

Úvod do databázových systémů

3. cvičení

Ing. Petr Lukáš
petr.lukas@nativa.cz

Ostrava, 2013

Opakování

- **Operátory relační algebry**
- **4 fáze vytváření databáze**
- **Rozdíl mezi konceptuálním a relačním modelem**

- **Primární klíč**

- **Cizí klíč**

- **Typy vztahů a jejich převod na relační model**

Opakování

- **Operátory relační algebry**
selekce, projekce, kartézský součin, sjednocení, průnik, rozdíl
- **4 fáze vytváření databáze**
zadání, konceptuální model, relační datový model, implementace
- **Rozdíl mezi konceptuálním a relačním modelem**
Konceptuální model neudává, jakým způsobem bude databáze řešena (zda použijeme tabulky, stromovou strukturu, objektově orientovanou databázi atd.)
V konceptuální modelu neuvídáme cizí klíče.
- **Primární klíč**
Atribut nebo více atributů, jejichž hodnota nebo kombinace hodnot je pro každý záznam v relaci jedinečná.
- **Cizí klíč**
Atribut nebo více atributů, které se odkazují na primární klíč z jiné (cizí) tabulky.
- **Typy vztahů a jejich převod na relační model**
1:N – převádíme přidáním cizího klíče na stranu N
M:N – řešíme vždy další vazební tabulkou

SQL

Structured Query Language

- 1. Rozdělení jazyků pro práci s daty**
- 2. Klauzule příkazu SELECT**
- 3. Příklad 1 – výběr tabulky**
- 4. Příklad 2 – filtrování**
- 5. Příklad 3 – výběr sloupců**
- 6. Příklad 4 – třídění**
- 7. Syntaxe SQL**
- 8. Štábní kultura**
- 9. Microsoft SQL Server Management Studio**

SQL

Jazyk SQL (Structured Query Language)

- Standard pro komunikaci s relačními databázemi
- Podporován většinou moderních relačních SŘBD

Jazyk SQL (Structured Query Language)

- Standard pro komunikaci s relačními databázemi
- Podporován většinou moderních relačních SŘBD



193 systems in ranking, September 2013

Rank	Last Month	DBMS	Database Model	Score	Changes
1.		1. Oracle	Relational DBMS	1529.61	-14.83
2.	↑	3. Microsoft SQL Server	Relational DBMS	1313.78	+8.82
3.	↓	2. MySQL	Relational DBMS	1305.76	-19.07

<http://db-engines.com>

Rozdělení jazyků pro práci s daty

Rozdělení jazyků pro práci s daty

- **DDL** (Data Definition Language)
Vytváření datových struktur (prázdných tabulek)
- **DML** (Data Manipulation Language)
Vkládání, úprava a mazání dat
- **DQL** (Data Query Language)
Dotazování nad daty

Rozdělení jazyků pro práci s daty

- **DDL** (Data Definition Language)
Vytváření datových struktur (prázdných tabulek)
- **DML** (Data Manipulation Language)
Vkládání, úprava a mazání dat
- **DQL** (Data Query Language)
Dotazování nad daty



SQL

Rozdělení jazyků pro práci s daty

- **DDL** (Data Definition Language)
Vytváření datových struktur (prázdných tabulek)
- **DML** (Data Manipulation Language)
Vkládání, úprava a mazání dat
- **DQL** (Data Query Language)
Dotazování nad daty



SQL

Klauzule příkazu SELECT

Klauzule příkazu SELECT

SELECT	<i>seznam sloupců na výstupu</i>
FROM	<i>vstupní tabulky</i>
WHERE	<i>podmínka</i>
GROUP BY	<i>seskupení</i>
HAVING	<i>podmínka skupiny</i>
ORDER BY	<i>seřazení výsledku</i>

Ne všechny klauzule jsou povinné, ale jejich pořadí je striktně dané!

Klauzule příkazu SELECT

SELECT	<i>seznam sloupců na výstupu</i>
FROM	<i>vstupní tabulky</i>
WHERE	<i>podmínka</i>
GROUP BY	<i>seskupení</i>
HAVING	<i>podmínka skupiny</i>
ORDER BY	<i>seřazení výsledku</i>

Ne všechny klauzule jsou povinné, ale jejich pořadí je striktně dané!

1. Příklad – výběr všech dat z tabulky

Student

login	jmeno	prijmeni	datum_narozeni
nov078	Jan	Novák	5.9.1987
luk194	Petr	Lukáš	7.6.1988
lak065	Luboš	Lakatoš	2.8.1991



Vyber vše z tabulky Student

1. Příklad – výběr všech dat z tabulky

Student

login	jmeno	prijmeni	datum_narozeni
nov078	Jan	Novák	5.9.1987
luk194	Petr	Lukáš	7.6.1988
lak065	Luboš	Lakatoš	2.8.1991



Vyber vše z tabulky Student

```
SELECT *  
FROM Student
```



login	jmeno	prijmeni	datum_narozeni
nov078	Jan	Novák	5.9.1987
luk194	Petr	Lukáš	7.6.1988
lak065	Luboš	Lakatoš	2.8.1991

2. Příklad – filtrování

Student

login	jmeno	prijmeni	datum_narozeni
nov078	Jan	Novák	5.9.1987
luk194	Petr	Lukáš	7.6.1988
lak065	Luboš	Lakatoš	2.8.1991



Vyber všechny studenty, kteří se narodili po 1.1.1988

2. Příklad – filtrování

Student

login	jmeno	prijmeni	datum_narozeni
nov078	Jan	Novák	5.9.1987
luk194	Petr	Lukáš	7.6.1988
lak065	Luboš	Lakatoš	2.8.1991



Vyber všechny studenty, kteří se narodili po 1.1.1988

```
SELECT *  
FROM Student  
WHERE datum_narozeni > '1.1.1988'
```



login	jmeno	prijmeni	datum_narozeni
luk194	Petr	Lukáš	7.6.1988
lak065	Luboš	Lakatoš	2.8.1991

2. Příklad – filtrování (srovnání se selekcí)

Student

login	jmeno	prijmeni	datum_narozeni
nov078	Jan	Novák	5.9.1987
luk194	Petr	Lukáš	7.6.1988
lak065	Luboš	Lakatoš	2.8.1991



```
SELECT *  
FROM Student  
WHERE datum_narozeni > '1.1.1988'
```

$\sigma_{datum_narozeni > "1.1.1988"} Student$



login	jmeno	prijmeni	datum_narozeni
luk194	Petr	Lukáš	7.6.1988
lak065	Luboš	Lakatoš	2.8.1991

3. Příklad – výběr sloupců

Student

login	jmeno	prijmeni	datum_narozeni
nov078	Jan	Novák	5.9.1987
luk194	Petr	Lukáš	7.6.1988
lak065	Luboš	Lakatoš	2.8.1991



Vyber jména a příjmení všech studentů

3. Příklad – výběr sloupců

Student

login	jmeno	prijmeni	datum_narozeni
nov078	Jan	Novák	5.9.1987
luk194	Petr	Lukáš	7.6.1988
lak065	Luboš	Lakatoš	2.8.1991



Vyber jména a příjmení všech studentů

```
SELECT jmeno, prijmeni  
FROM Student
```



jmeno	prijmeni
Jan	Novák
Petr	Lukáš
Luboš	Lakatoš

3. Příklad – výběr sloupců (srovnání s projekcí)

Student

login	jmeno	prijmeni	datum_narozeni
nov078	Jan	Novák	5.9.1987
luk194	Petr	Lukáš	7.6.1988
lak065	Luboš	Lakatoš	2.8.1991



```
SELECT jmeno, prijmeni  
FROM Student
```

$\pi_{jmeno, prijmeni} Student$



jmeno	prijmeni
Jan	Novák
Petr	Lukáš
Luboš	Lakatoš

4. Příklad – spojování tabulek

Student

login	jmeno	skola
nov078	Jan	1
luk194	Petr	1
lak065	Luboš	2

Skola

id	nazev
1	VŠB – TUO
2	Ostravská univerzita



Vyber jména studentů a k nim názvy škol, ve kterých studují

4. Příklad – spojování tabulek

Student

login	jmeno	skola
nov078	Jan	1
luk194	Petr	1
lak065	Luboš	2

Skola

id	nazev
1	VŠB – TUO
2	Ostravská univerzita



Vyber jména studentů a k nim názvy škol, ve kterých studují

```
SELECT jmeno, nazev  
FROM Student, Škola  
WHERE Student.skola = Skola.id
```



jmeno	nazev
Jan	VŠB – TUO
Petr	VŠB – TUO
Luboš	Ostravská univerzita

4. Příklad – spojování tabulek (srovnání s RA)

Student

login	jmeno	skola
nov078	Jan	1
luk194	Petr	1
lak065	Luboš	2

Skola

id	nazev
1	VŠB – TUO
2	Ostravská univerzita



```
SELECT jmeno, nazev  
FROM Student, Skola  
WHERE Student.skola = Skola.id
```

$$\pi_{jmeno, nazev}(\sigma_{skola=id} (Student \times Skola))$$


jmeno	nazev
Jan	VŠB – TUO
Petr	VŠB – TUO
Luboš	Ostravská univerzita

5. Příklad – řazení

Student

login	jmeno	prijmeni	datum_narozeni
nov078	Jan	Novák	5.9.1987
luk194	Petr	Lukáš	7.6.1988
lak065	Luboš	Lakatoš	2.8.1991



Vyber všechny studenty setříděné sestupně dle data narození

5. Příklad – řazení

Student

login	jmeno	prijmeni	datum_narozeni
nov078	Jan	Novák	5.9.1987
luk194	Petr	Lukáš	7.6.1988
lak065	Luboš	Lakatoš	2.8.1991



Vyber všechny studenty setříděné sestupně dle data narození

```
SELECT *  
FROM Student  
ORDER BY datum_narozeni DESC
```



login	jmeno	prijmeni	datum_narozeni
lak065	Luboš	Lakatoš	2.8.1991
luk194	Petr	Lukáš	7.6.1988
nov078	Jan	Novák	5.9.1987

5. Příklad – řazení

Student

login	jmeno	prijmeni	datum_narozeni
nov078	Jan	Novák	5.9.1987
luk194	Petr	Lukáš	7.6.1988
lak065	Luboš	Lakatoš	2.8.1991



Vyber všechny studenty setříděné sestupně dle data narození

Jak by vypadal odpovídající výraz relační algebry?

5. Příklad – řazení

Student

login	jmeno	prijmeni	datum_narozeni
nov078	Jan	Novák	5.9.1987
luk194	Petr	Lukáš	7.6.1988
lak065	Luboš	Lakatoš	2.8.1991



Vyber všechny studenty setříděné sestupně dle data narození

Jak by vypadal odpovídající výraz relační algebry?

- **Nijak**, relační algebra (tak, jak jsme si ji ukazovali minule) pracuje s množinami. Tzn. výstupem každého operátoru je množina. Žádný operátor na množině nezajistí její uspořádání.
- SQL nepracuje s tabulkami jako s množinami n-tic, ale jako se seznamy záznamů (uspořádanými multimnožinami)

- **Booleovské výrazy**

Používáme spojky **AND**, **OR**, **NOT** pro logický součin, součet a negaci.

- **Operátory porovnání**

Porovnání na shodnost pomocí **=** (nepoužívá se **==** jako v C++). Dále obvyklý význam mají znaky **<**, **>**, **<>** nebo **!=**. Speciální význam má porovnání s tzv. NULL hodnotou – výrazy **IS NULL** nebo **IS NOT NULL**.

Syntaxe SQL (2)

- **Aritmetika**

Standardní význam mají operátory **+**, **-**, *****, **/**, **%** (modulo).
Pozor, u dělení celých čísel vychází opět celé číslo!
+ může sloužit také pro konkatenci (spojení) řetězců.

- **Textové řetězce**

Textové **řetězce** se zapisují mezi jednoduché apostrofy.
Speciální operátor **LIKE** slouží k porovnání řetězce s regulárním výrazem, kde znak **%** zastupuje libovolný počet libovolných znaků.

*např. **jmeno LIKE 'P%'** – všechna jména začínající na "P".*

- **Klauzule SELECT**

Bezprostředně za klauzulí select se mohou nacházet (mimo jiné) následující modifikátory:

DISTINCT – omezení duplicit na výstupu.

```
SELECT DISTINCT jmeno  
FROM Student
```

Vrať všechna křestní jména studentů.

TOP n – zajímá nás jen prvních n výsledků. Používá se např. pokud chceme jen „nahlédnout“, jak data vypadají nebo velmi výhodně také v kombinaci s tříděním.

```
SELECT TOP 1 login  
FROM Student  
ORDER BY datum_narozeni
```

Vrať login nejstaršího studenta.

- **Množinové operace**

Přestože se SQL, jak jsme si ukazovali, nedívá na tabulky jako na množiny, podporuje množinové operace:

UNION – sjednocení

INTERSECT – průnik

EXCEPT – rozdíl

```
SELECT jmeno  
FROM Student1  
UNION  
SELECT jmeno  
FROM Student2
```

Vrať dokupy jména studentů z tabulek *Student1* a *Student2*.

Štábní kultura

```
select TOP 10 jmeno FROM STUDent,  
vyrobek  
    where prijmeni = 'Lukáš' ORDER  
BY  
rok_narozeni
```

vs.

```
SELECT TOP 10 jmeno  
FROM Student, Vyrobek  
WHERE prijmeni = 'Lukáš'  
ORDER BY rok_narozeni
```

- Každá **klauzule** začíná na novém řádku.
- U složitých dotazů je vhodné psát klíčová slova klauzulí na zvláštní řádky.
- **Klíčová slova** píšeme velkými písmeny.
- **Názvy tabulek a atributů** píšeme stejně, jako byly definovány. Tzn. máme-li např. tabulku Student, nebudeme se na ni odvolávat pomocí STUDENT.
- Kdykoli si nejsme jisti prioritou operátorů, použijeme **závorky**.

- **Rozdělení jazyků**

- Jazyky pro dotazování (DQL), jazyky pro modifikaci dat (DML), jazyky pro definici dat (DDL).

- **6 klauzulí příkazu SELECT**

- SELECT, FROM, WHERE, GROUP BY, HAVING, ORDER BY

- **Syntaxe SQL**

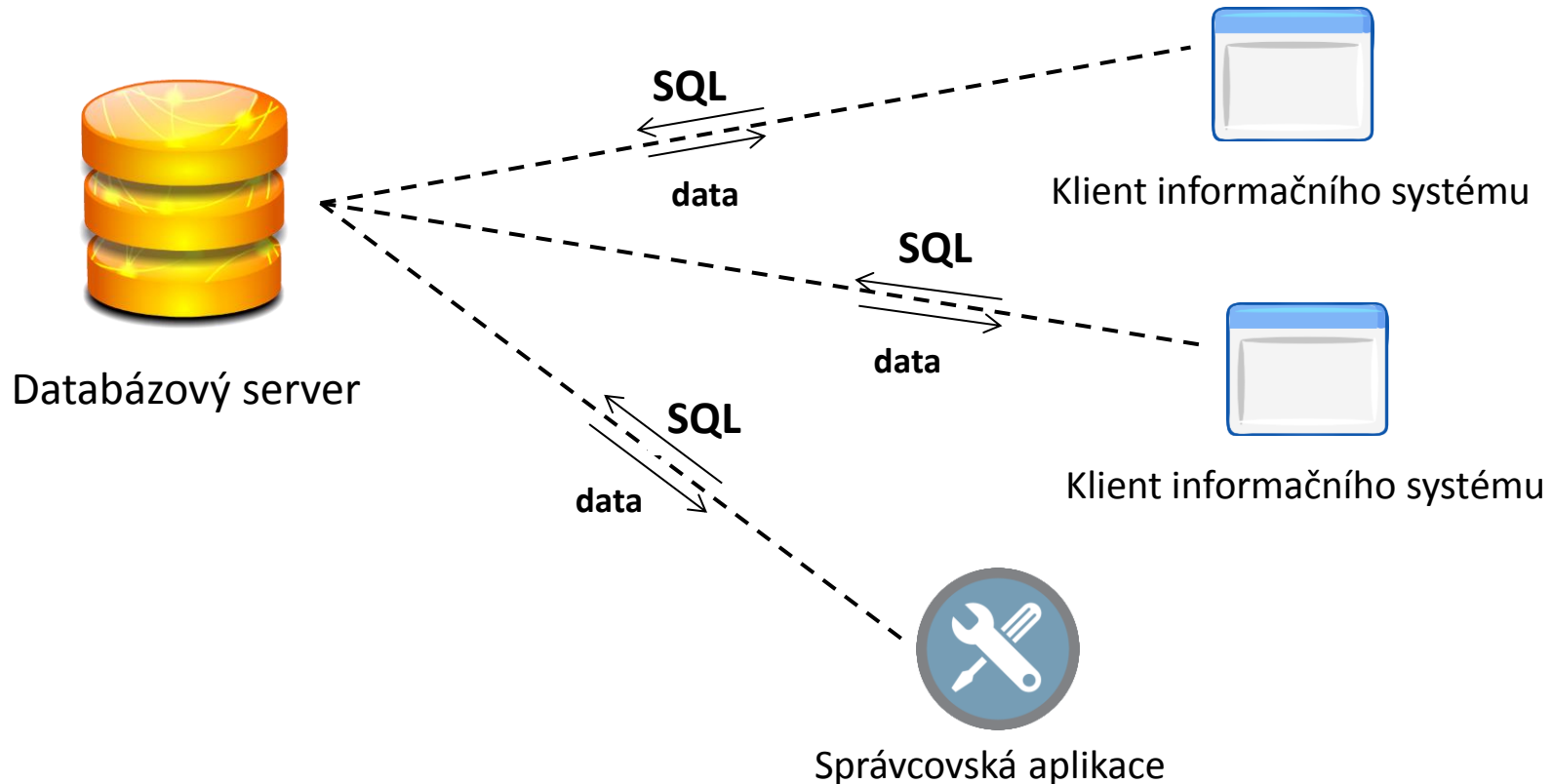
- Operátory, booleovské výrazy, práce s řetězci

- **Štábní kultura**

- Každá klauzule na novém řádku, klíčová slova velkými písmeny, tabulky a atributy pojmenováváme tak, jak byly definovány

Připojení na server

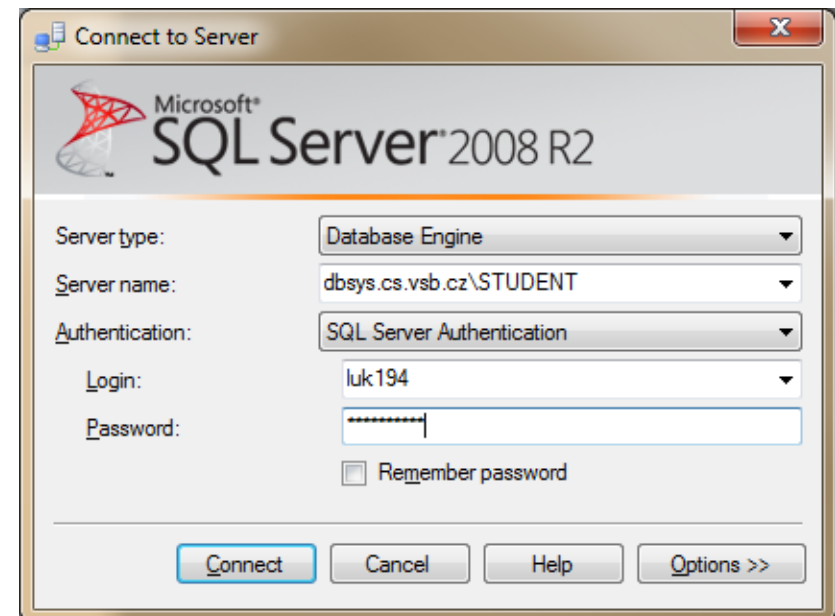
- Po celou dobu v rámci kurzu UDBS budeme pracovat se SŘBD **Microsoft SQL Server 2012**.
- Jako administrátoři databáze budeme využívat zejména správcovskou aplikaci **Microsoft SQL Server Management Studio**.





Microsoft SQL Server Management Studio (1)

- Po spuštění SSMS (SQL Server Management Studio) je potřeba připojit se ke správnému serveru.
 - Jako **Server name** uveďte “**dbsys.cs.vsb.cz\STUDENT**”
 - Vyberte typ autentifikace “**SQL Server Authentication**”
 - **Login** je váš login, kterým se přihlašujete do Edisonu
 - **Password** je heslo, které vám bylo **zasláno e-mailem**. Zkontrolujte tedy školní poštu.
-
- Ke školnímu SQL serveru je možné připojit se např. i z domu, ale je potřeba být ve školní VPN.



Microsoft SQL Server Management Studio (2)

Nový dotaz

Otevře novou
prázdnou záložku pro
psaní skriptů

Seznam databází

Na jednom serveru
může běžet více
databází

Seznam tabulek

Jednotlivé tabulky v
databázi

The screenshot shows the Microsoft SQL Server Management Studio interface. The 'Object Explorer' on the left displays the server structure for '(local) (SQL Server 10.50.1617 - Petr-NB\Petr)'. The 'Tables' folder under 'dbo' is expanded, showing 'dbo.koupil', 'dbo.uzivatel', and 'dbo.vyrobek'. The 'SQLQuery2.sql' window on the right contains the following SQL query:

```
SELECT *  
FROM uzivatel
```

The 'Results' pane at the bottom shows the output of the query, which is a table with the following data:

	login	rok_narzeni	mesto
1	kasa	1981	Ostrava
2	knedlik	1977	Bmo
3	kuchta	1996	Olomouc
4	malta	1985	Opava
5	pepik	1991	Praha
6	ruprt	1983	Praha
7	rychlarota	1984	Bmo
8	stelar	1994	Ostrava
9	vinetu	1976	Zlin

The status bar at the bottom indicates 'Query executed successfully.' and shows the server name '(local) (10.50 RTM) Petr-NB'.

Microsoft SQL Server Management Studio (3)

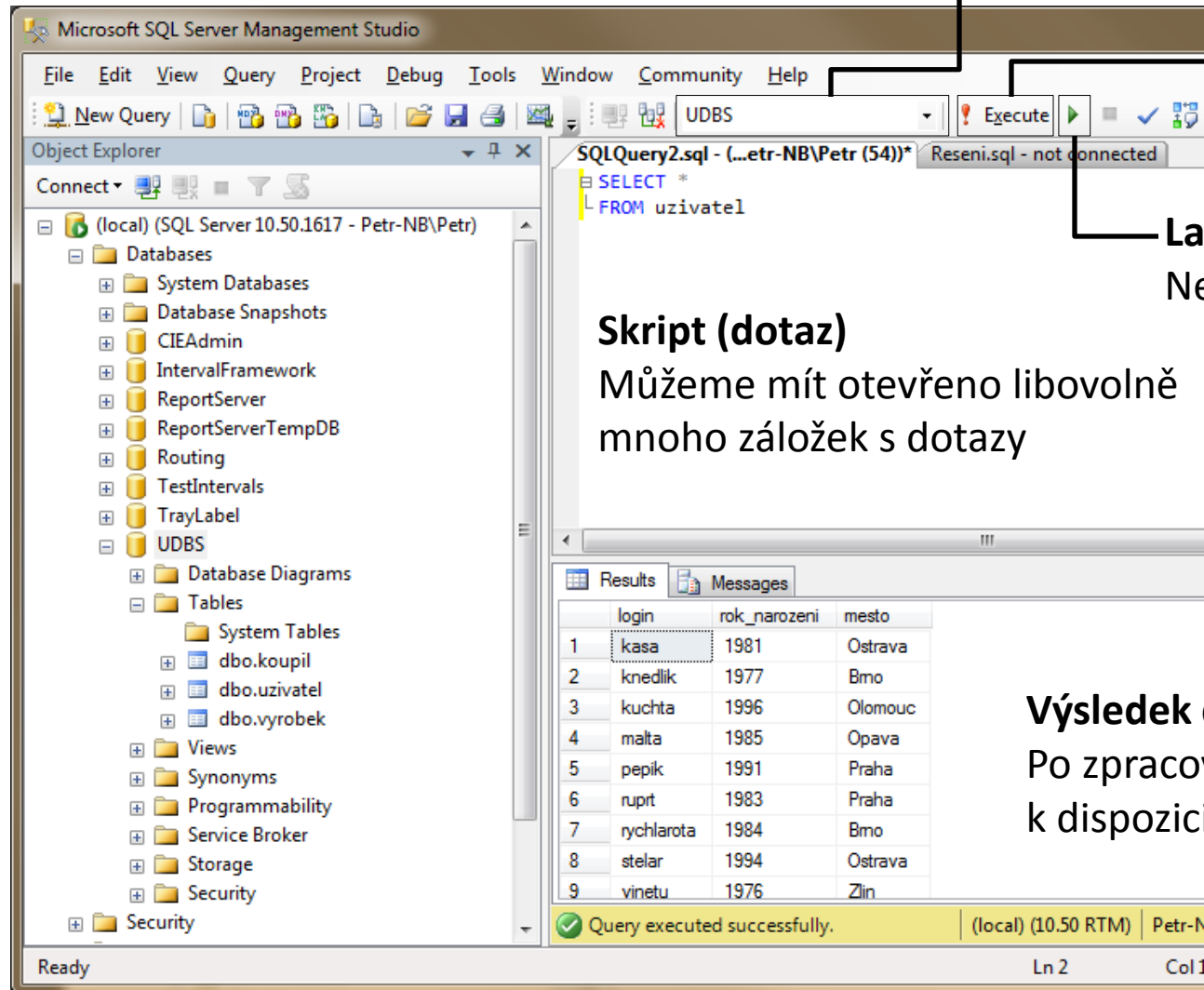
Aktuální databáze
Vždy kontrolujte, se kterou databází pracujete

Spuštění skriptu
Těž zkratka F5

Ladění skriptu
Nepoužívat!

Skript (dotaz)
Můžeme mít otevřeno libovolně mnoho záložek s dotazy

Výsledek dotazu
Po zpracování dotazu je výsledek k dispozici v tomto prostoru



The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows the server instance (local) (SQL Server 10.50.1617 - Petr-NB\Petr) with various databases and tables. The main window displays a query window titled 'SQLQuery2.sql - (...etr-NB\Petr (54))*' containing the following SQL query:

```
SELECT *  
FROM uzivatel
```

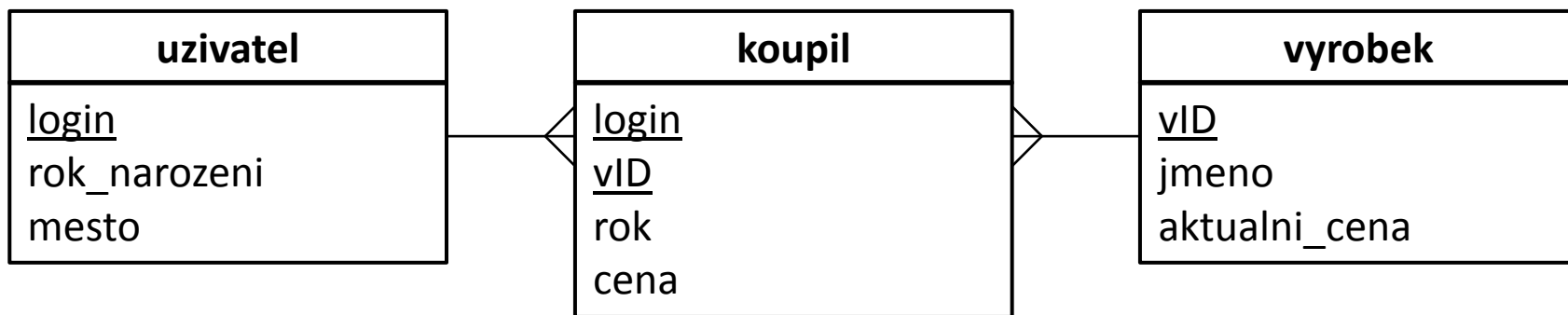
The Results pane shows the output of the query as a table with 9 rows and 3 columns: login, rok_narozeni, and mesto.

	login	rok_narozeni	mesto
1	kasa	1981	Ostrava
2	knedlik	1977	Bmo
3	kuchta	1996	Olomouc
4	malta	1985	Opava
5	pepik	1991	Praha
6	ruprt	1983	Praha
7	rychlarota	1984	Bmo
8	stelar	1994	Ostrava
9	vinetu	1976	Zlin

The status bar at the bottom indicates 'Query executed successfully.' and shows the current server instance as '(local) (10.50 RTM) | Petr-NB\Petr'.

Cvičení (1)

1. Z materiálů pro 3. cvičení na dbedu.cs.vsb.cz stáhněte skript Uživatel Koupil Výrobek.
2. Zkopírujte skrip do konzole SSMS a spusťte pomocí F5.
3. Skript nám připraví databázi se vzorovými daty, nad kterými budeme klást různé dotazy. Po provedení skriptu můžeme skript bez obav smazat.



Cvičení (2)

Než se pustíme do zadání, vyzkoušejme si základní dotazy:

1. Podívejte se postupně na data v jednotlivých tabulkách, tzn. zkuste napsat dotazy, které vám vyberou postupně všechna data z tabulek **uzivatel**, **koupil**, **vyrobek**.
2. Vyzkoušejte si filtrování záznamů pomocí klauzule **WHERE** – např. vyberte všechny výrobky s aktuální cenou 1500 Kč.
3. Vyzkoušejte si omezit výčet sloupců – z tabulky výrobků např. vyberte jen jejich názvy.
4. Vyzkoušejte si klauzuli **ORDER BY** pro třídění výsledku – vyberte např. všechny uživatele a setřídte je dle roku narození.

www.dbedu.cs.vsb.cz

- Přihlášení přes **jednotný login a heslo**
- Vpravo sloupec -> *České kurzy* -> *UDBS*