Basi di Dati

Esercitazione JDBC

28/05/2007

Sorgente di dati ODBC

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

Sorgente di dati ODBC

DSN

a. utente: disponibile solo per l'utente che lo crea.
b. DSN sistema: disponibile per tutti gli utenti.

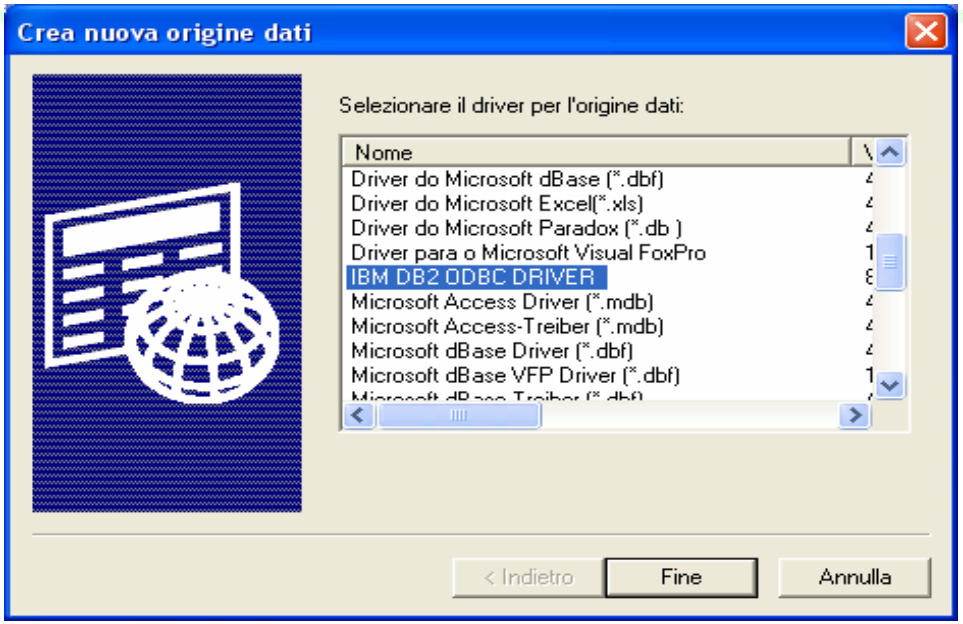
Click su ***Aggiungi***

Scelta del driver: "***IBM DB2 ODBC DRIVER***"

Scegliere il nome della sorgente: "***EsercSQL***"

Scegliere un (alias di) database, già esistente, cui connettersi: "***EsJDBC***"

Ok!



Esercizio 1 e 2

Considerando il seguente schema:

Fornitori (CodiceFornitore, Nome, Indirizzo, Citta)

Prodotti (CodiceProdotto, Tipo, Marca, Modello)

Catalogo (CodiceFornitore, CodiceProdotto, Costo)

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

Connessione al database

```
public class Esercizio_1_1 {
 public static void main(String[] arg){
  Connection con = null;
  try {// Caricamento del driver
        //(due driver diversi, scegliere; notare i msg diversi)
    //Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
    Class.forName("com.ibm.db2.jcc.DB2Driver");
  } catch (ClassNotFoundException exClass) {
  System.err.println("Fallito il caricamento del driver");
  return;
  }
  try {// Apertura della connessione
        //(scegliere il nome, coerentemente con il driver)
    //String url = "jdbc:odbc:EsercSQL";
    String url = "jdbc:db2:esempio";
    con = DriverManager.getConnection(url);
  } catch (SQLException exSQL) {
    System.err.println("Fallita connessione al database.");
    System.err.println("SQL Error Code: " + exSQL.getErrorCode()
      + ", SQL State" + exSQL.getSQLState());
    System.err.println("SQL Message: " + exSQL.getMessage());
  return;
  }
```

Connessione al database, 2

Due modalità equivalenti

```
try {
    Class.forName("com.ibm.db2.jcc.DB2Driver");
}
```

```
try {
    Driver driver = new com.ibm.db2.jcc.DB2Driver();
    DriverManager.registerDriver(driver);
}
```

Chiusura connessione al database

```
public class Esercizio_1_1 {
 public static void main(String[] arg){
  Connection con = null;
  ...
  // Apertura della connessione
  ...
  try {
    // Uso della connessione...
  } catch (SQLException exSQL) {
    // Cattura eccezioni nell'uso della connessione...
  }
  finally{
 // Chiudere SEMPRE le connessioni
    try {
      con.close();
      System.out.println("Chiusa connessione al databse.");
    } catch (SQLException exSQL) {
      System.err.println("Errore nella chiusura.");
      System.err.println("SQL Message: " + exSQL.getMessage());
    }
  }
  ...
```

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 …
  try { // Esecuzione dell'interrogazione SQL
    Statement UnaDDL = con.createStatement();
    //UnaDDL.executeUpdate( "DROP TABLE Fornitori "  );
    //System.out.println("Tabella Fornitori eliminata.");
    UnaDDL.executeUpdate( "CREATE TABLE Fornitori ("  +
      "CodiceFornitore   VARCHAR (8)  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.");
  } catch (SQLException exSQL){
    System.err.println("Errore nella creazione della tabella");
    System.err.println("SQL Error Code: " + exSQL.getErrorCode()
      + ", SQL State" + exSQL.getSQLState());
    System.err.println("SQL Message: " + exSQL.getMessage());
  }
…chiusura connessione …
```

Esercizio 2

2. Scrivere una applicazione Java che inserisce i fornitori

```
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
…
```

Esercizio 2

2. Scrivere una applicazione Java che inserisce i fornitori

```
  ...
Statement stmt = null;
  try { //Esecuzione dell'interrogazione SQL
    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
      + " ennuple inserite nalla relazione Fornitori.");
    stmt.close();
  } catch (SQLException exSQL){
    System.err.println("Errore nell'interrogazione");
    System.err.println("SQL Error Code: " + exSQL.getErrorCode()
      + ", SQL State" + exSQL.getSQLState());
    System.err.println("SQL Message: " + exSQL.getMessage());
  }
  ...
```

Esercizio 3

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

Esercizio 3

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

```
...
 connessione al database
...
try { // Esecuzione dell'interrogazione SQL
  Statement query = con.createStatement();
  String queryString =
    "SELECT F.Nome, Indirizzo, Citta,  " +
    "  Marca, Modello, Costo " +
    "FROM Fornitori AS F, Catalogo AS C, Prodotti AS P " +
    "WHERE C.CodiceFornitore = F.CodiceFornitore    " +
    "  AND C.CodiceProdotto = P.CodiceProdotto     " +
    "ORDER BY F.Nome, Marca, Modello ";

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

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("Nome: " + nomeF + ", Indirizzo: "
      + indirizzoF + ", Città: " + cittaF );
  System.out.println("\tMarca" + "\tModello" + "\tCosto");
  }
  marcaP = result.getString("Marca");
  modelloP = result.getString("Modello") ;
  costoC = result.getInt("Costo") ;  // sia costo un intero
  System.out.println(""
    + "\t" + marcaP
    + "\t" + modelloP
    + "\t" + costoC);
  nomeF_old = nomeF ;
} // chiudo while
```

Esercizio 3

```
...
      result.close();
      query.close();
    }
    catch (SQLException exSQL){
      System.err.println("Errore nell'interrogazione");
      System.err.println("SQL Error Code: " + exSQL.getErrorCode()
        + ", SQL State" + exSQL.getSQLState());
      System.err.println("SQL Message: " + exSQL.getMessage());
    }
    finally {
      try {
        con.close();
        System.out.println("Chiusa connessione al database.");
      }
      catch (SQLException exSQL) {
        System.err.println("Errore nella chiusura connessione.");
        System.err.println("SQL Error Code: " + exSQL.getErrorCode()
          + ", SQL State" + exSQL.getSQLState());
        System.err.println("SQL Message: " + exSQL.getMessage());
      }
    }
  }
}
```