Medii vizuale de programare
Curs 5

Conf. dr.ing. GENGE Béla

Universitatea "Petru Maior", Departamentul de Informatica
Tîrgu Mureș, Romania
bela.genge@ing.upm.ro

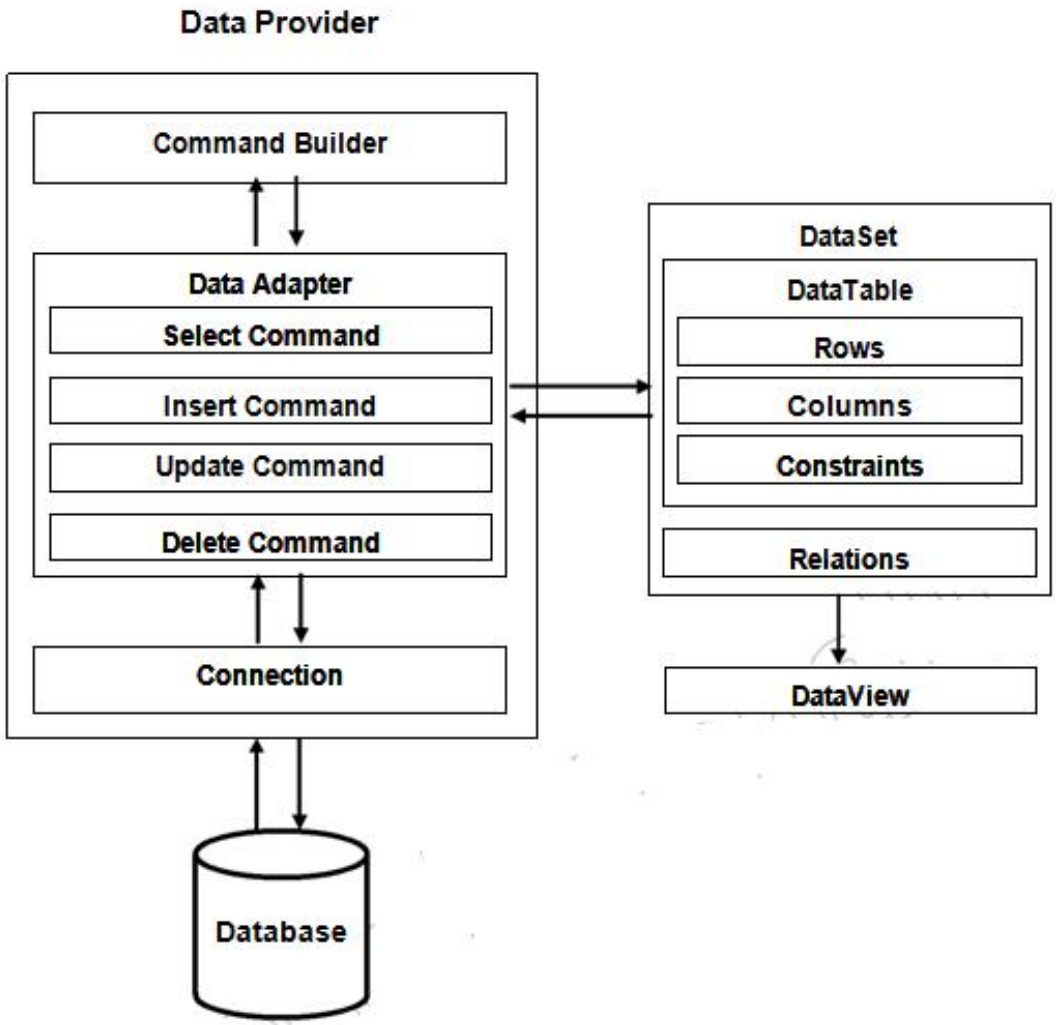
ADO.NET

- Un set de clase ce permite accesul la colecțiile de date
- Programatorului îi sunt oferite opțiuni bogate pentru conectarea la o baza de date
- Conexiunile pot fi permanente (scalabilitate redusa) sau temporare

ADO.NET - arhitectura

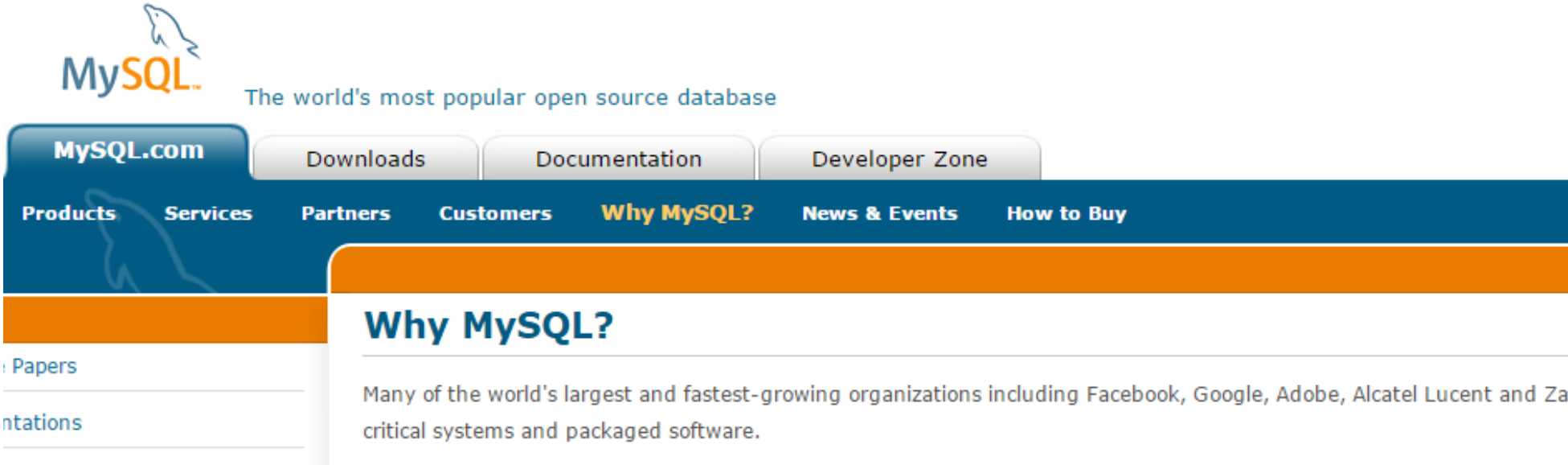
- Entitățile principale: `DataSet` și `DataProvider`
- `DataSet`:
  - Asigura o reprezentare în memorie a datelor
  - Implica realizarea unei copii locale a datelor asupra carora se lucrează
- `DataProvider`:
  - Asigura legatura cu baza de date
  - Furnizori de date:
    - Microsoft Access Database (OleDb .NET)
    - Microsoft ODBC
    - Microsoft SQL Server
    - Oracle

ADO.NET - arhitectura (sursa: programcall.com)



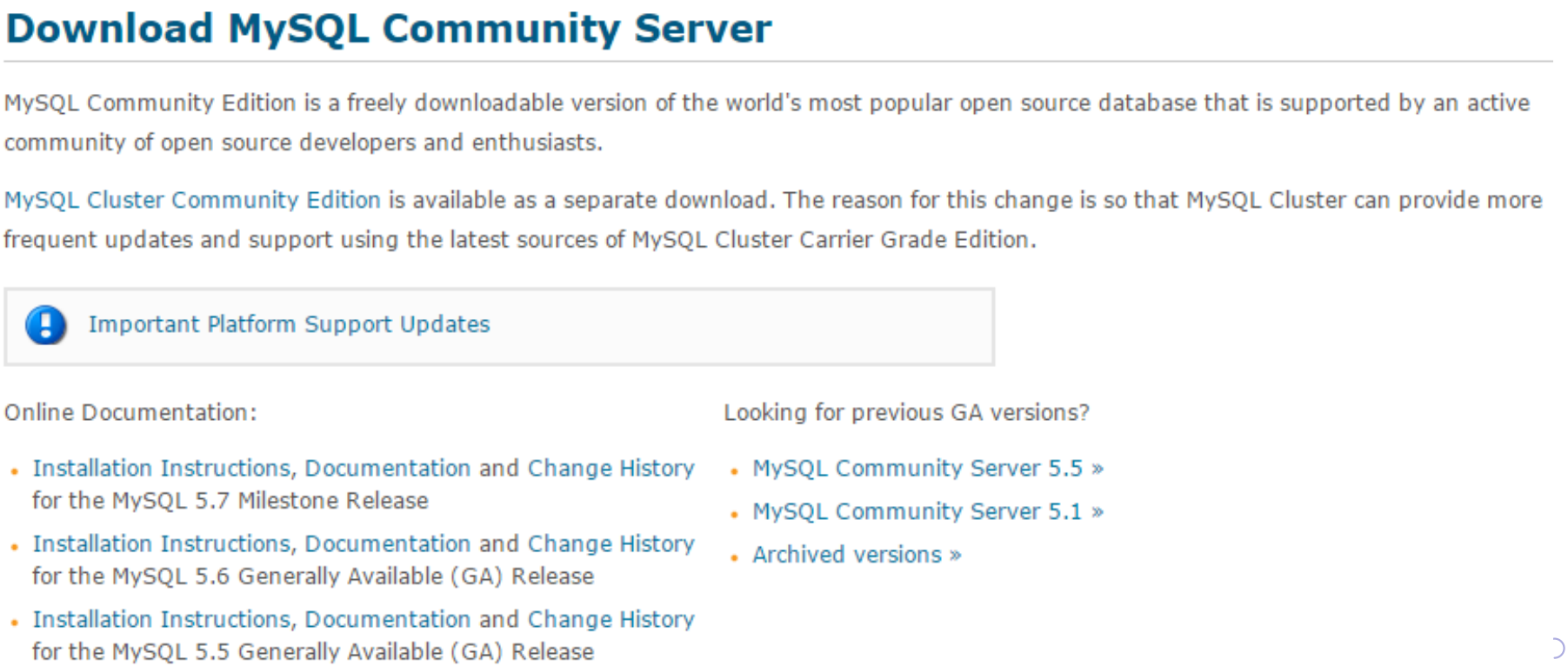
MySQL Server

- MySQL este cel mai popular server pentru baze de date.
- Proprietatea Oracle Corporation, proiectul este însă open source.
- Licența **Community edition** permite utilizarea gratuită a software-ului – trebuie însă plătită o licență dacă software-ul e inclus în aplicații comerciale.



MySQL Server - instalare

- Descărcare installer: `https://dev.mysql.com/downloads/mysql/`
- Instalare separat următoarele (full install automat poate eșua pe anumite sisteme - installer bug?):
  1. MySQL server (x86).
  2. .NET Connector (x86).
  3. MySQL Workbench (x86).



Download MySQL Community Server

MySQL Community Edition is a freely downloadable version of the world's most popular open source database that is supported by an active community of open source developers and enthusiasts.

MySQL Cluster Community Edition is available as a separate download. The reason for this change is so that MySQL Cluster can provide more frequent updates and support using the latest sources of MySQL Cluster Carrier Grade Edition.

Important Platform Support Updates

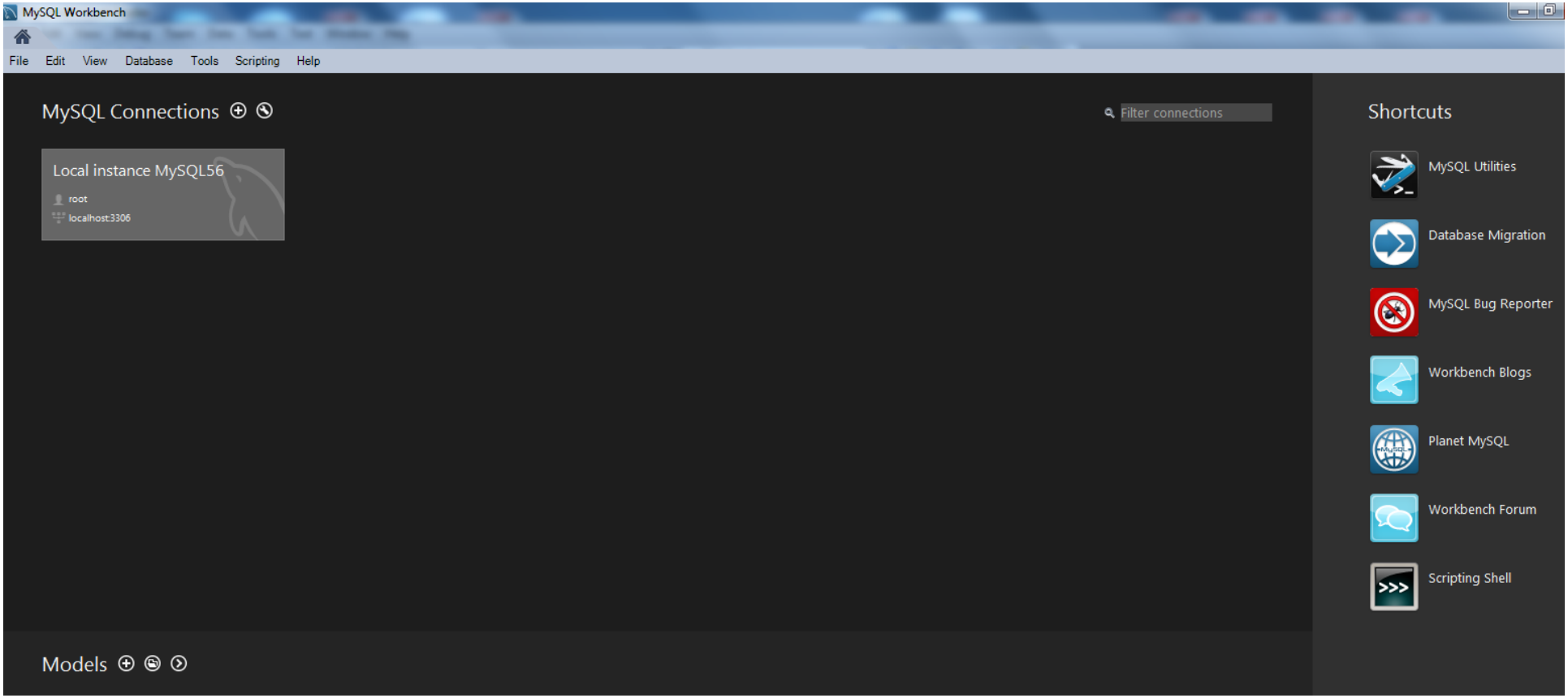
Online Documentation:

- Installation Instructions, Documentation and Change History for the MySQL 5.7 Milestone Release
- Installation Instructions, Documentation and Change History for the MySQL 5.6 Generally Available (GA) Release
- Installation Instructions, Documentation and Change History for the MySQL 5.5 Generally Available (GA) Release

Looking for previous GA versions?

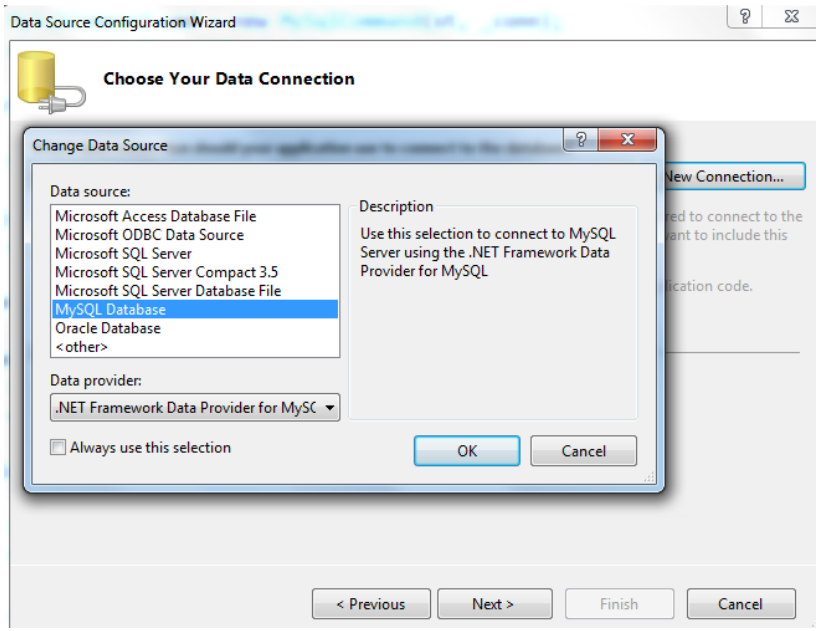
- MySQL Community Server 5.5 »
- MySQL Community Server 5.1 »
- Archived versions »

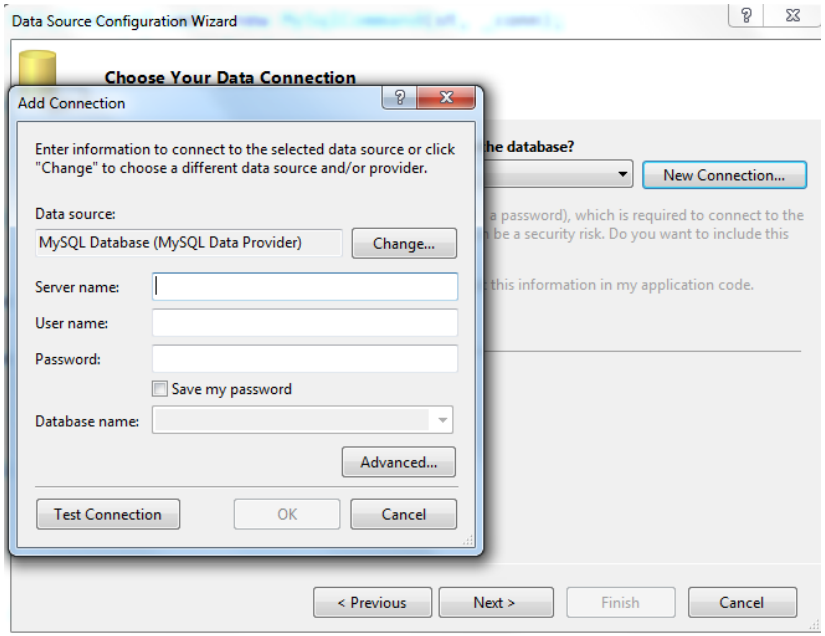
MySQL Workbench - crearea unei baze de date



MySQL Server - integrarea cu .NET

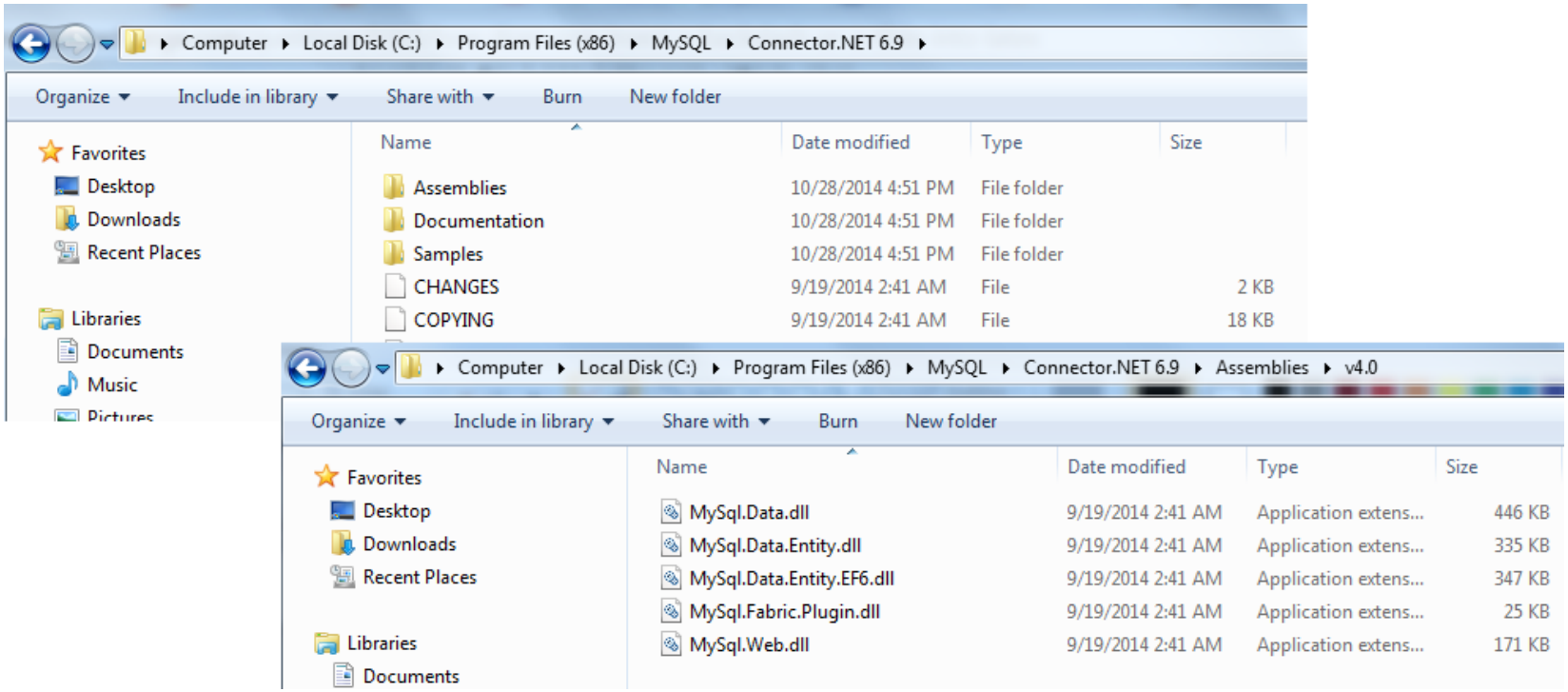
- MySQL .NET Connector.
- Adăugarea unei noi surse de tipul **MySQL data source** la proiect:
  - Bara de meniu: `Project` ⇒ `Add new data source` ⇒ `Database` ⇒ `DataSet` ⇒ `NewConnection` ... (String-ul de conexiune trebuie copiat, va fi folosit ulterior!!!)





MySQL Server - integrarea cu .NET

- Instalare MySQL .NET Connector.
- De regulă instalarea se face în: *C:\Program Files (x86)\MySQL\Connector.NET 6.9*
- Din directorul Assemblies se selectează versiunea framework-ului.
- În proiect se adaugă o referință către `MySql.Data.dll`



MySQL Server - crearea unei noi conexiuni

- Adăugarea namespace-ului: `using MySql.Data.MySqlClient`
- Instanțierea clase de conexiune: `_conn = new MySqlConnection()`
- Configurarea șirului de conectare:

```
_conn.ConnectionString =
"server=127.0.0.1;uid=root;pwd=root;database=student;";
```

- Deschiderea conexiunii: `_conn.Open()`
- Tratarea excepțiilor: `MySqlException`

Închidere conexiune

Închidere conexiune

```
if (null != _conn){
_conn.Close();
_conn.Dispose();
_conn = null;
}
```

MySQL Server - comenzi

- `ExecuteNonQuery`: comenzi care nu returnează date, e.g., `INSERT`, `UPDATE`, `DELETE`
- `ExecuteScalar`: comenzi ce returnează o singură valoare, e.g., count
- `ExecuteReader`: comenzi care returnează un set de date, e.g., `SELECT`, in `DataReader`

Comenzi SQL

- Toate comenzile SQL se construiesc sub forma unor șiruri de caractere (string).
- Execuția comenzilor cu `MySqlCommand`.

Creare comanda SQL

```
string stmt = "...";
MySqlCommand command = new MySqlCommand(stmt, _conn);
command.ExecuteNonQuery();
//sau
command.ExecuteScalar();
//sau
command.ExecuteReader();
```

Metode apelate din `MySqlCommand`

- `ExecuteNonQuery`: comenzi care nu returneaza înregistrari, e.g., `INSERT`, `UPDATE`, `DELETE`.
- `ExecuteScalar`: comenzi care returneaza o singura valoare, e.g., count.
- `ExecuteReader`: comenzi care returneaza un set de date, e.g., `SELECT`, în `DataReader`.

MySQL - creare tabel

- Construire comanda.
- Apel `ExecuteNonQuery`.
- Tratare eroare (excepție).

SQL Statement: creare tabel

```
string stmt="CREATE TABLE Users (id INT NOT NULL
AUTO_INCREMENT, Nume VARCHAR(45) NOT NULL, Prenume
VARCHAR(45) NOT NULL, AnNastere INT NOT NULL, PRIMARY
KEY(id))";
MySqlCommand cmd = new MySqlCommand(stmt, _conn);
try{
cmd.ExecuteNonQuery();
}
catch (MySqlException ex){
MessageBox.Show("Unable to create table! ERROR:" +
ex.ToString());
return;
```

MySQL - inserare înregistrare

- Construire comanda.
- Apel `ExecuteNonQuery`.
- Tratare eroare (excepție).

SQL Statement: inserare

```
string stmt="INSERT INTO Users(Nume, Prenume, AnNastere)
VALUES('Popescu', 'Adrian', '1980')";
MySqlCommand cmd = new MySqlCommand(stmt, _conn);
try{
cmd.ExecuteNonQuery();
}
catch (MySqlException ex){
MessageBox.Show("Unable to insert into table! ERROR:" +
ex.ToString());
return;
}
```

MySQL - Ștergere înregistrare

- Construire comanda.
- Apel `ExecuteNonQuery`.
- Tratare eroare (excepție).

SQL Statement: inserare

```
string stmt="DELETE FROM Users WHERE Nume='Popescu'";
MySqlCommand cmd = new MySqlCommand(stmt, _conn);
try{
cmd.ExecuteNonQuery();
}
catch (MySqlException ex){
MessageBox.Show("Unable to delete from table! ERROR:" +
ex.ToString());
return;
}
```

MySQL - citire înregistrari

- Construire comanda.
- Apel `ExecuteReader` - datele accesibile prin `MySqlDataReader`.
- Tratare eroare (excepție).
- Metode/proprietăți `MySqlDataReader`:
  - `Read()`: returnează `true` cât timp mai sunt înregistrari.
  - `FieldCount`: numărul de câmpuri.
  - `GetName(i)`: denumirea câmpului i.
  - `GetInt32(i)`: valoarea câmpului i pentru câmpuri de tipul INTEGER.
  - `GetString(i)`: valoarea câmpului i pentru câmpuri de tipul TEXT.
  - `Close()`: eliberează resursele alocate citirii - OBLIGATORIU trebuie apelat.
  - IMPORTANT! Alegerea apelului `GetInt32(i)` sau `GetString(i)` se face în cod în funcție de tipul coloanei.

MySQL - citire înregistrari

SQL Statement: citire înregistrari

```
string stmt="SELECT * FROM Users";
MySqlCommand cmd = new MySqlCommand(stmt, _conn);
MySqlDataReader reader = null;
try{reader = cmd.ExecuteReader();}catch(...){...}
string data = "Fields:";
for (int i = 0; i < reader.FieldCount; ++i)
  data += reader.GetName(i) + "\n";
data += "DATA: \n";
while (reader.Read()){
  data += reader.GetInt32(0).ToString() + " ";
  data += reader.GetString(1) + " ";
  data += reader.GetString(2) + " ";
  data += reader.GetInt32(3).ToString() + " ";
  data += "\n";}
reader.Close();
```

MySQL Server - select (cont.)

- Stocare date în DataGridView
- Ștergerea tuturor înregistrărilor: `grid.Rows.Clear()`
- Configurarea numărului de coloane: `grid.ColumnCount = data.FieldCount`
- Configurarea numelui coloanei: `grid.Columns[i].Name = data.GetName(i)`
- Adăugare înregistrare (rând nou): `idx = grid.Rows.Add()` - returnează indexul ultimului rând adăugat.
- Modificare valoare celulă: `grid.Rows[idx].Cells[i].Value = data.GetString(i)`

ADO.NET - SQL injection

- Prin preluarea neverificată a datelor introduse de utilizator se pot executa (injecta) comenzi SQL.
- Scopul: execuția unor comenzi aleatoare, vizualizare date, modify fields, execuția unor proceduri stocate, etc.
- Exemplu: SELECT bazat pe username.

Query

```
string sQuery = "SELECT Nume, Prenume, AnNastere FROM Users
where Nume='";
sQuery += textBox1.Text;
sQuery += "'";
```

ADO.NET - SQL injection

- Numele introdus: `nuconteaza' or 'x'='x`

Comanda construită

```
sQuery = "SELECT Nume, Prenume, AnNastere FROM Users
    WHERE Nume='nuconteaza' or 'x'='x'";
```

ADO.NET - Metode pentru prevenirea SQL injection

- Filtrarea caracterelor speciale (sanitizare), e.g., ”
- Transferul parametric al datelor - MySQL va filtra caracterele speciale.

Execuție comandă

```
string sQuery = "SELECT Nume, Prenume, AnNastere FROM Users
WHERE Nume=@Param1";
cmd.Parameters.AddWithValue("@Param1", textBox1.Text);
```