

# Informační systémy ve zdravotnictví

## 7. cvičení

Ing. Petr Lukáš  
[petr.lukas@nativa.cz](mailto:petr.lukas@nativa.cz)

Ostrava, 2014

# Opakování

- **Klauzule příkazu SELECT a jejich význam**
- **Co znamenají zkratky DDL, DML, DQL?**
- **Účel DISTINCT**

# Opakování

- **Klauzule příkazu SELECT a jejich význam**  
SELECT, FROM, WHERE, GROUP BY, HAVING, ORDER BY
- **Co znamenají zkratky DDL, DML, DQL?**  
Data **D**efinition Language – nové tabulky, sloupce, atd.  
Data **M**anipulation Language – vkládání, úprava a mazání záznamů, Data **Q**uery Language – dotazování
- **Účel DISTINCT**  
Omezí duplicity na výstupu SELECT.

# Základní klauzule SELECT – vyhodnocování dotazu

```
SELECT jmeno, prijmeni, obor  
FROM Student, Rocnik  
WHERE Student.idRocnik = Rocnik.idRocnik AND datum_nar > '1.1.1990'  
ORDER BY datum_nar
```

## Student

login	jmeno	prijmeni	datum_nar	idRocnik
ros057	Jan	Rostlina	4.6.1987	1
hrn072	Václav	Hrnc	8.9.1991	2
bet128	Zdislava	Betonová	27.3.1990	2

## Rocnik

idRocnik	obor	rok
1	FEI	2013/2014
2	EKF	2013/2014

# Základní klauzule SELECT – vyhodnocování dotazu

**SELECT** jmeno, prijmeni, obor

**FROM** Student, Rocnik

**WHERE** Student.idRocnik = Rocnik.idRocnik **AND** datum\_nar > '1.1.1990'

**ORDER BY** datum\_nar

## Student

login	jmeno	prijmeni	datum_nar	idRocnik
ros057	Jan	Rostlina	4.6.1987	1
hrn072	Václav	Hrnc	8.9.1991	2
bet128	Zdislava	Betonová	27.3.1990	2

## Rocnik

idRocnik	obor	rok
1	FEI	2013/2014
2	EKF	2013/2014

# Základní klauzule SELECT – vyhodnocování dotazu

**SELECT** jmeno, prijmeni, obor

**FROM** Student, Rocnik

**WHERE** Student.idRocnik = Rocnik.idRocnik **AND** datum\_nar > '1.1.1990'

**ORDER BY** datum\_nar

login	jmeno	prijmeni	datum_nar	idRocnik <sub>S</sub>	idRocnik <sub>R</sub>	obor	rok
ros057	Jan	Rostlina	4.6.1987	1	1	FEI	2013/2014
hrn072	Václav	Hrnc	8.9.1991	2	1	FEI	2013/2014
bet128	Zdislava	Betonová	27.3.1990	2	1	FEI	2013/2014
ros057	Jan	Rostlina	4.6.1987	1	2	EKF	2013/2014
hrn072	Václav	Hrnc	8.9.1991	2	2	EKF	2013/2014
bet128	Zdislava	Betonová	27.3.1990	2	2	EKF	2013/2014

*(kartézský součin všech tabulek v klauzuli FROM)*

# Základní klauzule SELECT – vyhodnocování dotazu

**SELECT** jmeno, prijmeni, obor

**FROM** Student, Rocnik

**WHERE** Student.idRocnik = Rocnik.idRocnik **AND** datum\_nar > '1.1.1990'

**ORDER BY** datum\_nar

login	jmeno	prijmeni	datum_nar	idRocnik <sub>S</sub>	idRocnik <sub>R</sub>	obor	rok
ros057	Jan	Rostlina	4.6.1987	1	1	FEI	2013/2014
hrn072	Václav	Hrnec	8.9.1991	2	1	FEI	2013/2014
bet128	Zdislava	Betonová	27.3.1990	2	1	FEI	2013/2014
ros057	Jan	Rostlina	4.6.1987	1	2	EKF	2013/2014
hrn072	Václav	Hrnec	8.9.1991	2	2	EKF	2013/2014
bet128	Zdislava	Betonová	27.3.1990	2	2	EKF	2013/2014

# Základní klauzule SELECT – vyhodnocování dotazu

**SELECT** jmeno, prijmeni, obor

**FROM** Student, Rocnik

**WHERE** Student.idRocnik = Rocnik.idRocnik **AND** datum\_nar > '1.1.1990'

**ORDER BY** datum\_nar

login	jmeno	prijmeni	datum_nar	idRocnik <sub>S</sub>	idRocnik <sub>R</sub>	obor	rok
res057	Jan	Rostlina	4.6.1987	1	1	FEI	2013/2014
hrn072	Václav	Hrnc	8.9.1991	2	1	FEI	2013/2014
bet128	Zdislava	Betonová	27.3.1990	2	1	FEI	2013/2014
res057	Jan	Rostlina	4.6.1987	1	2	EKF	2013/2014
hrn072	Václav	Hrnc	8.9.1991	2	2	EKF	2013/2014
bet128	Zdislava	Betonová	27.3.1990	2	2	EKF	2013/2014

*(vybereme pouze řádky odpovídající podmínce)*



# Základní klauzule SELECT – vyhodnocování dotazu

```
SELECT jmeno, prijmeni, obor  
FROM Student, Rocnik  
WHERE Student.idRocnik = Rocnik.idRocnik AND datum_nar > '1.1.1990'  
ORDER BY datum_nar
```

login	jmeno	prijmeni	datum_nar	idRocnik <sub>S</sub>	idRocnik <sub>R</sub>	obor	rok
hrn072	Václav	Hrnec	8.9.1991	2	2	EKF	2013/2014
bet128	Zdislava	Betonová	27.3.1990	2	2	EKF	2013/2014

# Základní klauzule SELECT – vyhodnocování dotazu

```
SELECT jmeno, prijmeni, obor  
FROM Student, Rocnik  
WHERE Student.idRocnik = Rocnik.idRocnik AND datum_nar > '1.1.1990'  
ORDER BY datum_nar
```

login	jmeno	prijmeni	datum_nar	idRocnik <sub>S</sub>	idRocnik <sub>R</sub>	obor	rok
bet128	Zdislava	Betonová	27.3.1990	2	2	EKF	2013/2014
hrn072	Václav	Hrnec	8.9.1991	2	2	EKF	2013/2014

*(setřídíme řádky podle požadovaných sloupců)*

# Základní klauzule SELECT – vyhodnocování dotazu

```
SELECT jmeno, prijmeni, obor
```

```
FROM Student, Rocnik
```

```
WHERE Student.idRocnik = Rocnik.idRocnik AND datum_nar > '1.1.1990'
```

```
ORDER BY datum_nar
```

login	jmeno	prijmeni	datum_nar	idRocnik <sub>S</sub>	idRocnik <sub>R</sub>	obor	rok
bet128	Zdislava	Betonová	27.3.1990	2	2	EKF	2013/2014
hrn072	Václav	Hrnec	8.9.1991	2	2	EKF	2013/2014

*(setřídíme řádky podle požadovaných sloupců)*

# Základní klauzule SELECT – vyhodnocování dotazu

```
SELECT jmeno, prijmeni, obor
```

```
FROM Student, Rocnik
```

```
WHERE Student.idRocnik = Rocnik.idRocnik AND datum_nar > '1.1.1990'
```

```
ORDER BY datum_nar
```

jmeno	prijmeni	obor
Zdislava	Betonová	EKF
Václav	Hrnc	EKF

*(vybereme pouze požadované sloupce)*

# Základní klauzule SELECT – vyhodnocování dotazu

```
SELECT jmeno, prijmeni, obor  
FROM Student, Rocnik  
WHERE Student.idRocnik = Rocnik.idRocnik AND datum_nar > '1.1.1990'  
ORDER BY datum_nar
```

jmeno	prijmeni	obor
Zdislava	Betonová	EKF
Václav	Hrnc	EKF

- **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*.

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

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