Course Name: A Level (2nd Sem) **Subject: JAVA**
Topic: JDBC(Part 4) **Date: 25-06-20**

2)To Read the data from existing table:

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

public class Read
{
public static void main(String arg[])
{
try
 {
Class.forName("com.mysql.jdbc.Driver");
Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/test","root"
,"root");
Statement stmt = con.createStatement();
System.out.println("Created DB Connection....");

ResultSet rs = stmt.executeQuery("select name, salary from emp");

while(rs.next())
{
System.out.println(rs.getString("name"));
System.out.println(rs.getInt("salary"));
}
rs.close();
con.close();
} catch (ClassNotFoundException e)
{
e.printStackTrace();
}
catch (SQLException e)
{
e.printStackTrace();
}
}
```

The **result set object** maintains the cursor to the row of a table, initially it is at first

3)Select statement

The select statement is used to fetch data stored in relational databases.

```
public class Select {
public static void main (String[] args) {
try {
String url = "com.mysql.jdbc.Driver";
Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/test","root","root");
Statement stmt = conn.createStatement();
ResultSet rs;
rs = stmt.executeQuery("SELECT name FROM emp WHERE salary =44444");
while ( rs.next() )
 {
String lastName = rs.getString("name");
System.out.println(lastName);
}
conn.close();
} catch (Exception e) {

System.out.println("Got an exception! ");
System.out.println(e.getMessage());
}
}
}
```

PreparedStatement In Java

PreparedStatement is a special type of Statement object which is used to execute parameterized SQL queries or Dynamic SQL queries. Parameterized SQL queries look like this,

select * from STUDENT where ID = ?; where '?' is called Parameter Index or Place Holder for real parameters to be passed while executing this query.

1)To Create Table using prepared statement:

```
public class CreateTable {
public static void main(String[] args) {
try {
Class.forName("com.mysql.jdbc.Driver");
Connection con =DriverManager.getConnection("jdbc:mysql://localhost:3306/test",
"root","root");
String sql = "create table emp5(name varchar(25),department varchar(5),salary
varchar(25))";
PreparedStatement stmt = con.prepareStatement(sql);
stmt.executeUpdate(sql);
con.close();
System.out.println("table created");
}
catch(SQLException sqlex){
sqlex.printStackTrace();
}
catch (ClassNotFoundException e) {
e.printStackTrace();
}
}}
```

2)Insert Data in table using prepared statement

```
public class PSC {
public static void main(String a[]){
try {
Class.forName("com.mysql.jdbc.Driver" );
Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/test","root","root");
String query = "insert into emp(name,salary) values(?,?)";
PreparedStatement prSt = con.prepareStatement(query);
prSt.setString(1, "c");
prSt.setInt(2, 10000);
//count will give you how many records got updated
int count = prSt.executeUpdate();
//Run the same query with different values
prSt.setString(1, "CROWN1");
prSt.setInt(2, 5000);
count = prSt.executeUpdate();
con.close();
} catch (ClassNotFoundException e) {
e.printStackTrace();
} catch (SQLException e) {
e.printStackTrace();
}}}
```