**JAVA DATABASE CONNECTIVITY(JDBC)**

**CONCEPTS LEARNED**

1. Fundamental steps in JDBC.
2. Creating a connection.
3. Creating a statement/prepared statement.
4. Executing statements and storing result set.
5. Processing the result set.
6. Closing the connection.
7. Problem statement.
8. Schema.
9. Functionality.

**FUNDAMENTAL STEPS IN JDBC**

1. Import JDBC packages.
2. Load and register the JDBC driver.
3. Open a connection to the database.
4. Create a statement/prepared statement object to perform a query.
5. Execute the statement/prepared statement object and return a query resultset.
6. Process the resultset.
7. Close the resultset and statement objects.
8. 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.

**SOLUTION**

**FUNCTIONALITY**

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

CREATE

◦ Get all the user credentials and insert into tables.

◦ Create a contactId for each new user automatically.

UPDATE

◦ 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.

DELETE

◦ 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.

READ

◦ 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.

**ENHANCED ENTITY-RELATIONSHIP DIAGRAM**

