protocol:subprotocol://localhost:port/dbname

(org.apache.derby.jdbc.ClientDriver) localhost:1527

private static String url="jdbc:derby://localhost:1527/InternCogDb;user=cog;password=pass@123";

(com.mysql.jdbc.Driver) localhost:3306

private static String url="jdbc:mysql://localhost:3306/InternCogDb;user=root;password=pass@123";

In DBMS , a prepared statement or parameterized statement is used for executing the repeated set of statements with good efficiency. (Template) :- values for the parameters are substituted during each execution or at run time.

1. Prepare :- at first, the application creates the statement / template and sends it to the server. The certain values are left undefined, called parameters. (?) :- undefined, placeholder

Insert into authors (authorid, authorname) values (?,?)

1. Parsing, compiling and translating
2. Execute (pass the values for the placeholder)

PreparedStatment :- is an interface , which is a sub interface of Statement. it is used for Parameterized query.

The performance of an application will be faster using prepared statement, because the query is compiled only once.

PreparedStatment Interface has various methods

1. Sets the integer value to the given parameter index.

setInt(int paramindex, int value); //setFloat(, float )

*setString*(int paramindex, String value); //setDouble(, double)

int executeUpdate(); (ddl / dml) // number of rows affected (int)

ResultSet executeQuery(); //retrieving data in the form of Result set

create void display(int empid) {

String query=”Select \* from employees where employeeid =?”;

}

public void insertEmployees(String id, String name, String address,int age, int salary)

{

/\*

String sql=”insert into Employees (employeeid,employeename,address,employeeage,employeesalary) values (‘“ + id + “’,’” + name + “’,’” + address + “’,” +age + “,” +salary+ “)”;

\*/

/// simplest way of writing a query using parameters (PreparedStatment)

String sql=”insert into Employees (employeeid,employeename,address,employeeage,employeesalary)

values (?,?,?,?,?)”

//con 🡪 Connection object

PreparedStatement pstmt= con.prepareStatment(sql);

pstmt.setString(1,”E001”);

pstmt.setString(2,”Aston”);

pstmt.setString(3,”Dadar”);

pstmt.setInt(4,20);

pstmt.setInt(5,25000);

int count=pstmt.executeUpdate();

sop(count + “ Rows affected”);

== Updating a record

String q=”Update employee set name=? where id=?”;

PreparedStatement ps=con.prepareStatement(q);

ps.setString(1,”rashi”);

ps.setInt(2,101);

int c=ps.executeUpdate();

sop(c + “ Rows affected”);

=================================== Retrieving the record

String q=”select \* from emp where city in (?)”;

PreparedStatement ps=con.prepareStatement(q);

Ps.setArray(1,arr);

ResultSet rs=ps.executeQuery();

while(rs.next())

{

Sop(rs.getInt(1) + “\t” + rs.getString(2));

}

========================Storing Images

Create table Student (studentid int, studentphoto varbinary(max))

PreparedStatement pstmt=con.prepareStatement(“insert into Student values (?,?)”);

FileInputStream fis= new FileInputStream(“D://student//student1.jpg”);

pstmt.setInt(1,10);

pstmt.setBinaryStream(2,fis,fis.available());

int count=pstmt.executeUpdate();

create table Emp (empid int, empname varchar(20))

create proc prcInsert @e int , @name varchar(20)

as

begin

insert into Emp values(@e,@name)

end

**============= calling stored procedure in JDBC  
CallableStatement interface :- is to call stored procedures and functions**

**//Connection con……. Class.forName()**

**con.AutoCommit(false);**

**CallableStatement cs=con.prepareCall(“{call prcInsert(?,?)}”);**

**cs.setInt(1,1);**

**cs.setString(2,”Aston”);**

**cs.execute(); //cs.executeBatch();**

con.commit();

con.close();

}