

Vysoká škola báňská – Technická univerzita Ostrava

Fakulta elektrotechniky a informatiky

Katedra informatiky

Database Research Group



# Úvod do databázových systémů

## Cvičení 3

Ing. Petr Lukáš

[petr.lukas@vsb.cz](mailto:petr.lukas@vsb.cz)

EA440

Ostrava, 2016



# SQL

## Structured Query Language



- **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*



<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
ble424	Adam	Bledý	7.6.1988
ska065	Luboš	Škaredý	2.8.1991

*Vyber vše z tabulky Student*

Vyber všechny studenty, kteří se narodili po 1.1.1988

Vyber jména a příjmení všech studentů

Vyber všechny studenty setříděné sestupně dle data narození





## Student

login	jmeno	prijmeni	datum_narozeni
nov078	Jan	Novák	5.9.1987
ble424	Adam	Bledý	7.6.1988
ska065	Luboš	Škaredý	2.8.1991

*Vyber vše z tabulky Student*

```
SELECT *  
FROM Student
```

Vyber všechny studenty, kteří se narodili po 1.1.1988

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

Vyber jména a příjmení všech studentů

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

Vyber všechny studenty setříděné sestupně dle data narození

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

## 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



- **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

- **Kultura psaní kódu**

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





- **Další konstrukce SQL**
- **Poddotazy**



## Student

login	jmeno	prijmeni	datum_narozeni	typ_studia
nov078	Jan	Novák	5.9.1987	prezenční
luk194	Petr	Lukáš	7.6.1988	prezenční
lak065	Luboš	Lakatoš	2.8.1991	kombinovaný
m1a452	Lukáš	Mladý	1.10.1992	dálkový



*Vyber všechny prezenční a kombinované studenty*



## Student

login	jmeno	prijmeni	datum_narozeni	typ_studia
nov078	Jan	Novák	5.9.1987	prezenční
luk194	Petr	Lukáš	7.6.1988	prezenční
lak065	Luboš	Lakatoš	2.8.1991	kombinovaný
m1a452	Lukáš	Mladý	1.10.1992	dálkový



*Vyber všechny prezenční a kombinované studenty*

```
SELECT *  
FROM Student  
WHERE  
  typ_studia = 'prezenční' OR  
  typ_studia = 'kombinovaný'
```



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



## Student

login	jmeno	prijmeni	datum_narozeni	typ_studia
nov078	Jan	Novák	5.9.1987	prezenční
luk194	Petr	Lukáš	7.6.1988	prezenční
lak065	Luboš	Lakatoš	2.8.1991	kombinovaný
m1a452	Lukáš	Mladý	1.10.1992	dálkový



*Vyber všechny prezenční a kombinované studenty*

```
SELECT *  
FROM Student  
WHERE  
  typ_studia = 'prezenční' OR  
  typ_studia = 'kombinovaný'
```

=

```
SELECT *  
FROM Student  
WHERE  
  typ_studia IN ('prezenční',  
                'kombinovaný')
```



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

### Student

login	jmeno	prijmeni	datum_narozeni	typ_studia
nov078	Jan	Novák	5.9.1987	P
luk194	Petr	Lukáš	7.6.1988	P
lak065	Luboš	Lakatoš	2.8.1991	K
mmla452	Lukáš	Mladý	1.10.1992	D

### Typ\_Studia

zkratka	název	dochazka
P	Prezenční	1
K	Kombinované	1
D	Dálkové	0



*Vyber všechny studenty, kteří podle typu studia musí mít pravidelnou docházku*



### Student

login	jmeno	prijmeni	datum_narozeni	typ_studia
nov078	Jan	Novák	5.9.1987	P
luk194	Petr	Lukáš	7.6.1988	P
lak065	Luboš	Lakatoš	2.8.1991	K
m1a452	Lukáš	Mladý	1.10.1992	D

### Typ\_Studia

zkratka	název	dochazka
P	Prezenční	1
K	Kombinované	1
D	Dálkové	0

*Vyber všechny studenty, kteří podle typu studia musí mít pravidelnou docházku*

```
SELECT *  
FROM Student  
WHERE typ_studia IN  
(  
    SELECT zkratka FROM Typ_Studia WHERE dochazka = 1  
)
```

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

### Student

login	jmeno	prijmeni	datum_narozeni	typ_studia
nov078	Jan	Novák	5.9.1987	P
luk194	Petr	Lukáš	7.6.1988	P
lak065	Luboš	Lakatoš	2.8.1991	K
m1a452	Lukáš	Mladý	1.10.1992	D

### Typ\_Studia

zkratka	název	dochazka
P	Prezenční	1
K	Kombinované	1
D	Dálkové	0



Vnější dotaz

```
SELECT *  
FROM Student  
WHERE typ_studia IN  
(  
  SELECT zkratka FROM Typ_Studia WHERE dochazka = 1  
)
```

Vnitřní dotaz (poddotaz)



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



## 2. příklad – konstrukce IN s poddotazem

Student

login	jmeno	prijmeni	datum_narozeni	typ_studia
nov078	Jan	Novák	5.9.1987	P
luk194	Petr	Lukáš	7.6.1988	P
lak065	Luboš	Lakatoš	2.8.1991	K
m1a452	Lukáš	Mladý	1.10.1992	D

Typ\_Studia

zkratka	název	dochazka
P	Prezenční	1
K	Kombinované	1
D	Dálkové	0



```
SELECT *  
FROM Student  
WHERE typ_studia IN  
(  
  SELECT zkratka FROM Typ_Studia WHERE dochazka = 1  
)
```



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





### Student

login	jmeno	prijmeni	datum_narozeni	typ_studia
nov078	Jan	Novák	5.9.1987	P
luk194	Petr	Lukáš	7.6.1988	P
lak065	Luboš	Lakatoš	2.8.1991	K
mla452	Lukáš	Mladý	1.10.1992	D

### Typ\_Studia

zkratka	název	dochazka
P	Prezenční	1
K	Kombinované	1
D	Dálkové	0



```
SELECT *  
FROM Student  
WHERE typ_studia IN  
(  
  'P', 'K'  
)
```



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

## Student

login	jmeno	prijmeni	datum_narozeni	typ_studia
nov078	Jan	Novák	5.9.1987	prezenční
luk194	Petr	Lukáš	7.6.1988	prezenční
lak065	Luboš	Lakatoš	2.8.1991	kombinovaný
mla452	Lukáš	Mladý	1.10.1992	dálkový



*Vyberte všechny studenty, jejichž příjmení je zároveň jménem.*



## Student

login	jmeno	prijmeni	datum_narozeni	typ_studia
nov078	Jan	Novák	5.9.1987	prezenční
luk194	Petr	Lukáš	7.6.1988	prezenční
lak065	Luboš	Lakatoš	2.8.1991	kombinovaný
m1a452	Lukáš	Mladý	1.10.1992	dálkový



*Vyberte všechny studenty, jejichž příjmení je zároveň jménem.*

```
SELECT *  
FROM Student S1  
WHERE EXISTS  
(  
  SELECT * FROM Student S2 WHERE S2.jmeno = S1.prijmeni  
)
```



login	jmeno	prijmeni	datum_narozeni	typ_studia
luk194	Petr	Lukáš	7.6.1988	prezenční



## Student

login	jmeno	prijmeni	datum_narozeni	typ_studia
nov078	Jan	Novák	5.9.1987	prezenční
luk194	Petr	Lukáš	7.6.1988	prezenční
lak065	Luboš	Lakatoš	2.8.1991	kombinovaný
m1a452	Lukáš	Mladý	1.10.1992	dálkový



```
SELECT *
FROM Student S1
WHERE EXISTS
(
  SELECT *
  FROM Student S2
  WHERE S2.jmeno = S1.prijmeni
)
```



login	jmeno	prijmeni	datum_narozeni	typ_studia
luk194	Petr	Lukáš	7.6.1988	prezenční

- Nemůžeme si dovolit představu, že se nejprve provede vnitřní a potom vnější dotaz.



## Student

login	jmeno	prijmeni	datum_narozeni	typ_studia
nov078	Jan	Novák	5.9.1987	prezenční
luk194	Petr	Lukáš	7.6.1988	prezenční
lak065	Luboš	Lakatoš	2.8.1991	kombinovaný
m1a452	Lukáš	Mladý	1.10.1992	dálkový

```
SELECT *  
FROM Student S1  
WHERE EXISTS  
(  
  SELECT *  
  FROM Student S2  
  WHERE S2.jmeno = S1.prijmeni  
)
```



- Nemůžeme si dovolit představu, že se nejprve provede vnitřní a potom vnější dotaz.
- **Vnitřní dotaz je závislý na vnějším.**
- Je nutné si uvědomit, že podmínka WHERE se vyhodnocuje zvlášť pro každý vznikající záznam výstupu.



login	jmeno	prijmeni	datum_narozeni	typ_studia
luk194	Petr	Lukáš	7.6.1988	prezenční

## Student

login	jmeno	prijmeni		typ_studia
nov078	Jan	Novák		prezenční
luk194	Petr	Lukáš		prezenční
lak065	Luboš	Lakatoš		kombinovaný
m1a452	Lukáš	Mladý		dálkový



```
SELECT *
FROM Student S1
WHERE EXISTS
(
  SELECT *
  FROM Student S2
  WHERE S2.jmeno = S1.prijmeni
)
```



login	jmeno	prijmeni	datum_narozeni	typ_studia
luk194	Petr	Lukáš	7.6.1988	prezenční

Připrav prázdný seznam *result*;

```
for (s1 in Student)
{
  bool exists = false;
  for (s2 in Student)
  {
    if (s2.jmeno == s1.prijmeni)
    {
      exists = true;
    }
  }
  if (exists)
  {
    přidej s1 do result;
  }
}
```



## Student

	login	jmeno	prijmeni	datum_narozeni	typ_studia
<b>S1</b>	nov078	Jan	Novák	5.9.1987	prezenční
	luk194	Petr	Lukáš	7.6.1988	prezenční
	lak065	Luboš	Lakatoš	2.8.1991	kombinovaný
	mla452	Lukáš	Mladý	1.10.1992	dálkový

```
SELECT *  
FROM Student S1  
WHERE EXISTS  
(  
  SELECT *  
  FROM Student S2  
  WHERE S2.jmeno = 'Novák'  
  )  
  S1.prijmeni
```



Pro záznam o Janu Novákovi, existuje nějaký student s křestním jménem Novák?

**NEEXISTUJE**

Jan Novák tedy nebude na výstupu.



login	jmeno	prijmeni	datum_narozeni	typ_studia
-------	-------	----------	----------------	------------



### Student

	login	jmeno	prijmeni	datum_narozeni	typ_studia
	nov078	Jan	Novák	5.9.1987	prezenční
<b>S1</b>	luk194	Petr	Lukáš	7.6.1988	prezenční
	lak065	Luboš	Lakatoš	2.8.1991	kombinovaný
<b>S2</b>	mla452	Lukáš	Mladý	1.10.1992	dálkový

```
SELECT *  
FROM Student S1  
WHERE EXISTS  
(  
  SELECT *  
  FROM Student S2  
  WHERE S2.jmeno = 'Lukáš'  
  )  
  S1.prijmeni
```



Pro záznam o Petru Lukášovi, existuje nějaký student s křestním jménem Lukáš?

**ANO, EXISTUJE**

Petr Lukáš tedy bude na výstupu.



login	jmeno	prijmeni	datum_narozeni	typ_studia
luk194	Petr	Lukáš	7.6.1988	prezenční





## Student

login	jmeno	prijmeni	datum_narozeni	typ_studia
nov078	Jan	Novák	5.9.1987	prezenční
luk194	Petr	Lukáš	7.6.1988	prezenční
lak065	Luboš	Lakatoš	2.8.1991	kombinovaný
m1a452	Lukáš	Mladý	1.10.1992	dálkový

S1

```

SELECT *
FROM Student S1
WHERE EXISTS
(
  SELECT *
  FROM Student S2
  WHERE S2.jmeno = 'Lakatoš'
)
                S1.prijmeni
    
```



Pro záznam o Luboši Lakatošovi, existuje nějaký student s křestním jménem Lakatoš?

**NEEXISTUJE**

Luboš Lakatoš tedy nebude na výstupu.



login	jmeno	prijmeni	datum_narozeni	typ_studia
luk194	Petr	Lukáš	7.6.1988	prezenční



## Student

login	jmeno	prijmeni	datum_narozeni	typ_studia
nov078	Jan	Novák	5.9.1987	prezenční
luk194	Petr	Lukáš	7.6.1988	prezenční
lak065	Luboš	Lakatoš	2.8.1991	kombinovaný
m1a452	Lukáš	Mladý	1.10.1992	dálkový

**S1**

```
SELECT *
FROM Student S1
WHERE EXISTS
(
  SELECT *
  FROM Student S2
  WHERE S2.jmeno = 'Mladý'
  )
  S1.prijmeni
```



Pro záznam o Lukáši Mladém, existuje nějaký student se jménem Mladý?

**NEEXISTUJE**

Lukáš Mladý tedy nebude na výstupu.



login	jmeno	prijmeni	datum_narozeni	typ_studia
luk194	Petr	Lukáš	7.6.1988	prezenční



## Student

login	jmeno	prijmeni	datum_narozeni
nov078	Jan	Novák	5.9.1987
luk194	Petr	Lukáš	7.6.1988
lak065	Luboš	Lakatoš	2.8.1991
m1a452	Lukáš	Mladý	1.10.1992

## Ucitel

login	jmeno	prijmeni	datum_narozeni
nov078	Ivan	Hrozivý	1.2.1965
luk194	Karel	Přísný	5.12.1972
lak065	Zdeněk	Veliký	25.6.1990
m1a452	Václav	Důsledný	1.10.1985



*Vyber všechny učitele, kteří jsou starší než všichni studenti*



## Student

login	jmeno	prijmeni	datum_narozeni
nov078	Jan	Novák	5.9.1987
luk194	Petr	Lukáš	7.6.1988
lak065	Luboš	Lakatoš	2.8.1991
m1a452	Lukáš	Mladý	1.10.1992

## Ucitel

login	jmeno	prijmeni	datum_narozeni
nov078	Ivan	Hrozivý	1.2.1965
luk194	Karel	Přísný	5.12.1972
lak065	Zdeněk	Veliký	25.6.1990
m1a452	Václav	Důsledný	1.10.1985



*Vyber všechny učitele, kteří jsou starší než všichni studenti*

```
SELECT *  
FROM Ucitel  
WHERE datum_narozeni < ALL(SELECT datum_narozeni FROM Student)
```



login	jmeno	Prijmeni	datum_narozeni
nov078	Ivan	Hrozivý	1.2.1965
luk194	Karel	Přísný	5.12.1972
m1a452	Václav	Důsledný	1.10.1985



**Student**

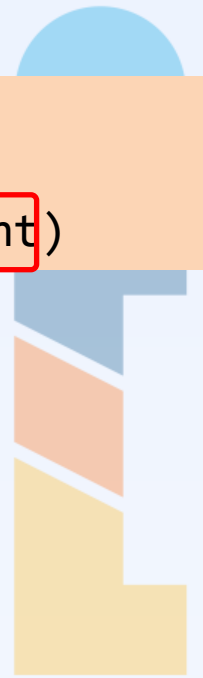
login	jmeno	prijmeni	datum_narozeni
nov078	Jan	Novák	5.9.1987
luk194	Petr	Lukáš	7.6.1988
lak065	Luboš	Lakatoš	2.8.1991
m1a452	Lukáš	Mladý	1.10.1992

**Ucitel**

login	jmeno	prijmeni	datum_narozeni
nov078	Ivan	Hrozivý	1.2.1965
luk194	Karel	Přísný	5.12.1972
lak065	Zdeněk	Veliký	25.6.1990
m1a452	Václav	Důsledný	1.10.1985

```
SELECT *
FROM Ucitel
WHERE datum_narozeni < ALL(SELECT datum_narozeni FROM Student)
```

login	jmeno	Prijmeni	datum_narozeni
nov078	Ivan	Hrozivý	1.2.1965
luk194	Karel	Přísný	5.12.1972
m1a452	Václav	Důsledný	1.10.1985



### Student

login	jmeno	prijmeni	datum_narozeni
nov078	Jan	Novák	5.9.1987
luk194	Petr	Lukáš	7.6.1988
lak065	Luboš	Lakatoš	2.8.1991
m1a452	Lukáš	Mladý	1.10.1992

### Ucitel

login	jmeno	prijmeni	datum_narozeni
nov078	Ivan	Hrozivý	1.2.1965
luk194	Karel	Přísný	5.12.1972
lak065	Zdeněk	Veliký	25.6.1990
m1a452	Václav	Důsledný	1.10.1985

```
SELECT *
FROM Ucitel
WHERE datum_narozeni < ALL(5.9.1987, 7.6.1988, 2.8.1991, 1.10.1992)
```

login	jmeno	Prijmeni	datum_narozeni
nov078	Ivan	Hrozivý	1.2.1965
luk194	Karel	Přísný	5.12.1972
m1a452	Václav	Důsledný	1.10.1985



# 5. Příklad – konstrukce ANY (SOME)

**Student**

login	jmeno	prijmeni	datum_narozeni
nov078	Jan	Novák	5.9.1987
luk194	Petr	Lukáš	7.6.1988
lak065	Luboš	Lakatoš	2.8.1991
m1a452	Lukáš	Mladý	1.10.1992

**Ucitel**

login	jmeno	prijmeni	datum_narozeni
nov078	Ivan	Hrozivý	1.2.1965
luk194	Karel	Přísný	5.12.1972
lak065	Zdeněk	Veliký	25.6.1990
m1a452	Václav	Důsledný	1.10.1985

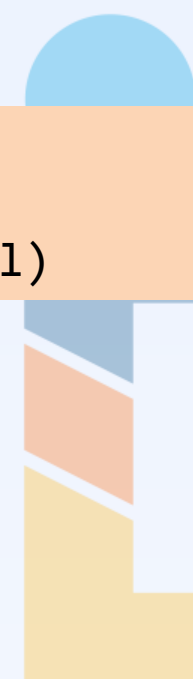


*Vyber studenty, kteří jsou starší než nějaký učitel*

```
SELECT *  
FROM Student  
WHERE datum_narozeni < ANY(SELECT datum_narozeni FROM Ucitel)
```



login	jmeno	prijmeni	datum_narozeni
nov078	Jan	Novák	5.9.1987
luk194	Petr	Lukáš	7.6.1988



# 5. Příklad – konstrukce ANY (SOME)

**Student**

login	jmeno	prijmeni	datum_narozeni
nov078	Jan	Novák	5.9.1987
luk194	Petr	Lukáš	7.6.1988
lak065	Luboš	Lakatoš	2.8.1991
m1a452	Lukáš	Mladý	1.10.1992

**Ucitel**

login	jmeno	prijmeni	datum_narozeni
nov078	Ivan	Hrozivý	1.2.1965
luk194	Karel	Přísný	5.12.1972
lak065	Zdeněk	Veliký	25.6.1990
m1a452	Václav	Důsledný	1.10.1985

*Vyber studenty, kteří jsou starší než nějaký učitel*

```
SELECT *  
FROM Student  
WHERE datum_narozeni < ANY(SELECT datum_narozeni FROM Ucitel)
```

login	jmeno	prijmeni	datum_narozeni
nov078	Jan	Novák	5.9.1987
luk194	Petr	Lukáš	7.6.1988



# 5. Příklad – konstrukce ANY (SOME)

**Student**

login	jmeno	prijmeni	datum_narozeni
nov078	Jan	Novák	5.9.1987
luk194	Petr	Lukáš	7.6.1988
lak065	Luboš	Lakatoš	2.8.1991
m1a452	Lukáš	Mladý	1.10.1992

**Ucitel**

login	jmeno	prijmeni	datum_narozeni
nov078	Ivan	Hrozivý	1.2.1965
luk194	Karel	Přísný	5.12.1972
lak065	Zdeněk	Veliký	25.6.1990
m1a452	Václav	Důsledný	1.10.1985

*Vyber studenty, kteří jsou starší než nějaký učitel*

```
SELECT *  
FROM Student  
WHERE datum_narozeni < ANY(1.2.1965, 5.12.1972, 25.6.1990, 1.10.1985)
```

login	jmeno	prijmeni	datum_narozeni
nov078	Jan	Novák	5.9.1987
luk194	Petr	Lukáš	7.6.1988

- **IN** Ptám se, zda hodnota atributu spadá do nějakého výčtu.
- **EXISTS** Test na existenci záznamu
- **ALL** Porovnání s výčtem hodnot, porovnání musí platit pro všechny hodnoty ve výčtu
- **ANY** Porovnání s výčtem hodnot, porovnání musí platit alespoň jednou



- Uvedené konstrukce se velmi často kombinují s použitím **poddotazu**.
- Pokud je vnitřní poddotaz **nezávislý** na vnějším dotazu (tj. nenachází se tam žádná podmínka nebo výraz, který by se na vnější dotaz odkazoval), můžeme si představit vyhodnocení celého dotazu tak, že se **nejprve vyhodnotí vnitřek, potom vnějšek**.
- Pokud je vnitřní poddotaz **závislý**, musíme si vyhodnocení představit tak, že se **vnitřní dotaz provede pro každý** např. **testovaný záznam z dotazu vnějšího**.



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

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

