

Jazyk C# 2

6. seminář

Radek Janošík

Univerzita Palackého v Olomouci

22. 3. 2024

Reakce na úkoly

- Návrátová hodnota `FirstOrDefault`
- Nastavení jazyku porovnání
- Rozličná, různorodá řešení unikátnosti
- Porovnávání objektů pomocí `==` (řešení unikátnosti)

Databáze – úvod

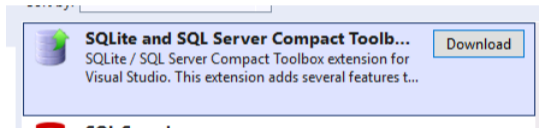
- = (ve stručnosti) persistentní úložiště dat
- Neřešíme konkrétní implementaci databáze, zajímá nás rozhraní
- SQL(Structured Query Language) – dotazovací jazyk, různé mutace
- Zde v kurzu – MSSQL, SQLite(Linux, MacOS)
- Data v tabulkách, každý sloupec pevný datový typ
- Řádky \approx objekty, Sloupce \approx vlastnosti objektů
- Více v kurzu databází – dnes jen zjednodušený pohled z C#

Databáze – vytvoření

- MSSQL Express – zdarma (určitý limit paměti, jader)
- Správa – MS SQL Server Management Studio
- Možnost testovat na `ts.inf.upol.cz` – připojení pomocí vzdálené plochy
- Integrace ve Visual Studiu – Server Explorer → Database Connection
- Alternativa MS SQL Server Database File
 - ▶ Databáze v souboru .mdf
 - ▶ Snadno přenositelné
 - ▶ Pro nás postačující
- Pro připojení k DB slouží tzv. Connection string (k nahlédnutí v Properties)

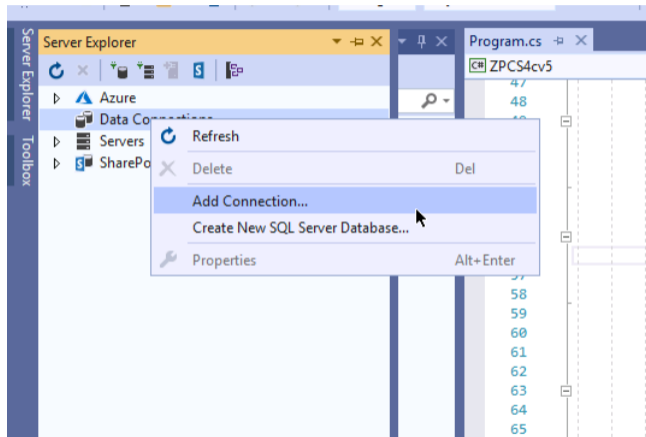
SQLite – VisualStudio

- Podpora SQLite ve VisualStudios není plnohodnotná
 - ▶ Nejde třeba vytvářet a spouštět libovolné dotazy
 - ▶ Nutno doinstalovat SQLite and SQL Server Compact Toolbox



- V ServerExplorer se poté objeví ikonka rozšíření
- V Rider podpora lepší

Připojení existující databáze ze souboru



Zvolíme soubor s databází

Add Connection ? X

Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.

Data source:
Microsoft SQL Server Database File (SqlClient) Change...

Database file name (new or existing):
\\tsclient\share\teaching\ZP4CS\2020\06\db.m Browse...

Log on to the server

Use Windows Authentication
 Use SQL Server Authentication

User name:
Password:

Save my password

Advanced...

Test Connection OK Cancel

Prohlížení DB

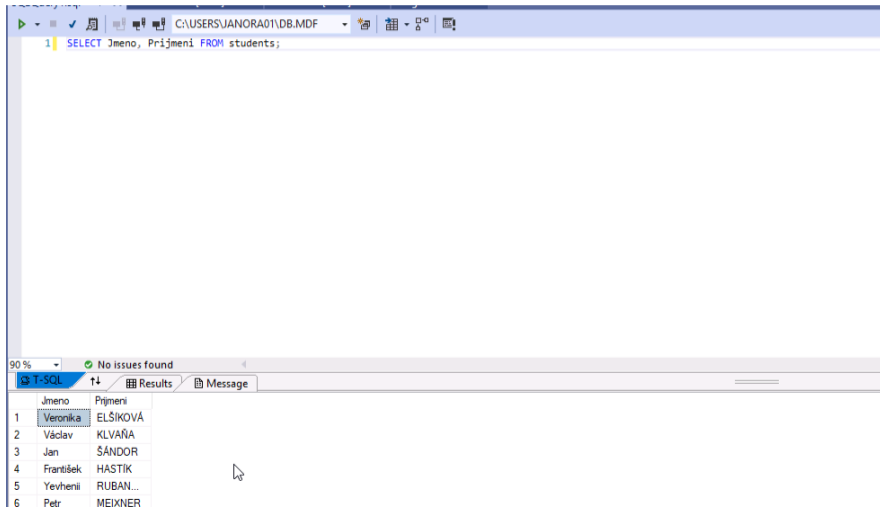
The screenshot shows the Visual Studio IDE with a SQL Server Enterprise edition connection. The Server Explorer on the left shows the database structure, with a context menu open over the 'dbo.exams' table. The menu options include: Add New Table, Add New Trigger, New Query, Open Table Definition (highlighted), Show Table Data, Copy (Ctrl+C), Delete (Del), Refresh, and Properties (Alt+Enter). The main window displays the 'dbo.exams' table data with columns: OsCislo, Jmeno, Prijmeni, UserName, Rocnik, and OborKomb. The table contains 30 rows of student data.

OsCislo	Jmeno	Prijmeni	UserName	Rocnik	OborKomb
R18557	Veronika	ELŠÍKOVÁ	elsive00	3	APLINF
R18569	Václav	KLVAŇA	klvava00	2	APLINF
R18603	Jan	ŠÁNDOR	sandja00	3	APLINF
R190175	František	HASTÍK	hastfr00	2	INF
R190176	Yevhenii	RUBANSKYI	rubaye00	2	INF
R190685	Petr	MEIXNER	meixpe00	2	INF
R190733	Lukáš	NETŘEBA	netrlu00	2	INF
R19118	Ondřej	BERČÍK	bercon00	2	INF
R19129	Zdeněk	RIEGL	riegzd00	2	INF
R19136	Martin	VÍTEK	vitema03	2	INF
R19173	Michal	ŠÍMČÍK	simcmi03	2	IVma-Zmi
R19548	Michal	KRUPÍK	krupmi02	2	INF
R19705	Michal	BEZDĚK	bezdm01	2	APLINF
R19707	Jakub	BRÁZDIL	brajja01	2	APLINF
R19709	Dominik Ján	BYSTRJANSKÝ	bystdo01	2	APLINF
R19733	Iveta	KOVAŘIKOVÁ	kovaii04	2	APLINF
R197333	Martin	KRÚZA	kruzma01	2	APLINF
R19734	Martin	KRÚZA	kruzma00	2	APLINF
R1973444	Martin	KRÚZA	kruzma02	2	INF
R19736	David	KUČERA	kucedo07	2	APLINF
R19739	Rostislav	LÍŠKA	liskro00	2	APLINF
R19743	Erik Daniel	MURGAŠ	murger00	2	APLINF
R19744	Tomáš	NÁDVORNÍK	nadvto01	2	APLINF
R19747	Katerina	OLEIKOVÁ	otejka02	2	APLINF
R19748	Ondřej	PAVELKA	paveon03	2	APLINF
R19750	Václav	PROCHÁZKA	procv01	2	APLINF
R19752	Karyna	ROZNIUK	rozka02	2	APLINF
R19753	Marek	SCHINDLER	schima09	2	APLINF

- Můžeme se podívat na data v tabulce nebo zobrazit její definici.

Spouštění SQL dotazů

- Pravým tlačítkem myši na databázi nebo tabulku ⇒ New Query
- Tlačítkem „Play“ dotaz odešleme na server



The screenshot shows a SQL Server Enterprise Manager interface. At the top, the file path is C:\USERS\JANORA01\DB.MDF. The query window contains the following SQL query:

```
1 | SELECT Jmeno, Prijmeni FROM students;
```

Below the query window, the results are displayed in a table with the following data:

	Jmeno	Prijmeni
1	Veronika	ELŠIKOVÁ
2	Václav	KLVAŇA
3	Jan	ŠÁNDOR
4	František	HASTIK
5	Yevhenii	RUBAN...
6	Petr	MEIXNER

Jazyk SQL (1/3) – Základní dotazy

- Výběr dat

```
1 SELECT sloupec1, sloupec2 FROM tabulka WHERE podminka;
```

- Např.:

```
1 SELECT name, surname FROM students WHERE year=3;
```

```
2 SELECT * FROM subjects WHERE name = 'Jan';
```

- Řazení

```
1 SELECT name, surname FROM students ORDER BY surname DESC;
```

```
2 SELECT * FROM subjects WHERE obor = 'KMI' ORDER BY year ASC;
```

- Pozor – porovnání je pouze = místo ==

Jazyk SQL (2/3) – Základní dotazy

- Editace záznamu

```
1 UPDATE tabulka SET sloupec1=hodnota1, sloupec2=hodnota2 WHERE  
   podmínka;
```

- Např.:

```
1 UPDATE students SET year=3, obor = 'MI' WHERE year=3;  
2 UPDATE subjects SET name='XXXXXX';
```

- Vložení záznamu

```
1 INSERT INTO tabulka (sloupec1, sloupec2, ...) VALUES (hodnota1,  
   hodnota2, ...);
```

- Např.:

```
1 INSERT INTO students (name, surname) VALUES ('Petr', 'Novak');
```

Jazyk SQL (3/3) – Základní dotazy

- Smazání záznamu

```
1 DELETE FROM tabulka WHERE podminka;
```

- Např.:

```
1 DELETE FROM students WHERE id=35;  
2 DELETE FROM students;
```

- Další dotazy (`GROUP BY`, `CREATE TABLE`, `JOIN`,...) viz. dokumentace

- [https://technet.microsoft.com/en-us/library/ms189826\(v=sql.90\).aspx](https://technet.microsoft.com/en-us/library/ms189826(v=sql.90).aspx)

Zjištění connection stringu

- Abychom se mohli k DB připojit z kódu, potřebujeme tzv. **Connection String**
- Najdeme jej v **Properties** databáze:

The screenshot shows the Visual Studio IDE. On the left, the Solution Explorer shows a project named 'ConsoleApp23' with files 'Properties', 'References', 'App.config', and 'Program.cs'. The Properties window is open, showing the 'db.mdf Connection' properties. The 'Connection String' property is highlighted, showing the value: `Data Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\janora01\db.mdf;Integrated Security=True;Connect Timeout=30`. Below the Properties window, a SQL query result is displayed in a table with 7 rows and 2 columns: 'Jmeno' and 'Příměří'. The output shows names like 'Veronika ELŠIKOVÁ', 'Václav KLVARA', 'Jan ŠANDOR', 'František HASTIK', 'Yevhenii RUBAN...', 'Petr MEIXNER', and 'Lukáš NETŘEBA'. A status bar at the bottom indicates 'Query executed successfully at 14:24:39'.

Jmeno	Příměří
1 Veronika	ELŠIKOVÁ
2 Václav	KLVARA
3 Jan	ŠANDOR
4 František	HASTIK
5 Yevhenii	RUBAN...
6 Petr	MEIXNER
7 Lukáš	NETŘEBA

Připojení DB v C#

- Namespace `Microsoft.Data.SqlClient`
- Pro SQLite `Microsoft.Data.Sqlite` <https://learn.microsoft.com/cs-cz/dotnet/standard/data/sqlite/?tabs=netcore-cli>
 - ▶ Možná nutné stáhnout přes NuGet
- Připojení a čtení výsledků dotazu
- Vytvoříme si `SqlConnection`, otevřeme spojení
- Poté vytvoříme příkaz `SqlCommand` a vytvoříme reader, ze kterého přečteme všechna data
- Data indexována podle pořadí sloupců v definici tabulky

Dotaz s parametry

```
1 try {
2     using (SqlConnection conn = new SqlConnection(connectionString))
3     {
4         conn.Open();
5         SqlCommand command = new SqlCommand("SELECT * FROM students;",
6             conn);
7         using (SqlDataReader dr = command.ExecuteReader())
8         {
9             while (dr.Read())
10            {
11                Console.WriteLine($"{dr[0]}, {dr[1]}, {dr[2]}");
12            }
13        }
14    } catch (Exception e) {
15        ...
16    }
```

Dotaz s parametry

- Proč je špatně:

```
1 string someName <- nacteme data od uzivatele
2 string someSurname <- nacteme data od uzivatele
3 SqlCommand command = new SqlCommand($"SELECT * FROM students
4     WHERE name='{someName}' OR surname='{someSurname}';",
    conn);
```


Dotaz s parametry

- Proč je špatně:

```
1 string someName <- nacteme data od uzivatele
2 string someSurname <- nacteme data od uzivatele
3 SqlCommand command = new SqlCommand($"SELECT * FROM students
4     WHERE name='{someName}' OR surname='{someSurname}';",
    conn);
```

- A správně

```
1 SqlCommand command = new SqlCommand("SELECT * FROM students
2     WHERE name = @someParam OR surname=@surname", conn);
3 command.Parameters.Add(new SqlParameter("someParam", someName));
4 command.Parameters.Add(new SqlParameter("surname", someSurname));
```

Dotaz s parametry

- Proč je špatně:

```
1 SqlCommand command = new SqlCommand($"SELECT * FROM students
2     WHERE name='{someName}' OR surname='{someSurname}';",
3     conn);
```

- A správně

```
1 SqlCommand command = new SqlCommand("SELECT * FROM students
2     WHERE name = @someParam OR surname=@surname", conn);
3 command.Parameters.Add(new SqlParameter("someParam", someName));
4 command.Parameters.Add(new SqlParameter("surname", someSurname));
```

- Odpověď: nikdy bychom neměli vkládat do dotazů surový obsah proměnných. Zvláště, pokud je může modifikovat uživatel aplikace.
- Parametrické dotazy pomocí @ pomůžou escapovat obsahy proměnných – zamezení SQL Injection útoku
- Viz https://www.w3schools.com/sql/sql_injection.asp

Vyhodnocení dotazu bez výsledků

- Metoda `ExecuteNonQuery()` – vrací počet ovlivněných prvků
- Lze volat asynchronně
- Vložení prvku

```
1 SqlCommand command = new SqlCommand("INSERT INTO students (id,  
    name, surname) VALUES (@id, @name, @surname);", conn);  
2 command.Parameters.Add(new SqlParameter("id", 3));  
3 command.Parameters.Add(new SqlParameter("name", "Alois"));  
4 command.Parameters.Add(new SqlParameter("surname", "Fridrich"));  
5 int affected = command.ExecuteNonQuery();
```

Vyhodnocení dotazu bez výsledků

- Metoda `ExecuteNonQuery()` – vrací počet ovlivněných prvků
- Lze volat asynchronně
- Vložení prvku

```
1 SqlCommand command = new SqlCommand("INSERT INTO students (id,  
    name, surname) VALUES (@id, @name, @surname);", conn);  
2 command.Parameters.Add(new SqlParameter("id", 3));  
3 command.Parameters.Add(new SqlParameter("name", "Alois"));  
4 command.Parameters.Add(new SqlParameter("surname", "Fridrich"));  
5 int affected = command.ExecuteNonQuery();
```

- Vymazání prvku(ů)

```
1 SqlCommand command = new SqlCommand("DELETE FROM students WHERE  
    surname=@srn", conn);  
2 command.Parameters.Add(new SqlParameter("srn", "SOUKUP"));  
3 int affected = command.ExecuteNonQuery();  
4 Console.WriteLine("Affected: " + affected);
```

Úkol

- Stáhnout soubor db.mdf z <https://apollo.inf.upol.cz/~janostik/data/db.mdf>, připojit ve VS a zjistit connection string
 - ▶ Případně <https://apollo.inf.upol.cz/~janostik/data/db.sqlite> pro Rider/MacOS
- Z databáze „vytáhnout“ abecedně vzestupně seříděné osoby dle příjmení a vypsat 6 unikátních dvojic jmen a příjmení, které jsou 3. - 8. v pořadí
- Vložit dva nové studenty (smysluplná data)
- Vymazat studenty, kteří studují IT
- Změnit userId studenta s OsČíslem "R21390" na "holuma08"
- Pro zbylé studenty zjistit jejich známky ze všech zkoušek
- **Při řešení co nejvíce použít SQL**
- Vše vypsat do konzole v pořadí úkolů a ošetřit sql chyby odchycením výjimek