Ανοικτά Ακαδημαϊκά Μαθήματα

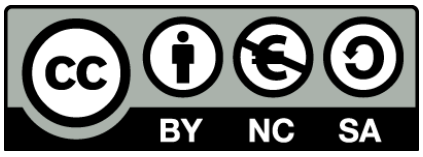
Τεχνολογικό Εκπαιδευτικό Ίδρυμα Αθήνας

Βάσεις Δεδομένων και Web-based Εφαρμογές

Ενότητα: Επισκόπηση της χρήσης του jdbc API

Χ. Σκουρλάς

Τμήμα Μηχανικών Πληροφορικής ΤΕ



Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.

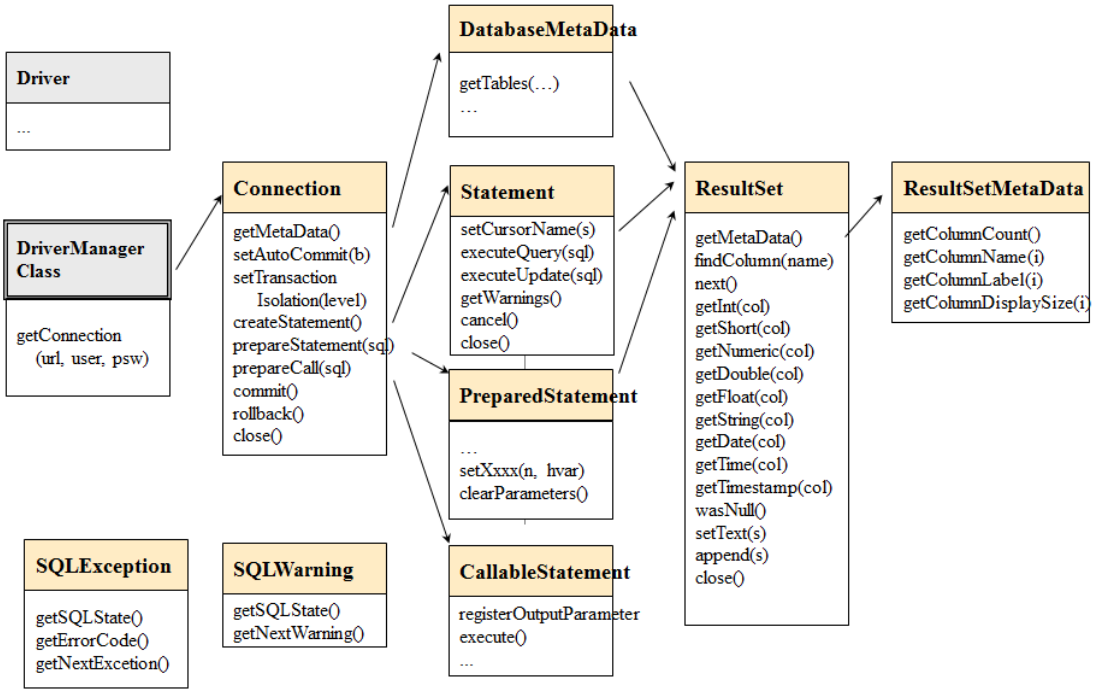
Περιεχόμενα

Στόχος της ενότητας είναι η επισκόπηση της χρήσης του jdbc API για τη δημιουργία εφαρμογών – jsp pages και τη διαχείριση βάσεων δεδομένων. Μετά την επεξεργασία της ενότητας ο ενδιαφερόμενος θα έχει κατανοήσει τα θέματα δημιουργίας μιας εφαρμογής jsp pages.

Λέξεις κλειδιά: jdbc API, netbeans, jsp pages

1. Επισκόπηση της χρήσης jdbc API

Στο σχήμα (Martii Leiho) προσφέρεται μία οπτική υπενθύμιση για μία σειρά από σημαντικές κλάσεις, αντικείμενα, μεθόδους (class, objects, methods) του jdbc API και με τα βέλη περιγράφεται η επικοινωνία / σύνδεσή / διάταξή τους.



Η βασική αρχή χρήσης του jdbc API είναι απλή:
Ας εστιάσουμε στα βέλη. Για να ορίσω ένα αντικείμενο μιάς κλάσης στην αιχμή ενός βέλους χρησιμοποιώ την κατάλληλη μέθοδο και αντικείμενο της «προηγούμενης» κλάσης που βρίσκεται στην αρχή του ιδίου βέλους.

```
Statement myStatement = myconnection.createStatement();
```

Δηλαδή για να ορίσουμε το αντικείμενο myStatement «εφαρμόζουμε» τη μέθοδο createStatement στο αντικείμενο myconnection.

Τα αντικείμενα της βάσης δεδομένων ορίζονται σαν συμβολοσειρές (string). Ακολουθούν παραδείγματα:

```
String testDatabase =
“jdbc:mysql://localhost:3306/mydb?user=admin&password=1234";
```

Προσοχή! Η παρακάτω εντολή είναι λανθασμένη

String sqlString = "INSERT INTO USERS VALUES ('user1', 'password1')";

Η σωστή διαμορφώνεται ως εξής: **… VALUES('"+user1+"','"+password1+"')**

Άρα, **String sql="insert into users values ('"+user1+"','"+password1+"') ";**

2. Απλά λειτουργικά παραδείγματα χρήσης του API:

Η DriverManagerClass παρέχει τη μέθοδο getConnection. Μπορούμε να ορίσουμε ένα αντικείμενο, για παράδειγμα το αντικείμενο myConnection τύπου Connection, ως εξής:

```
Connection myConnection = DriverManager.getConnection(testDatabase);
```

Αυτό προϋποθέτει ότι έχουμε ορίσει ένα αντικείμενο testDatabase τύπου string που «ορίζει» τον driver που χρησιμοποιούμε, το URL, τη βάση μας, username και password. Άρα θα έχουμε:

```
String testDatabase =
"jdbc:mysql://localhost:3306/mydb?user=admin&password=1234";
Connection myConnection = DriverManager.getConnection(testDatabase);
```

Σημείωση: Το αντικείμενο myConnection είναι τύπου Connection και μπορούμε να χρησιμοποιήσουμε τη μέθοδο CreateStatement της κλάσης Connection για να ορίσουμε ένα αντικείμενο, για παράδειγμα το αντικείμενο myStatement τύπου Statement. Μέσω του αντικειμένου myStatement θα γράφουμε και θα εκτελούμε δηλώσεις SQL (INSERT, UPDATE, DELETE, SELECT).

```
Statement myStatement = myconnection.createStatement();
```

Το αντικείμενο myStatement είναι τύπου Statement και μπορούμε να το «εκτελέσουμε» χρησιμοποιώντας μία από τις δύο σχετικές μεθόδους της κλάσης δηλαδή τη μέθοδο executeUpdate ή τη μέθοδο executeQuery. Ποια είναι κατάλληλη μέθοδος εξαρτάται από τη δήλωση SQL. Στην περίπτωση δήλωσης INSERT/UPDATE/DELETE έχουμε:

```
myStatement.executeUpdate(sqlString);
```

Η συγγραφή της SQL δήλωσης γίνεται σε ένα αντικείμενο τύπου string, για παράδειγμα στο αντικείμενο με όνομα sqlString.

```
String sqlString = "INSERT INTO USERS VALUES ('"+user1+"','"+password1+"')
('user1', 'password1')";
```

Έτσι για να εκτελέσουμε μία SQL δήλωση:

```
String sql="insert into users  values ('"+user1+"','"+password1+"')";
myStatement.executeUpdate(sqlString);
```

Ειδικά στην περίπτωση δήλωσης SELECT πρέπει να ορίσουμε ένα αντικείμενο, για παράδειγμα το αντικείμενο με όνομα rs, τύπου ResultSet. Στο αντικείμενο «αποθηκεύονται» τα αποτελέσματα της δήλωσης SELECT. Προηγουμένως, θα χρειασθεί να ορίσουμε το αντικείμενο sqlString που είναι μια συμβολοσειρά που «περιέχει» τη δήλωση SELECT:

```
String sqlString = "SELECT * FROM MYTABLE WHERE PARAM = '"+paramValue+"'";
ResultSet rs = myStatement.executeQuery(sqlString);
```

Η κλάση ResultSet παρέχει μεθόδους για να διαχειριστούμε τις γραμμές που «αποθηκεύονται» στο rs:

next(), previous(), first(), last()

Επίσης, παρέχει μία σειρά από μεθόδους κάθε μία από τις οποίες μπορεί να χρησιμοποιηθεί για να πάρουμε πληροφορία από μια στήλη:

getString, getInt, getFloat, getDate.

3. Σύνδεση σε βάση δεδομένων στην περίπτωση του προϊόντος MySQL

Για να συνδεθούμε στη βάση δεδομένων στην περίπτωση του προϊόντος MySQL και να γράψουμε δηλώσεις SQL (SQL statements) απαιτούνται οι εξής ενέργειες:

1. ***Δήλωση του JDBC driver***

libraries>δεξί κλικ> Add Jar/Folder και επιλέγουμε το αρχείο mysql-connector-java-5.1.13 -bin.jar ή mysql.jar

2. ***Εισαγωγή του package java.sql στο πρόγραμμά μας***

<%@page import="java.sql.*" %> (χρησιμοποιούμε .* έτσι ώστε να μπορούμε να χρησιμοποιήσουμε όλες τις κλάσεις του ***package java.sql***)

3. ***«Φόρτωση» του JDBC driver στο πρόγραμμά μας***

```
Class.forName("com.mysql.jdbc.Driver");
```

// Φορτώνουμε την κλάση Driver από το package *com.mysql.jdbc*.

4. **Σύνδεση στη βάση δεδομένων**

```
String testDatabase =
"jdbc:mysql://localhost:3306/mydb?user=admin&password=1234";
```

// Ορίζουμε για τη βάση δεδομένων μας (έστω ότι την ονομάζουμε mydb) ένα αντικείμενο testDatabase τύπου συμβολοσειράς (string) της μορφής: "*jdbc:mysql://ServerName:ServerPort/DatabaseName?user=myUsername&password=myPassword*";

```
Connection myConnection = DriverManager.getConnection(testDatabase);
```

// Δημιουργούμε ένα αντικείμενο τύπου *Connection*. Το αντικείμενό μας, myConnection, είναι μια ανοιχτή σύνδεση με τη βάση δεδομένων

5. **Δημιουργία αντικειμένου για να γράψουμε SQL statements**

```
Statement myStatement = myconnection.createStatement();
```

// Δημιουργούμε το αντικείμενο myStatement τύπου Statement. Μέσω του αντικειμένου myStatement θα γράφουμε δηλώσεις SQL.

6. ***Εγγραφή SQL statements***

Λανθασμένη εντολή

~~String sqlString = "INSERT INTO USERS VALUES ('user1', 'password1')";~~

Σωστή εντολή

String sql="insert into users values ('"+user1+"','"+password1+"')";

// Ορίζουμε ένα αντικείμενο sqlString τύπου string για να γράψουμε τη δήλωση SQL (που θέλουμε να εκτελεστεί στη συνέχεια).

7. ***Εκτέλεση SQL statements***

```
myStatement.executeUpdate(sqlString);
```

// Εκτελούμε τη δήλωση SQL που γράψαμε και πιο συγκεκριμένα χρησιμοποιούμε το αντικείμενο myStatement που δημιουργήσαμε. Επειδή έχουμε στην περίπτωση αυτή εντολή INSERT χρησιμοποιήσαμε τη μέθοδο executeUpdate. Το ίδιο θα κάναμε και σε δήλωση UPDATE ή DELETE.

Αν η sqlString αφορούσε μια SQL statement τύπου select, η μέθοδός μας θα ήταν η *executeQuery:*

```
myStatement.executeQuery(sqlString);
```

8. ***Κλείσιμο του αντικειμένου «αναγραφής» SQL statements***

```
myStatement.close();
```

// Αφού εκτελέσουμε την SQL statement πρέπει να κλείσουμε το αντικείμενο που χρησιμοποιείται για την εγγραφή της SQL statements.

9. ***Κλείσιμο της σύνδεσης***

```
myConnection.close();
```

// Κλείνουμε τη σύνδεση με τη βάση δεδομένων.

4. Παραδείγματα διαχείρισης δηλώσεων SQL - Insert-delete-update-select statements

Για να γράψουμε μια εντολή SQL, χρησιμοποιούμε τη μέθοδο

```
myStatement.methodName(sqlString);
```

όπου myStatement είναι ένα αντικείμενο τύπου Statement και methodName είναι μια από τις μεθόδους που παρέχει η κλάση Statement.

Θα μελετήσουμε διάφορες μορφές της μεθόδου methodName.

Στη συνέχεια, διορθώστε το διαγραμμένο μέρος των δηλώσεων.

Εισαγωγή δεδομένων (insert statement)

Χρησιμοποιούμε τη μέθοδο executeUpdate μέσω της εντολής:

```
myStatement.executeUpdate(sqlString);
```

όπου *sqlString* είναι μια συμβολοσειρά της μορφής:
`String sqlString = "INSERT INTO MYTABLE VALUES ('value1', 'value2')";`

Διαγραφή δεδομένων (delete statement)

Χρησιμοποιούμε τη μέθοδο executeUpdate μέσω της εντολής:

```
myStatement.executeUpdate(sqlString);
```

όπου το *sqlString* είναι της μορφής:

```
String sqlString = "DELETE FROM MYTABLE WHERE PARAM = '~~paramValue~~'";
```

Ενημέρωση δεδομένων (update statement)

Χρησιμοποιούμε τη μέθοδο executeUpdate μέσω της εντολής:

```
myStatement.executeUpdate(sqlString);
```

όπου το `sqlString` είναι της μορφής:

```
String sqlString = "UPDATE MYTABLE SET PARAM1='~~value1~~' WHERE
PARAM2='~~value2~~'";
```

Επιλογή δεδομένων (select statement)

Για επιλογή δεδομένων χρησιμοποιούμε τη μέθοδο executeQuery μέσω της εντολής:

```
ResultSet rs = myStatement.executeQuery(sqlString);
```

όπου sqlString είναι μια συμβολοσειρά της μορφής:

```
String sqlString = "SELECT * FROM MYTABLE WHERE PARAM = '~~paramValue~~'";
```

Η μέθοδος *executeQuery* μας δίνει τη δυνατότητα να ανακτήσουμε γραμμές δεδομένων από τη βάση δεδομένα. Τα δεδομένα αυτά πρέπει να τα αποθηκεύσουμε προσωρινά σε κάποια δομή ώστε να τα χρησιμοποιήσουμε.

Η Java παρέχει μια εύχρηστη δομή μέσω της κλάσης *ResultSet*. Πιο συγκεκριμένα, η μέθοδος *ExecuteQuery* μας επιστρέφει ένα αντικείμενο τύπου *ResultSet*. Ας ονομάσουμε το συγκεκριμένο αντικείμενο *rs*. Μέσα στο αντικείμενο αυτό αποθηκεύονται προσωρινά όλες οι γραμμές δεδομένων που επιστρέφει η δήλωση SELECT που εκτελέσαμε. Μέσω του αντικειμένου αυτού μπορούμε να επεξεργαστούμε τις γραμμές στο πρόγραμμά μας

Το αντικείμενο *rs* μας παρέχει και έναν δείκτη ο οποίος αρχικά δείχνει πρίν από την πρώτη γραμμή αποτελεσμάτων που είναι αποθηκευμένα στο αντικείμενο *rs*.

Το rs (ως αντικείμενο της κλάσης ResultSet) μας παρέχει μεθόδους για να επεξεργαζόμαστε τις γραμμές που ανακτήσαμε από τη βάση δεδομένων. Οι πιο σημαντικές μέθοδοι αναφέρονται παρακάτω:

Για να χρησιμοποιήσουμε την πληροφορία που είναι καταχωρημένη στο αντικείμενο rs, θα πρέπει:

- Να μετακινήσουμε το δείκτη στη γραμμή που θέλουμε να επεξεργασθούμε ή να τον μετακινούμε διαδοχικά αν θέλουμε να επεξεργασθούμε όλες τις αποθηκευμένες γραμμές. Αυτό γίνεται με χρήση των μεθόδων: **next(), previous(), first(), last()**
- Να χρησιμοποιήσουμε την κατάλληλη μέθοδο για να πάρουμε πληροφορία από μια στήλη. Αυτό γίνεται με τις συναρτήσεις **getString, getInt, getFloat getDate** κ.τ.λ.

Πρακτικός κανόνας για τη συγγραφή δήλωσης SQL στα προγράμματά μας:

Δοκιμάζουμε τη δήλωση με συγκεκριμένες τιμές στις μεταβλητές.

Τοποθετούμε την εντολή σε " " και αντικαθιστούμε κάθε συγκεκριμένη τιμή, για παράδειγμα την τιμή 'ANALYST'με το όνομα της μεταβλητής γραμμένο με "+ +", ως εξής: "'+job+'"

Δείτε και το παράδειγμα:

```
String sql="insert into emp(ename,job,sal,deptno) values
("'+name+"','"+job+"','"+sal+"','"+deptno+")";
```

5. Παραδείγματα χρήσης embedded statement

Ακολουθεί παράδειγμα χρήσης της executeUpdate για Insert statement. Επιπλέον, δίδεται και παράδειγμα χρήσης της executeQuery που βοηθά στην κατανόηση του τρόπου που χρησιμοποιούμε τις διάφορες μεθόδους της κλάσης ResultSet.

Αρχικά παραθέτουμε μία To-do list για τη σύνδεση σελίδων με βάση δεδομένων.

To-do list – Σύνδεση σελίδων με βάση.

1) Δήλωση του JDBC driver στο περιβάλλον Netbeans

2) Εισάγετε το package java.sql στο πρόγραμμά σας:

```
<%@page import="java.sql.*" %>
```

3) Σύνδεση στη βάση

Για παράδειγμα, οι παράμετροι της βάσης είναι:

IP= localhost, Port= 3306, Database= users_login, Username= root, Password= 1234

```
String DB = "jdbc:mysql://localhost:3306/users_login?user=root&password=1234";
```

Connection myConnection = DriverManager.getConnection(DB);

4) Δημιουργία αντικειμένου για να γράψουμε SQL statement:

Statement SMT = myConnection.createStatement();

5) Αναγραφή SQL statement στο αντικείμενο SMT και εκτέλεση:

String sql="SELECT * FROM user WHERE Uname='"+a_user+"'AND Upass='"+a_pass+"' ";

SMT.executeUpdate(sql); -- insert, update

ResultSet rs=SMT.executeQuery(sql); -- select

found= rs.first();

if (found){ … } else { … }

6) Κλείσιμο αντικειμένου «αναγραφής» SQL statement:

SMT.close();

7) Κλείσιμο σύνδεσης:

myConnection.close();

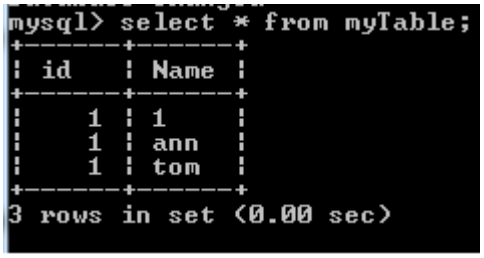
Προβολή του περιεχομένου του πίνακα

Έστω ότι έχουμε τον παραπάνω πίνακα myTable της βάσης δεδομένων mydb:

myTable

ID	*Int*
Name	*Text*

select * from myTable



1. Ορίζουμε δυο παραμέτρους.
2. Στις παραμέτρους αυτές αποθηκεύουμε τις τιμές των δυο στηλών του πίνακα mytable.
   Θυμίζουμε ότι η εντολή String name=new String(); δημιουργεί το αντικείμενο name που είναι instance της κλάσης String.

   Η εντολή int m[] = new int[5000]; δημιουργεί τον πίνακα αριθμών 5000 θέσεων που περιέχουν τιμές ακεραίων αριθμών. Όπως δημιουργείται το αντικείμενο πίνακας m[] (με τον τελεστή new) σε όλες τις θέσεις εκχωρείται τιμή 0.

   Η εντολή String s[] = new String[5000]; δημιουργεί πίνακα συμβολοσειρών 5000 θέσεων που περιέχουν τιμές συμβολοσειρών (String). Όπως δημιουργείται το αντικείμενο πίνακας s[] (με τον τελεστή new) σε όλες τις θέσεις εκχωρείται τιμή NULL.
3. Φορτώνουμε το driver και προετοιμάζουμε τη συμβολοσειρά για σύνδεση.
4. Συνδεόμαστε, στην περίπτωσή μας, με username root και password 1234, στη βάση δεδομένων mydb.
5. Συνδεόμαστε στη βάση δεδομένων, mydb, δημιουργούμε το αντικείμενο τύπου statement και προετοιμάζουμε την SQL string.
6. Εκτελούμε το SQL Statement.
7. Μέχρι στιγμής έχουμε αποθηκεύσει όλον τον πίνακα αποτελεσμάτων στο αντικείμενο rs. Μένει να «διαβάσουμε» τα στοιχεία του rs. Αυτό γίνεται με έναν βρόχο while.
8. Η εντολή while(rs.next()) έχει διπλό ρόλο: Εκτελεί την εντολή rs.next(). Επιπλέον, ελέγχει αν rs.next()=true. Αυτό σημαίνει ότι, αν rs.next()=true, τότε ο βρόχος θα εκτελεστεί. Αν rs.next()=false τότε ο βρόχος δε θα εκτελεστεί και το πρόγραμμα θα προχωρήσει παρακάτω. Η μέθοδος next() της resultSet επιστρέφει true αν υπάρχει επόμενη γραμμή για να διαβάσει. Αν δεν υπάρχει επόμενη γραμμή, δηλαδή έχουμε φτάσει στο τέλος των αποτελεσμάτων, τότε επιστρέφει false. Με τον τρόπο αυτό, αν υπάρχει επόμενη γραμμή, ο βρόχος εκτελείται, και σταματά να εκτελείται όταν οι γραμμές «τελειώσουν». Ο βρόχος κάνει το εξής:
   a. Διαβάζει, τις στήλες της τρέχουσας γραμμής και τις καταχωρεί στις αντίστοιχες μεταβλητές του προγράμματος (s[j], (m[j],), μέσω της μεθόδου getString() ή getInt().
   b. Τυπώνει στην οθόνη τις παραπάνω μεταβλητές.

Δημιουργείστε ένα αρχείο test.jsp και γράψτε τον κώδικα. Εκτελέστε το πρόγραμμα.

j=3
1 1
1 ann
1 tom

Δημιουργείστε ένα αρχείο test.jsp και γράψτε τον παρακάτω κώδικα

```
<%--
    Document   : test
    Created on : April 31, 2014, 8:24:04 AM
    Author     :
--%>
// Import the package java.sql – all the classes
<%@page import="java.sql.*" %>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%
int j=0;
// Define parameters for retrieving database results
int m[] = new int[5000];
String s[] = new String[5000];
// Load the driver
Class.forName("com.mysql.jdbc.Driver");
// Define the connection parameters
String myDatabase =
"jdbc:mysql://localhost:3306/personnel?user=root";
// Connect to the database
Connection myConnection = DriverManager.getConnection(myDatabase);
// Create object to execute statements
Statement myStatement = myConnection.createStatement();
// Select everything from the database table
String sqlString = "SELECT * FROM myTable ";
ResultSet rs=myStatement.executeQuery(sqlString);
// Store the results in vectors in order to pass them to the JSP
while(rs.next()){
m[j]=rs.getInt("id");
s[j]=rs.getString("name");
j++;
}
// Close the connection to the database
myStatement.close();
myConnection.close();
// Print
out.println("j="+j+"<br>");
```

```
for(int i=0; i<j; i++)
{
out.println(m[i]+"");
out.println(s[i]+"<br>");
}
%>
```

Εισαγωγή γραμμών

Δημιουργείστε ένα αρχείο ins.jsp και γράψτε τον παρακάτω κώδικα

```
<%@page import="java.sql.*" %>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
INSERT INTO mytable VALUES(4, 'Jim')
<%
// Load the driver
Class.forName("com.mysql.jdbc.Driver");
// Define the connection parameters
String myDatabase =
"jdbc:mysql://localhost:3306/personnel?user=root&password=1234";
// Connect to the database
Connection myConnection = DriverManager.getConnection(myDatabase);
// Create object to execute statements
Statement myStatement = myConnection.createStatement();
// Select everything from the database table
String sqlString = "INSERT INTO mytable VALUES(4, 'Jim')";
myStatement.executeUpdate(sqlString);
// Close the connection to the database
myStatement.close();
myConnection.close();
%>
```

Στη σελίδα ins.jsp γράψαμε τη δήλωση SQL με πραγματικές τιμές και στην επόμενη σελίδα ins_param.jsp την ξαναγράφουμε με παραμετρικές τιμές. Αντίστοιχα γράφουμε:

```
String sqlString = "INSERT INTO mytable(id) VALUES('"+param+"')";
```

```
String sqlString = "INSERT INTO mytable VALUES(4, 'Jim')";
```

Εισαγωγή γραμμών με παραμετρικές τιμές

Δημιουργείστε ένα αρχείο ins_param.jsp και γράψτε τον παρακάτω κώδικα

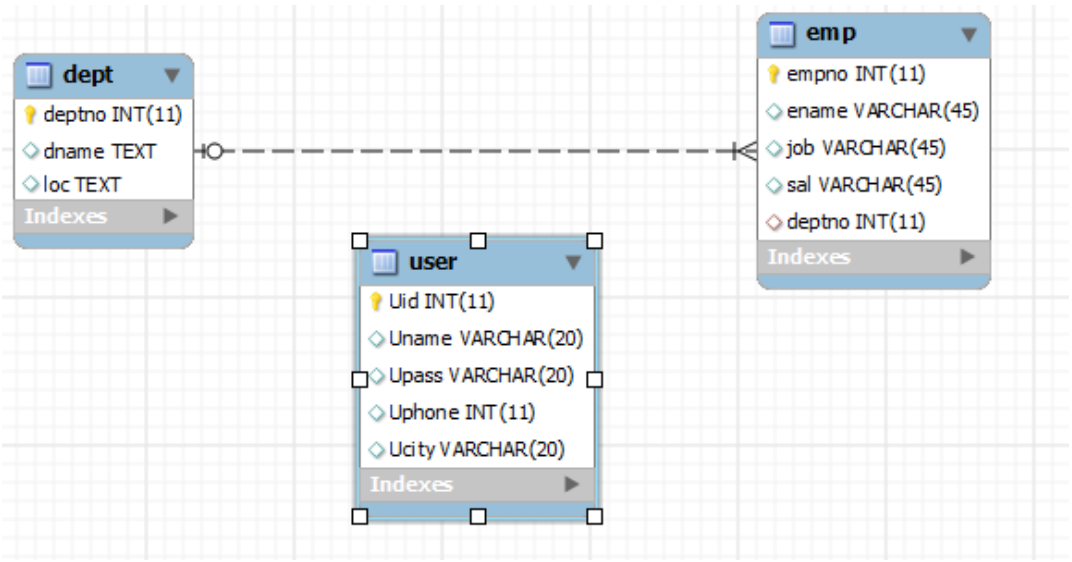
```
<%@page import="java.sql.*" %>

<%@page contentType="text/html" pageEncoding="UTF-8"%>

INSERT INTO mytable VALUES(4, 'Jim') -- html

<% int param=7; %>

<%

// Load the driver

Class.forName("com.mysql.jdbc.Driver");

// Define the connection parameters

String myDatabase =

"jdbc:mysql://localhost:3306/personnel?user=root&password=1234";

// Connect to the database

Connection myConnection = DriverManager.getConnection(myDatabase);

// Create object to execute statements

Statement myStatement = myConnection.createStatement();

// Select everything from the database table

String sqlString = "INSERT INTO mytable(id) VALUES('"+param+"')";

myStatement.executeUpdate(sqlString);

// Close the connection to the database

myStatement.close();

myConnection.close();

%>
```

6. Case Study «Διαχείριση βάσης προσωπικού»

Σας αναθέτουν να σχεδιάσετε και μα υλοποιήσετε εφαρμογή η οποία αφορά τη διαχείριση του προσωπικού εταιρείας. Στην εφαρμογή θα έχουν πρόσβαση μόνο όσοι χρήστες είναι εγγεγραμμένοι και διαθέτουν username –password. Αρχικά θα εμφανίζεται στο χρήστη μια login φόρμα όπου θα δίνεται η δυνατότητα στον χρήστη να συνδεθεί. Αφού συνδεθεί ο χρήστης θα έχει τις παρακάτω δυνατότητες:

1. Προβολή όλων των τμημάτων της εταιρείας μαζί με τα στοιχεία εργαζομένων που ανήκουν σε αυτά.
2. Εισαγωγή ενός νέου εργαζόμενου.
3. Διαγραφή ενός εργαζόμενου
4. Ενημέρωση των στοιχείων των εργαζομένων.

Οι πίνακες της εφαρμογής θα είναι οι παρακάτω:



```
DROP DATABASE personnel_db;

CREATE DATABASE personnel;

USE personnel;

CREATE TABLE Dept(DEPTNO INT(2) NOT NULL,

                DNAME VARCHAR(14), LOC VARCHAR(14),

                NO_OF_EMPLOYEES INT(3),

                PRIMARY KEY(DEPTNO));

CREATE TABLE Emp(EMPNO INT(4) NOT NULL AUTO_INCREMENT,

               ENAME VARCHAR(10), JOB VARCHAR(9),

               SAL FLOAT(7,2),

               DEPTNO INT(2), PRIMARY KEY(EMPNO),

               FOREIGN KEY(DEPTNO) REFERENCES Dept(DEPTNO));


INSERT INTO Dept(DEPTNO, DNAME, LOC)

        VALUES (10, 'ACCOUNTING', 'NEW YORK');

INSERT INTO Dept(DEPTNO, DNAME, LOC)

        VALUES (20, 'RESEARCH', 'DALLAS');

INSERT INTO Dept(DEPTNO, DNAME, LOC)

        VALUES (30, 'SALES', 'CHICAGO');

INSERT INTO Dept(DEPTNO, DNAME, LOC)

        VALUES (40, 'OPERATIONS', 'BOSTON');
```

```
INSERT INTO Emp( ENAME, JOB, SAL,  DEPTNO)  VALUES ( 'SMITH', 'CLERK', 800, 20);

INSERT INTO Emp( ENAME, JOB, SAL,  DEPTNO) VALUES ( 'ALLEN', 'SALESMAN', 1600, 30);

INSERT INTO Emp( ENAME, JOB, SAL,  DEPTNO) VALUES ( 'WARD', 'SALESMAN', 1250, 30);

INSERT INTO Emp( ENAME, JOB, SAL,  DEPTNO) VALUES ( 'JONES', 'MANAGER', 2975, 20);

INSERT INTO Emp( ENAME, JOB, SAL,  DEPTNO)  VALUES ( 'MARTIN', 'SALESMAN', 1250, 30);

INSERT INTO Emp( ENAME, JOB, SAL,  DEPTNO) VALUES ( 'BLAKE', 'MANAGER', 2850, 30);

INSERT INTO Emp( ENAME, JOB, SAL,  DEPTNO) VALUES ( 'CLARK', 'MANAGER', 2450, 10);

INSERT INTO Emp( ENAME, JOB, SAL,  DEPTNO) VALUES ( 'SCOTT', 'ANALYST', 3000, 20);

INSERT INTO Emp( ENAME, JOB, SAL,  DEPTNO) VALUES ( 'KING', 'PRESIDENT', 5000, 10);

INSERT INTO Emp( ENAME, JOB, SAL,  DEPTNO) VALUES ( 'TURNER', 'SALESMAN', 1500, 30);

INSERT INTO Emp( ENAME, JOB, SAL,  DEPTNO) VALUES ( 'ADAMS', 'CLERK', 1100, 20);

INSERT INTO Emp( ENAME, JOB, SAL,  DEPTNO) VALUES ( 'JAMES', 'CLERK', 950, 30);

INSERT INTO Emp( ENAME, JOB, SAL,  DEPTNO) VALUES ( 'FORD', 'ANALYST', 3000, 20);

INSERT INTO Emp( ENAME, JOB, SAL,  DEPTNO) VALUES ( 'MILLER', 'CLERK', 1300, 10);

INSERT INTO Emp( ENAME, JOB, SAL,  DEPTNO) VALUES ( 'BATES', 'ANALYST', 1300, NULL);


UPDATE dept

SET no_of_employees =

                (SELECT COUNT(*)

                 FROM emp

                 WHERE emp.deptno = dept.deptno);

Select * from dept;

Select * from  emp;
```

```
mysql> Select * from dept;
+--------+------------+----------+------------------+
| DEPTNO | DNAME      | LOC      | NO_OF_EMPLOYEES |
+--------+------------+----------+------------------+
|     10 | ACCOUNTING | NEW YORK |                3 |
|     20 | RESEARCH   | DALLAS   |                5 |
|     30 | SALES      | CHICAGO  |                6 |
|     40 | OPERATIONS | BOSTON   |                0 |
+--------+------------+----------+------------------+
4 rows in set (0.00 sec)

mysql> Select * from  emp;
+-------+--------+-----------+---------+--------+
| EMPNO | ENAME  | JOB       | SAL     | DEPTNO |
+-------+--------+-----------+---------+--------+
|     1 | SMITH  | CLERK     |  800.00 |     20 |
|     2 | ALLEN  | SALESMAN  | 1600.00 |     30 |
|     3 | WARD   | SALESMAN  | 1250.00 |     30 |
|     4 | JONES  | MANAGER   | 2975.00 |     20 |
|     5 | MARTIN | SALESMAN  | 1250.00 |     30 |
|     6 | BLAKE  | MANAGER   | 2850.00 |     30 |
|     7 | CLARK  | MANAGER   | 2450.00 |     10 |
|     8 | SCOTT  | ANALYST   | 3000.00 |     20 |
|     9 | KING   | PRESIDENT | 5000.00 |     10 |
|    10 | TURNER | SALESMAN  | 1500.00 |     30 |
|    11 | ADAMS  | CLERK     | 1100.00 |     20 |
|    12 | JAMES  | CLERK     |  950.00 |     30 |
|    13 | FORD   | ANALYST   | 3000.00 |     20 |
|    14 | MILLER | CLERK     | 1300.00 |     10 |
|    15 | BATES  | ANALYST   | 1300.00 |   NULL |
+-------+--------+-----------+---------+--------+
15 rows in set (0.02 sec)
```

CREATE TABLE user(

uname text,

upass text,

Uid int(11),

Uphone varchar(45), Ucity varchar(45));

INSERT INTO `user` VALUES ('admin','1234',1,NULL,NULL);

Select * from user;

```
mysql> Select * from user;
+-------+-------+------+--------+-------+
| uname | upass | Uid  | Uphone | Ucity |
+-------+-------+------+--------+-------+
| admin | 1234  |    1 | NULL   | NULL  |
+-------+-------+------+--------+-------+
1 row in set (0.00 sec)
```

Χρήση Cursor

```
DELIMITER $$
DROP PROCEDURE IF EXISTS CursorProc$$
CREATE PROCEDURE CursorProc()
BEGIN
DECLARE  no_more_depts, available_employees INT DEFAULT 0;
DECLARE  dept_code VARCHAR(255);
DECLARE  cur_dept CURSOR FOR
 SELECT  deptno FROM dept;
 DECLARE  CONTINUE HANDLER FOR NOT FOUND
 SET  no_more_depts = 1;
 /* for  loggging information */
CREATE  TABLE infologs (
Id int(11) NOT NULL AUTO_INCREMENT,
Msg varchar(255) NOT NULL, PRIMARY KEY (Id));
OPEN  cur_dept;
FETCH  cur_dept  INTO dept_code;
 REPEAT
   SELECT  no_of_employees INTO available_employees
   FROM  dept
   WHERE  deptno = dept_code;
    IF  available_employees < 5 THEN
      INSERT  INTO infologs(msg)VALUES  (dept_code);
    END  IF;
    FETCH  cur_dept INTO dept_code;
 UNTIL  no_more_depts = 1
 END REPEAT;
```

```
CLOSE  cur_dept;

 SELECT *  FROM infologs;

 DROP TABLE  infologs;

END$$

DELIMITER ;

CALL  CURSORPROC();
```

7. Υλοποίηση εφαρμογής

Τα στοιχεία της σύνδεσης είναι:

Ip:localhost

Port:3306

Database: personnel

User=root

Password=

Υπενθυμίζουμε ότι ο χρήστης (βλέπε και πίνακα user) έχει:

User=root

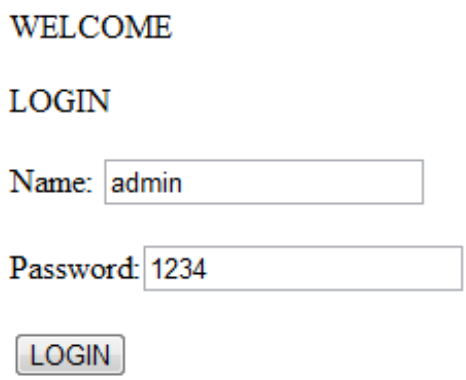
Password=1234

Στάδια:

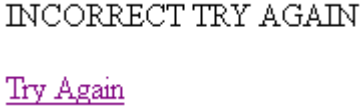
I. Διαχείριση Σύνδεσης

Υλοποιείται με χρήση δύο (2) σελίδων: index.jsp, check.jsp

Index.jsp



Αν δοθούν λανθασμένα στοιχεία τότε



```
<%--
    Document   : index
    Created on : April 30, 2014, 2:04:01 PM
    Author     :
--%>


<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">
        <title>JSP Page</title>
    </head>
    <body>
        WELCOME<p></p>
        LOGIN<p></p>
        <p></p>
        <form name="formName" method="post" action="check.jsp" >
            Name: <input type="text" name="y"><p></p>
            Password:<input type="text" name="k"><P></P>
             <input type="submit" value="LOGIN"><p></p>
        </form>
    </body>
</html>
```

```
<%--
    Document   : check
    Created on : April 30, 2014, 2:12:02 PM
    Author     :
--%>

<%@page import="java.sql.*" %>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">
        <title>JSP Page</title>
    </head>
    <body>
        <%
        Boolean found;
        String x =request.getParameter("y");
        String p = request.getParameter("k");
         String  URL;
                Class.forName("com.mysql.jdbc.Driver");
        String DB =
"jdbc:mysql://localhost:3306/personnel?user=root";
        Connection myConnection = DriverManager.getConnection(DB);
        Statement SMT = myConnection.createStatement();
        String sql="SELECT * FROM user WHERE Uname='"+x+"'AND
Upass='"+p+"' ";
        ResultSet rs=SMT.executeQuery(sql);
        found= rs.first();
        if (found){
               URL = "home.jsp?p1="+x+"";
```

```
            response.sendRedirect(URL);
          }
          else
                                {
          out.println("INCORRECT TRY AGAIN");%><P></P>
          <a href="index.jsp">Try Again</a><%
                        }
             SMT.close();
          myConnection.close();
          %>
      </body>
</html>
```

Σημαντική σημείωση

Έστω ότι τα στοιχεία της σύνδεσης είναι:

Ip:localhost

Port:3306

Database: personnel

User=root

Password=1234

Τότε αντί της εντολής

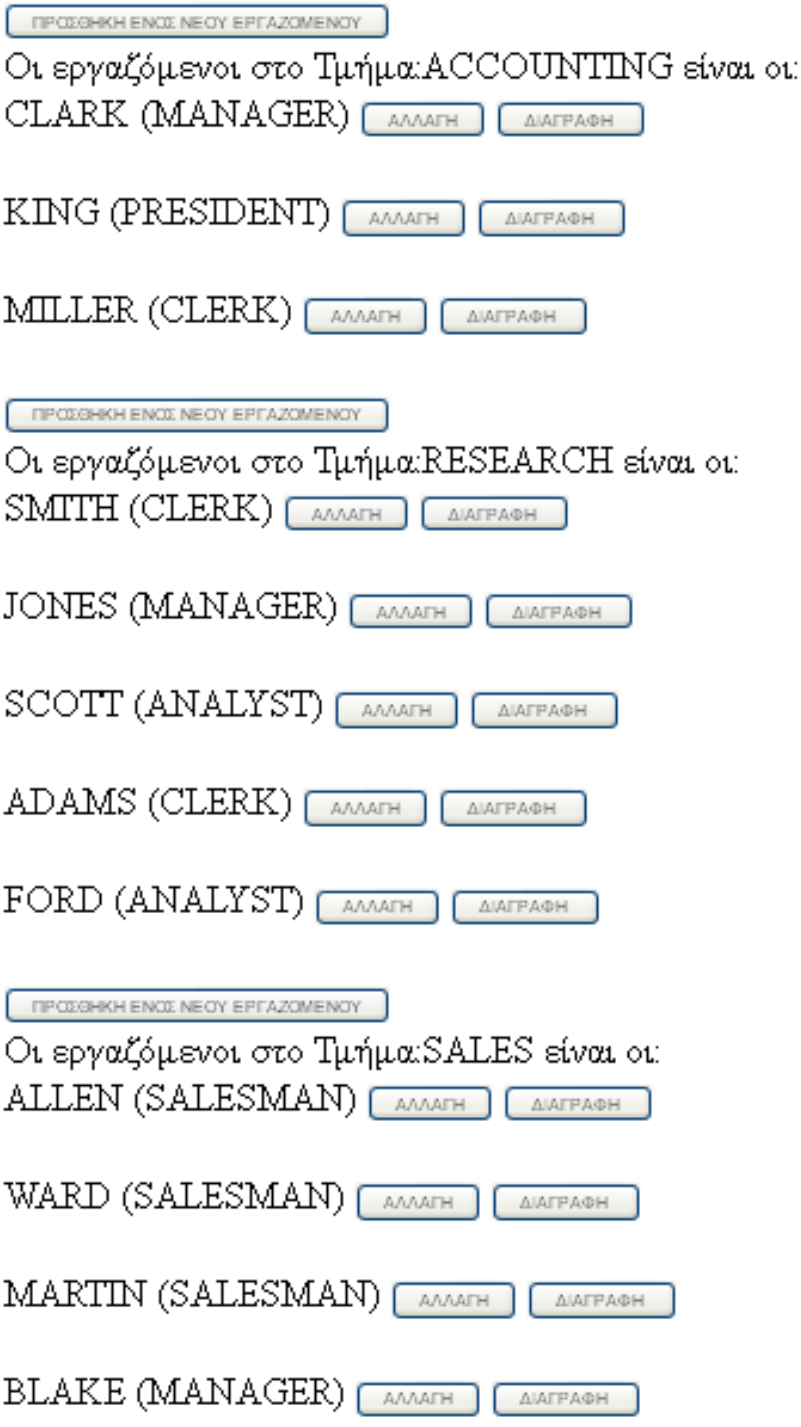
String DB = "jdbc:mysql://localhost:3306/personnel?user=root";

Χρησιμοποιούμε την εντολή:

String DB = "jdbc:mysql://localhost:3306/personnel_db?user=root&password=1234";

II. Προβολή όλων των εργαζομένων και των τμημάτων στα οποία ανήκουν.

Υλοποιείται με χρήση της σελίδας (home.jsp)



κ.λπ.

```
<%--
    Document   : home
    Created on : April 30, 2014, 2:26:58 PM
    Author     :
--%>

<%@page import="java.sql.*" %>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
 <%
  Integer empno[]=new Integer[100];
 String name[]=new String[100];
  String job[]=new String[100];
int i;
 Class.forName("com.mysql.jdbc.Driver");
        String DB =
"jdbc:mysql://localhost:3306/personnel?user=root";
        Connection myConnection = DriverManager.getConnection(DB);
        Statement SMT = myConnection.createStatement();
        Statement SMT1 = myConnection.createStatement();

        %>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">
        <title>JSP Page</title>
    </head>
        <body>
      <%
       String sql="SELECT * FROM dept ";
```

```
        ResultSet rs=SMT.executeQuery(sql);
         String dname;
        int count=0;
        while (rs.next())
        {
            i=rs.getInt("deptno");
            String sql1="SELECT * FROM emp WHERE deptno='"+i+"' ";
        ResultSet rs1=SMT1.executeQuery(sql1);
        dname=rs.getString("dname");
        %>
       <br> <INPUT TYPE="BUTTON" VALUE="ΠΡΟΣΘΗΚΗ ΕΝΟΣ ΝΕΟΥ
ΕΡΓΑΖΟΜΕΝΟΥ" style="font-size:7px"
      ONCLICK="window.location.href='insert.jsp?deptno=<%= i%>'">
    <%
        out.println(" <br>Οι εργαζόμενοι στο Τμήμα:"+dname+" είναι
οι:");

       while(rs1.next())
                       {
       name[count]=rs1.getString("ename");
        job[count]=rs1.getString("job");
        empno[count]=rs1.getInt("empno");
                %>        <br>
            <%=  name[count]%>  (<%=  job[count]%>)

           <INPUT TYPE="BUTTON" VALUE="ΑΛΛΑΓΗ" style="font-size:7px"
      ONCLICK="window.location.href='update.jsp?empno=<%=
empno[count]%>'">
         <INPUT TYPE="BUTTON" VALUE="ΔΙΑΓΡΑΦΗ " style="font-size:7px"
      ONCLICK="window.location.href='delete.jsp?empno=<%=
empno[count]%>'">
```

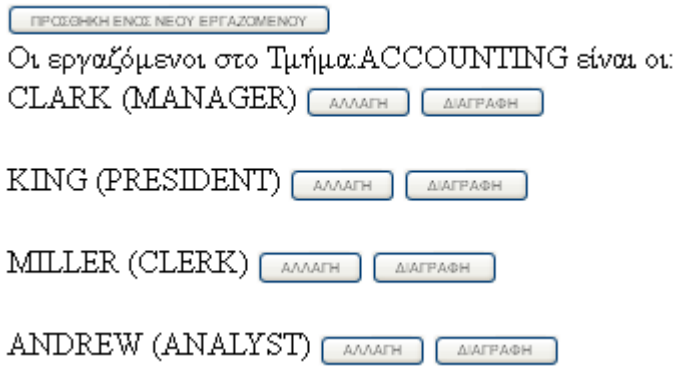
```
                    <br>
            <%  String x="o";
        }}%>
                                <%
            SMT.close();
            SMT1.close();
        myConnection.close();
                %>
    </body>
</html>
```

III. Εισαγωγή Νέου εργαζόμενου (insert.jsp, ins_submit.jsp)

Για παράδειγμα

IV. Ενημέρωση στοιχείων εργαζομένων (update.jsp, up_submit.jsp)

Αν συνδεθούμε εκ νέου θα δούμε



κ.λπ.

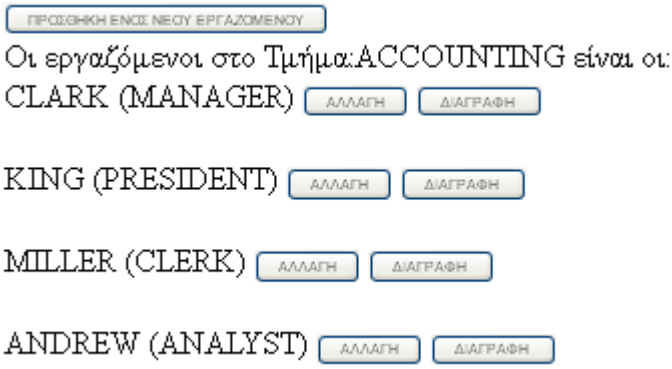
Επιλέγουμε ΑΛΛΑΓΗ για τον υπάλληλο ANDREW.

Αλλάζουμε τα στοιχεία

Επιλέγουμε update οπότε ενημερώνεται η γραμμή και επιστρέφουμε στην αρχική φόρμα:

V. Διαγραφή ενός εργαζομένου (delete.jsp)

Αν συνδεθούμε εκ νέου θα δούμε



κ.λπ.

Αν επιλέξουμε ΔΙΑΓΡΑΦΗ για τον υπάλληλο ANDREW διαγράφεται ο υπάλληλος.

Σημείωση: Θα έπρεπε να βλέπαμε όλα τα στοιχεία του πριν τον διαγράψουμε. Επιπλέον, θα έπρεπε να υπάρχει και επιβεβαίωση πριν από τη διαγραφή.

Γενικά θα μπορούσαμε να βελτιώσουμε σε πολλά σημεία την εφαρμογή με σχετικά απλό τρόπο!

```
<%--
    Document   : insert
    Created on : April 30, 2014, 2:32:55 PM
    Author     :
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>

        <html>
    <head>
        <meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">
        <title>JSP Page</title>
    </head>
      <body>
        <form name="insert" method="get" action="ins_submit.jsp">
            Name<input type="text" name="ename"><br>
            Job<input type="text" name="job"><br>
            Salary<input type="text" name="sal"><br>
            <input type="hidden" name="deptno" value="<%=
request.getParameter("deptno") %>">
            <input type="submit" value="insert">
        </form>
    </body>
</html>
```

```
<%--
    Document   : ins_submit
    Created on : April 30, 2014, 2:34:18 PM
    Author     :
--%>


<%@page import="java.sql.*"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">
        <title>JSP Page</title>
    </head>
    <body>
       <%
 Class.forName("com.mysql.jdbc.Driver");
        String DB =
"jdbc:mysql://localhost:3306/personnel?user=root";
        Connection myConnection = DriverManager.getConnection(DB);
        Statement SMT = myConnection.createStatement();
     String name=request.getParameter("ename");
     String job=request.getParameter("job");
     String sal=request.getParameter("sal");
     int deptno=Integer.parseInt(request.getParameter("deptno"));
     String sql="insert into emp(ename,job,sal,deptno) values
('"+name+"','"+job+"','"+sal+"',"+deptno+")";
     SMT.executeUpdate(sql);
        String URL = "index.jsp";
         response.sendRedirect(URL);
        %>
```

```
            <%
                SMT.close();
            myConnection.close();
                    %>
                </body>
</html>
```

```
<%--
    Document   : update
    Created on : April 30, 2014, 2:35:38 PM
    Author     :
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>

<%@page import="java.sql.*"%>

        <html>
    <head>
        <meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">
        <title>JSP Page</title>
    </head>
     <body>
    <%
 Class.forName("com.mysql.jdbc.Driver");
        String DB =
"jdbc:mysql://localhost:3306/personnel?user=root";
        Connection myConnection = DriverManager.getConnection(DB);
        Statement SMT = myConnection.createStatement();

 String job="",sal="",ename="";
     int empno=Integer.parseInt(request.getParameter("empno"));
     String sql="select * from emp where empno="+empno;
     ResultSet rs= SMT.executeQuery(sql);
     while(rs.next()){
```

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

      sal=rs.getString("sal");

      ename=rs.getString("ename");

      }


         %>

             <form name="insert" method="get" action="up_submit.jsp">

             Name<input type="text" name="ename" value="<%=
ename%>"><br>

             Job<input type="text" name="job"value="<%= job%>"><br>

             Salary<input type="text" name="sal" value="<%=
sal%>"><br>

             <input type="hidden" name="empno" value="<%= empno%>">

             <input type="submit" value="update">

         </form>


    </body>

</html>
```

```
<%--
    Document   : up_submit
    Created on : April 30, 2014, 2:37:10 PM
    Author     :
--%>


<%@page import="java.sql.*"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">
        <title>JSP Page</title>
    </head>
    <body>
       <%
 Class.forName("com.mysql.jdbc.Driver");
        String DB =
"jdbc:mysql://localhost:3306/personnel?user=root";
        Connection myConnection = DriverManager.getConnection(DB);
        Statement SMT = myConnection.createStatement();


     String name=request.getParameter("ename");
     String job=request.getParameter("job");
     String sal=request.getParameter("sal");
     int empno=Integer.parseInt(request.getParameter("empno"));



     String sql="update emp set ename ='"+name+"' where
empno="+empno;
     SMT.executeUpdate(sql);
```

```
       sql="update emp set job ='"+job+"' where empno="+empno;
      SMT.executeUpdate(sql);
      sql="update emp set sal ='"+sal+"' where empno="+empno;
      SMT.executeUpdate(sql);
         String URL = "index.jsp";
          response.sendRedirect(URL);
                 SMT.close();
         myConnection.close();
         %>
     </body>
</html>
```

```
<%--
    Document   : delete
    Created on : April 30, 2014, 2:38:19 PM
    Author     :
--%>
<%@page import="java.sql.*"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">
        <title>JSP Page</title>
    </head>
    <body>
       <%
 Class.forName("com.mysql.jdbc.Driver");
        String DB =
"jdbc:mysql://localhost:3306/personnel?user=root";
        Connection myConnection = DriverManager.getConnection(DB);
        Statement SMT = myConnection.createStatement();
    int empno=Integer.parseInt(request.getParameter("empno"));
     String sql="delete from emp  where empno="+empno;
      SMT.executeUpdate(sql);
        String URL = "index.jsp";
         response.sendRedirect(URL);
            SMT.close();
        myConnection.close();
        %>
    </body>
</html>
```

Δημιουργία βάσης δεδομένων

```
CREATE DATABASE personnel;

USE personnel;

CREATE TABLE Dept(DEPTNO INT(2) NOT NULL,
                  DNAME VARCHAR(14), LOC VARCHAR(14),
                  NO_OF_EMPLOYEES INT(3),
                  PRIMARY KEY(DEPTNO));
CREATE TABLE Emp(EMPNO INT(4) NOT NULL AUTO_INCREMENT,
                 ENAME VARCHAR(10), JOB VARCHAR(9),
                 SAL FLOAT(7,2),
                 DEPTNO INT(2), PRIMARY KEY(EMPNO),
                 FOREIGN KEY(DEPTNO) REFERENCES Dept(DEPTNO));


INSERT INTO Dept(DEPTNO, DNAME, LOC)
        VALUES (10, 'ACCOUNTING', 'NEW YORK');
INSERT INTO Dept(DEPTNO, DNAME, LOC)
        VALUES (20, 'RESEARCH', 'DALLAS');
INSERT INTO Dept(DEPTNO, DNAME, LOC)
        VALUES (30, 'SALES', 'CHICAGO');
INSERT INTO Dept(DEPTNO, DNAME, LOC)
        VALUES (40, 'OPERATIONS', 'BOSTON');
INSERT INTO Emp( ENAME, JOB, SAL,  DEPTNO)  VALUES ( 'SMITH', 'CLERK', 800, 20);
INSERT INTO Emp( ENAME, JOB, SAL,  DEPTNO) VALUES ( 'ALLEN', 'SALESMAN', 1600, 30);
INSERT INTO Emp( ENAME, JOB, SAL,  DEPTNO) VALUES ( 'WARD', 'SALESMAN', 1250, 30);
INSERT INTO Emp( ENAME, JOB, SAL,  DEPTNO) VALUES ( 'JONES', 'MANAGER', 2975, 20);
INSERT INTO Emp( ENAME, JOB, SAL,  DEPTNO)  VALUES ( 'MARTIN', 'SALESMAN', 1250, 30);
INSERT INTO Emp( ENAME, JOB, SAL,  DEPTNO) VALUES ( 'BLAKE', 'MANAGER', 2850, 30);
INSERT INTO Emp( ENAME, JOB, SAL,  DEPTNO) VALUES ( 'CLARK', 'MANAGER', 2450, 10);
```

```
INSERT INTO Emp( ENAME, JOB, SAL,  DEPTNO) VALUES ( 'SCOTT', 'ANALYST', 3000, 20);

INSERT INTO Emp( ENAME, JOB, SAL,  DEPTNO) VALUES ( 'KING', 'PRESIDENT', 5000, 10);

INSERT INTO Emp( ENAME, JOB, SAL,  DEPTNO) VALUES ( 'TURNER', 'SALESMAN', 1500, 30);

INSERT INTO Emp( ENAME, JOB, SAL,  DEPTNO) VALUES ( 'ADAMS', 'CLERK', 1100, 20);

INSERT INTO Emp( ENAME, JOB, SAL,  DEPTNO) VALUES ( 'JAMES', 'CLERK', 950, 30);

INSERT INTO Emp( ENAME, JOB, SAL,  DEPTNO) VALUES ( 'FORD', 'ANALYST', 3000, 20);

INSERT INTO Emp( ENAME, JOB, SAL,  DEPTNO) VALUES ( 'MILLER', 'CLERK', 1300, 10);

INSERT INTO Emp( ENAME, JOB, SAL,  DEPTNO) VALUES ( 'BATES', 'ANALYST', 1300, NULL);

UPDATE dept

SET no_of_employees =

                (SELECT COUNT(*)

                 FROM emp

                 WHERE emp.deptno = dept.deptno);

Select * from dept;

Select * from  emp;

CREATE TABLE user(

  uname text,

  upass text,

  Uid int(11),

  Uphone varchar(45), Ucity varchar(45));

INSERT INTO `user` VALUES ('admin','1234',1,NULL,NULL);

Select * from user;
```