JAVA DATABASE CONNECTIVITY(JDBC)

[FUNDAMENTAL STEPS IN JDBC:](#_wig3uuhm36lu)

[CREATING CONNECTION:](#_y0zz6jvcccx4)

[CREATING STATEMENT/PREPARED STATEMENT:](#_ikgnfnu1eakl)

[EXECUTING STATEMENTS AND STORING IN RESULT SET:](#_ex07nt1ks7d3)

[PROCESS THE RESULT SET:](#_8id58ynt81z4)

[CLOSE CONNECTION:](#_1o5yzptubz4f)

[PROBLEM STATEMENT:](#_s4dejyhox40t)

[APPROACH:](#_t2nts8mtuvg5)

[PROPER JDBC CONNECTION :](#_y9kvb0cn3vue)

[FUNCTIONALITY:](#_rucz2ygqhpg4)

[SCHEMA :](#_519f1nt372wi)

[SOURCE CODE:](#_5apo36orf7eh)

### FUNDAMENTAL STEPS IN JDBC:

* Import **JDBC** packages.
* Load and register the **JDBC** driver.
* Open a connection to the database.
* Create a statement/prepared statement object to perform a query.
* Execute the statement/prepared statement object and return a query resultset.
* Process the resultset.
* Close the resultset and statement objects.
* Close the connection.

#### 

#### CREATING CONNECTION:

Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/database\_name",”username”, “password”);

#### CREATING STATEMENT/PREPARED STATEMENT:

Statement stmt = stmt conn.createStatement( );

String SQL = "Update Employees SET age = ? WHERE id = ?";  
PreparedStatement pstmt = conn.prepareStatement(SQL);

#### EXECUTING STATEMENTS AND STORING IN RESULT SET:

ResultSet rs=stmt.executeQuery("select \* from table\_name");

#### 

#### PROCESS THE RESULT SET:

ResultSet rs=stmt.executeQuery("select \* from table\_name");

**while**(rs.next()){

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

#### CLOSE CONNECTION:

conn.close();

### PROBLEM STATEMENT:

Create an application that allows you to create edit and update contact information of list of

people. Also this should have the functionality to sort by first name or last name

Contact Information includes

First Name

Last Name

Phone Number – Should have office , home and mobile number

Office – extension number + number

Mobile number – country code + number

Home – area code, country code and number

Email Address

Give appropriate data types to the above fields and perform validation checks.

#### APPROACH:

* Have 5 packages
  + Com.some\_name.beans
  + Com.some\_name.constants
  + Com.some\_name.crud
  + Com.some\_name.dao
  + Com.some\_name.utils
* Com.some\_name.beans will have a pojo class that have all the user details declared and this class has getters and setters.
* Com.some\_name.constants will have a constants class that has all the constants declared and initialised. Constants are made public and static.
* Com.some\_name.crud will have two java files
  + Index.java--main class that loads first which based on user choice redirects to a specific method of redirect.java.
  + Redirect.java--has 4 methods create ,edit,delete,display that gets the input from the user and sends it to the dao.
* Com.some\_name.dao will have a dao class that has the functionality of create,edit,delete,display.
* Com.some\_name.utils will have the database connectivity essentials.

Connectivity code is maintained separately for reusablity.

#### PROPER JDBC CONNECTION :

* Create a connection object and connect it to database using getconnection(connectionString,username,password) method.

public static Connection getConnection() {

Connection conn = null;

try {

conn = DriverManager.getConnection(Constants.connectionString, Constants.username, Constants.password);

logger.log(Level.INFO, "connected");

} catch (SQLException e) {

logger.log(Level.SEVERE, "Error connecting with SQL Driver");

}

return conn;

}

* ConnectionString,username,password should be declared in constants.
* While Closing connection close connection,statement,resultset object.

public static void closeConnection(Connection conn, PreparedStatement pst, ResultSet rs) {

try {

if (rs != null) {

rs.close();

}

if (pst != null) {

pst.close();

}

if (conn != null) {

conn.close();

}

} catch (SQLException e) {

logger.log(Level.SEVERE, "Error closing the connection variables");

}

}

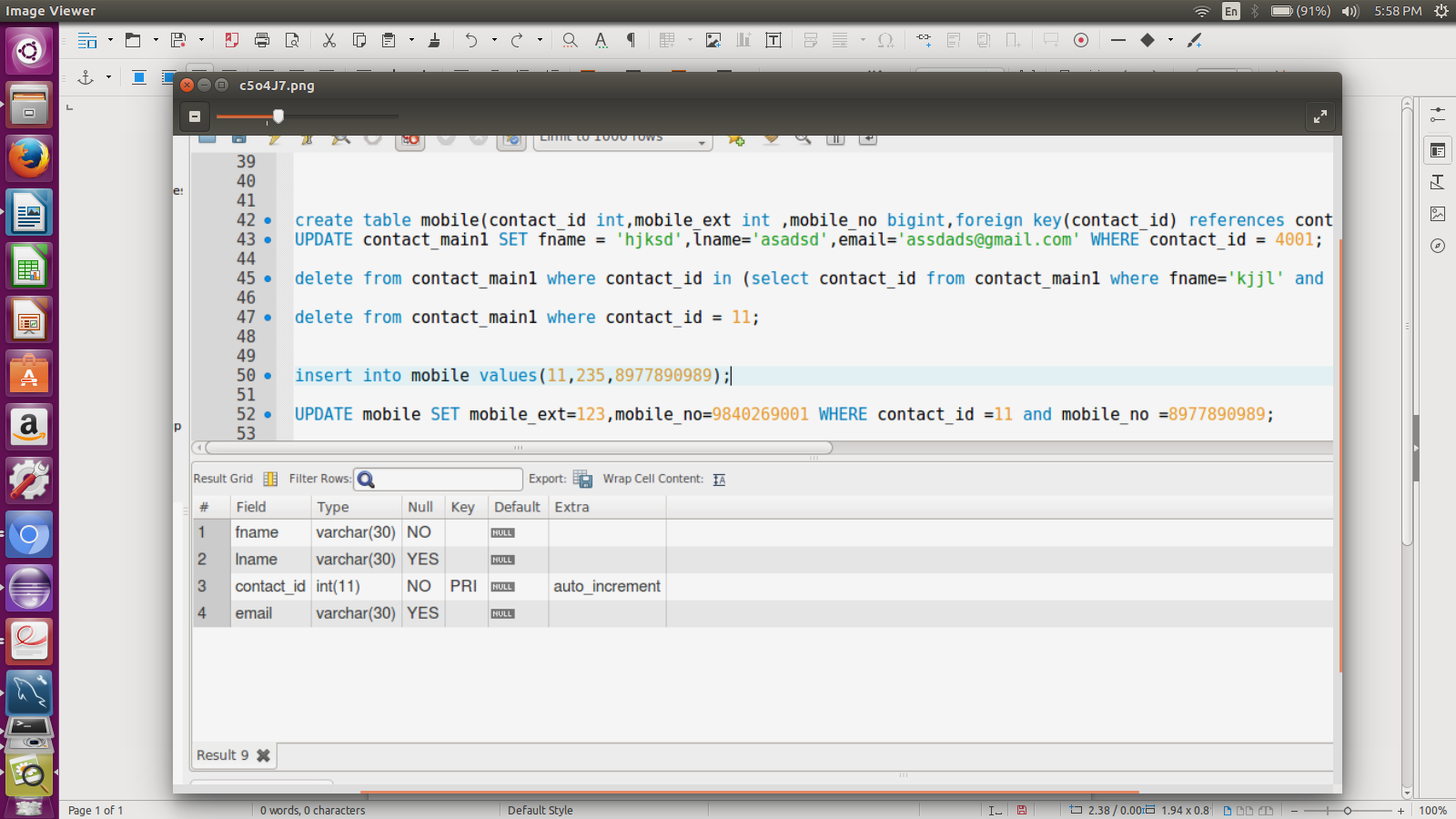
#### FUNCTIONALITY:

ContactDAO inside com.some\_name.constants will contain the business logic of create,edit,delete,display.

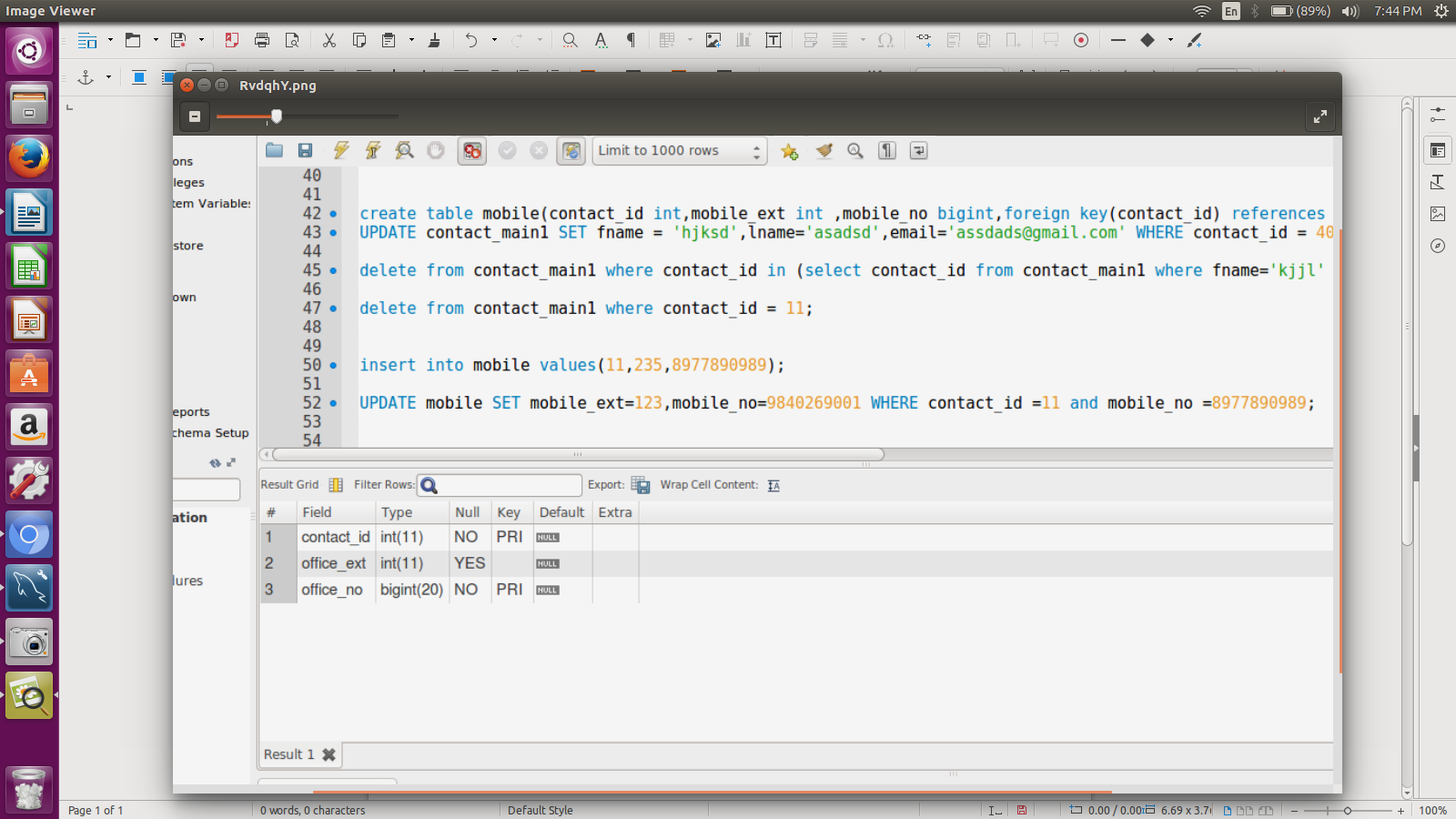
* For create method:
  + Get all the user credentials and insert into tables.
  + Create a contactId for each new user automatically.
* For edit method:
  + Get the email from the user which he wants to edit.
  + Match the email with the contactId in the database and proceed with the edit process.
* For delete method:
  + Get the email from the user which he wants to delete.
  + Match the email with the contactId in the database and proceed with the delete process.
* For display method:
  + Get the email from the user which he wants to display.
  + Match the email with the contactId in the database and proceed with the display process.

#### SCHEMA :

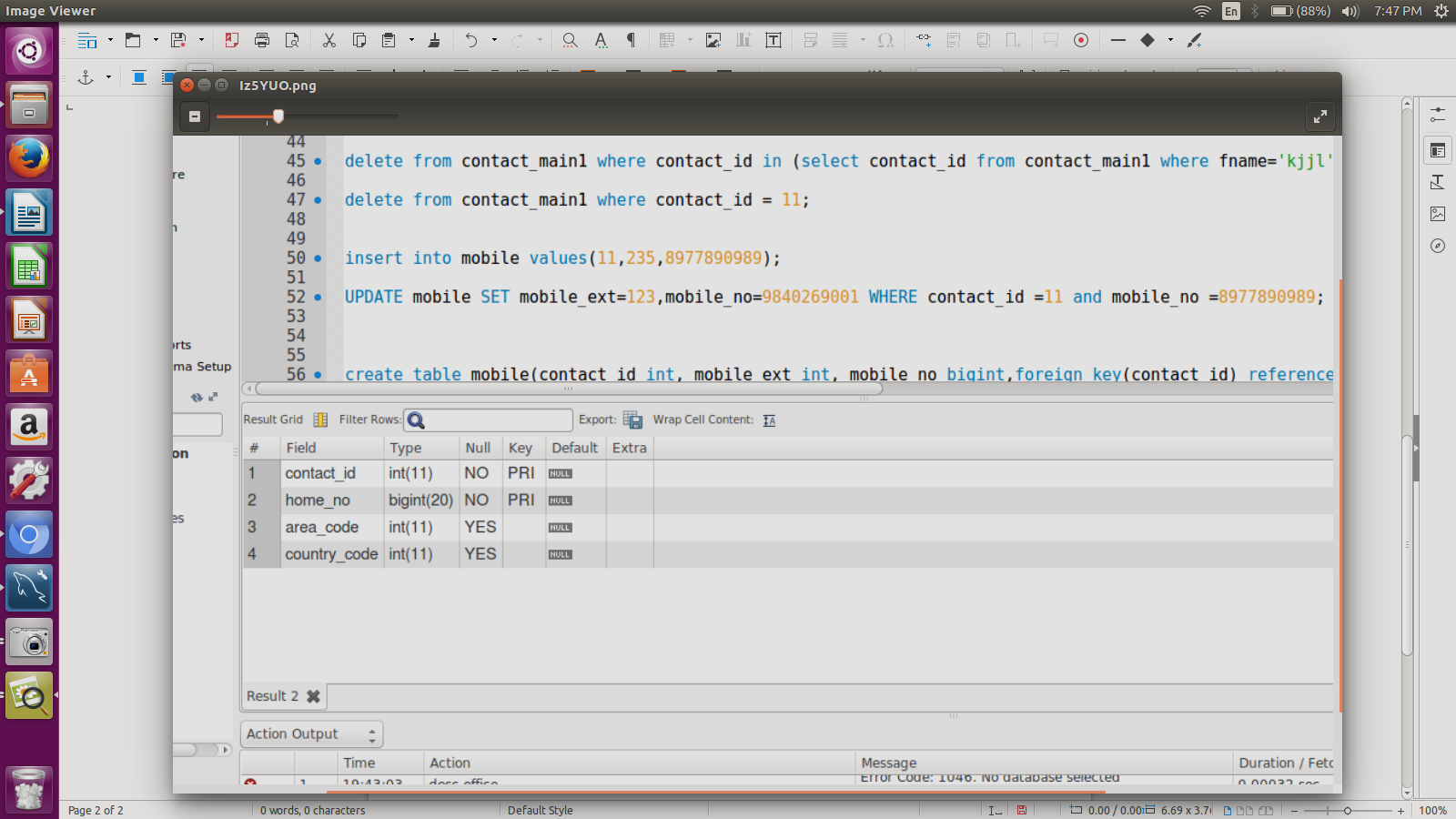
Contact\_main



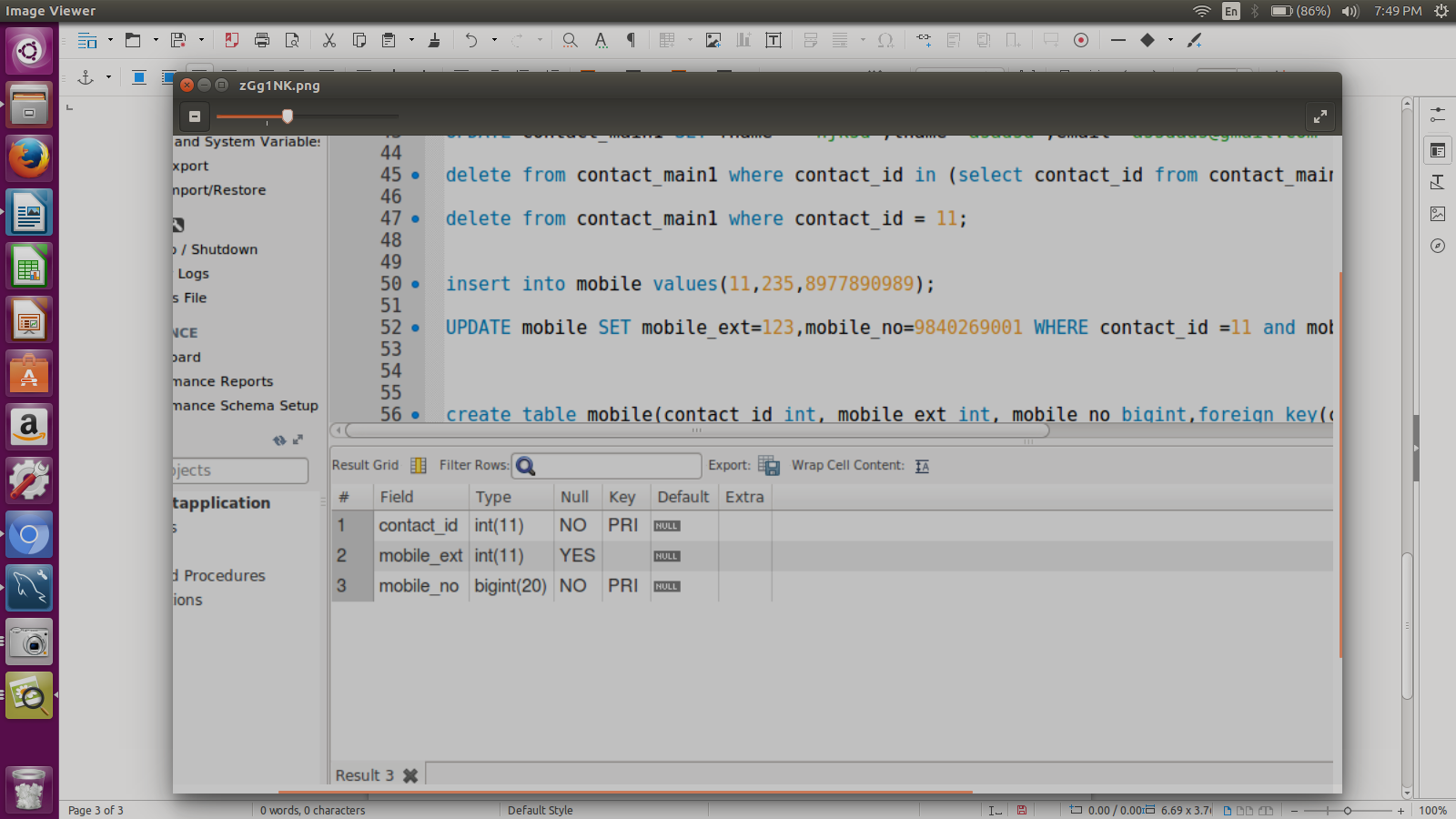
Office



Home



Mobile



### SOURCE CODE:

<https://github.com/sruthiviswanathan/Zterns-19-Sruthi/tree/master/Contact_application>