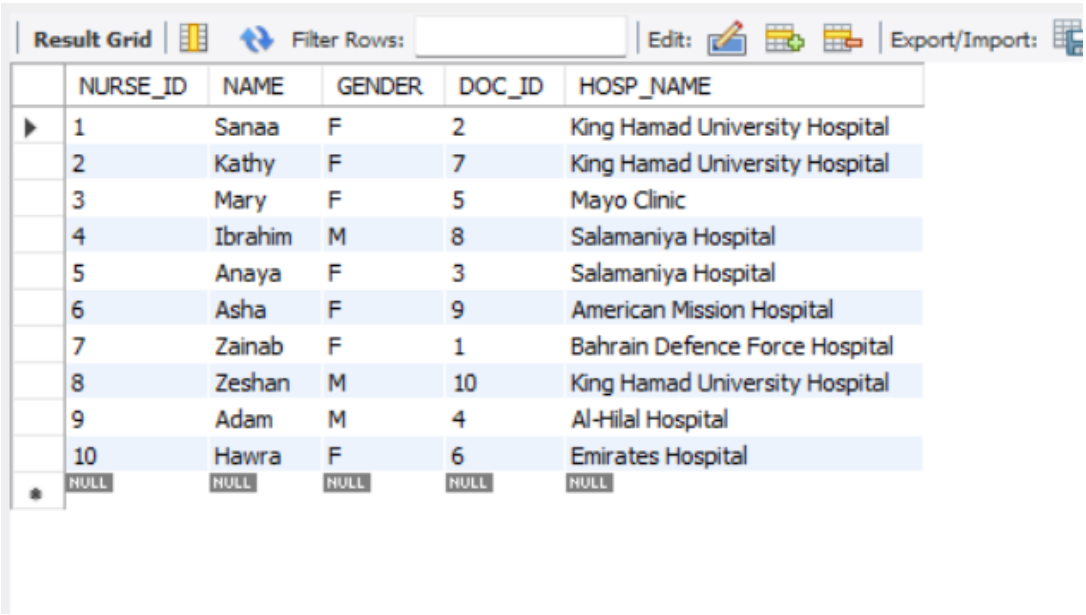
TALE MEDICINE

REG_NO	MED_NAME	PRICE	EXP_DATE
20	Adol Syrup	0.3	2023-05-15
21	Amoxil Capsules	0.7	2023-03-10
22	Aspirin	1	2023-12-09
23	Bonjela Gel	0.4	2024-09-23
24	Paracetamol	1.2	2024-05-17
25	Atorvastatin	1.5	2023-06-15
26	Coversyl	1	2023-05-15
27	Diamicron	0.9	2024-02-03
28	Lipitor	1.4	2025-03-12
29	Glucophage	1.7	2024-07-19
NULL	NULL	NULL	NULL

TABLE DOCTOR

	DOC_ID	DNAME	GENDER	QUALIFICATION	JOB_SPECIFICATION	HOSP_NAME	NEW_DOC_ID
1	1	Abdullah	M	MBBS	Dermatologist	Bahrain Defence Force Hospital	1
2	2	Ahmed	M	Board Certified	Ophthalmologist	King Hamad University Hospital	2
3	3	Ameera	F	MD	Pediatrician	Salamaniya Hospital	3
4	4	Ali	M	BPT	Physiotherapist	Al-Hilal Hospital	4
5	5	Carolina	F	MS	Surgeon	Mayo Clinic	5
6	6	Sarah	F	MD	Anatomy	Emirates Hospital	6
7	7	Qasim	M	MD	Radiologist	King Hamad University Hospital	7
8	8	Fatema	F	Residency	Anesthetist	Salamaniya Hospital	8
9	9	Khalid	M	MBBS	Psychiatrist	American Mission Hospital	9
10	10	Amal	F	D.M.	Nephrologist	King Hamad University Hospital	10

TABLE NURSE



NURSE_ID	NAME	GENDER	DOC_ID	HOSP_NAME
1	Sanaa	F	2	King Hamad University Hospital
2	Kathy	F	7	King Hamad University Hospital
3	Mary	F	5	Mayo Clinic
4	Ibrahim	M	8	Salamaniya Hospital
5	Anaya	F	3	Salamaniya Hospital
6	Asha	F	9	American Mission Hospital
7	Zainab	F	1	Bahrain Defence Force Hospital
8	Zeshan	M	10	King Hamad University Hospital
9	Adam	M	4	Al-Hilal Hospital
10	Hawra	F	6	Emirates Hospital
NULL	NULL	NULL	NULL	NULL

TABLE RECEPTION

REC_ID	TEL_NO	EMAIL	HOSP_NAME
41	39123456	khuh@gmail.com	King Hamad University Hospital
42	39123456	khuh@gmail.com	King Hamad University Hospital
43	39456780	mayoclinic@facebook.com	Mayo Clinic
44	33456780	emirateshospital@yahoo.com	Emirates Hospital
45	33123456	bdf@yahoo.com	Bahrain Defence Force Hospital
46	33678901	alhilal@facebook.com	Al-Hilal Hospital
47	39678901	daralshifa@gmail.com	Dar Al Shifa Hospital
48	33912045	st.thomashospital@gmail.com	St Thomas' Hospital London
49	39912045	kingfaisalshrc@yahoo.com	King Faisal Specialist Hospital & Research Centre
50	39678901	daralshifa@gmail.com	Dar Al Shifa Hospital
NULL	NULL	NULL	NULL

TABLE PATIENT

Result Grid | Filter Rows: | Edit: | Export/Im

SSN	FNAME	LNAME	AGE	GENDER	NURSE_ID	REC_ID
100000001	Sara	Majeed	27	F	7	46
100000002	Ahmed	Jamaal	59	M	3	43
100000003	Abdulla	Hameed	45	M	9	48
100000004	Mariam	Muhammad	40	F	1	41
100000005	Fatema	Hasan	64	F	6	47
100000006	Zainab	Abdulla	55	F	2	49
100000007	Khalil	Ibrahim	35	M	8	44
100000008	Alyaa	Husain	57	F	5	42
100000009	Khalid	Ahmed	60	M	4	50
100000010	Jawad	Ali	20	M	10	45
NULL	NULL	NULL	NULL	NULL	NULL	NULL

TABLE APPOINTMENT

APPOINT_NO	APPOINT_DATE	APPOINT_TIME	REC_ID
1	2022-05-03	07:05:03	45
2	2022-03-17	13:30:00	49
3	2022-12-22	09:45:00	44
4	2022-07-09	17:20:00	48
5	2022-05-03	12:15:00	41
6	2022-11-20	08:05:00	47
7	2022-07-12	15:40:00	42
8	2022-04-01	14:00:00	50
9	2022-08-30	10:25:00	43
10	2022-06-15	16:50:00	46
NULL	NULL	NULL	NULL

TABLE VISIT

VISIT_DATE	SSN	HOSP_NAME
2021-10-10	100000009	Dar Al Shifa Hospital
2020-05-15	100000005	Dar Al Shifa Hospital
2022-01-30	100000006	King Faisal Specialist Hospital & Research Centre
2019-06-04	100000002	Mayo Clinic
2020-12-24	100000010	Bahrain Defence Force Hospital
2020-11-11	100000004	King Hamad University Hospital
2021-02-06	100000001	Al-Hilal Hospital
2022-04-21	100000003	St Thomas' Hospital London
2022-04-19	100000007	Emirates Hospital
2021-02-07	100000008	King Hamad University Hospital

TABLE DIAGNOSIS

DIAGNOS_NO	ISSUE_DATE	TREATMENT	REMARKS	NURSE_ID	DOC_ID
31	2022-06-04	Physiotherapy	Once a month	4	8
32	2022-05-29	Aromatherapy	Twice a week	7	1
33	2022-06-18	Cyrotherapy	Twice a month	5	3
34	2022-07-07	Phototherapy	Once a month	10	6
35	2022-08-13	Radiotherapy	Once in 3 months	2	7
36	2022-12-03	Immunotherapy	Once a month	6	9
37	2022-07-04	Monotherapy	Once a month	1	2
38	2022-06-04	Pharmacotherapy	Once a month	8	10
39	2022-07-19	Oxygen therapy	Once a week	3	5
40	2022-06-25	Gene therapy	Once a month	9	4
NULL	NULL	NULL	NULL	NULL	NULL

TABLE PURCHASE

Result Grid | Filter Rows

REG_NO	SSN
20	100000007
23	NULL
28	100000004
25	100000009
21	100000003
29	100000006
22	100000008
27	100000002
24	100000007
26	NULL

TABLE EXAMINE

Result Grid | Filter Rows

DOC_ID	SSN
1	100000004
2	100000006
3	100000002
4	100000009
5	100000008
6	100000005
7	100000001
8	100000007
9	100000003
10	100000010

TABLE MEDICINE_COUNTRY

	MAN_COUNTRY	REG_NO
▶	United States	20
	Australia	21
	United States	22
	Germany	23
	United Kingdom	24
	United Kingdom	25
	Kingdom of Saudi Arabia	26
	Germany	27
	Switzerland	28
	Germany	29

Now we are done with the creation of the tables; hence, we will now start with the queries

1. updateAppointment: Use “PreparedStatement” to insert 2 new entries statically into the Appointment table, then execute the updates. (Hint: use setString)

Code for the Query

```
public void updateAppointment() {
    try {
        String sql = "INSERT INTO APPOINTMENT VALUES (?, ?,?,?)";
        PreparedStatement preparedStatement = connection.prepareStatement(sql);

        // Set the values for the placeholders
        preparedStatement.setInt(parameterIndex:1, x:12);
        preparedStatement.setString(parameterIndex:2, x:"2022-10-15");
        preparedStatement.setString(parameterIndex:3, x:"08:09:23");
        preparedStatement.setInt(parameterIndex:4, x:49);

        // Execute the update
        preparedStatement.executeUpdate();

        preparedStatement.setInt(parameterIndex:1, x:13);
        preparedStatement.setString(parameterIndex:2, x:"2022-10-15");
        preparedStatement.setString(parameterIndex:3, x:"09:09:23");
        preparedStatement.setInt(parameterIndex:4, x:44);

        // Execute the update
        preparedStatement.executeUpdate();
    } catch (SQLException e) {
        e.printStackTrace();
    }
}
```

Result:-

As proposed in the above static update of the table appointment, where we have predefined the value of the row details that need to be updated, they are updated after we complete the execution of the above function.

APPOINT_NO	APPOINT_DATE	APPOINT_TIME	REC_ID
4	2022-07-09	17:20:00	48
5	2022-05-03	12:15:00	41
6	2022-11-20	08:05:00	47
7	2022-07-12	15:40:00	42
8	2022-04-01	14:00:00	50
9	2022-08-30	10:25:00	43
10	2022-06-15	16:50:00	46
11	2022-03-05	NULL	42
12	2022-10-15	08:09:23	49
13	2022-10-15	09:09:23	44

Therefore, Rows 12 and 13 got updated according to the data, which was not taken as input from the user.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\shash\Desktop\SQL Scripts\B21CS093_Lab7\Assignment7>  & 'C:\Program Files\Java\jdk-21\bin\jav
s' '-cp' 'C:\Users\shash\Desktop\SQL Scripts\B21CS093_Lab7\Assignment7\bin' 'DatabaseHandler'
java.sql.SQLException: No suitable driver found for jdbc:mysql://localhost:3306/Hospital
        at java.sql/java.sql.DriverManager.getConnection(DriverManager.java:708)
        at java.sql/java.sql.DriverManager.getConnection(DriverManager.java:230)
        at DatabaseHandler.<init>(DatabaseHandler.java:22)
        at DatabaseHandler.main(DatabaseHandler.java:98)
Updating Appointment (Static)...
```

2. updateAppointmentDynamic: Now, use “PreparedStatement” to insert 3 new entries dynamically into the appointment table, then execute the updates. (Hint: (Use BufferedReader.)

Code for the Query:-

```
public void updateAppointmentDynamic(BufferedReader reader) {
    for(int i=0;i<3;i++){
        try {
        String sql = "INSERT INTO APPOINTMENT VALUES (?, ?,?,?)";
        PreparedStatement preparedStatement = connection.prepareStatement(sql);

        System.out.println(x:"Enter APPOINT_NO: ");
        int appointNo = Integer.parseInt(reader.readLine());

        System.out.println(x:"Enter APPOINT_DATE (YYYY-MM-DD): ");
        String appointDate = reader.readLine();

        System.out.println(x:"Enter APPOINT_TIME (HH:MM:SS): ");
        String appointtime = reader.readLine();

        System.out.println(x:"Enter REC_ID: ");
        int recId = Integer.parseInt(reader.readLine());

        preparedStatement.setInt(parameterIndex:1, appointNo);
        preparedStatement.setString(parameterIndex:2, appointDate);
        preparedStatement.setString(parameterIndex:3, appointtime);
        preparedStatement.setInt(parameterIndex:4, recId);

        preparedStatement.executeUpdate();
        }
        catch (IOException | SQLException e) {
        e.printStackTrace();
        }
```

Result:-

Input

```
        at DatabaseHandler.main(DatabaseHandler.java:126)
Enter APPOINT_NO:
14
Enter APPOINT_DATE (YYYY-MM-DD):
2022-09-03
Enter APPOINT_TIME (HH:MM:SS):
21:09:08
Enter REC_ID:
45
Enter APPOINT_NO:
15
Enter APPOINT_DATE (YYYY-MM-DD):
2022-09-06
Enter APPOINT_TIME (HH:MM:SS):
21:08:21
Enter REC_ID:
46
Getting Medicine Countries Starting with...
U
Manufactured in: United States, REG_NO: 20
Manufactured in: United States, REG_NO: 22
Manufactured in: United Kingdom, REG_NO: 24
Manufactured in: United Kingdom, REG_NO: 25
PS C:\Users\shash\Desktop\SQL Scripts\B21CS093 Lab7\Assignment7>  c:; cd 'c:
```

Output

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

APPOINT_NO	APPOINT_DATE	APPOINT_TIME	REC_ID
1	2022-05-03	07:05:03	45
2	2022-03-17	13:30:00	49
3	2022-12-22	09:45:00	44
4	2022-07-09	17:20:00	48
5	2022-05-03	12:15:00	41
6	2022-11-20	08:05:00	47
7	2022-07-12	15:40:00	42
8	2022-04-01	14:00:00	50
9	2022-08-30	10:25:00	43
10	2022-06-15	16:50:00	46
11	2022-03-05	NULL	42
12	2022-10-15	08:09:23	49
13	2022-10-15	09:09:23	44
14	2022-09-03	21:09:08	45
15	2022-09-06	21:08:21	46
NULL	NULL	NULL	NULL

3. updateVisitDynamic & setVisitConstraints: Change the Constraints for Visit table to only allow integer entries for SSN and date format entries for Visit_date, and then create updateVisitDynamic to dynamically enter 3 entries to the table, along with error handling for inputs violating the newly set constraints.

Code for the Query:-

```
public void updateVisitDynamic(BufferedReader reader) {
    try {
        String sql = "INSERT INTO Visit VALUES (?, ?, ?)";
        PreparedStatement preparedStatement = connection.prepareStatement(sql);
        for (int i = 1; i <= 3; i++) {
            System.out.println("Entry" + i + "\n");

            System.out.println(x:"Enter Visit_date (YYYY-MM-DD): ");
            String visitDateStr = reader.readLine();

            System.out.println(x:"Enter SSN: ");
            int ssn = Integer.parseInt(reader.readLine());

            System.out.println(x:"Enter HOSP_NAME: ");
            String hospName = reader.readLine();
            java.util.Date visitDate = null;
            try {
                SimpleDateFormat dateFormat = new SimpleDateFormat(pattern:"yyyy-MM-dd");
                visitDate = dateFormat.parse(visitDateStr);
            } catch (ParseException e) {
                System.out.println(x:"Invalid date format. Use YYYY-MM-DD.");
                continue;
            }
            preparedStatement.setDate(parameterIndex:1, new java.sql.Date(visitDate.getTime()));
            preparedStatement.setInt(parameterIndex:2, ssn);
            preparedStatement.setString(parameterIndex:3, hospName);

            int rowsAffected = preparedStatement.executeUpdate();
            if (rowsAffected > 0) {
                System.out.println(x:"Visit record added successfully.");
            } else {
                System.out.println(x:"Failed to add visit record.");
            }
        }
    } catch (IOException | SQLException e) {
        e.printStackTrace();
    }
}
```

Result:-

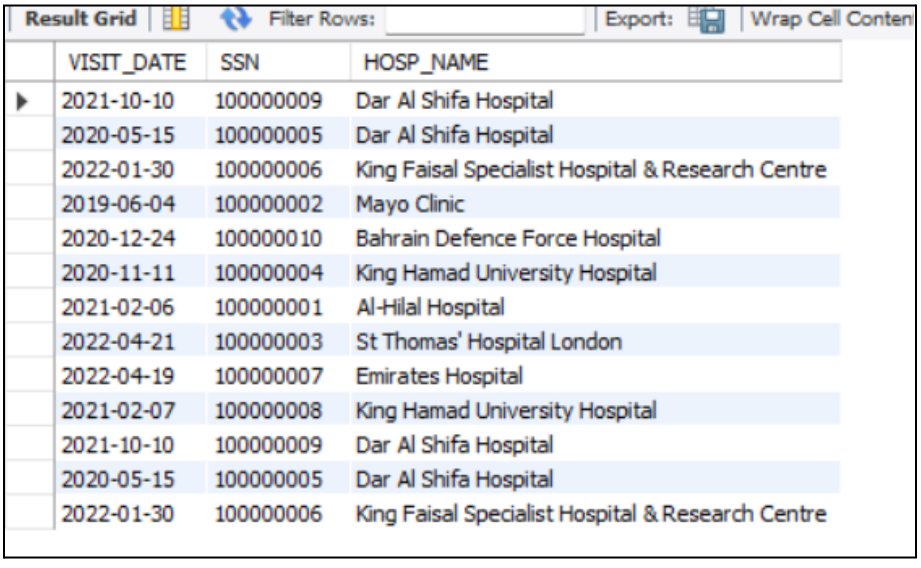
Input

```
Enter Visit_date (YYYY-MM-DD):
2021-10-10
Enter SSN:
100000009
Enter HOSP_NAME:
Dar Al Shifa Hospital
Visit record added successfully.
Entry2

Enter Visit_date (YYYY-MM-DD):
2020-05-15
Enter SSN:
100000005
Enter HOSP_NAME:
Dar Al Shifa Hospital
Visit record added successfully.
Entry3

Enter Visit_date (YYYY-MM-DD):
2022-01-30
Enter SSN:
100000006
Enter HOSP_NAME:
King Faisal Specialist Hospital & Research Centre
Visit record added successfully.
PS C:\Users\shash\Desktop\SQL Scripts\B21CS093_Lab7\Assignment7>
```

Output



Result Grid | Filter Rows: | Export: | Wrap Cell Conten

VISIT_DATE	SSN	HOSP_NAME
2021-10-10	100000009	Dar Al Shifa Hospital
2020-05-15	100000005	Dar Al Shifa Hospital
2022-01-30	100000006	King Faisal Specialist Hospital & Research Centre
2019-06-04	100000002	Mayo Clinic
2020-12-24	100000010	Bahrain Defence Force Hospital
2020-11-11	100000004	King Hamad University Hospital
2021-02-06	100000001	Al-Hilal Hospital
2022-04-21	100000003	St Thomas' Hospital London
2022-04-19	100000007	Emirates Hospital
2021-02-07	100000008	King Hamad University Hospital
2021-10-10	100000009	Dar Al Shifa Hospital
2020-05-15	100000005	Dar Al Shifa Hospital
2022-01-30	100000006	King Faisal Specialist Hospital & Research Centre

4. getMediCountryFromLetter: Create this method to get the records of the Medicine_country table starting with the given alphabet (For e.g., for input ‘I’, we should retrieve medicines with manufacturing country ‘India’ and other countries starting with I).

Code for the query:-

```
public void getMediCountryFromLetter(BufferedReader reader) throws IOException {
    try {
        String sql = "SELECT * FROM MEDICINE_COUNTRY WHERE MAN_COUNTRY LIKE ?";
        PreparedStatement preparedStatement = connection.prepareStatement(sql);

        String letter = reader.readLine();

        preparedStatement.setString(parameterIndex:1, letter + "%");

        ResultSet resultSet = preparedStatement.executeQuery();
        while (resultSet.next()) {
            String country = resultSet.getString(columnLabel:"MAN_COUNTRY");
            int regNo = resultSet.getInt(columnLabel:"REG_NO");
            System.out.println("Manufactured in: " + country + ", REG_NO: " + regNo);
        }
    } catch (SQLException e) {
        e.printStackTrace();
    }
}
```

Result:-

```
Getting Medicine Countries Starting with...
U
Manufactured in: United States, REG_NO: 20
Manufactured in: United States, REG_NO: 22
Manufactured in: United Kingdom, REG_NO: 24
Manufactured in: United Kingdom, REG_NO: 25
PS C:\Users\shash\Desktop\SQL Scripts\B21CS093_Lab7\Assignment7>
@C:\Users\shash\AppData\Local\Temp\cp_62lyu3r8ygjyjf3vi9bkcw4gw.a
```

5. deleteAffiliation: Create this method to delete 2 records from the Affiliation table (which you created in Assignment 6). (Hint: Use “PreparedStatement”)

Code for the query:-

```
public void deleteAffiliation(BufferedReader reader) throws NumberFormatException, IOException {
    for(int i=0;i<2;i++){
        try {
        // Define the DELETE statement
        String sql = "DELETE FROM Affiliation WHERE doc_id = ?"; // Assuming you want to delete 2 records for the given Doctor_ID

        int doctorId = Integer.parseInt(reader.readLine());
        PreparedStatement preparedStatement = connection.prepareStatement(sql);

        // Set the value for the placeholder
        preparedStatement.setInt(parameterIndex:1, doctorId);

        // Execute the delete operation
        int rowsDeleted = preparedStatement.executeUpdate();

        if (rowsDeleted > 0) {
            System.out.println("Deleted " + rowsDeleted + " records from Affiliation for Doctor ID " + doctorId);
        } else {
            System.out.println("No records found for Doctor ID " + doctorId);
        }
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

}
```

Result:-

```
' 'DatabaseHandler'
Deleting entries from Affiliation Table...
9
Deleted 1 records from Affiliation for Doctor ID 9
8
Deleted 1 records from Affiliation for Doctor ID 8
PS C:\Users\shash\Desktop\SQL Scripts\B21CS093_Lab7\Assignment7>  &
' 'DatabaseHandler'
```