
Basi di Dati

Esercitazione JDBC

01 giugno 2004

Ing. Paolo Cappellari
Ing. Paolo Papotti

Esercizio 1

Considerando il seguente schema:

Fornitori (CodiceFornitore, Nome, Indirizzo, Città)

Prodotti (CodiceProdotto, Tipo, Marca, Modello)

Catalogo (CodiceFornitore, CodiceProdotto, 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:

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

Esercizio 1 (continua)

1. Scrivere una applicazione Java che stampa, per ogni fornitore, l'elenco dei prodotti forniti (Marca, Tipo prodotto, 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
...
```

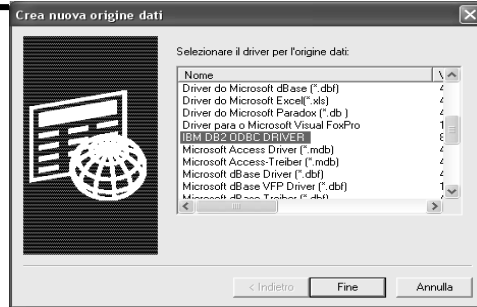
Sorgente di dati ODBC

1. Andare su: **Avvio** → **Pannello di Controllo** → **Strumenti di Amministrazione** → **Origine dati (ODBC)**.



Sorgente di dati ODBC

1. DSN
 - a. utente: disponibile solo per l'utente che lo crea.
 - b. DSN sistema: disponibile per tutti gli utenti.
2. Click su **Aggiungi**
3. Scelta del driver: **"IBM DB2 ODBC DRIVER"**
4. Scegliere il nome della sorgente: **"EsercSQL"**
5. Scegliere un (alias di) database, già esistente, cui connettersi: **"SQL2"**
6. Ok!



Connessione al database

```
import java.sql.*;
public class Esercizio_1_1 {
    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. Driver " +
                " non trovato");
        }

        String url = "jdbc:odbc:EsercSQL";
        String username = "";
        String pwd = "";
        try { // Apertura della connessione
            con = DriverManager.getConnection(url, username, pwd);
        }
        catch (SQLException exSQL) {
            System.err.println("Fallita connessione al database. La " +
                " sorgente dati (ODBC) non esiste.");
        }
    }
}
```

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("La tabella Fornitori e' stata creata.");
            createTable.close();
            con.close();
        }
        catch (SQLException exQuery){
            System.err.println("Errore nell'interrogazione.");
        }
    }
}
```

Soluzione Esercizio 1.2 (1/2)

1. Scrivere una applicazione Java che inserisce ...

```
import java.sql.*;

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)

1. 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 (SQLException exQuery){
    System.err.println("Errore nell'interrogazione.");
}
...
```

Soluzione Esercizio 1.3 (1/4)

1. 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 AS F, Catalogo AS C, Prodotti AS P " +
                "WHERE C.CodiceFornitore = F.CodiceFornitore " +
                "    AND C.CodiceProdotto = P.CodiceProdotto " +
                "ORDER BY Nome, Marca, Tipo, Modello ";

            ResultSet result = query.executeQuery(queryString);
            ...
        }
```

Soluzione Esercizio 1.3 (2/4)

Centro comandi

Tipo comando
Istruzioni SQL e comandi CLP DB2

Sistema

Interattivo Script Risultati Plan di accesso

Eseguire le modifiche richieste e fare clic sul pulsante Commit aggiornamento.

NOME	INDIRIZZO	CITTÀ	MARCA	TIPO	MODELLO	COSTO
Ladroni	Via Ostense	Roma	ACER	Desktop	730	2.200
Ladroni	Via Ostense	Roma	IBM	Desktop	510	3.200
Risparmietti	Viale Marco...	Roma	ACER	Desktop	730	1.800
Risparmietti	Viale Marco...	Roma	IBM	Desktop	510	2.500
Risparmietti	Viale Marco...	Roma	IBM	Notebook	390 x	1.900
Teloporto	Via Roma	Milano	ACER	Desktop	730	2.000
Teloporto	Via Roma	Milano	IBM	Notebook	390 x	2.200

Avanti Righe nella memoria 7 [1 - 7]

Commit aggiornamenti automatico Commit aggiornamento Rollback

Soluzione Esercizio 1.3 (3/4)

```
...
// Elaborazione del risultato
boolean newFornitore = true, firstRow = true ;
String nomeF, indirizzoF, cittaF, marcaP, tipoP, modelloP;
int 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.getInt("Costo") ; // sia costo un intero
    System.out.println("\t" + marcaP + " " + tipoP + " " +
        modelloP + " " + costoC);

    nomeF_old = nomeF ;
    newFornitore = true ;
} // chiudo while
...

```

Soluzione Esercizio 1.3 (4/4)

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