

```
package lab12;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.Statement;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.math.BigDecimal;

import java.util.Scanner;

class lab12
{
    public static void main(String args[])
    {
        Connection c = null;
        try
        {
            // Load Postgresql Driver class
            Class.forName("org.postgresql.Driver");

            // Using Driver class connect to databased on localhost, port=5432, database=Lab_5, user=postgres,
            password=password. If cannot connect then exception will be generated (try-catch block)
            c = DriverManager.getConnection("jdbc:postgresql://localhost:5432/Lab_5","postgres", "password");

            System.out.println("Opened database successfully");

            // Create instance of this class to call other methods
            lab12 p = new lab12();
            p.setSearchPath(c);
            p.insertMedicinesTable(c);
```

```
p.queryMedicineTable(c);
p.updateMedTable(c);
p.deleteMedTable(c);
c.close();
}
catch (Exception e)
{
e.printStackTrace();
System.err.println(e.getClass().getName()+": "+e.getMessage());
System.exit(0);
}
}
```

```
void setSearchPath(Connection c)
{
Statement stmt = null;
try
{
stmt = c.createStatement();
String sql = "SET search_path TO hospital;";
stmt.executeUpdate(sql);
stmt.close();
System.out.println("Changed Search Path successfully");
}
catch (Exception e)
{
e.printStackTrace();
System.err.println(e.getClass().getName()+": "+e.getMessage());
System.exit(0);
}
```

```
}  
}
```

```
void insertMedicinesTable(Connection c)
```

```
{
```

```
    PreparedStatement stmt = null;
```

```
    String sql = "INSERT INTO
```

```
    medicines(medicine_name,medicine_id,cost_per_unit,amount_in_unit,amount_available,company_name) VALUES (?, ?, ?, ?, ?, ?);";
```

```
    try
```

```
    {
```

```
        Scanner in = new Scanner(System.in);
```

```
        stmt = c.prepareStatement(sql);
```

```
        System.out.println("Enter Medicine name:");
```

```
        String s = in.nextLine();
```

```
        System.out.println("You entered string "+s);
```

```
        stmt.setString(1, s);
```

```
        System.out.println("Enter medicine id:");
```

```
        int a = in.nextInt();
```

```
        System.out.println("You entered integer "+a);
```

```
        stmt.setInt(2, a);
```

```
        System.out.println("Enter cost per unit:");
```

```
        int b = in.nextInt();
```

```
        System.out.println("You entered integer "+b);
```

```
        stmt.setInt(3, b);
```

```
System.out.println("Enter amount in unit:");  
int x = in.nextInt();  
System.out.println("You entered integer "+x);  
stmt.setInt(4, x);
```

```
System.out.println("Enter Amount Available:");  
int d = in.nextInt();  
System.out.println("You entered integer "+d);  
stmt.setInt(5, d);
```

```
System.out.println("Enter Company name:");  
String e = in.nextLine();  
e = in.nextLine();  
System.out.println("You entered string "+e);  
stmt.setString(6, e);
```

```
in.close();  
int affectedRows = stmt.executeUpdate();  
stmt.close();  
System.out.println("Table Inserted successfully: Rows Affected: " + affectedRows);  
}  
catch (Exception e)  
{  
e.printStackTrace();  
System.err.println(e.getClass().getName()+": "+e.getMessage());  
System.exit(0);  
}  
}
```

```
void queryMedicineTable(Connection c)
{
    Statement stmt = null;

    try
    {
        stmt = c.createStatement();

        ResultSet rs = stmt.executeQuery("SELECT medicine_id, medicine_name FROM medicines WHERE
        amount_available = 0");

        while(rs.next())
        {
            String med_nm;
            Long med_no;

            med_nm = rs.getString("medicine_name");
            med_no = rs.getLong(1);

            System.out.println("Medicine ID: " + med_no + ", Medicine Name: " + med_nm);
        }

        stmt.close();

        System.out.println("Table Queried successfully");
    }
    catch (Exception e)
    {
        e.printStackTrace();

        System.err.println(e.getClass().getName()+" : "+e.getMessage());

        System.exit(0);
    }
}
```

```
}
```

```
void updateMedTable(Connection c)
```

```
{
```

```
PreparedStatement stmt = null;
```

```
String sql = "UPDATE medicines SET amount_available = ? WHERE medicine_id = ?";
```

```
try
```

```
{
```

```
stmt = c.prepareStatement(sql);
```

```
stmt.setBigDecimal(1, new BigDecimal(40));
```

```
stmt.setBigDecimal(2, new BigDecimal(4));
```

```
int affectedRows = stmt.executeUpdate();
```

```
stmt.close();
```

```
System.out.println("Table Updated successfully: Rows Updated: " + affectedRows);
```

```
}
```

```
catch (Exception e)
```

```
{
```

```
e.printStackTrace();
```

```
System.err.println(e.getClass().getName()+" : "+e.getMessage());
```

```
System.exit(0);
```

```
}
```

```
}
```

```
void deleteMedTable(Connection c)
```

```
{
```







```
PreparedStatement stmt = null;
```

```
String sql = "DELETE FROM medicines WHERE medicine_id = ?";
```

```
try
```

```
{  
stmt = c.prepareStatement(sql);  
stmt.setBigDecimal(1, new BigDecimal(44));  
int affectedRows = stmt.executeUpdate();  
stmt.close();  
System.out.println("Table Updated successfully: Rows Updated: " + affectedRows);  
  
}  
catch (Exception e)  
{  
e.printStackTrace();  
System.err.println(e.getClass().getName()+": "+e.getMessage());  
System.exit(0);  
}  
}  
  
}
```

Before running Java program:

Data Output		Explain	Messages	Notifications		
	 <b>medicine_id</b> [PK] integer	 <b>medicine_name</b> character varying (40)	 <b>cost_per_unit</b> numeric (8,2)	 <b>amount_in_unit</b> smallint	 <b>amount_available</b> integer	 <b>company_name</b> character varying (40)
1	1	Lithium Carbonate	185.00	10	56	Krajcik Inc
2	2	Staples Instant Hand Sanitizer	715.00	20	65	Osinski Group
3	3	Hand Cleanser	236.00	20	43	Heller Group
4	4	Dexilant	275.00	5	40	Bins-Jacobi
5	5	Naproxen	460.00	10	49	Leuschke and Sons
6	6	MICRELL Sp	200.00	12	25	Aufderhar Inc
7	7	Less Relief	590.00	10	67	Lehner-Thompson
8	8	Levofloxacin	920.00	5	49	Hayes LLC
9	9	hyoscyamine sulfate	975.00	5	87	Kreiger-Greenholt
10	10	METFORMIN HYDROCHLORI...	200.00	10	50	Hauck, Lowe and Steuber
11	11	Cheston Cold	35.00	10	60	Krajcik Inc
12	15	Cold Go	185.00	10	0	Krajcik Inc
13	44	levocit	40.00	40	40	krx

## Java Program output:

```

<terminated> lab12 [Java Application] C:\Users\91972\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32
Opened database successfully
Changed Search Path successfully
Enter Medicine name:
citrizin
You entered string citrizin
Enter medicine id:
41
You entered integer 41
Enter cost per unit:
50
You entered integer 50
Enter amount in unit:
10
You entered integer 10
Enter Amount Available:
40
You entered integer 40
Enter Company name:
ciplla
You entered string cipla
Table Inserted successfully: Rows Affected: 1
Medicine ID: 15, Medicine Name: Cold Go
Table Queried successfully
Table Updated successfully: Rows Updated: 1
Table Updated successfully: Rows Updated: 1

```

## After running Java Program:



	<div>medicine_id</div> <div>[PK] integer</div>	<div>medicine_name</div> <div>character varying (40)</div>	<div>cost_per_unit</div> <div>numeric (8,2)</div>	<div>amount_in_unit</div> <div>smallint</div>	<div>amount_available</div> <div>integer</div>	<div>company_name</div> <div>character varying (40)</div>
1	1	Lithium Carbonate	185.00	10	56	Krajcik Inc
2	2	Staples Instant Hand Sanitizer	715.00	20	65	Osinski Group
3	3	Hand Cleanser	236.00	20	43	Heller Group
4	4	Dexilant	275.00	5	40	Bins-Jacobi
5	5	Naproxen	460.00	10	49	Leuschke and Sons
6	6	MICRELL Sp	200.00	12	25	Aufderhar Inc
7	7	Less Relief	590.00	10	67	Lehner-Thompson
8	8	Levofloxacin	920.00	5	49	Hayes LLC
9	9	hyoscyamine sulfate	975.00	5	87	Kreiger-Greenholt
10	10	METFORMIN HYDROCHLORI...	200.00	10	50	Hauck, Lowe and Steuber
11	11	Cheston Cold	35.00	10	60	Krajcik Inc
12	15	Cold Go	185.00	10	0	Krajcik Inc
13	41	citrizin	50.00	10	40	cipla