

# MySQL + Γλώσσα Προγραμματισμού

# Database drivers

Για να χρησιμοποιήσουμε μια βάση δεδομένων από μια γλώσσα προγραμματισμού χρειαζόμαστε έναν “driver”.

- **JDBC** είναι το API για τη *Java* και καθορίζει πως ένας πελάτης μπορεί να συνδεθεί με μια βάση δεδομένων

Σύνδεση με τη βδ, εκτέλεση ερωτήσεων και τροποποιήσεων και συλλογή των αποτελεσμάτων

Τεχνικά: είναι ένα σύνολο από κλάσεις στο `java.sql` package.

- Υπάρχει ένα JDBC για κάθε διαφορετικό ΣΔΒΔ
  - Για τη **MySQL: Connector/J**

✓ Για MySQL και C++: **Connector/C++**

# Βήμα 0:

## Εγκατάσταση του driver (Connector/J)

- Κατεβάστε Connector/J using binary distribution από:  
<http://dev.mysql.com/downloads/connector/j/5.0.html>
- Εγκατάσταση: unzip (or untar) και θέστε το mysql-connector-java-*[version]*-bin.jar στο class path
- Αν χρησιμοποιείτε Netbeans IDE, υπάρχει ήδη
  - Οδηγίες: «Inside the Projects tab, right click on the Libraries node and select Add Library option. From the list of options, select MySQL JDBC Driver.»

# Παράδειγμα

Δημιουργούμε μια βάση δεδομένων

```
mysql> CREATE DATABASE testdb;  
Query OK, 1 row affected (0.02 sec)
```

Φτιάχνουμε ένα νέο χρήστη και του δίνουμε δικαιώματα σε όλους τους πίνακες της testdb

```
mysql> CREATE USER 'testuser'@'localhost' IDENTIFIED BY 'test623';  
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> USE testdb;
```

Database changed

```
mysql> GRANT ALL ON testdb.* TO 'testuser'@'localhost';  
Query OK, 0 rows affected (0.00 sec)
```

# Παράδειγμα

```
DROP TABLE IF EXISTS Books, Authors, Testing, Images;

CREATE TABLE IF NOT EXISTS Authors(Id INT PRIMARY KEY AUTO_INCREMENT,
  Name VARCHAR(25)) ENGINE=InnoDB;
CREATE TABLE IF NOT EXISTS Books(Id INT PRIMARY KEY AUTO_INCREMENT,
  AuthorId INT, Title VARCHAR(100),
  FOREIGN KEY(AuthorId) REFERENCES Authors(Id) ON DELETE CASCADE)
ENGINE=InnoDB;
CREATE TABLE IF NOT EXISTS Testing(Id INT) ENGINE=InnoDB;
CREATE TABLE IF NOT EXISTS Images(Id INT PRIMARY KEY AUTO_INCREMENT,
  Data MEDIUMBLOB);

INSERT INTO Authors(Id, Name) VALUES(1, 'Jack London');
INSERT INTO Authors(Id, Name) VALUES(2, 'Honore de Balzac');
INSERT INTO Authors(Id, Name) VALUES(3, 'Lion Feuchtwanger');
INSERT INTO Authors(Id, Name) VALUES(4, 'Emile Zola');
INSERT INTO Authors(Id, Name) VALUES(5, 'Truman Capote');

INSERT INTO Books(Id, AuthorId, Title) VALUES(1, 1, 'Call of the Wild');
INSERT INTO Books(Id, AuthorId, Title) VALUES(2, 1, 'Martin Eden');
INSERT INTO Books(Id, AuthorId, Title) VALUES(3, 2, 'Old Goriot');
INSERT INTO Books(Id, AuthorId, Title) VALUES(4, 2, 'Cousin Bette');
INSERT INTO Books(Id, AuthorId, Title) VALUES(5, 3, 'Jew Sues');
INSERT INTO Books(Id, AuthorId, Title) VALUES(6, 4, 'Nana');
INSERT INTO Books(Id, AuthorId, Title) VALUES(7, 4, 'The Belly of Paris');
INSERT INTO Books(Id, AuthorId, Title) VALUES(8, 5, 'In Cold blood');
INSERT INTO Books(Id, AuthorId, Title) VALUES(9, 5, 'Breakfast at Tiffany');
```

# Σύνδεση με τη βάση δεδομένων

```
package name.mysql.first;  
  
import java.sql.Connection;  
import java.sql.DriverManager;  
import java.sql.PreparedStatement;  
import java.sql.ResultSet;  
import java.sql.SQLException;  
import java.sql.Statement;
```

# Σύνδεση με τη βάση δεδομένων

```
Connection con = null;  
  
String url = "jdbc:mysql://localhost:3306/testdb";  
String user = "testuser";  
String password = "test623";  
  
con = DriverManager.getConnection(url, user, password);
```

url = "jdbc:mysql://localhost:3306/testdb"

Host                      Port                      όνομα βδ

# Εκτέλεση SQL εντολών στη βδ

Η μέθοδος `createStatement()` του connection object δημιουργεί ένα Statement object για να στέλνουμε SQL εντολές στη βάση δεδομένων

```
Statement st = null;  
  
st = con.createStatement();
```

Για να εκτελέσουμε SQL εντολές

1. Ερωτήσεις (select queries)

```
rs = st.executeQuery(query);
```

2. Τροποποιήσεις (insert, update deletes)

```
st.executeUpdate(query);
```



# Τροποποίηση

Η εκτέλεση της εντολής δεν επιστρέφει αποτελέσματα

```
Statement st = null;
```

```
st = con.createStatement();
```

```
String query = "INSERT INTO Authors(Id, Name) VALUES(6, 'Steven King');"
```

```
st.executeUpdate(query);
```

# Ερώτηση

Η εκτέλεση της εντολής επιστρέφει αποτελέσματα.

Τα αποτελέσματα εισάγονται σε ένα ResultSet object – ένας πίνακας

```
Statement st = null;
```

```
ResultSet rs = null;
```

```
st = con.createStatement();
```

```
String query = "SELECT * FROM Authors";
```

```
rs = pst.executeQuery(query);
```

```
pst.close();
```

# Ερώτηση: Cursor

Τα δεδομένα από το ResultSet διαβάζονται μια πλειάδα (γραμμή, εγγραφή) τη φορά  
Η μέθοδος `next()` προχωρά το δείκτη (cursor) στην επόμενη εγγραφή  
Επιστρέφει null όταν δεν υπάρχουν άλλες πλειάδες

```
Statement st = null;

ResultSet rs = null;

st = con.createStatement();

String query = "SELECT * FROM Authors";
rs = pst.executeQuery(query);
pst.close();

while (rs.next()) {
    System.out.print(rs.getInt(1));
    System.out.print(": ");
    System.out.println(rs.getString(2));
}
```

# Prepared Statements

Αντί να γράφουμε μέσα στην SQL έκφραση τις πραγματικές τιμές χρησιμοποιούμε placeholder (?)

Προσδιορίζουμε τις τιμές αργότερα

```
PreparedStatement pst = null;  
ResultSet rs = null;  
String author = "Trygve Gulbrandsen";  
  
pst = con.prepareStatement("INSERT INTO Authors(Name) VALUES(?)")
```

Προσδιορισμός τιμής

```
pst.setString(1, author);  
pst.executeUpdate();  
pst.close();
```

# Περισσότερες πηγές

Στο παρακάτω link μπορείτε να δείτε ένα ολοκληρωμένο πρόγραμμα Java

[http://www.cs.uoi.gr/~pvassil/courses/db\\_III/exercises/JavaExamples/Simple/Example1.java](http://www.cs.uoi.gr/~pvassil/courses/db_III/exercises/JavaExamples/Simple/Example1.java)

Επίσης:

<http://zetcode.com/db/mysqljava/>

Online documentation :

<http://dev.mysql.com/doc/refman/5.0/en/connector-j.html>

# Ερωτήσεις;