
Basi di Dati

Esercitazione JDBC

Giugno 2007

Esercizi

Considerando il seguente schema:

Fornitori (CodiceFornitore, Nome, Indirizzo, Citta)

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:

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

Esercizi

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

OUTPUT

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

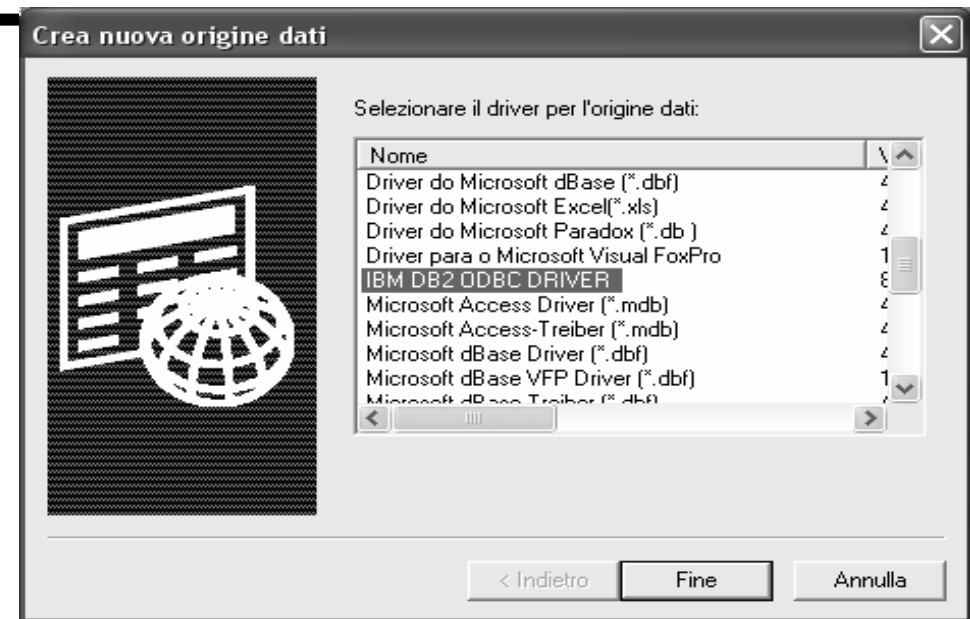
Sorgente di dati ODBC

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



Sorgente di dati ODBC

1. DSN (data source name)
 - 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: "**EsJDBC**"
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. 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 (20)    NOT NULL, " +
                "Nome                VARCHAR (20) NOT NULL, " +
                "Indirizzo           VARCHAR (30)           , " +
                "Citta                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.");
        }
    }
}
```

2. 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
        ...
    }
}
```


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

```

Soluzione Esercizio 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, Citta, " +
            "  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 3

The screenshot shows a window titled "Editor comandi 1" with a menu bar (Editor comandi, Selezionato, Modifica, Vista, Strumenti, ?) and a toolbar. Below the toolbar are tabs for "Comandi", "Risultati dell'interrogazione" (selected), and "Plan di accesso". The main area contains a table with 7 columns: NOME, INDIRIZZO, CITTA, MARCA, TIPO, MODELLO, and COSTO. The table has 7 rows of data. To the right of the table are buttons for "Aggiungi riga" and "Cancella riga". At the bottom, there are buttons for "Commit", "Rollback", and "Letture sequenziale altre righe". A checkbox for "Commit aggiornamenti automatico" is unchecked, and the text "7 righe in memoria" is displayed at the bottom right.

NOME	INDIRIZZO	CITTA	MARCA	TIPO	MODELLO	COSTO
Ladroni	Via Ostense	Roma	ACER	Desktop	730	2200
Ladroni	Via Ostense	Roma	IBM	Desktop	510	3200
Risparmietti	Viale Marconi	Roma	ACER	Desktop	730	1800
Risparmietti	Viale Marconi	Roma	IBM	Desktop	510	2500
Risparmietti	Viale Marconi	Roma	IBM	Notebook	390x	1900
Teloporto	Via Roma	Milano	ACER	Desktop	730	2000
Teloporto	Via Roma	Milano	IBM	Notebook	390x	2200

Soluzione Esercizio 3

```
...
// Elaborazione del risultato
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("Citta");
    if (!nomeF_old.equals(nomeF))
        System.out.println(nomeF + " " + indirizzoF + " " + cittaF );
    marcaP = result.getString("Marca");
    tipoP = result.getString("Tipo") ;
    modelloP = result.getString("Modello") ;
    costoC = result.getInt("Costo") ; // prendo costo come intero
    System.out.println("\t" + marcaP + " " + tipoP + " " +
        modelloP + " " + costoC);

    nomeF_old = nomeF ;
} // chiudo while
...
```

Soluzione Esercizio 3

```
...
    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
```