



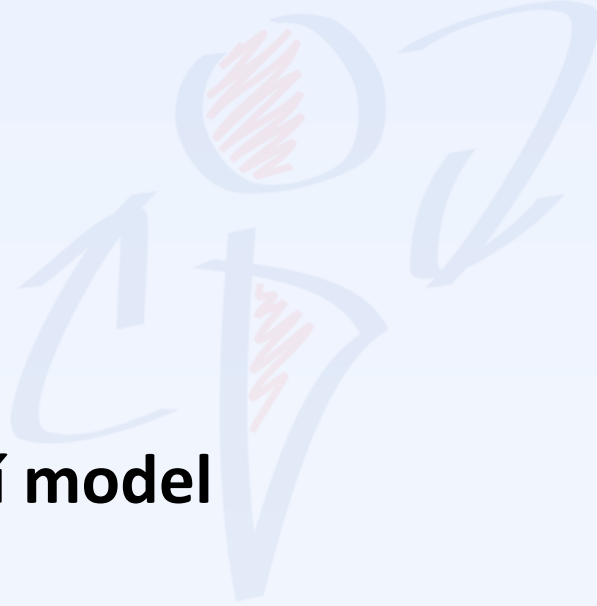
Vysoká škola báňská – Technická univerzita Ostrava  
Fakulta elektrotechniky a informatiky



# Úvod do databázových systémů

## Cvičení 3

Ing. Petr Lukáš  
[petr.lukas@vsb.cz](mailto:petr.lukas@vsb.cz)  
Ostrava, 2014

- **4 fáze vytváření databáze**
  - **Rozdíl mezi konceptuálním a relačním modelem**
  - **Primární klíč**
  - **Cizí klíč**
  - **Typy vztahů a jejich převod na relační model**
- 

- **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**



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



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



229 systems in ranking, October 2014

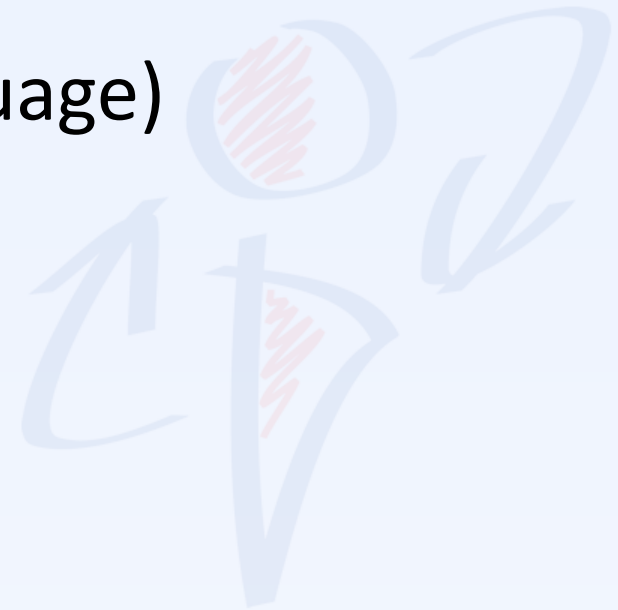
Rank	Last Month	DBMS	Database Model	Score	Changes
1.	1.	Oracle	Relational DBMS	1471.90	+4.99
2.	2.	MySQL	Relational DBMS	1262.97	-34.17
3.	3.	Microsoft SQL Server	Relational DBMS	1219.60	+10.73

<http://db-engines.com>





- **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*



- **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**

- **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**



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

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

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

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

**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*



## 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



### 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



### 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

**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

## 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



## 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

### Student

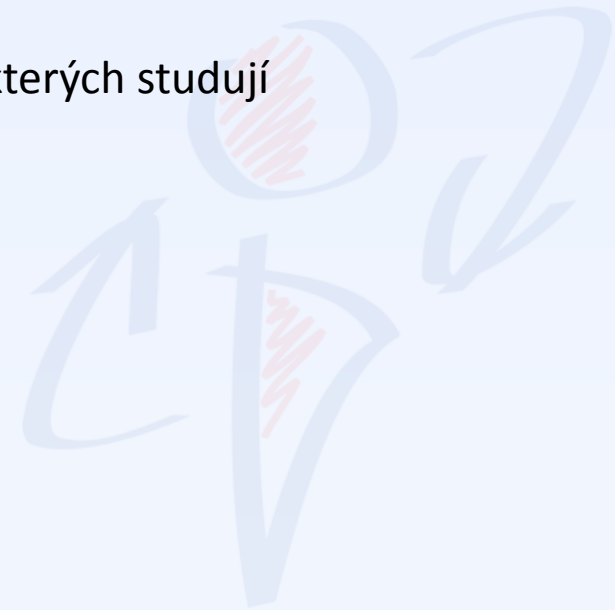
login	jmeno	skola
nov078	Jan	1
luk194	Petr	1
lak065	Luboš	2

### Skola

id	nazev
1	VŠB – TUO
2	Ostravská univerzita



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



### 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, Skola  
WHERE Student.skola = Skola.id
```



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

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

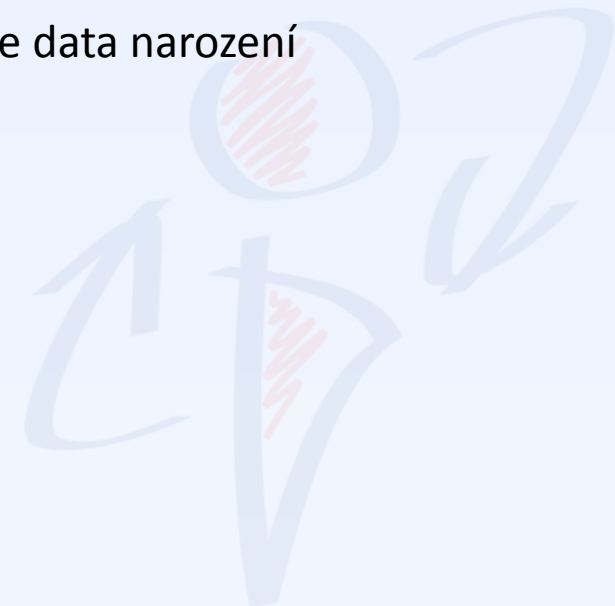


## Student

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



**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

- **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**.

- **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.

```
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

[www.dbedu.cs.vsb.cz](http://www.dbedu.cs.vsb.cz)

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

