

## Task 9: Database Interaction

1. Write code to establish a connection to your SQL database. Create a class DBConnection in a package connectionutil with a static variable connection of Type Connection and a static method getConnection() which returns connection. Connection properties supplied in the connection string should be read from a property file.

### CODE:

DBConnection class is created with a static variable of Connection Type and defined a static method and connection properties established from a property file.

### DBConnection

```
package connectionutil;
```

```
import java.io.FileInputStream;
```

```
import java.io.IOException;
```

```
import java.sql.*;
```

```
import java.util.*;
```

```
public class DBConnection {
```

```
    private static Connection connection;
```

```
    //The constructor is created as private inorder to prevent  
    instantiation
```

```
    private DBConnection(){};
```

```
    //Static method which returns connection
```

```
    //To get connection properties load from properties file
```

```
    public static Connection getConnection() throws  
    ClassNotFoundException, SQLException, IOException {
```

```
        FileInputStream fis=new
```

```

FileInputStream("C:\\Users\\ramya\\IdeaProjects\\CourierManagemen
t_OOps\\Assignment_OOps_Exception_JDBC\\connectionutil\\db.pr
operties");
    Properties properties=new Properties();
    properties.load(fis);
    String driver = properties.getProperty("db.driver");
    String url = properties.getProperty("db.url");
    String username = properties.getProperty("db.user");
    String password = properties.getProperty("db.password");
    connection = DriverManager.getConnection(url, username,
password);
    Class.forName(driver);
    System.out.println("Driver Loaded Successfully");
    connection=
DriverManager.getConnection(url,username,password);
    return connection;

}
}

```

## PROPERTIES FILE

```

db.url=jdbc:mysql://localhost:3306/courier_management
db.user=root
db.password=Nivi2movi#
db.driver=com.mysql.cj.jdbc.Driver

```

2. Create a Service class CourierServiceDb in dao with a static variable named connection of type Connection which can be assigned in the constructor by invoking the method in DBConnection Class.

```

package DAO;
import connectionutil.DBConnection;

import java.io.IOException;
import java.sql.*;
import java.util.HashMap;
import java.util.Map;

```

```

import java.util.Scanner;

public class CourierServiceDb {
    public static Connection connection;

    // Constructor to initialize the connection
    public CourierServiceDb() {
        try {
            connection = DBConnection.getConnection();//method invoking
        }
        catch (IOException | ClassNotFoundException | SQLException e)
        {
            System.out.println("Error establishing database connection: "
+ e.getMessage());
        }
    }
}

```

3. Include methods to insert, update, and retrieve data from the database (e.g., inserting a new order, updating courier status).

***insert\_record(connection);***

```

public static void insert_record(Connection connection) throws
SQLException {
    PreparedStatement ps = connection.prepareStatement("insert into
courier(courierID, senderName, senderAddress, receiverName,
receiverAddress, weight, status, trackingNumber, deliveryDate)
values (?,?,?,?,?,?,?,?,?)");
    Scanner sc = new Scanner(System.in);
    System.out.println("Enter Courier ID: ");
    int courierID = sc.nextInt();
    sc.nextLine();

    System.out.println("Enter Sender Name: ");
    String senderName = sc.nextLine();
}

```

```
System.out.println("Enter Sender Address: ");  
String senderAddress = sc.nextLine();
```

```
System.out.println("Enter Receiver Name: ");  
String receiverName = sc.nextLine();
```

```
System.out.println("Enter Receiver Address: ");  
String receiverAddress = sc.nextLine();
```

```
System.out.println("Enter Package Weight (kg): ");  
double weight = sc.nextDouble();  
sc.nextLine();
```

```
System.out.println("Enter Courier Status: ");  
String status = sc.nextLine();
```

```
System.out.println("Enter Tracking NUmber: ");  
String tnum = sc.nextLine();
```

```
System.out.println("Enter Delivery Date: ");  
String deli_date = sc.nextLine();
```

```
ps.setInt(1,courierID);  
ps.setString(2,senderName);  
ps.setString(3,senderAddress);  
ps.setString(4,receiverName);  
ps.setString(5,receiverAddress);  
ps.setDouble(6,weight);  
ps.setString(7,status);  
ps.setString(8,tnum);  
ps.setString(9,deli_date);  
int c=ps.executeUpdate();  
System.out.println("Number of Rows Inserted: "+c);  
}
```

## OUTPUT

```
Run CourierServiceDb x
C:\Users\ramya\.jdk\openjdk-23.0.1\bin\java.exe "-javaagent
Driver Loaded Successfully
Enter 1:Insertion of Record
Enter 2:Updation of Record
Enter 3:Retrieval of Record
Enter 4:To Get Delivery History
Enter 5:To Get Shipment Status Report
Enter 6:To Get Revenue Report
1
Enter Courier ID:
12
Enter Sender Name:
Vijay
Enter Sender Address:
Tiruvannamalai
Enter Receiver Name:
Akash
Enter Receiver Address:
Porur Chennai
Enter Package Weight (kg):
2.3
Enter Courier Status:
Ordered
Enter Tracking Number:
TN74888594
Enter Delivery Date:
2025-04-15
Number of Rows Inserted: 1
```

27 • `select * from Courier;`

CourierID	SenderName	SenderAddress	ReceiverName	ReceiverAddress	Weight	Status	TrackingNumber
5	Ravi Kumar	Gandhi Street, Vellore, Tamil Nadu	Anitha P	Thillai Nagar, Trichy, Tamil Nadu	4.10	Dispatched	TN741852963
6	Arun Kumar	Anna Nagar, Chennai, Tamil Nadu	Karthik S	Race Course, Trichy, Tamil Nadu	2.80	Delivered	TN369258147
7	Priya Raj	Gandhipuram, Coimbatore, Tamil Nadu	Vikram R	KK Nagar, Madurai, Tamil Nadu	1.55	In Transit	TN852963741
8	Divya Shree	Sathyamangalam, Erode, Tamil Nadu	Sundar C	Simmakkal, Madurai, Tamil Nadu	3.00	Shipped	TN159357486
9	Meena Lakshmi	Kovaiipudur, Coimbatore, Tamil Nadu	Ravi Kumar	Gandhi Street, Vellore, Tamil Nadu	2.40	delivered	TN753951486
10	Revathi R	MGR Nagar, Salem, Tamil Nadu	Anitha P	Thillai Nagar, Trichy, Tamil Nadu	1.20	Pending	TN369147258
12	Vijay	Tiruvannamalai	Akash	Porur Chennai	2.30	Ordered	TN74888594
32	cvf	junygbvbh	degghg	nyhtyu	4.00	vdrgr	sdr56547
•	NULL	NULL	NULL	nyhtyu	NULL	NULL	NULL

***update\_record(connection);***

```
public static void update_record(Connection connection) throws
SQLException {
    PreparedStatement ps= connection.prepareStatement("update
courier set status=? where courierID=?");
    Scanner sc = new Scanner(System.in);
    System.out.println("Enter New Status to be updated for specific
courier ID");
    String status=sc.next();
    System.out.println("Enter Courier ID");
    int courierID=sc.nextInt();
    ps.setString(1,status);
    ps.setInt(2,courierID);
    int c= ps.executeUpdate();
    System.out.println("Status Updated as "+status+" for Courier ID
"+courierID);

}
```

**OUTPUT**

```

Run CourierServiceDb x
C:\Users\ramya\.jdk\openjdk-23.0.1\bin\java.exe "-javaagent:C:\Pro
Driver Loaded Successfully
Enter 1:Insertion of Record
Enter 2:Updation of Record
Enter 3:Retrieval of Record
Enter 4:TO Get Delivery History
Enter 5:To Get Shipment Status Report
Enter 6:To Get Revenue Report
2
Enter New Status to be updated for specific courier ID
Delivered
Enter Courier ID
8
Status Updated as Delivered for Courier ID 8

Process finished with exit code 0

```

## Before

7	Priya Raj	Gandhipuram, Coimbatore, Tamil Nadu	Vikram R	KK Nagar, Madurai, Tamil Nadu	1.55	In Transit	TN852963741
8	Divya Shree	Sathyamangalam, Erode, Tamil Nadu	Sundar C	Simmakkal, Madurai, Tamil Nadu	3.00	Shipped	TN159357486
9	Meena Lakshmi	Kovaipudur, Coimbatore, Tamil Nadu	Ravi Kumar	Gandhi Street, Vellore, Tamil Nadu	2.40	delivered	TN753951486

## After

7	Priya Raj	Gandhipuram, Coimbatore, Tamil Nadu	Vikram R	KK Nagar, Madurai, Tamil Nadu	1.55	In Transit	TN852963741
8	Divya Shree	Sathyamangalam, Erode, Tamil Nadu	Sundar C	Simmakkal, Madurai, Tamil Nadu	3.00	Delivered	TN159357486
9	Meena Lakshmi	Kovaipudur, Coimbatore, Tamil Nadu	Ravi Kumar	Gandhi Street, Vellore, Tamil Nadu	2.40	delivered	TN753951486
10	Revathi R	MGR Nagar, Salem, Tamil Nadu	Anitha P	Thillai Nagar, Trichy, Tamil Nadu	1.20	Pending	TN369147258

***retrieve\_record(connection);***

public static void retrieve\_record(Connection connection) throws  
SQLException {

Statement st=connection.createStatement();

ResultSet rs=st.executeQuery("select courierID,senderName from  
courier");

while(rs.next())

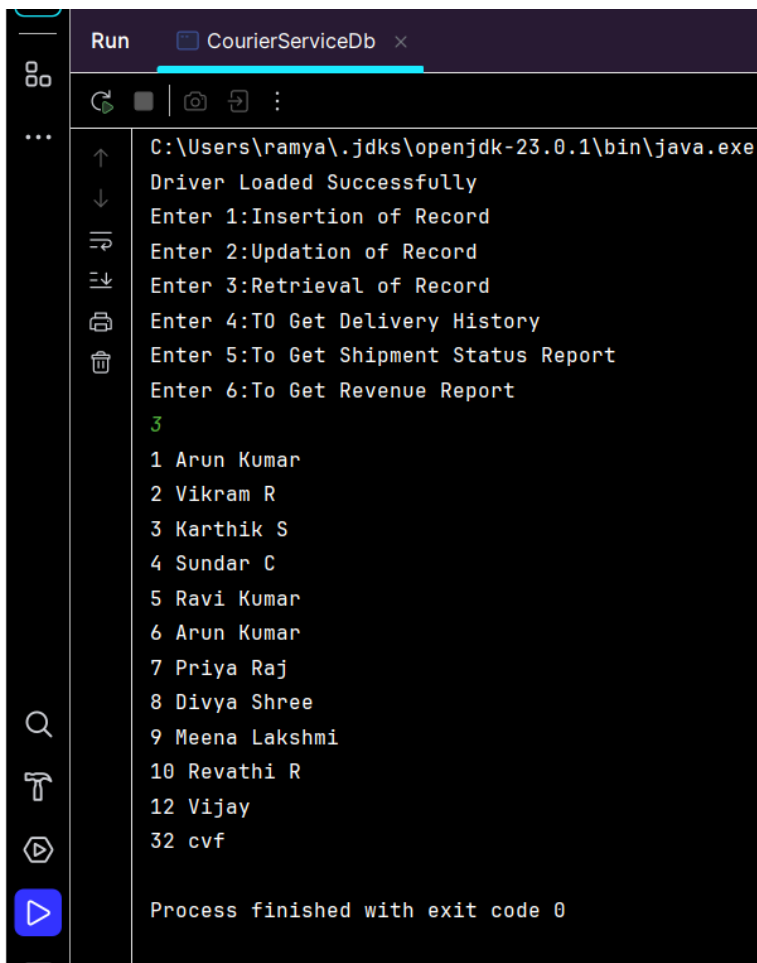
{

System.out.println(rs.getInt(1)+" "+rs.getString(2));

}

}

## OUTPUT



```
Run CourierServiceDb x
C:\Users\ramya\.jdk\openjdk-23.0.1\bin\java.exe
Driver Loaded Successfully
Enter 1:Insertion of Record
Enter 2:Updation of Record
Enter 3:Retrieval of Record
Enter 4:T0 Get Delivery History
Enter 5:To Get Shipment Status Report
Enter 6:To Get Revenue Report
3
1 Arun Kumar
2 Vikram R
3 Karthik S
4 Sundar C
5 Ravi Kumar
6 Arun Kumar
7 Priya Raj
8 Divya Shree
9 Meena Lakshmi
10 Revathi R
12 Vijay
32 cvf
Process finished with exit code 0
```

4. Implement a feature to retrieve and display the delivery history of a specific parcel by querying the database. 1. Generate and display reports using data retrieved from the database (e.g., shipment status report, revenue report).

***delivery\_history(connection);***

public static void delivery\_history(Connection connection) throws SQLException {

Scanner sc = new Scanner(System.in);

System.out.println("Enter Courier ID to view Delivery History");

int cid=sc.nextInt();

PreparedStatement pst= connection.prepareStatement("select status

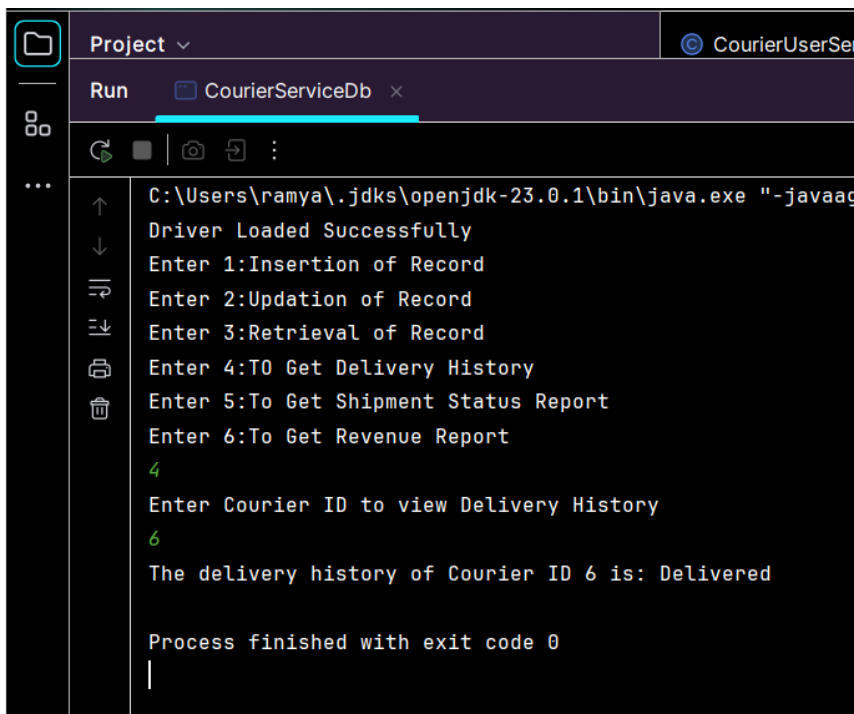


```

from courier where courierID=?");
    pst.setInt(1,cid);
    ResultSet rs = pst.executeQuery();
    if (rs.next()) {
        System.out.println("The delivery history of Courier ID " + cid +
" is: " + rs.getString("status"));
    } else {
        System.out.println("No delivery history found for Courier ID " +
cid);
    }
}
}

```

## OUTPUT



```

C:\Users\ramya\.jdk\openjdk-23.0.1\bin\java.exe "-javaag
Driver Loaded Successfully
Enter 1:Insertion of Record
Enter 2:Updation of Record
Enter 3:Retrieval of Record
Enter 4:To Get Delivery History
Enter 5:To Get Shipment Status Report
Enter 6:To Get Revenue Report
4
Enter Courier ID to view Delivery History
6
The delivery history of Courier ID 6 is: Delivered

Process finished with exit code 0

```

*shipment\_status\_report(connection);*

```

public static void shipment_status_report(Connection connection)
throws SQLException {
    Map<String,Integer> statusCount=new HashMap<>();
    Statement st= connection.createStatement();
    ResultSet rs= st.executeQuery("select status from courier");

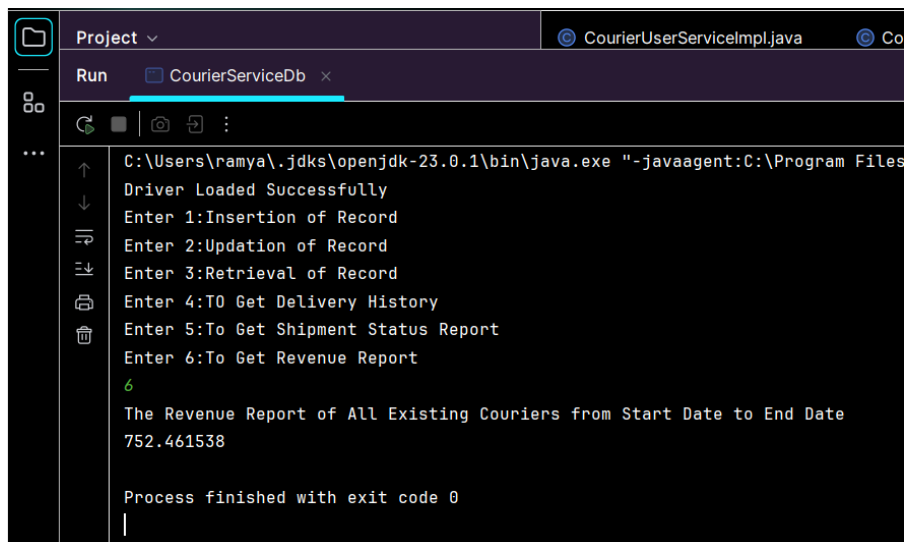
```



***revenue\_report(connection);***

```
public static void revenue_report(Connection connection) throws
SQLException {
    Statement st= connection.createStatement();
    ResultSet rs=st.executeQuery("select avg(amount) from payment
where paymentdate between '2025-01-01' and '2025-03-06'");
    if (rs.next()) {
        double avgRevenue = rs.getDouble(1);
        System.out.println("The Revenue Report of All Existing
Couriers from Start Date to End Date\n" + avgRevenue);
    }
}
```

## OUTPUT



```
C:\Users\ramya\.jdk\openjdk-23.0.1\bin\java.exe "-javaagent:C:\Program Files
Driver Loaded Successfully
Enter 1:Insertion of Record
Enter 2:Updation of Record
Enter 3:Retrieval of Record
Enter 4:T0 Get Delivery History
Enter 5:To Get Shipment Status Report
Enter 6:To Get Revenue Report
6
The Revenue Report of All Existing Couriers from Start Date to End Date
752.461538
Process finished with exit code 0
```

## MAIN METHOD

1.To select the option is obtained using switch case

```
public static void main(String[] args) throws SQLException {

    CourierServiceDb csd=new CourierServiceDb();
    Scanner sc=new Scanner(System.in);
```

```
        System.out.println("Enter 1:Insertion of Record\nEnter
2:Updation of Record" +
        "\nEnter 3:Retrieval of Record\nEnter 4:TO Get Delivery
History\nEnter 5:To Get Shipment Status Report" +
        "\nEnter 6:To Get Revenue Report");
        int value = sc.nextInt();
        switch (value) {
        case 1:
            insert_record(connection);
            break;
        case 2:
            update_record(connection);
            break;
        case 3:
            retrieve_record(connection);
            break;
        case 4:
            delivery_history(connection);
            break;
        case 5:
            shipment_status_report(connection);
            break;
        case 6:
            revenue_report(connection);
            break;
        case 7:
            System.out.println("Invalid Option");
        }
    }
}
```