



Uttara InfoSolutions

www.uttarainfo.com

JDBC Practicals

0) Copy TestPS.java as it contains all the sample code you will need for this lab.

1) First install HSQLDB and test it (Look into HowToHSQLDB.txt for further info). Start the server (runServer.bat) and GUI (runManagerSwing.bat -> select 3rd option).

2) Then test SQL Create/insert/update/delete/select data from tables.

Ex:

```
create table names(fname varchar(100),lname varchar(100))
```

```
insert into names(fname,lname) values('amitab','bacchan')
```

```
update names set fname='salman' where lname='bhai'
```

```
delete from names
```

```
select * from names order by fname
```

3) Eclipse->File->New->Java Project -> TestJDBC. Set the dependencies (copy the driver jar file into the classpath) -> right click on project -> create new folder -> name it lib -> copy paste the hsqldb.jar from the lib folder of hsqldb folder into this. Right click on project -> properties -> build path -> libraries ->add jars -> select project -> select lib folder

Right click-> file -> new -> java class -> TestDb.java -> main()

a) Load the driver (by doing Class.forName()) and loading class by looking into documentation). You can click connect in GUI of hsqldb and choose 3rd option. It will give you the details of URL, driver name, uid and password.

b) Establish connection by using DriverManager. Test whether connection is established!

```

Class.forName("org.hsqldb.jdbc.JDBCDriver");

con=DriverManager.getConnection("jdbc: hsqldb:hsql://
localhost/", "SA", "");

System.out.println(con);

```

Test the code. If there are any exceptions, identify problem (if classnotfound, problem is driver jar is not in classpath. If connection ex, then db is not running. If SQLException, maybe the url, uid, pwd are wrong)

Once that is done, embed correct try..catch..finally:

Before try, declare your resource variables (Connection, ResultSet, PreparedStatement, etc to null). Inside try, initialize them. Then apply business logic. In catch, catch SQLException and invoke e.printStackTrace(). In finally, test the resource variable for null and then call close() on it. Embed finally block code also in try..catch(SQLException e)

Apply the logic to create PreparedStatement for insert, update, delete and for select. Using the ResultSet loop over the results and display on the console.

Ex: Assuming there is a table called enemies with name and reason as columns.

```
ps_ins = con.prepareStatement("insert into enemies(name,reason) values(?,?)");
```

```
String name = <take input>
```

```
String reason = <take input>
```

```
ps_ins.setString(1, name);
```

```
ps_ins.setString(2, reason);
```

```
ps_ins.execute();
```

```
ps_sel = con.prepareStatement("select * from enemies");
```

```
ps_sel.execute();
```

```
while(rs.next())
```

```
{
```

```
    name = rs.getString("name");
```

```
    reason = rs.getString("reason");
```

```
    System.out.println("Name : "+name+" Reason : "+reason);
```

```
}
```

Test the code. Verify if you understand how the code works. If you have any doubts, ask the Lab Instructor!

Create update and delete PreparedStatements as well and see how it works.

Build a Helper class with static methods getConnection() and close() overloaded methods. Use this in your main method.

4) Now build a menu based program to allow user to enter fname,lname of his friends into the db (you can reuse what you have built earlier to do this but try to not copy paste code from the demoed ex java file):

Press 1 to enter names

Press 2 to delete a name

Press 3 to update a name

Press 4 to display the names

Press 5 to exit

5) I would like you to create a simple contacts application. Create a table contacts with following columns: sl_no,name,email,phone,dob, created date. You should store phone number in a separate table as one contact can have many phone numbers. That table can be called contact_phoneinfo with columns sl_no(PK), contacts_sl (FK), phone (char), type (varchar indicating mobile/landline/office etc). If you do not know how to retrieve generated primary key to store in secondary table, store data into first table only.

Now build a Menu based main() application to provide a menu to User for him to add/delete/ retrieve/update contacts info in the db. Give him the ability to display based on sorting by name, dob, created date.