MASARYKOVA UNIVERZITA
INSTITUT BIOSTATISTIKY A ANALÝZ

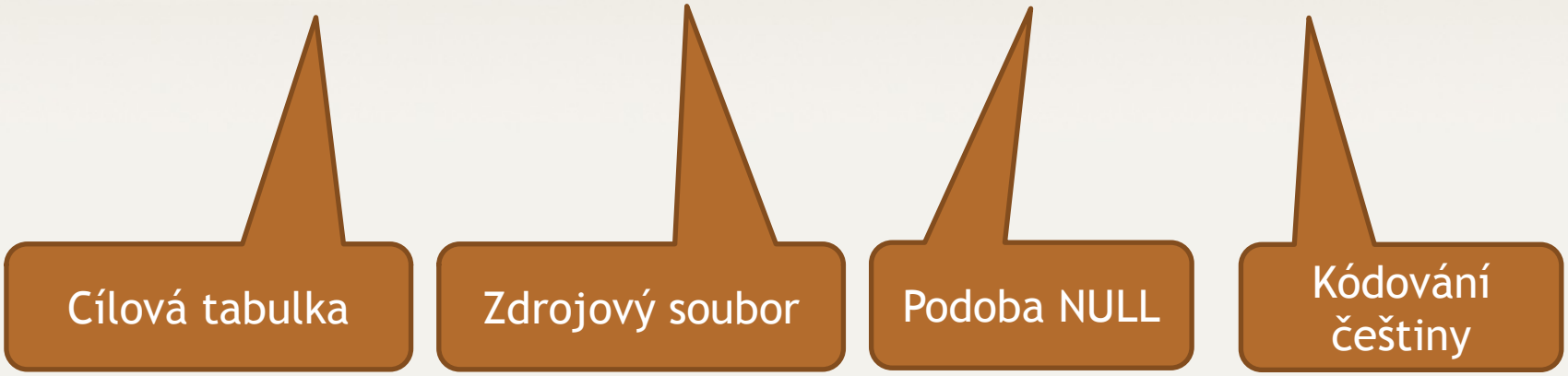
Databázové systémy a SQL

Lekce 5

Daniel Klimeš

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:

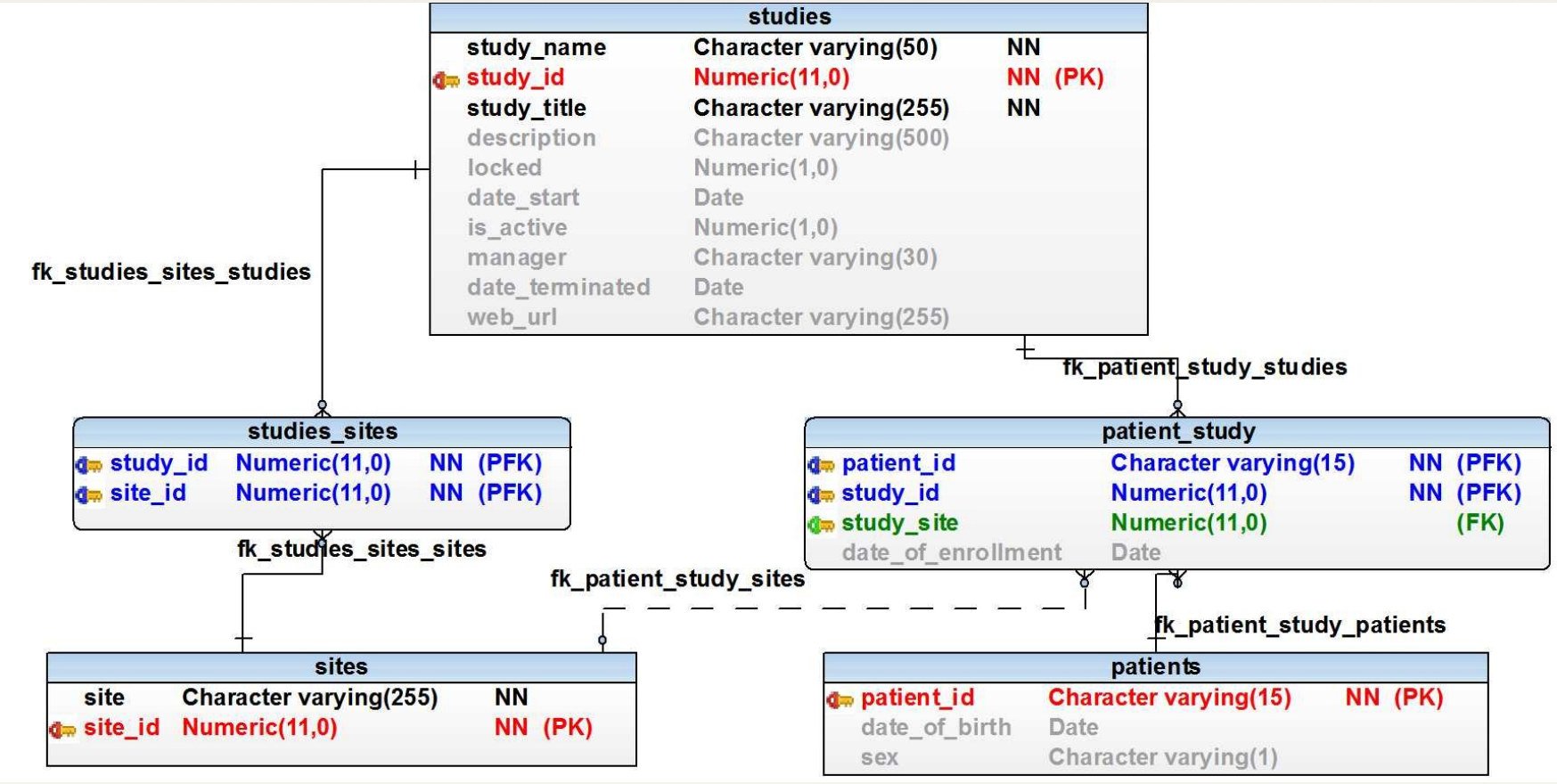
text, csv (Comma Separated Values), or binary. The default is text.

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

TRIALDB - datový model

Vazba pacienti – studie m-n => „mezitabulka“ PATIENT_STUDY
Vazba studie – pracoviště m-n => „mezitabulka“ STUDIES_SITES



Cvičení

Spusťte skript3.sql

Zjistěte počet pacientů v jednotlivých studiích
STUDY_NAME, počet pacientů

Zjistěte počet pacientů dle pohlaví v jednotlivých studiích
STUDY_NAME, pohlaví, počet pacientů

Zjistěte počet zapojených pracovišť do jednotlivých studií
STUDY_NAME, počet pracovišť

Vypište pracoviště zapojená do více studií
SITE, počet studií

Vypište všechny studie a počet zařazených pacientů v jednotlivých letech
STUDY_NAME, rok(DATE_OF_ENROLLMENT), počet pacientů

Poddotazy SQL

Zanořené dotazy

- uzavřené v kulatých závorkách **()**
- poddotazem je myšlen příkaz SELECT

SELECT sloupec
FROM tabulka
WHERE podmínka

GROUP BY
HAVING
ORDER BY

- místo názvu sloupce
- místo názvu tabulky
- v sekci WHERE

Subdotazy SQL - místo sloupce

Vnořený dotaz na pozici sloupce musí vrátit právě jeden řádek a právě jeden sloupec!

```
SELECT COUNT(student_uco),
        (SELECT COUNT (*) FROM student)
FROM vyuka;

SELECT COUNT(patient_id),
        (SELECT COUNT  (*) FROM  patients)
FROM patient_study;
```

Subdotazy SQL - místo sloupce

CVIČENÍ:

Napište dotaz, který vrátí seznam všech studentů,
počet jejich registrovaných předmětů
a kolik je to procent ze všech dostupných předmětů.

Subdotazy SQL - místo sloupce

CVIČENÍ:

Napište dotaz, který vrátí seznam všech studentů,
počet jejich registrovaných předmětů
a kolik je to procent ze všech dostupných předmětů

```
SELECT s.uco,  COUNT(v.predmet_id),
            ROUND(100.0 * (COUNT(v.predmet_id)) /
            (SELECT COUNT(*) FROM predmet) )
FROM student s JOIN vyuka v
ON s.uco=v.student_uco
GROUP BY s.uco;
```

Zanořený dotaz - místo názvu tabulky

Poddotaz na pozici FROM nahrazuje tabulku.
V postgreSQL musí být poddotaz na pozici tabulky **VŽDY** pojmenován!

```
SELECT  COUNT(*)  FROM (
   SELECT study_id, COUNT(patient_id)
     FROM patient_study GROUP BY study_id
   ) sub
```

Zanořený dotaz – místo názvu tabulky

CVIČENÍ:

Napište dotaz, který vrátí seznam studentů, kteří jsou registrováni do více než jednoho předmětu.

Zanořený dotaz - místo názvu tabulky

CVIČENÍ:

Varianta 1

```
SELECT * FROM (
        SELECT s.jmeno, s.prijmeni, s.uco, COUNT(v.predmet_id) pocet
        FROM student s JOIN vyuka v ON s.uco=v.student_uco
        GROUP BY s.jmeno, s.prijmeni, s.uco) sub
WHERE pocet>1;
```

Varianta 2

```
SELECT s.jmeno, s.prijmeni, s.uco, COUNT(v.predmet_id)
        FROM student s JOIN vyuka v ON s.uco=v.student_uco
        GROUP BY s.jmeno, s.prijmeni, s.uco
        HAVING COUNT(v.predmet_id)>1
        ORDER BY s.jmeno;
```

Vnořený dotaz za WHERE

Varianty:

- WHERE sloupec = (SELECT sloupec FROM…
  zanořený dotaz musí vrátit **právě 1 řádek a 1 sloupec**

- WHERE sloupec = **ANY** (SELECT sloupec FROM…
- WHERE sloupec **IN** (SELECT sloupec FROM …
- WHERE sloupec > **ALL** (SELECT sloupec FROM …
  zanořený dotaz musí vrátit **1 sloupec a libovolný počet řádků**

- WHERE **EXISTS** (SELECT * FROM….
- WHERE **NOT EXISTS** (SELECT * FROM…
  zanořený dotaz může vracet **libovolný počet řádků i sloupců**

Zanořené dotazy se obvykle propojují s nadřazeným dotazem pomocí podmínky v sekci WHERE

Vnořený dotaz za WHERE sloupec = (SELECT …

Varianty:

- WHERE sloupec = (SELECT sloupec FROM…
  zanořený dotaz musí vrátit právě 1 řádek a 1 sloupec

SELECT * FROM patients

WHERE date_of_birth = (SELECT MAX(date_of_birth) FROM patients);

Vnořený dotaz za WHERE sloupec ANY/IN/ALL

Varianty:

- WHERE sloupec = **ANY** (SELECT sloupec FROM…
- WHERE sloupec **IN** (SELECT sloupec FROM …
- WHERE sloupec > **ALL** (SELECT sloupec FROM …

zanořený dotaz musí vrátit 1 sloupec a libovolný počet řádků

```
SELECT * FROM student
WHERE uco = ANY (SELECT student_uco FROM vyuka WHERE
                                    predmet_id=10);
SELECT * FROM student
WHERE uco IN    (SELECT student_uco FROM vyuka WHERE
                                    predmet_id=10);
```

Vnořený dotaz za WHERE EXISTS/NOT EXISTS

Varianty:

- WHERE **EXISTS** (SELECT * FROM….
- WHERE **NOT EXISTS** (SELECT * FROM…

  zanořený dotaz může vracet libovolný počet řádků i sloupců

```
SELECT * FROM student s
WHERE EXISTS (SELECT * FROM vyuka v
                    WHERE predmet_id=10
                    AND s.uco=v.student_uco);
```

Vnořený dotaz za WHERE

Nejmladší student:

```
SELECT * FROM student WHERE  datum_narozeni  = (
     SELECT MAX(datum_narozeni) FROM student);

SELECT * FROM student WHERE
               datum_narozeni >= ALL (
     SELECT  datum_narozeni  FROM student);

SELECT * FROM student tab1 WHERE  NOT EXISTS  (
    SELECT  * FROM student tab2
    WHERE tab2. datum_narozeni > tab1.datum_narozeni );
```

Pozor na NULL hodnoty !

Subdotazy SQL - Vnořený dotaz za WHERE

CVIČENÍ:

Vypište seznam studentů, kteří nemají registrovaný žádný předmět.

Subdotazy SQL - Vnořený dotaz za WHERE

CVIČENÍ:

Vypište seznam studentů, kteří nemají registrovaný žádný předmět.

```
SELECT * FROM student s
WHERE NOT EXISTS (
                    SELECT * FROM vyuka v
                    WHERE s.uco=v.student_uco
                    );
```

Cvičení

CVIČENÍ

Cvičení

Najděte všechny učitele, kteří nevyučují žádný předmět.

Cvičení

Najděte všechny učitele, kteří nevyučují žádný předmět.

```
SELECT * FROM ucitel u
WHERE NOT EXISTS (
                    SELECT * FROM predmet p
                    WHERE u.ucitel_uco=p.ucitel_uco);
```

Cvičení

Vypište všechny studenty, kteří mají zapsaný předmět Databáze v biomedicíně i Černou magii. (predmet_id 1 a 10)

Cvičení

Vypište všechny studenty, kteří mají zapsaný předmět Databáze v biomedicíně i Černou magii. (predmet_id 1 a 10)

```
SELECT * FROM student s
WHERE EXISTS (SELECT * FROM vyuka v
                    WHERE s.uco=v.student_uco AND predmet_id=1)
INTERSECT
SELECT * FROM student s
WHERE EXISTS (SELECT * FROM vyuka v
                    WHERE s.uco=v.student_uco AND predmet_id=10);
```

Cvičení

Vypište všechny studenty, kteří mají zapsaný předmět Databáze v biomedicíně i Černou magii. (predmet_id 1 a 10)

```
SELECT * FROM student s
WHERE EXISTS (SELECT * FROM vyuka v
                    WHERE s.uco=v.student_uco AND predmet_id=1)
AND EXISTS (SELECT * FROM vyuka v
                    WHERE s.uco=v.student_uco AND predmet_id=10);
```

Cvičení

Vypište všechny studenty, kteří mají zapsaný předmět Databáze v biomedicíně (predmet_id 1), ale nemají zapsanou Černou magii (predmet_id 10).

Cvičení

Vypište všechny studenty, kteří mají zapsaný předmět Databáze v biomedicíně (predmet_id 1), ale nemají zapsanou Černou magii (predmet_id 10).

```
SELECT * FROM student s
WHERE EXISTS (SELECT * FROM vyuka v
                    WHERE s.uco=v.student_uco AND predmet_id=1)
INTERSECT
SELECT * FROM student s
WHERE NOT EXISTS (SELECT * FROM vyuka v
                    WHERE s.uco=v.student_uco AND predmet_id=10);
```

Cvičení

Vypište všechny studenty, kteří mají zapsaný předmět Databáze v biomedicíně (predmet_id 1), ale nemají zapsanou Černou magii (predmet_id 10).

```
SELECT * FROM student s
WHERE EXISTS (SELECT * FROM vyuka v
                    WHERE s.uco=v.student_uco AND predmet_id=1)
AND NOT EXISTS (SELECT * FROM vyuka v
                    WHERE s.uco=v.student_uco AND predmet_id=10);
```

Cvičení

Vypište všechna pracoviště, která v roce 2010 nezařadila do studie žádného pacienta.

Cvičení

Vypište všechna pracoviště, která v roce 2010 nezařadila do studie žádného pacienta.

```
SELECT * FROM sites s
WHERE NOT EXISTS (
        SELECT * FROM patient_study ps
        WHERE EXTRACT(YEAR FROM date_of_enrollment)=2010
        AND ps.study_site=s.site_id
        );
```

Cvičení

Vypište všechna pracoviště, která zařadila pacienta naposledy v roce 2010.

Cvičení

Vypište všechna pracoviště, která zařadila pacienta naposledy v roce 2010.

```
SELECT * FROM sites s
WHERE EXISTS (
        SELECT * FROM patient_study ps
        WHERE EXTRACT(YEAR FROM date_of_enrollment)=2010
        AND ps.study_site=s.site_id
        )
AND NOT EXISTS (
        SELECT * FROM patient_study ps
        WHERE EXTRACT(YEAR FROM date_of_enrollment)>2010
        AND ps.study_site=s.site_id
        );
```

Cvičení

Vypište všechna pracoviště, která zařadila pacienta naposledy v roce 2010.

```
SELECT s.site, s.site_id,
        MAX(EXTRACT(YEAR FROM ps.date_of_enrollment)) rok
FROM sites s JOIN patient_study ps
ON s.site_id=ps.study_site
GROUP BY s.site, s.site_id
HAVING MAX(EXTRACT(YEAR FROM ps.date_of_enrollment))=2010;
```

Cvičení

Najděte předměty, kam se přihlásil alespoň jeden student (muž) a vypište celkový počet přihlášených studentů.

Cvičení

Najděte předměty, kam se přihlásil alespoň jeden student (muž) a vypište celkový počet přihlášených studentů.

```
SELECT predmet_id, COUNT(*) FROM student s JOIN vyuka v
ON  s.uco = v.student_uco
WHERE EXISTS  (
        SELECT predmet_id FROM student s2 JOIN vyuka v2
        ON s2.uco = v2.student_uco
        AND s2.pohlavi = ‘muž' AND v.predmet_id=v2.predmet_id
        )
GROUP BY predmet_id;
```

Domácí úkol

Zjistěte počet pacientů v jednotlivých studiích po pracovištích a dle pohlaví
STUDY_NAME, SITE, SEX, počet pacientů

Děkuji za pozornost.