

The background of the slide is a solid light brown color, decorated with a repeating pattern of dark brown coffee beans. Each bean has a lighter brown, curved line running down its center, and a soft shadow is cast to its right, giving it a three-dimensional appearance.

JDBC-CRUD

Operations

By Poojith sai Geesala

Enrolment no: EBEON0322579270

Aim: To Perform (CRUD) Create, Read, Update, Delete operations in IDE (Eclipse) by using MySQL database and java JDBC in Eclipse IDE.

Requirements:

- *JDK (Java SE 6 to Java SE 18) supported
- *Eclipse IDE
- *MySQL Workbench
- *MySQL JDBC DRIVER Type -4 (Native Protocol)

Note:

- Type-4 driver is also called native protocol driver. This driver interacts directly with the database. It does not require any native database library, which is why it is also known as Thin Driver.
- Does not require any native library and Middleware server, so no client-side or server-side installation.
- It is fully written in Java language, hence they are portable drivers.

JDBC Basic steps:

- Import the necessary classes
- Load the JDBC driver
- Identify the data source (Define the Connection URL)
- Establish the Connection
- Create a Statement Object
- Execute query string using Statement Object
- Retrieve data from the returned ResultSet Object

Procedure: (Procedure to create the program)

- Create a Database from workbench
- Create a Table – Employ from workbench
- Load the driver in your java program by using Driver Class Name
- The loaded class will throw Exception So Surround the Class.forName() with try-catch block.

```
Class.forName("com.mysql.cj.jdbc.Driver");  
  
try  
{  
    Class.forName("com.mysql.cj.jdbc.Driver");  
}  
catch (ClassNotFoundException e)  
{  
    System.out.println(e.getMessage());  
    e.printStackTrace();  
}
```

- Create a Connection Object and Statement Object in Try-with-Recourses Statement.

```
try(Connection con = DriverManager.getConnection  
    ("jdbc:mysql://localhost:3306/database1jdev","root","Poojith@13");  
    Statement St=con.createStatement();)
```

- Write the query in the String like as in MySQL workbench.

```
String query = "insert into employ  
                values(10,'Saikiran',2021,'capgemini')";
```

- Execute the query by using executeQuery() and executeUpdate() method by passing the String.

- To get the

```
String query = "insert into employ  
                values(11,'Sachin',2021,'capgemini')"
```

```
St.executeUpdate(query);
```

```
String query = "select * from employ";
```

```
St.executeQuery(query);
```

- Don't forget to Print all the caught Exceptions i.e. SQL and Non-SQL Exceptions

```
catch(SQLException Se)
{
    System.out.println(Se.getMessage());
    Se.printStackTrace();
}
catch (Exception e)
{
    System.out.println(e.getMessage());
    e.printStackTrace();
}
```

- To get the operation feedback Create an if-else condition and the condition will be the executionUpdate() method as an int value.

```
int count = St.executeUpdate(query);
if(count==0)
```

```
{  
    System.out.println("Not Inserted");  
}  
else  
{  
    System.out.println("Inserted");  
}
```

- This is the same steps for INSERT,DELETE and UPDATE.
- The Select query requires ResultSet object, getString() and While loop to print required number of Columns.

```
ResultSet rs=St.executeQuery(query)  
while(rs.next())  
{  
    System.out.println(rs.getInt(1)+" "+rs.getString(2));  
}
```


- Run every Method of Crud operations one by one

```
Employee E= new Employee();  
System.out.println("***Employ Details***");  
  
System.out.println("***please enter numbers between(1-4)  
                    for the CRUD operation you required ;) ***");  
  
System.out.println("***Enter corresponding number(1.Create/Insert,  
                    2.Read/Select,3.Update,4.Delete)***");  
  
Scanner s=new Scanner(System.in);  
int Opp=s.nextInt();  
switch(Opp)  
{  
case 1:  
    E.insert();  
    break;  
case 2:  
    E.Select();  
    -  
    -  
}
```

```
        E.update();  
        break;  
case 4:  
    E.delete();  
    break;  
default:  
    System.out.println("***Invalid Input please  
                        Enter numbers (1-4)***");  
}  
s.close();
```

- This is a basic program so it is static i.e. the queries can't be inserted into compiler at runtime
- Enter different Queries Of CRUD-Create or Insert, Read or Select, Update, Delete inside of the Query strings and run the SQL through the Eclipse

Thank You