

# Databázové systémy a SQL

## Lekce 3

Daniel Klimeš

# Operátory za WHERE

=	Rovná se
<>	Nerovná se
IS NULL/ IS NOT NULL	Testování prázdné/neprázdné hodnoty
[NOT] IN (hodnota, hodnota, ...)	Rovnost [NEROVNOST] se skupinou hodnot
LIKE	Podobný řetězec

SELECT \* FROM tabulka WHERE sloupec IN (1,5,7)

SELECT \* FROM tabulka WHERE sloupec NOT IN ('a', 'd', 'j')

SELECT \* FROM tabulka WHERE sloupec LIKE ('Jan%')

% = žádný nebo libovolné znaky

WHERE jmeno = 'Jan' AND prijmeni = 'Novák'

AND, OR, NOT

AND	TRUE	FALSE	NULL
TRUE	TRUE	FALSE	NULL
FALSE	FALSE	FALSE	FALSE

OR	TRUE	FALSE	NULL
TRUE	TRUE	TRUE	TRUE
FALSE	TRUE	FALSE	NULL

NOT TRUE = FALSE  
NOT FALSE = TRUE  
NOT NULL = NULL

**AND se vyhodnocuje před OR !**

X AND FALSE => FALSE

X OR TRUE => TRUE

FALSE AND FALSE OR TRUE => TRUE

FALSE AND (FALSE OR TRUE) => FALSE

5 > 1 AND NULL IS NOT NULL OR 1 = 1 => TRUE

DELETE FROM student WHERE

Prijmeni = 'Novák' AND jmeno = 'Jiří' OR jmeno = 'Jan'

Prijmeni = 'Novák' AND (jmeno = 'Jiří' OR jmeno = 'Jan')

- CASE WHEN podmínka THEN vysledek  
    WHEN podmínka2 THEN vysledek 2  
    ELSE vysledek 3 END
- ELSE nepovinné,
- Vyhodnocování **končí na první splněné podmínce**
- Všechny výsledky musí být stejného datového typu

### Příklad:

```
SELECT vek,  
      CASE WHEN vek IS NULL THEN 'neznamo'  
          WHEN vek < 20 THEN 'kat < 20'  
          WHEN vek < 25 THEN 'kat 20-24'  
          WHEN vek < 30 THEN 'kat 25-29'  
          ELSE 'kat 30 a starsi' END kategorie  
FROM  
student
```



## Agregace

---

# GROUP BY, HAVING

Seskupení položek

```
SELECT sloupec, COUNT(*), MAX(sloupec2), MIN(sloupec2) FROM tabulka  
GROUP BY sloupec;
```

```
SELECT sloupec, COUNT(*), MAX(sloupec2), MIN(sloupec2) FROM tabulka  
WHERE sloupec2 > 1 and ...  
GROUP BY sloupec;
```

```
SELECT sloupec, COUNT(*), MAX(sloupec2), MIN(sloupec2) FROM tabulka  
GROUP BY sloupec  
HAVING count(*) > 1
```

Funkce	Popis	Pozn.
COUNT(*)	Počet	
AVG(sloupec)	Aritmetický průměr	
MIN(sloupec)	Minimum	
MAX(sloupec)	Maximum	
STDDEV(sloupec)	Směrodatná odchylka	
SUM(sloupec)	Suma	
MEDIAN(sloupec)	Medián	ORACLE



```
SELECT  COUNT(*), --všechny řádky
        COUNT(sloupec), -- všechny NOT NULL řádky
        COUNT(DISTINCT sloupec) -- počet unikátních hodnot
```

FROM tabulka;

```
SELECT COUNT(*), COUNT(jmeno), COUNT(DISTINCT jmeno)
FROM student
```

SELECT DISTINCT sloupec1 FROM tabulka; -- unikátní hodnoty sloupce  
 SELECT DISTINCT sloupec1, sloupec2 FROM tabulka; -- unikátní kombinace sloupců

SELECT DISTINCT jmeno FROM student

SELECT jmeno FROM student  
 GROUP BY jmeno

## Zjistěte

- Počet jednotlivých křestních jmen v tabulce student
- Průměrný věk studenta, součet věků
- Počet studentů a průměrný věk studenta podle sloupce stupen
  - pouze skupiny, které mají víc jak 3 studenty
    - pouze ženy

## WHERE x HAVING

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



---

# PRÁCE S VÍCE TABULKAMI

# Práce s více tabulkami

ID pacienta	Jmeno	Prijmeni
1	Jan	Novák
2	Jana	Nová
3	Karel	Starý

+

ID pacienta	Datum vysetreni	Vysledek vysetreni
1	12.1.2011	39,5
1	15.3.2011	36,8
2	2.2.2011	37,5

ENTITY = tabulky

RELATIONSHIP = vazba

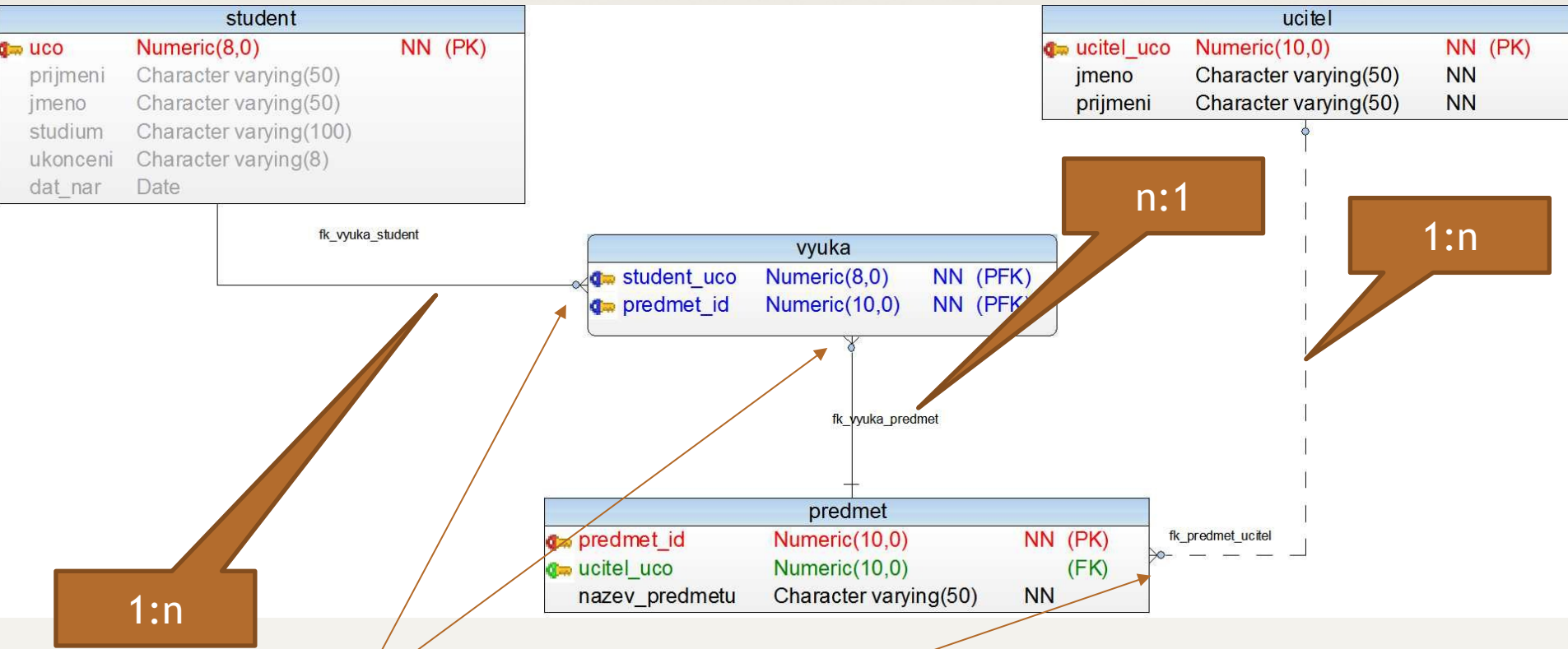
E-R diagramy = datové modely

1:1 – jeden řádek tabulky A má vazbu s jedním řádkem tabulky B

**1:n – k jednomu řádku tabulky A se váže 0 až N řádků tabulky B**

m:n – k jednomu řádku tabulky A se váže 0 až N řádků tabulky B  
**ale zároveň** k jednomu řádku z B se váže 0 až N řádků A

# ER diagram



1:n

n:1

1:n

Vidlička = dětská závislá tabulka



- Definice entit (tabulek)
- Stanovení primárních klíčů všech tabulek
- Tvorba vazeb
  - **Migrace primárního klíče rodičovské tabulky do dětské tabulky**
  - **Cizí klíč může, ale nemusí být součástí primárního klíče dětské tabulky**

Spustte skript2.sql

- Vytvořte si vlastní předmět (řádek v tabulce předmět)
- Zkuste vytvořit předmět s neexistujícím UCO\_ucitele
- Přihlaste se do zvolených předmětů
- Odhlašte se ze všech předmětů
- Přihlaste se do **všech** dostupných předmětů
- Zkuste smazat všechny učitele