

**AMITY UNIVERSITY, PATNA**

**AMITY INSTITUTE OF INFORMATION TECHNOLOGY**

**Advanced Java Assignment**

**BCA**



Name : Aryan Amar  
Program/Semester : BCA – 6 ‘A’  
Enroll. Number : A45304821047  
Submitted to : Dr. Naveen Kumar Singh

**Problem statement :** Connect to the java program by using as many databases as you can.

**Introduction :**

Java, being a versatile programming language, provides extensive support for connecting to various types of databases. This assignment explores the process of connecting Java with six different databases: Oracle, PostgreSQL, MySQL, MariaDB, MS SQL Server, and SQLite. The assignment demonstrates connections to Oracle, PostgreSQL, MySQL, and MariaDB through a single project utilizing different packages, while connections to MS SQL Server and SQLite are demonstrated in separate projects.

**1. Connecting Java to Oracle, PostgreSQL, MySQL, and MariaDB:**

**a. Oracle Database Connection:**

- i. Utilize the Oracle JDBC driver to connect Java to an Oracle database.
- ii. Import the necessary packages for JDBC connectivity.
- iii. Establish a connection using the DriverManager class and the appropriate JDBC URL, username, and password.

**b. PostgreSQL Database Connection:**

- i. Import the PostgreSQL JDBC driver.
- ii. Set up the JDBC connection using DriverManager with the PostgreSQL-specific JDBC URL format.
- iii. Provide authentication credentials to establish the connection.

**c. MySQL and MariaDB Database Connections:**

- i. Import the JDBC drivers for MySQL and MariaDB respectively.
- ii. Establish connections using their respective JDBC URLs and authentication credentials.
- iii. Note any differences in JDBC URL formats or authentication mechanisms between MySQL and MariaDB.

**2. Connecting Java to MS SQL Server and SQLite:**

**a. MS SQL Server Connection:**

- i. Import the JDBC driver for MS SQL Server.

**ii.** Set up the connection using DriverManager with the appropriate JDBC URL format for MS SQL Server.

**iii.** Provide necessary authentication details to establish the connection.

**b. SQLite Database Connection:**

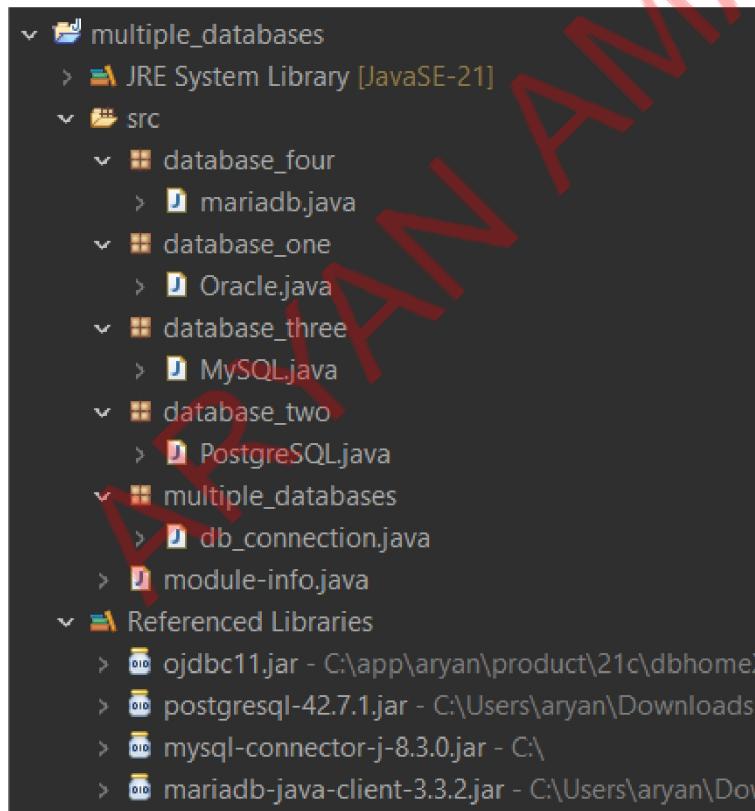
**i.** Import the JDBC driver for SQLite.

**ii.** Establish a connection to the SQLite database file using DriverManager with the file path as the JDBC URL.

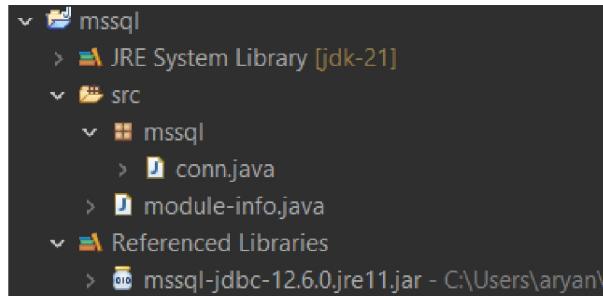
**iii.** Authenticate the connection if necessary.

**Design**

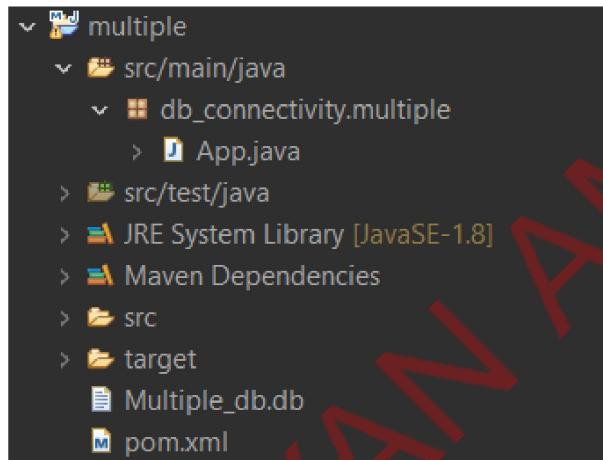
**Folder Structure (for oracle, mysql, postgresql and mariadb)**



### **Folder Structure (for ms sql)**



### **Folder Structure (for sqlite)**



## CODE

### db\_connection.java

```
package multiple_databases;

import java.sql.SQLException;
import java.util.Scanner;
import database_four.mariadb;
import database_one.Oracle;
import database_two.PostgreSQL;
import database_three.MySQL;

public class db_connection {

    public static void main(String[] args) throws
ClassNotFoundException, SQLException {
        // TODO Auto-generated method stub
        Scanner sc = new Scanner(System.in);
        //object for oracle database connectivity
        Oracle O = new Oracle();
        //object for PostgreSQL database connectivity
        PostgreSQL P = new PostgreSQL();
        //object for MySQL database connectivity
        MySQL M = new MySQL();
        //object for Mariadb database connectivity
        mariadb maria = new mariadb();
        while (true) {
            System.out.println("\nChoose the database you
want to connect to -");
            System.out.println("1. Oracle Database");
            System.out.println("2. MySQL");
```

## CODE

```
System.out.println("3. PostgreSQL");
System.out.println("4. MariaDB");
System.out.println("5. Exit");
System.out.println("\nPlease enter your choice
:");

int choice = sc.nextInt();

switch (choice) {
case 1:
    System.out.println("\nORACLE
Database.....\n");
    O.oracle_connectivity();
    break;
case 2:
    System.out.println("\nMySQL
Database.....\n");
    M.MySQL_connectivity();
    break;
case 3:
    System.out.println("\nPostgreSQL
Database.....\n");
    P.postgreSQL_connectivity();
    break;
case 4:
    System.out.println("\nMariaDB
Database.....\n");
    maria.Mariadb_connectivity();
    break;
case 5:
```

CODE

## Oracle.java

```
package database_one;

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

public class Oracle {

    public void oracle_connectivity() throws
ClassNotFoundException, SQLException

    {
        Class.forName("oracle.jdbc.driver.OracleDriver");

        String url = "jdbc:oracle:thin:@localhost:1521:xe";
        String username = "SYSTEM";
        String password = "12345678";
        Connection con = DriverManager.getConnection(url,username,password);
        Statement st = con.createStatement();
        String query = "select * from emp";
        ResultSet rs = st.executeQuery(query);
        while(rs.next())
        {
            System.out.println(rs.getString(1));
            System.out.println(rs.getString(2));
            System.out.println(rs.getString(3));
            System.out.println(rs.getString(4));
            System.out.println(rs.getString(5));
        }
    }
}
```

## CODE

```
String pwd = "aryan123";  
Connection con = DriverManager.getConnection(url, username,  
pwd);  
Statement st = con.createStatement();  
String query = "Create Table table_ora(t_id int, t_name  
varchar(50));"  
st.executeUpdate(query);  
System.out.println("Table has been created  
successfully!!!");  
}  
}  
  
PostgreSQL.java
```

```
package database_two;  
import java.sql.Connection;  
import java.sql.DriverManager;  
import java.sql.SQLException;  
import java.sql.Statement;  
  
public class PostgreSQL {  
    public void postgreSQL_connectivity() throws  
ClassNotFoundException, SQLException  
    {  
        Class.forName("org.postgresql.Driver");  
        String url =  
"jdbc:postgresql://localhost:5432/multiple_db";  
        String username = "postgres";  
        String pwd = "aryan123";  
        Connection con = DriverManager.getConnection(url, username,  
pwd);  
    }  
}
```

## CODE

```
Statement st = con.createStatement();

String query = "Create Table table_two(t_id int, t_name
varchar(50))";

st.executeUpdate(query);

System.out.println("Table has been created
successfully!!!!");

}

}
```

### MySQL.java

```
package database_three;

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

public class MySQL {

    public void MySQL_connectivity() throws
ClassNotFoundException, SQLException
    {
        Class.forName("com.mysql.cj.jdbc.Driver");
        String url = "jdbc:mysql://localhost:3306/multiple_db";
        String username = "root";
        String pwd = "aryan123";
        Connection con = DriverManager.getConnection(url, username,
pwd);
        Statement st = con.createStatement();
        String query = "Create Table table_three(t_id int, t_name
varchar(50))";
```

## CODE

```
        st.executeUpdate(query);

        System.out.println("Table has been created
successfully!!!");

    }

}
```

### mariadb.java

```
package database_four;

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

public class mariadb {

    public void Mariadb_connectivity() throws
ClassNotFoundException, SQLException

    {

        Class.forName("org.mariadb.jdbc.Driver");

        String url = "jdbc:mariadb://localhost:3307/multiple_db";
        String username = "root";
        String pwd = "aryan123";

        Connection con = DriverManager.getConnection(url, username,
pwd);

        Statement st = con.createStatement();

        String query = "Create Table table_maria(t_id int, t_
varchar(50))";

        st.executeUpdate(query);

        System.out.println("Table has been created
successfully!!!");
```

## CODE

```
}
```

### conn.java (for MS Sql Server)

```
package mssql;

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

public class conn {

    public static void main(String[] args) throws
ClassNotFoundException, SQLException {
        // TODO Auto-generated method stub

        Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");
        String connection_string = "jdbc:sqlserver://LAPTOP-
6EPRPCI6\\SQLEXPRESS;databaseName=Multiple_db;integratedSecurity
=true;encrypt=false;";
        Connection con =
        DriverManager.getConnection(connection_string);
        Statement st = con.createStatement();
        String query = "Create Table table_ms_sql(t_id int,
t_name varchar(50));";
        st.executeUpdate(query);
        System.out.println("Table has been created
successfully!!!!");
    }
}
```

AAMAN MAR(47)

## CODE

```
}
```

### App.java (for SQLite)

```
package db_connectivity.multiple;

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

public class App
{
    public static void main( String[] args ) throws
ClassNotFoundException, SQLException
    {
        Class.forName("org.sqlite.JDBC");
        Connection con =
DriverManager.getConnection("jdbc:sqlite:Multiple_db.db");
        Statement st = con.createStatement();
        String query = "Create Table table_light(t_id int, t_name
varchar(50));";
        st.executeUpdate(query);
        System.out.println("Table has been created
successfully!!!!");
    }
}
```

ANSWER(4)

## CODE

### pom.xml (for SQLite)

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">

    <modelVersion>4.0.0</modelVersion>

    <groupId>db_connectivity</groupId>
    <artifactId>multiple</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <packaging>jar</packaging>

    <name>multiple</name>
    <url>http://maven.apache.org</url>

    <properties>
        <project.build.sourceEncoding>UTF-
8</project.build.sourceEncoding>
    </properties>

    <dependencies>
        <dependency>
            <groupId>junit</groupId>
            <artifactId>junit</artifactId>
            <version>3.8.1</version>
            <scope>test</scope>
        </dependency>
        <!--
https://mvnrepository.com/artifact/org.xerial/sqlite-jdbc -->
```

## CODE

```
<dependency>
    <groupId>org.xerial</groupId>
    <artifactId>sqlite-jdbc</artifactId>
    <version>3.45.1.0</version>
</dependency>
<dependency>
    <groupId>org.apache.logging.log4j</groupId>
    <artifactId>log4j-api</artifactId>
    <version>2.20.0</version>
</dependency>
<dependency>
    <groupId>org.apache.logging.log4j</groupId>
    <artifactId>log4j-core</artifactId>
    <version>2.20.0</version>
</dependency>
<dependency>
    <groupId>org.apache.logging.log4j</groupId>
    <artifactId>log4j-slf4j-impl</artifactId>
    <version>2.20.0</version>
</dependency>
</dependencies>
</project>
```

## INPUT/OUTPUT

### Connection to oracle database

```
Problems Servers Terminal Data Source Explorer Properties Console × Progress
db_connection [Java Application] C:\Users\aryan\p2\pool\plugins\org.eclipse.jdt.core\src\main\java\com\example\db\connection.java
Choose the database you want to connect to -
1. Oracle Database
2. MySQL
3. PostgreSQL
4. MariaDB
5. Exit

Please enter your choice :
1
|
ORACLE Database.....
Table has been created successfully!!!
```

```
SQL> desc table_ora;
ERROR:
ORA-04043: object table_ora does not exist

SQL> desc table_ora;
Name          Null?    Type
-----        -----
T_ID          NUMBER(38)
T_NAME        VARCHAR2(50)
```

## INPUT/OUTPUT

### Connection to postgresql database

```
Choose the database you want to connect to -  
1. Oracle Database  
2. MySQL  
3. PostgreSQL  
4. MariaDB  
5. Exit  
  
Please enter your choice :  
3  
  
PostgreSQL Database.....  
  
Table has been created successfully!!!
```

```
postgres=# \c multiple_db  
You are now connected to database "multiple_db" as user "postgres".  
multiple_db=# \dt  
      List of relations  
 Schema |     Name      | Type | Owner  
-----+-----+-----+-----  
 public | postgresql_table | table | postgres  
 public | testtable    | table | postgres  
(2 rows)  
  
multiple_db=# \dt  
      List of relations  
 Schema |     Name      | Type | Owner  
-----+-----+-----+-----  
 public | postgresql_table | table | postgres  
 public | table_two     | table | postgres  
 public | testtable    | table | postgres  
(3 rows)  
  
multiple_db=# \d table_two  
          Table "public.table_two"  
 Column |       Type        | Collation | Nullable | Default  
-----+-----+-----+-----+-----  
 t_id  | integer         |           |          |  
 t_name | character varying(50) |           |          |
```

## INPUT/OUTPUT

### Connection to mysql database

```
Choose the database you want to connect to -  
1. Oracle Database  
2. MySQL  
3. PostgreSQL  
4. MariaDB  
5. Exit  
  
Please enter your choice :  
2  
  
MySQL Database.....  
  
Table has been created successfully!!!
```

```
mysql> use multiple_db;  
Database changed  
mysql> show tables;  
+-----+  
| Tables_in_multiple_db |  
+-----+  
| emp_details           |  
| employee               |  
| ipl_teams              |  
| mysql_table            |  
| politics                |  
+-----+  
5 rows in set (0.02 sec)  
  
mysql> show tables;  
+-----+  
| Tables_in_multiple_db |  
+-----+  
| emp_details           |  
| employee               |  
| ipl_teams              |  
| mysql_table            |  
| politics                |  
| table_three             |  
+-----+  
6 rows in set (0.00 sec)  
  
mysql> desc table_three;  
+-----+-----+-----+-----+-----+-----+  
| Field   | Type    | Null | Key | Default | Extra |  
+-----+-----+-----+-----+-----+-----+  
| t_id    | int     | YES  |      | NULL    |       |  
| t_name  | varchar(50) | YES  |      | NULL    |       |  
+-----+-----+-----+-----+-----+-----+  
2 rows in set (0.01 sec)
```

## INPUT/OUTPUT

### Connection to mariadb database

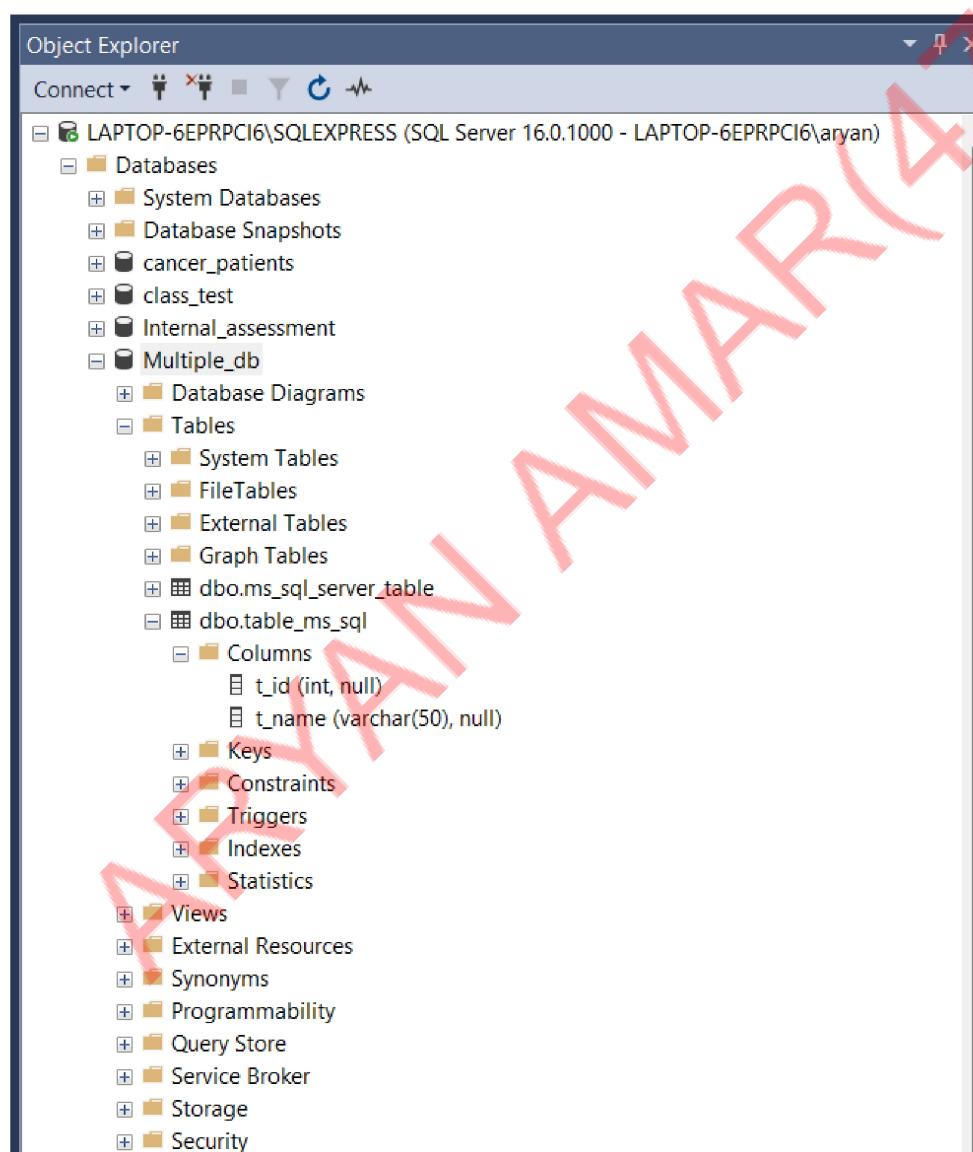
```
Choose the database you want to connect to -  
1. Oracle Database  
2. MySQL  
3. PostgreSQL  
4. MariaDB  
5. Exit  
  
Please enter your choice :  
4  
|  
MariaDB Database.....  
  
Table has been created successfully!!!
```

```
MariaDB [multiple_db]> show tables;  
+-----+  
| Tables_in_multiple_db |  
+-----+  
| mariadb_table          |  
| table_four             |  
+-----+  
2 rows in set (0.007 sec)  
  
MariaDB [multiple_db]> show tables;  
+-----+  
| Tables_in_multiple_db |  
+-----+  
| mariadb_table          |  
| table_four             |  
| table_maria            |  
+-----+  
3 rows in set (0.001 sec)  
  
MariaDB [multiple_db]> desc table_maria;  
+-----+-----+-----+-----+-----+  
| Field   | Type      | Null | Key | Default | Extra |  
+-----+-----+-----+-----+-----+  
| t_id    | int(11)   | YES  |     | NULL    |       |  
| t_name  | varchar(50)| YES  |     | NULL    |       |  
+-----+-----+-----+-----+-----+  
2 rows in set (0.025 sec)
```

# INPUT/OUTPUT

## Connection to ms sql database

```
Problems Servers Terminal Data Source Explorer Properties Console X Progress
<terminated> conn [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (Feb 18, 2024, 11:22:08 AM)
Table has been created successfully!!!
```



## **INPUT/OUTPUT**

## Connection to sqlite database

Table has been created successfully!!!

```
CREATE TABLE table_light(t_id int, t_name varchar(50))k  
/♦tabletable_li
```