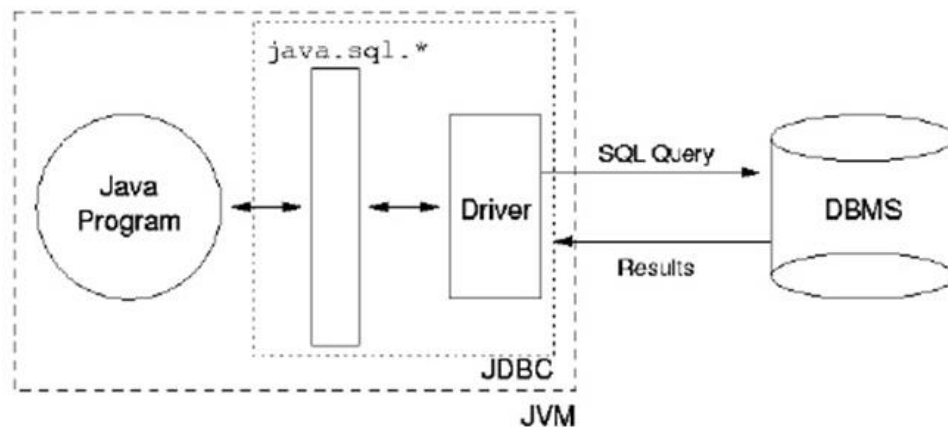


JDBC Driver

Ορισμένα από τα ακόλουθα παραδείγματα
προέρχονται από το βιβλίο Java Servlets και JSP 3rd
Edition, Εκδόσεις Murach

JDBC Architecture



Τύποι Οδηγών JDBC

The four types of JDBC database drivers

- | | |
|--------|---|
| Type 1 | <i>A JDBC-ODBC bridge driver</i> converts JDBC calls into ODBC calls that access the DBMS protocol. For this data access method, the ODBC driver must be installed on the client machine. |
| Type 2 | <i>A native protocol partly Java driver</i> converts JDBC calls into calls in the native DBMS protocol. Since this conversion takes place on the client, some binary code must be installed on the client machine. |
| Type 3 | <i>A net protocol all Java driver</i> converts JDBC calls into a net protocol that's independent of any native DBMS protocol. Then, middleware software running on a server converts the net protocol to the native DBMS protocol. Since this conversion takes place on the server side, no installation is required on the client machine. |
| Type 4 | <i>A native protocol all Java driver</i> converts JDBC calls into a native DBMS protocol. Since this conversion takes place on the server side, no installation is required on the client machine. |

Database Connection

Database URL syntax

```
jdbc:subprotocolName:databaseURL
```

How to connect to a MySQL database with automatic driver loading

```
try {  
    String dbURL = "jdbc:mysql://localhost:3306/murach";  
    String username = "root";  
    String password = "sesame";  
    Connection connection = DriverManager.getConnection(  
        dbURL, username, password);  
} catch(SQLException e) {  
    for (Throwable t : e)  
        t.printStackTrace();  
}
```

How to connect to an Oracle database with automatic driver loading

```
Connection connection = DriverManager.getConnection(  
    "jdbc:oracle:thin@localhost/murach", "scott", "tiger");
```

Αρχική Μέθοδος Ενεργοποίησης ενός Οδηγού JDBC

How to load a MySQL database driver prior to JDBC 4.0

```
try {  
    Class.forName("com.mysql.jdbc.Driver");  
} catch(ClassNotFoundException e) {  
    e.printStackTrace();  
}
```

Απλή εκτέλεση εντολής SQL

How to create a result set that contains 1 row and 1 column

```
Statement statement = connection.createStatement();  
ResultSet userIDResult = statement.executeQuery(  
    "SELECT UserID FROM User " +  
    "WHERE Email = 'jsmith@gmail.com'");
```

Προσπέλαση ResultSet

How to move the cursor to the first row in the result set

```
boolean userIDExists = userIDResult.next();
```

How to loop through a result set

```
while (products.next()) {  
    // statements that process each row  
}
```

Μέθοδοι αντικειμένου ResultSet

ResultSet methods for forward-only, read-only result sets

Method	Description
next()	Moves the cursor to the next row in the result set.
last()	Moves the cursor to the last row in the result set.
close()	Releases the result set's resources.
getRow()	Returns an int value that identifies the current row of the result set.

Μέθοδοι για λήψη τιμών στηλών

Methods of a ResultSet object that return data from a result set

Method	Description
<code>getXXX(int columnIndex)</code>	Returns data from the specified column number.
<code>getXXX(String columnName)</code>	Returns data from the specified column name.

Code that uses indexes to return columns from the products result set

```
String code = products.getString(1);  
String description = products.getString(2);  
double price = products.getDouble(3);
```

Code that uses names to return the same columns

```
String code = products.getString("ProductCode");  
String description = products.getString("ProductDescription");  
double price = products.getDouble("ProductPrice");
```

Απλή εκτέλεση εντολής SQL insert

How to use the executeUpdate method to modify data

How to add a row

```
String query =  
    "INSERT INTO Product (ProductCode, ProductDescription, ProductPrice) " +  
    "VALUES ('" + product.getCode() + "', " +  
        "'" + product.getDescription() + "', " +  
        "'" + product.getPrice() + "');"   
Statement statement = connection.createStatement();  
int rowCount = statement.executeUpdate(query);
```

Απλή εκτέλεση SQL Update

How to update a row

```
String query = "UPDATE Product SET " +  
    "ProductCode = '" + product.getCode() + "', " +  
    "ProductDescription = '" + product.getDescription() + "', " +  
    "ProductPrice = '" + product.getPrice() + "' " +  
    "WHERE ProductCode = '" + product.getCode() + "'";  
Statement statement = connection.createStatement();  
int rowCount = statement.executeUpdate(query);
```

Απλή εκτέλεση SQL Delete

How to delete a row

```
String query = "DELETE FROM Product " +  
               "WHERE ProductCode = '" + productCode + "'";  
Statement statement = connection.createStatement();  
int rowCount = statement.executeUpdate(query);
```

Δημιουργία προετοιμασμένης εντολής (PreparedStatement)

How to use a prepared statement

To return a result set

```
String preparedSQL = "SELECT ProductCode, ProductDescription, ProductPrice "  
                    + "FROM Product WHERE ProductCode = ?";  
PreparedStatement ps = connection.prepareStatement(preparedSQL);  
ps.setString(1, productCode);  
ResultSet product = ps.executeQuery();
```

Εκτέλεση προετοιμασμένης εντολής SQL Update

To modify a row

```
String preparedSQL = "UPDATE Product SET "  
                    + "    ProductCode = ?, "  
                    + "    ProductDescription = ?, "  
                    + "    ProductPrice = ?"  
                    + "WHERE ProductCode = ?";  
  
PreparedStatement ps = connection.prepareStatement(preparedSQL);  
ps.setString(1, product.getCode());  
ps.setString(2, product.getDescription());  
ps.setDouble(3, product.getPrice());  
ps.setString(4, product.getCode());  
ps.executeUpdate();
```

Εκτέλεση προετοιμασμένης εντολής SQL Insert

To insert a row

```
String preparedQuery =  
    "INSERT INTO Product (ProductCode, ProductDescription, ProductPrice) "  
    + "VALUES (?, ?, ?)";  
PreparedStatement ps = connection.prepareStatement(preparedQuery);  
ps.setString(1, product.getCode());  
ps.setString(2, product.getDescription());  
ps.setDouble(3, product.getPrice());  
ps.executeUpdate();
```

Εκτέλεση προετοιμασμένης εντολής SQL Delete

To delete a row

```
String preparedQuery = "DELETE FROM Product "  
                        + "WHERE ProductCode = ?";  
PreparedStatement ps = connection.prepareStatement(preparedQuery);  
ps.setString(1, productCode);  
ps.executeUpdate();
```

Δεξαμενές Συνδέσεων (Connection Pooling)

How connection pooling works

