

Databázové systémy v biomedicíně

Lekce III

Cvičení 7 / task 7

- Sloupec příjmení převed'te na velká písmena / convert lastname to uppercase
 - Smažte řádky s lichým/sudým ročníkem / delete row
 - Odháčkujte sloupec jmeno
 - Zkrat'te jméno na 1. znak a tečku / extract first letter from firstname and add dot
-
- Kolikátého bude za 7 let 7 měsíců a 7 dní /
what date will be after 7 years 7 months and 7 days
 - Ověřte, zda ženy mají v příjmení „ová“ / compare the end of lastname with sex
-
- Vymažte řádky ze **své** tabulky / delete all rows from your table
 - Zrušte **svoji** tabulku / remove your table

Operators for WHERE

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

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 / none or any character(s)

_ = právě jeden znak / any character, just one

Logical operators

WHERE firstname= 'Jan' AND lastname= '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 !

<https://www.postgresql.org/docs/15/static/functions-logical.html>

Logical operators

X AND FALSE => FALSE

X OR TRUE => TRUE

FALSE AND FALSE OR TRUE => TRUE

FALSE AND (FALSE OR TRUE) => FALSE

5 > 1 AND 1+1=3 OR 1 = 1 => TRUE

DELETE FROM student WHERE

lastname = 'Novák' AND firstname= 'Jiří' OR firstname= 'Jan'

x

lastname= 'Novák' AND (firstname= 'Jiří' OR firstname = 'Jan')

Conditional expression CASE

- 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
```

<https://www.postgresql.org/docs/15/static/functions-conditional.html>

GROUP BY, HAVING

Agregace

GROUP BY

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
```

<https://www.postgresql.org/docs/15/static/tutorial-aggr.html>

Agregační funkce

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

COUNT

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

```
FROM tabulka;
```

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

MODIFIKÁTOR DISTINCT / DISTINCT Clause

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

```
SELECT DISTINCT lastname FROM student
```

```
--srovnej
```

```
SELECT lastname FROM student  
GROUP BY lastname
```

```
SELECT DISTINCT sex, lastname FROM student
```

Task - aggregation

Zjistěte / compute from table student

- Počet jednotlivých křestních jmen v tabulce student

List of unique firstnames and number of students

- Průměrný věk studenta, součet věků

Average age of student, sum of age for all students

- Počet studentů a průměrný věk studenta podle sloupce stupen

Number of students and average age group by study

- Ponechte pouze skupiny, které mají víc jak 3 studenty

Result filter for groups with minimum 3 students

SELECT

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

Import dat

Import/export dat z/do textového souboru/file

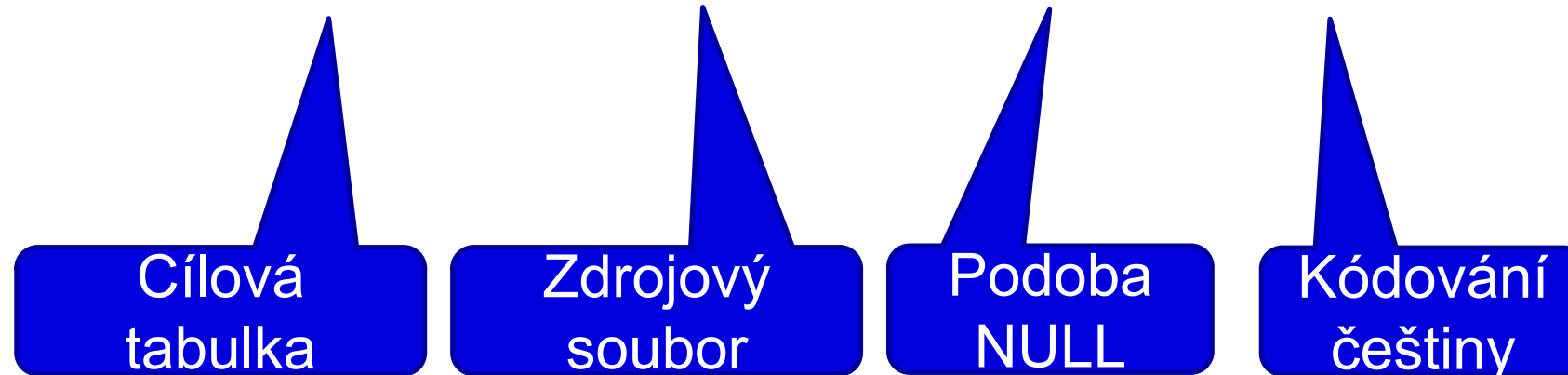
- Příkaz/command COPY FROM/TO
 - Ve Windows nastavit oprávnění na složku pro NETWORK_SERVICE
Set permission for source folder on disk for system user NETWORK_SERVICE
- Export dat
- COPY student TO 'c:\aa\student.txt' --export all data from table to a file
- COPY (SELECT uco, firstname FROM student) TO 'c:\aa\student_jmena.txt'
export result of SQL to a file

- Import dat
- COPY patients FROM 'c:/Users/student/Documents/data/patients.txt' NULL "
ENCODING 'UTF8';
- Před importem musí tabulka existovat
Table must exists before import

<https://www.postgresql.org/docs/current/static/sql-copy.html>

Import dat z textových souborů

```
COPY patients FROM 'Z:/DBM/patients.txt' NULL " ENCODING 'UTF8';
```



Další parametry příkazu COPY

FORMAT

Selects the data format to be read or written:

text, csv (Comma Separated Values), or binary. The default is text.

DELIMITER

Specifies the character that separates columns within each row (line) of the file.

The default is a tab character in text format, a comma in CSV format.

This must be a single one-byte character

Řádkový klient PSQL

- Spuštění z příkazové řádky
- V učebně:
 - `psql -h 147.251.145.6 -U studentucebna -d ucebnaarcx`
- Na vlastním počítači:
 - `psql -h localhost -U postgres`

- Příkaz `\copy`
- Platí stejné parametry jako v případě COPY příkazu
- Nevyžaduje oprávnění superuser

Import - task

Import data from the file ukol.csv

1. Read header of the file
2. Create table
3. Import data

First row:

id;datnar;datdg;datumrti;rc;lecbaporadi;lecbaod;lecbado;druhlecby;zaver;leu

Import raw data

```
CREATE TABLE ukol  
(  
id text,  
datnar text,  
datdg text,  
datumrti text,  
rc text,  
lecbaporadi text,  
lecbaad text,  
lecbado text,  
druhlecby text,  
zaver text,  
leu text  
);
```

Import "clean" data without header

```
CREATE TABLE ukol  
(  
id text,  
datnar date,  
datdg date,  
datumrti date,  
rc text,  
lecbaporadi text,  
lecbaad date,  
lecbado date,  
druhlecby text,  
zaver text,  
leu text  
);
```

```
COPY ukol FROM 'c:/aa/ukol.csv ' DELIMITER ';' NULL '' ENCODING 'UTF8';
```