

# Basi di Dati

## Esercitazione JDBC

22 ottobre 2002

**Ing. Tino Ferretti**

### Esercizio 1

Considerando il seguente schema:

**Fornitori** (CodiceFornitore, Nome, Indirizzo, Città)

**Prodotti** (CodiceProdotto, Tipo, Marca, Modello)

**Catalogo** (CodiceF, CodiceP, Costo)

Utilizzando per la connessione al database un driver di *Tipo 1*:

1. Scrivere una applicazione Java che crea la tabella Fornitori.
2. Scrivere una applicazione Java che inserisce i seguenti Fornitori:

CodiceFornitore	Nome	Indirizzo	Città
001	Ladroni	Via Ostense	Roma
002	Risparmietti	Viale Marconi	Roma
010	Teloporto	Via Roma	Milano

## Esercizio 1 (continua)

3. Scrivere una applicazione Java che stampa, per ogni fornitore, l'elenco dei prodotti forniti (Marca, Tipo, Modello, Costo).

### OUTPUT

```
Nome Indirizzo Città
    Marca, Tipo, Modello, Costo
...
    Marca, Tipo, Modello, Costo
Nome Indirizzo Città
    Marca, Tipo, Modello, Costo
...
    Marca, Tipo, Modello, Costo
...
```

## Connessione al database

```
public static void main(String[] arg){
    Connection con = null;
    try { // Caricamento del driver
        Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
    }
    catch (ClassNotFoundException exClass) {
        System.err.println("Fallita connessione al database.");
        System.exit(-1);
    }
    try { // Apertura della connessione
        String url = "jdbc:odbc:Eserc_3";
        String username = "";
        String pwd = "";
        con = DriverManager.getConnection(url, username, pwd);
    }
    catch (SQLException exSQL) {
        System.err.println("Fallita connessione al database");
        System.err.println(exSQL.getErrorCode() + " "
            + exSQL.getSQLState() + exSQL.getMessage() );
        System.exit(-1);
    }
}
```

## Soluzione Esercizio 1.1

### 1. Scrivere una applicazione Java che crea la tabella Fornitori.

```
import java.sql.*;
public class Esercizio_1_1 {
    public static void main(String[] arg){
        ...
        connessione al database
        ...
        try { // Esecuzione dell'interrogazione SQL
            Statement CreateTable = con.createStatement();
            CreateTable.executeUpdate( "CREATE TABLE Fornitori ( " +
                                     "CodiceFornitore  VARCHAR (8) NOT NULL, " +
                                     "Nome              VARCHAR (20) NOT NULL, " +
                                     "Indirizzo          VARCHAR (30)           , " +
                                     "Città              VARCHAR (20)           , " +
                                     "PRIMARY KEY( CodiceFornitore )" + ")" );
            System.out.println("Tabella Fornitori creata.");
            CreateTable.close();
            con.close();
        } catch (Exception exQuery)
        {System.err.println("Errore nell'interrogazione");}
    }
}
```

## Soluzione Esercizio 1.2 (1/2)

### 2. Scrivere una applicazione Java che inserisce ...

```
public class Esercizio_1_2 {
    // Array contenente i record da inserire
    static String[] SQLData = {
        "('001', 'Ladroni',      'Via Ostense',   'Roma')",
        "('002', 'Risparmietti', 'Viale Marconi', 'Roma')",
        "('010', 'Teloporto',    'Via Roma',    'Milano')",
    };
    //
    public static void main(String[] arg){
        ...
        connessione al database
        ...
    }
}
```

## Soluzione Esercizio 1.2 (2/2)

### 2. Scrivere una applicazione Java che inserisce ...

```
...
try { // Esecuzione dell'interrogazione SQL
    Statement stmt = con.createStatement();
    int iRowCount = 0;
    for (int i = 0; i < SQLData.length; i++) {
        iRowCount +=
            stmt.executeUpdate(
                "INSERT INTO Fornitori VALUES " + SQLData[i] );
    }
    System.out.println( iRowCount +
        " righe inserite nella tabella Fornitori.");
    stmt.close();
    con.close();
}
catch (Exception exQuery){
    System.err.println("Errore nell'interrogazione");}
}
```

## Soluzione Esercizio 1.3 (1/3)

### 3. Scrivere una applicazione Java che stampa, per ogni fornitore, l'elenco dei prodotti forniti (Marca, Tipo, Modello, Costo).

```
import java.sql.*;
public class Esercizio_1_3 {
    public static void main(String[] arg){
        ...
        connessione al database
        ...
        try { // Esecuzione dell'interrogazione SQL
            Statement query = con.createStatement();
            String queryString =
                "select Nome, Indirizzo, Città,      " +
                "      Marca, Tipo, Modello, Costo    " +
                "from Fornitori, Catalogo, Prodotti " +
                "where CodiceFornitore = CodiceF      " +
                "      and CodiceProdotto = CodiceP    " +
                "order by Nome, Marca, Tipo, Modello";
            ResultSet result = query.executeQuery(queryString);
```

## Soluzione Esercizio 1.3 (2/3)

```
// Elaborazione del risultato
boolean newFornitore = true ;
boolean firstRow = true ;
String nomeF, indirizzoF, cittaF, marcaP, tipoP, modelloP, costoC ;
String nomeF_old = "";
while (result.next()){
    nomeF = result.getString("Nome");
    indirizzoF = result.getString("Indirizzo");
    cittaF = result.getString("Città");
    if (nomeF_old.equals(nomeF)){ newFornitore = false ; }
    if (newFornitore){
        System.out.println(nomeF + " " + indirizzoF + " " + cittaF );
    }
    marcaP = result.getString("Marca");
    tipoP = result.getString("Tipo") ;
    modelloP = result.getString("Modello") ;
    costoC = result.getString("Costo") ;
    System.out.println("    " + marcaP +
        " " + tipoP + " " + modelloP + " " + costoC);
    nomeF_old = nomeF ;
    newFornitore = true ;
}
```

## Soluzione Esercizio 1.3 (3/3)

```
query.close();
con.close();
}
catch (SQLException exQuery){
    System.err.println("Errore nell'interrogazione");
    System.err.println(exQuery.getErrorCode() + " " +
        exQuery.getSQLState() + "\n" +
        exQuery.getMessage() );
}
}
```