++++++++
JDBC API
++++++++
 => It is an API given by SUNMS to interact with the database.
 => To interact with the database SUNMS had given a jar called rt.jar which is available for the programmer during the installation of jdk s/w.
 => To use JDBC in java program we take the support of a package called "java.sql.*" and "javax.sql.*".
 => API refers to set of rules and guideliness which has interfaces.
 => To get the implementation for these interface abstract methods we need to take the help of "DB-Vendor".
 => DB-Vendor will give the implementation for "SRS" and release those implementation classes in the form of "jars" to the java
    develop community people.
 => Depending upon the database we use in our project we need to use the respective jars supplied by "DB-Vendor".

++
Steps given by SUNMS to interact with Database
++
1. Load and register the driver
2. Establish the Connection
3. Create Statement/PreparedStatement/CallableStatement to transfer the query.
4. Execute the query
5. Process the result
6. close the resources
7. Handle SQLExceptions if it occurs.

To execute select Query we need to use
      public ResultSet executeQuery(String sqlQuery) throws SQLException

To execute non-select query like insert,update,delete we need to use
      public int executeUpdate(String sqlNonSelectQuery) throws SQLException

++++++++++++++++++++++++++
Code to execute SelectQuery
++++++++++++++++++++++++++
```
package in.pwskills.main;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

public class TestApp {

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

            // load and register the driver
            Class.forName("com.mysql.cj.jdbc.Driver");

            // Establish the Connection
            String url = "jdbc:mysql://localhost:3306/pwskillsbatch";
            String user = "root";
            String password = "root123";
```

```
            Connection connection = DriverManager.getConnection(url, user,
password);

            // Create a Statement object
            Statement statement = connection.createStatement();

            // Execute the query
            String sqlSelectQuery = "select sid,sname,sage,saddress from student";
            ResultSet resultSet = statement.executeQuery(sqlSelectQuery);

            System.out.println("SID\tSNAME\tSAGE\tSADDRESS");
            // Process the resultSet
            while (resultSet.next()) {
                  System.out.println(resultSet.getInt(1) + "\t" +
resultSet.getString(2) + "\t" + resultSet.getInt(3) + "\t"
                              + resultSet.getString(4));
            }

            //close the resources
            resultSet.close();
            statement.close();
            connection.close();

      }
}
```

++++++++++++++++++++++++++
Code to execute InsertQuery
++++++++++++++++++++++++++

```
package in.pwskills.main;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

public class TestApp {

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

            // load and register the driver
            Class.forName("com.mysql.cj.jdbc.Driver");

            // Establish the Connection
            String url = "jdbc:mysql://localhost:3306/pwskillsbatch";
            String user = "root";
            String password = "root123";
            Connection connection = DriverManager.getConnection(url, user,
password);

            // Create a Statement object
            Statement statement = connection.createStatement();

            // Execute the query
            String sqlInsertQuery = "insert into student(sid,sname,sage,saddress)
values(18,'kohli',49,'MI')";
            int rowAffected = statement.executeUpdate(sqlInsertQuery);
```

```
            //process the result
            if (rowAffected==1) {
                  System.out.println("No of rows affected is :: "+rowAffected);
            } else {
                  System.out.println("Not succesfull in insertion...");
            }

            //close the resources
            statement.close();
            connection.close();

      }
}
```

++++++++++++++++++++++++++++
Code to execute delete Query
++++++++++++++++++++++++++++

```
package in.pwskills.main;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.sql.Statement;

public class TestApp {

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

            // load and register the driver
            Class.forName("com.mysql.cj.jdbc.Driver");

            // Establish the Connection
            String url = "jdbc:mysql://localhost:3306/pwskillsbatch";
            String user = "root";
            String password = "root123";
            Connection connection = DriverManager.getConnection(url, user,
password);

            // Create a Statement object
            Statement statement = connection.createStatement();

            // Execute the query
            String sqlDeleteQuery = "delete from student where sid = 9";
            int rowAffected = statement.executeUpdate(sqlDeleteQuery);

            //process the result
            if (rowAffected==1) {
                  System.out.println("No of rows deleted are :: "+rowAffected);
            } else {
                  System.out.println("No record found for deletion");
            }

            //close the resources
            statement.close();
            connection.close();
      }
}
```

```
++++++++++++++++++++
Code for Update Query
++++++++++++++++++++
package in.pwskills.main;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.sql.Statement;

public class TestApp {

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

		// load and register the driver
		Class.forName("com.mysql.cj.jdbc.Driver");

		// Establish the Connection
		String url = "jdbc:mysql://localhost:3306/pwskillsbatch";
		String user = "root";
		String password = "root123";
		Connection connection = DriverManager.getConnection(url, user,
password);

		// Create a Statement object
		Statement statement = connection.createStatement();

		// Execute the query
		String sqlUpdateQuery = "update student set sname='sachin' where sid =
10";
		int rowAffected = statement.executeUpdate(sqlUpdateQuery);

		//process the result
		if (rowAffected==1) {
			System.out.println("No of rows updated are :: "+rowAffected);
		} else {
			System.out.println("No record found for upation");
		}

		//close the resources
		statement.close();
		connection.close();

	}
}

++++++++++++++++++++++++++++++++++++++++
Industry standard way of writing the code
++++++++++++++++++++++++++++++++++++++++
package in.pwskills.utility;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
```

```
public class JdbcUtil {

	static {
		// load and register the driver
		try {
			Class.forName("com.mysql.cj.jdbc.Driver");
		} catch (ClassNotFoundException e) {
			e.printStackTrace();
		}
	}

	public static Connection getDbConnection() throws SQLException {
		// Establish the Connection
		String url = "jdbc:mysql://localhost:3306/pwskillsbatch";
		String user = "root";
		String password = "root123";
		return DriverManager.getConnection(url, user, password);
	}

	public static void closeResources(ResultSet resultSet, Statement statement,
Connection connection)
			throws SQLException {

		if (resultSet != null)
			resultSet.close();

		if (statement != null)
			statement.close();

		if (connection != null)
			connection.close();
	}

}
```

TestApp.java
============

```
package in.pwskills.main;

import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

import in.pwskills.utility.JdbcUtil;

public class TestApp {

	public static void main(String[] args) {

		//Resources used
		Connection connection = null;
		Statement statement = null;
		ResultSet resultSet = null;

		try {
			//Getting the connection
			connection = JdbcUtil.getDbConnection();
```

```
                  if (connection != null)
                        statement = connection.createStatement();

                  if (statement != null)
                        resultSet = statement.executeQuery("select
sid,sname,sage,saddress from student");

                  if (resultSet != null) {
                        System.out.println("SID\tSNAME\tSAGE\tSADDRESS");
                        while (resultSet.next()) {
                              System.out.println(resultSet.getInt(1) + "\t" +
resultSet.getString(2) + "\t" + resultSet.getInt(3)
                                          + "\t" + resultSet.getString(4));
                        }
                  }
            } catch (SQLException e) {
                  e.printStackTrace();
            } catch (Exception e) {
                  e.printStackTrace();
            } finally {
                  //Closing the resources
                  try {
                        JdbcUtil.closeResources(resultSet, statement, connection);
                  } catch (SQLException e) {
                        e.printStackTrace();
                  } catch (Exception e) {
                        e.printStackTrace();
                  }
            }
      }
}
```

++++++++++++++++++
PreparedStatement
++++++++++++++++++
 => These statements are used to transfer the query during the compilation only.
 => These statements are also called as Pre-Compiled Query.
 => Performance of these object are high compared to Statement object.
 => In case of PreparedStatement the query will be incomplete and the values will be injected to the query at the runtime.
 => Values to the PreparedStatement object will be injected using setXXXXX(int pos,xxxxx value) throws SQLException method.

Syntax ::
 insert into tb_name(col1,col2,...) values (?,?,?,?);
 select col1,col2,col3,... from tb_name where col1 =?
 update table set col2 = ? where coln = ?
 delete from table where col = ?

eg#1.

JdbcUtil.java
++++++++++++
```
package in.pwskills.utility;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
```

```
import java.sql.SQLException;
import java.sql.Statement;

public class JdbcUtil {

	static {
		// load and register the driver
		try {
			Class.forName("com.mysql.cj.jdbc.Driver");
		} catch (ClassNotFoundException e) {
			e.printStackTrace();
		}
	}

	public static Connection getDbConnection() throws SQLException {
		// Establish the Connection
		String url = "jdbc:mysql://localhost:3306/pwskillsbatch";
		String user = "root";
		String password = "root123";
		return DriverManager.getConnection(url, user, password);
	}

	public static void closeResources(ResultSet resultSet, Statement statement,
Connection connection)
			throws SQLException {

		if (resultSet != null)
			resultSet.close();

		if (statement != null)
			statement.close();

		if (connection != null)
			connection.close();
	}

}

TestApp.java
++++++++++++
package in.pwskills.main;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.Scanner;

import in.pwskills.utility.JdbcUtil;

public class TestApp {
	public static void main(String[] args) {

		// Resources used
		Connection connection = null;
		PreparedStatement pstmt = null;
		ResultSet resultSet = null;
		Scanner scanner = null;
```

```
            try {
                  // Getting the connection
                  connection = JdbcUtil.getDbConnection();

                  String sqlInsertQuery = "insert into
student(sid,sname,sage,saddress) values(?,?,?,?)";
                  if (connection != null)
                        pstmt = connection.prepareStatement(sqlInsertQuery);

                  if (pstmt != null) {
                        scanner = new Scanner(System.in);

                        System.out.print("Enter the value of sname :: ");
                        String sname = scanner.next();

                        System.out.print("Enter the value of sid :: ");
                        Integer sid = scanner.nextInt();

                        System.out.print("Enter the value of saddress :: ");
                        String saddress = scanner.next();

                        System.out.print("Enter the value of sage :: ");
                        Integer sage = scanner.nextInt();

                        // Setting the values for PreparedStatement
                        pstmt.setInt(1, sid);
                        pstmt.setString(2, sname);
                        pstmt.setInt(3, sage);
                        pstmt.setString(4, saddress);

                        int rowAffected = pstmt.executeUpdate();
                        if (rowAffected == 1) {
                              System.out.println("Insertion succesfull");
                        } else {
                              System.out.println("Record not inserted to
database...");
                        }
                  }
            } catch (SQLException e) {
                  e.printStackTrace();
            } catch (Exception e) {
                  e.printStackTrace();
            } finally {
                  // Closing the resources
                  try {
                        JdbcUtil.closeResources(resultSet, pstmt, connection);
                        scanner.close();
                  } catch (SQLException e) {
                        e.printStackTrace();
                  } catch (Exception e) {
                        e.printStackTrace();
                  }
            }

      }
}

++++++++++++++++++
Code for Updation
```

```
++++++++++++++++++
package in.pwskills.main;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.Scanner;

import in.pwskills.utility.JdbcUtil;

public class UpdateApp {

	public static void main(String[] args) {

		// Resources used
		Connection connection = null;
		PreparedStatement pstmt = null;
		ResultSet resultSet = null;
		Scanner scanner = null;

		try {
			// Getting the connection
			connection = JdbcUtil.getDbConnection();

			String sqlUpdateQuery = "update student set sname = ? where sid =
?";
			if (connection != null)
				pstmt = connection.prepareStatement(sqlUpdateQuery);

			if (pstmt != null) {

				scanner = new Scanner(System.in);

				System.out.print("Enter the value of sname :: ");
				String sname = scanner.next();

				System.out.print("Enter the value of sid :: ");
				Integer sid = scanner.nextInt();

				// Setting the values for PreparedStatement
				pstmt.setString(1, sname);
				pstmt.setInt(2, sid);

				int rowAffected = pstmt.executeUpdate();

				if (rowAffected == 1) {
					System.out.println("updation succesfull");
				} else {
					System.out.println("Record not available for updation
with the id::  "+sid);
				}
			}

		} catch (SQLException e) {
			e.printStackTrace();
		} catch (Exception e) {
			e.printStackTrace();
		} finally {
```

```
                  // Closing the resources
                  try {
                        JdbcUtil.closeResources(resultSet, pstmt, connection);
                        scanner.close();
                  } catch (SQLException e) {
                        e.printStackTrace();
                  } catch (Exception e) {
                        e.printStackTrace();
                  }
            }

      }
}
```

++++++++++++++++++
Code for Deletion
++++++++++++++++++

```
package in.pwskills.main;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.Scanner;

import in.pwskills.utility.JdbcUtil;

public class DeleteApp {

      public static void main(String[] args) {

            // Resources used
            Connection connection = null;
            PreparedStatement pstmt = null;
            ResultSet resultSet = null;
            Scanner scanner = null;

            try {
                  // Getting the connection
                  connection = JdbcUtil.getDbConnection();

                  String sqlDeleteQuery = "delete from student where sid = ? ";
                  if (connection != null)
                        pstmt = connection.prepareStatement(sqlDeleteQuery);

                  if (pstmt != null) {

                        scanner = new Scanner(System.in);


                        System.out.print("Enter the value of sid :: ");
                        Integer sid = scanner.nextInt();

                        // Setting the values for PreparedStatement
                        pstmt.setInt(1, sid);

                        int rowAffected = pstmt.executeUpdate();
```

```
                        if (rowAffected == 1) {
                              System.out.println("deletion succesfull");
                        } else {
                              System.out.println("Record not available for deletion
with the id::  "+sid);
                        }
                  }

            } catch (SQLException e) {
                  e.printStackTrace();
            } catch (Exception e) {
                  e.printStackTrace();
            } finally {

                  // Closing the resources
                  try {
                        JdbcUtil.closeResources(resultSet, pstmt, connection);
                        scanner.close();
                  } catch (SQLException e) {
                        e.printStackTrace();
                  } catch (Exception e) {
                        e.printStackTrace();
                  }
            }

      }
}

++++++++++++++++++
Code for Selection
++++++++++++++++++
package in.pwskills.main;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.Scanner;

import in.pwskills.utility.JdbcUtil;

public class SelectApp {

      public static void main(String[] args) {

            // Resources used
            Connection connection = null;
            PreparedStatement pstmt = null;
            ResultSet resultSet = null;
            Scanner scanner = null;
            Integer sid = null;

            try {
                  // Getting the connection
                  connection = JdbcUtil.getDbConnection();

                  String sqlSelectQuery = "select sid,sname,sage,saddress from
student where sid = ? ";
                  if (connection != null)
```

```
                        pstmt = connection.prepareStatement(sqlSelectQuery);

                  if (pstmt != null) {

                        scanner = new Scanner(System.in);

                        System.out.print("Enter the value of sid :: ");
                        sid = scanner.nextInt();

                        // Setting the values for PreparedStatement
                        pstmt.setInt(1, sid);

                        resultSet = pstmt.executeQuery();

                  }
                  if (resultSet != null) {
                        if (resultSet.next()) {
                              System.out.println("SID\tSNAME\tSAGE\tSADDRESS");
                              System.out.println(resultSet.getInt(1) + "\t" +
resultSet.getString(2) + "\t" + resultSet.getInt(3)
                                          + "\t" + resultSet.getString(4));
                        } else {
                              System.out.println("Record not available for the give
id :: "+sid);
                        }
                  }

            } catch (SQLException e) {
                  e.printStackTrace();
            } catch (Exception e) {
                  e.printStackTrace();
            } finally {

                  // Closing the resources
                  try {
                        JdbcUtil.closeResources(resultSet, pstmt, connection);
                        scanner.close();
                  } catch (SQLException e) {
                        e.printStackTrace();
                  } catch (Exception e) {
                        e.printStackTrace();
                  }
            }

      }
}
```

While working with database, we perform the following operations

a. C ----> Create(insert)
b. R ----> Read(select)
c. U ----> Update(updation)
d. D ----> Delete(deletion)

The above mentioned operations we call as "CRUD/CURD/SCUD" operation.

++++++++++++++++++++++++++
Working with Date Operation
++++++++++++++++++++++++++

```
 Formats of Date in MySQL  :: YYYY-MM-DD
 Formats of Date in Oracle :: DD-MM-YY
```

While writing the program, we expect the date information from the user as per there timezone,so while writing the code we can write
by keeping particular database in mind, so we following the conversion format to store the date information in database.

```
EndUser(Input) --SimpleDateFormat(parse)------> java.util.Date -------> java.sql.Date
                                                                    |->  use preparedStatement and Set Date ----> DB specific format.
```

++++++++++++++++++++++++++++++++++++
Code for Inserting the date in Database
++++++++++++++++++++++++++++++++++++
TestApp.java
++++++++++++

```
package in.pwskills.main;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.Scanner;

import in.pwskills.utility.JdbcUtil;

public class InsertApp {

	public static void main(String[] args) {

		// Resources used
		Connection connection = null;
		PreparedStatement pstmt = null;
		ResultSet resultSet = null;
		Scanner scanner = null;

		try {
			// Getting the connection
			connection = JdbcUtil.getDbConnection();

			String sqlInsertQuery = "insert into person(name,dob) values (?,?)";
			if (connection != null)
				pstmt = connection.prepareStatement(sqlInsertQuery);

			if (pstmt != null) {

				scanner = new Scanner(System.in);

				System.out.print("Enter the value of sname :: ");
				String sname = scanner.next();

				System.out.print("Enter the value of dob(dd-mm-yyyy) :: ");
```

```
                        String dob = scanner.next();


                        SimpleDateFormat sdf = new SimpleDateFormat("dd-MM-yyyy");
                        Date utilDate = sdf.parse(dob);

                        java.sql.Date sqlDate = new
java.sql.Date(utilDate.getTime());

                        // Setting the values for PreparedStatement
                        pstmt.setString(1, sname);
                        pstmt.setDate(2, sqlDate);


                        int rowAffected = pstmt.executeUpdate();

                        if (rowAffected == 1) {
                              System.out.println("Insertion succesfull");
                        } else {
                              System.out.println("Record not inserted to
database...");
                        }

                  }

            } catch (SQLException e) {
                  e.printStackTrace();
            } catch (Exception e) {
                  e.printStackTrace();
            } finally {

                  // Closing the resources
                  try {
                        JdbcUtil.closeResources(resultSet, pstmt, connection);
                        scanner.close();
                  } catch (SQLException e) {
                        e.printStackTrace();
                  } catch (Exception e) {
                        e.printStackTrace();
                  }
            }

      }
}

++++++++++++++++++++++++++++++++++++++++
Code for Retreiving the data from Database
++++++++++++++++++++++++++++++++++++++++
  => Database -----> java.sql.Date ------> use SimpleDateFormat(format)----->
String format output(end user)

TestApp.java
============
package in.pwskills.main;

import java.sql.Connection;
import java.sql.Date;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
```

```
import java.sql.SQLException;
import java.text.SimpleDateFormat;
import java.util.Scanner;

import in.pwskills.utility.JdbcUtil;

public class SelectApp {

	public static void main(String[] args) {

		// Resources used
		Connection connection = null;
		PreparedStatement pstmt = null;
		ResultSet resultSet = null;
		Scanner scanner = null;
		String sname = null;

		try {
			// Getting the connection
			connection = JdbcUtil.getDbConnection();

			String sqlSelectQuery = "select name,dob from person where name =
?";

			if (connection != null)
				pstmt = connection.prepareStatement(sqlSelectQuery);

			if (pstmt != null) {

				scanner = new Scanner(System.in);

				System.out.print("Enter the name of the person:: ");
				sname = scanner.next();

				// Setting the values for PreparedStatement
				pstmt.setString(1, sname);

				resultSet = pstmt.executeQuery();

			}
			if (resultSet != null) {
				if (resultSet.next()) {
					System.out.println("NAME\tDOB");
					Date sqlDate = resultSet.getDate(2);

					SimpleDateFormat sdf = new SimpleDateFormat("dd-MMM-
yyy");
					String stringDate = sdf.format(sqlDate);
					System.out.println(resultSet.getString(1)+"\t"+
stringDate);
				} else {
					System.out.println("Record not available for the give
name :: "+sname);
				}
			}

		} catch (SQLException e) {
			e.printStackTrace();
		} catch (Exception e) {
```

```
                e.printStackTrace();
            } finally {

                // Closing the resources
                try {
                    JdbcUtil.closeResources(resultSet, pstmt, connection);
                    scanner.close();
                } catch (SQLException e) {
                    e.printStackTrace();
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        }
}
output
Enter the name of the person:: divya
NAME  DOB
divya 02-Jan-2001
```