SQL

Databázové praktikum, 2019/2020

SQL

- Structured Query Language
- Základná syntax:
  - **SELECT** <zoznam_atributov>
    FROM <zoznam_suborov>
    WHERE <podmienka>
    [ORDER BY <stlpec1> ASC/DESC, <stlpec2> ASC/DESC, ...]
    [LIMIT 100]
    [OFFSET 0]

- SQL je v súčastnosti najpoužívanejší dotazovací jazyk
  - Štandard už z 80. tych rokov (1986)
  - Posledných 10 rokov sa štandard veľmi nemenil
  - Aj keď existuje štandard, „SQL“ je rôzne pre každý databázový systém
  - Dotazy napísané pre jeden databázový systém (PosgreSQL) nemusia fungovať v inom (MySQL)
    - Hlavne kvôli rôznym užitočným rozšíreniam SQL, ktoré poskytuje daný databázový systém

Príklady SQL dotazov

- ```
  SELECT *
  FROM emp
  WHERE deptno='20'
  ORDER BY ename DESC
  ```
- ```
  SELECT DISTINCT deptno FROM emp
  ```
- ```
  SELECT e.ename, d.dname
  FROM emp as e, dept as d
  WHERE e.deptno=d.deptno
  ```
- ```
  SELECT
      ename,
      sal*0.19 as 'taxes'
  FROM emp
  WHERE sal>=1000 and ename like 'john%'
  ```

Príklady SQL dotazov

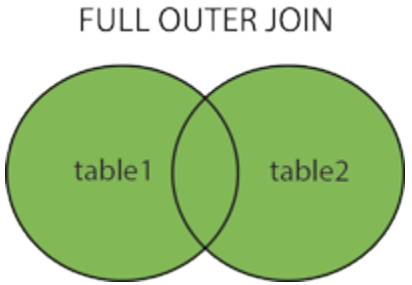
```
SELECT
    concat(e.firstname,’ ‘,e.lastname) as ename,
    (case
        when e.comm IS NULL
            then e.sal
        else e.comm+e.sal
    ) as ‘total_salary’
FROM emp
WHERE deptno>=20 and lower(e.firstname)=‘john’
```

JOINs

- Spojenie dvoch tabuliek
- Podmnožina kartézskeho súčinu tabuliek
  - Kartézsky súčin – každý riadok s každým

- INNER JOIN alebo iba JOIN
- LEFT JOIN
- RIGHT JOIN
- FULL JOIN

Kartézsky súčin (FULL JOIN):



Name	Deptno
John	10
Thomas	20
Joe	40

X

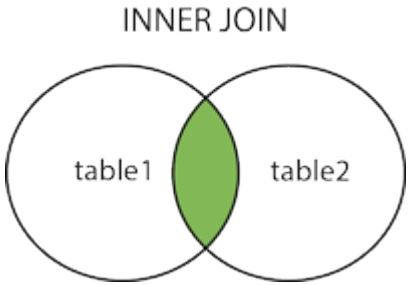
Deptno	Dept. name
10	Accounting
20	PR
30	Development

=

Name	Deptno	Deptno	Dept. name
John	10	10	Accounting
John	10	20	PR
John	10	30	Development
Thomas	20	10	Accounting
Thomas	20	20	PR
Thomas	20	30	Development
Joe	40	10	Accounting
Joe	40	20	PR
Joe	40	30	Development

SELECT * FROM emp, dept

INNER JOIN = JOIN:



Name	Deptno
John	10
Thomas	20
Joe	40

JOIN

Deptno	Dept. name
10	Accounting
20	PR
30	Development

=

Name	Deptno	Deptno	Dept. name
John	10	10	Accounting
~~John~~	~~10~~	~~20~~	~~PR~~
~~John~~	~~10~~	~~30~~	~~Development~~
~~Thomas~~	~~20~~	~~10~~	~~Accounting~~
Thomas	20	20	PR
~~Thomas~~	~~20~~	~~30~~	~~Development~~
~~Joe~~	~~40~~	~~10~~	~~Accounting~~
~~Joe~~	~~40~~	~~20~~	~~PR~~
~~Joe~~	~~40~~	~~30~~	~~Development~~

```
SELECT * FROM emp as e, dept as d
WHERE e.deptno=d.deptno

SELECT *
FROM emp as e
        join dept as d on e.deptno=d.deptno

SELECT * FROM emp as e natural join dept as d
```

INNER JOIN = JOIN:

Name	Deptno
John	10
Thomas	20
Joe	40

JOIN

Deptno	Dept. name
10	Accounting
20	PR
30	Development
10	Human res.

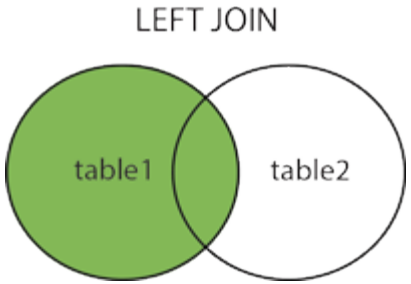
=

Name	Deptno	Deptno	Dept. name
John	10	10	Accounting
~~John~~	~~10~~	~~20~~	~~PR~~
~~John~~	~~10~~	~~30~~	~~Development~~
John	10	10	Human res.
~~Thomas~~	~~20~~	~~10~~	~~Accounting~~
Thomas	20	20	PR
~~Thomas~~	~~20~~	~~30~~	~~Development~~
~~Thomas~~	~~20~~	~~10~~	~~Human res.~~
~~Joe~~	~~40~~	~~10~~	~~Accounting~~
~~Joe~~	~~40~~	~~20~~	~~PR~~
~~Joe~~	~~40~~	~~30~~	~~Development~~
~~Joe~~	~~40~~	~~10~~	~~Human res.~~

```
SELECT *
FROM emp as e
        join dept as d
          on e.deptno=d.deptno
```

Ako by ste zapísali JOIN v Datalogu?

LEFT [outer] JOIN:



Name	Deptno
John	10
Thomas	20
Joe	40

LEFT JOIN

Deptno	Dept. name
10	Accounting
20	PR
30	Development
10	Human res.

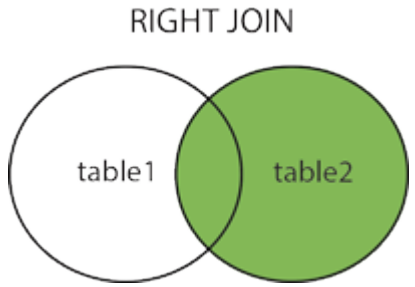
=

Name	Deptno	Deptno	Dept. name
John	10	10	Accounting
John	10	10	Human res.
Thomas	20	20	PR
Joe	40	null	null

```
SELECT *
FROM emp as e
        left join dept as d
                on e.deptno=d.deptno
```

Ako by ste zapísali LEFT JOIN v Datalogu?

RIGHT [outer] JOIN:



Deptno	Dept. name
10	Accounting
20	PR
30	Development
10	Human res.

RIGHT JOIN

Name	Deptno
John	10
Thomas	20
Joe	40

=

Name	Deptno	Deptno	Dept. name
John	10	10	Accounting
John	10	10	Human res.
Thomas	20	20	PR
Joe	40	null	null

To isté ako LEFT JOIN, akurát v obrátenom poradí

```
SELECT *
FROM dept as d
        right join emp as e
                on e.deptno=d.deptno
```

Operátory, výrazy a funkcie

- Vo WHERE časti môžete používať operátory:
  - =, <> (resp. !=), >, <, >=, <=, BETWEEN, LIKE, IN, IS NULL, IS NOT NULL
  - AND, OR, ! (not)
- Taktiež môžete používať aritmetické výrazy a hromadu ďalších funkcií
  - Napr. concat(e.firstname,' ', e.lastname)
  - Funkcie na prácu s dátumami, číslami a pod.

  - Zoznam podporovaných funkcií a ich syntax závisí na konkrétnom databázovom systéme
    - https://www.postgresql.org/docs/current/static/functions.html

SELECT v SELECTe (“podselecty”)

- SELECT name FROM emp as e WHERE e.ID IN (SELECT ID FROM managers)
- SELECT name FROM emp as e WHERE exists (SELECT * FROM managers WHERE id=e.id)

- V prípade podselektov si treba dávať pozor na efektívnosť
  - JOIN operácie vie databázový systém vcelku dobre optimalizovať (ak máte správne navrhnutú DB – o tom neskôr)
  - Optimalizovať podselekty však môže byť náročnejšie
  - Pomocou podselektov ľahšie napíšete „neefektívny“ dotaz
- Čo myslíte, ktorý dotaz vyššie má väčšiu šancu byť menej efektívny?
  - Ako by ste dotazy prepísali iba s použitím JOINov?

UNION

- SELECT name
  FROM emp_dallas WHERE sal>=1000

  UNION [ALL]

  SELECT name
  FROM emp_huston WHERE sal>=500

- Typy a počet atribútov v SELECT časti musia byť rovnaké

Pomocné selecty / tabuľky

- WITH emp_huston AS (
  SELECT * FROM emp as e, dept as d
  WHERE e.deptno=d.deptno and d.dname='huston'
)
  SELECT * FROM emp_huston WHERE sal>=1000

- CREATE TEMPORARY TABLE emp_huston (
  SELECT * FROM emp as e, dept as d
  WHERE e.deptno=d.deptno and d.dname='huston'
  );
- SELECT * FROM emp_huston WHERE sal>=1000

DML a DDL

- INSERT
- UPDATE
- DELETE

- CREATE [TEMPORARY] TABLE
- ALTER TABLE
- DROP TABLE
- …
- Bude to predmetom ďalších prednášok

PostgreSQL

- Na cvičenia budeme pracovať s databázovým systémom PostgreSQL
- Väčšina databázových systémov funguje formou client-server
  - **Server**
    - obsahuje dáta
    - vie chápať SQL dotazy
    - Klienti sa na neho pripájajú väčšinou cez SOCKETove spojenie (pripadne named pipes / iné kanály podporované OS)
  - **Client**
    - Aplikácia, ktorá potrebuje pracovať s dátami
    - Serveru posiela dotazy v SQL jazyku
    - Zobrazí / spracuje odpoveď od servera

PostgreSQL

- Momentálne sa budeme na databázový server pripájať cez „builtin“ klienta
  - Spustíte ho v konzole príkazom „psql“
  - Dokumentácia: https://www.postgresql.org/docs/current/static/app-psql.html
- Máme otvorené dve terminálové okná na cvika.dcs.fmph.uniba.sk
  - V jednom okne editujeme súbor so zadaním, napr. vim queries_emp.sql
  - V druhom okne editovaný súbor spustíme (všetky dotazy v ňom) príkazom
    - psql -f queries_emp.sql
- Každý pracujete nad svojou databázov emp, ktorá je automaticky vybratá po spustení psql

Práca s konzolov PostgreSQL

- Konzolu PostgreSQL sputíte príkazov psql
- V konzole následne môžete písať dotazy. Napr. SELECT * FROM emp;
- Zaujímavé špeciálne príkazy:
  - \d emp resp. \d+ emp - zobrazí štruktúru tabuľky
  - \d - zobrazí zoznam tabuliek v aktuálnej databáze
  - \db - zobrazí zoznam databáz
  - \c emp - pripojí sa k databáze emp
  - \q - ukončenie konzoly
- Dokumentácia konzolových príkazov: http://www.postgresql.org/docs/current/static/app-psql.html