číslo: 12 Otázka: Jazyk SQL – SELECT, VIEW (spojování tabulek, agregační funkce, seskupování záznamů)
předmět: DS

- <u>Select s klauzulemi</u>
  - **povinné:**
    - Select [distinct] - seznam atributů
    - From – název tabulky nebo datové spojení tabulek
  - **klauzule:**
    - Where – podmínka pro from
    - Group by – atribut1 ; atribut2- seskupení podle atributů
    - Having - podmínka group by
    - Order by – atribut1 ; atribut2 - seřazení atributů

1. From – určí množinu dat
2. Where – omezí množinu podle podmínky
3. Group by – rozdělí do skupin podle hodnot atributů
4. Having – omezí skupiny podle atributu
5. Order by – seřadí podle atributů
6. Select – vypíše data

Klauzole where:
- A) využití matematického operátoru:
- B) využití log. operátorů:
- C) vyhledání podle množiny
- D) vyhledávání podle vzoru:
- E) podmínka (Not Null)

př. výpis všech zaměstnanců s platem vyšším než 40 000,-.

```
SELECT zam.prijmeni, zam.jmeno, zam.plat
FROM zam
WHERE zam.plat > 40000;
```

Klauzole Order by:
- řízení řazení dat při výpisu
- skládá se z jmen sloupců, podle nichž má být seřazen
- umožňuje řadit vzestupně (ASC) nebo sestupně (DESC)

```
SELECT zam.prijmeni, zam.jmeno, zam.pozice, zam.plat
FROM zam
ORDER BY zam.pozice ASC, zam.plat DESC;
```

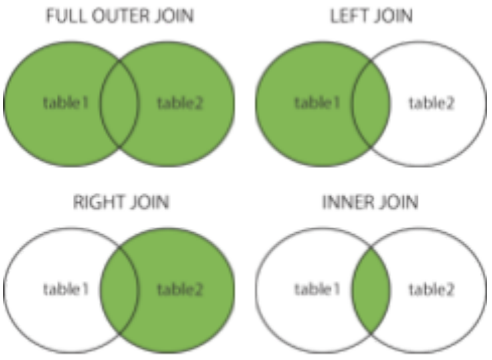
Klauzole Group by:
- seskupený dotaz, pro každou skupinu vytvoří jednoduchý souhrn
- sloupec (atributy) jsou uvedeny za příkazem Group by a rozdělení se podle nich provede zleva doprava

```
SELECT zam.odd_id, COUNT( zam.id ) AS pocet_zam_v_oddeleni,
    SUM( zam.plat ) AS plat_vsech_zam_v_oddeleni
FROM zam
GROUP BY zam.odd_id
ORDER BY zam.odd_id;
```

Klauzole Having:
- je určena jako podmínka (omezení) pouze skupin (tj. Bez Group by nemá smysl)
- jinak pracuje jako Where, ale může v ní být samostatně Agregační fce

```
SELECT zam.odd_id, COUNT( zam.id ) AS
pocet_zam_v_oddeleni,
        FROM zam
        GROUP BY zam.odd_id
        HAVING COUNT( zam.id ) > 5
        ORDER BY zam.odd_id;
```

Agregační funkce:
- funkce, které pracují s jedním argumentem a vrátí jen jednu výslednou hodnotu
- nelze je samostatně použít v klauzoli where (jedině v pod selectu)
- pokud chceme odstranit duplicity, používáme klíčové slovo distinct
- Count () … vrátí celkový počet řádků zadaného sloupce (může být i *)
- Sum () … vrací součet hodnot zadaného sloupce (sloupec musí mít číselný dat. typ)
- Avg () … vrací průměr zadaného sloupce (musí být číselný dat. typ)
- Min () … vrací nejmenší hodnotu ze sloupce (-„- řetězce a datum)
- Max () … vrací největší hodnotu ze sloupce (-„-)



SQL – spojení tabulek „Joins“
- A) Inner join – výpis průniku množin dat z obou tabulek
- B) left (right) join – výpis všech dat z levé (pravé) tabulky a průnik
- C) outher join – výpis všech dat z obou tabulek (sjednocení)

Syntax: **`inner join Tabulka on Tabulka.Id=TabulkaNaKterouJoinuju.FK;`**

View (pohled)

- Pohled je virtuální tabulka, jejíž definice je uložena v databázi na straně serveru. Tato tabulka často bývá kombinací více skutečných db. tabulek (např. pomocí joinů)
- Narozdíl od tabulky však pohled neobsahuje skutečná data, ale slouží pro uložení často používaných složitých dotazů nad daty (tzv. uložený select). Data se automaticky zobrazí (vykonání těchto příkazů) při použití pohledu.
- To znamená, že uložený pohled (view) můžeme použít „normální tabulku" v dalších dotazech v části From...
- Pohled může být použit jako ochranný mechanismus, který dovolí uživatelům přístup pouze k viditelným datům v pohledu a nikoliv ke všem skutečným datům v tabulce/kách. Tedy např. do definice pohledu nezahrne všechny atributy.
    - Syntax:
        - **create view<view name> as <select statement>;**
        - **alter view <view name> as <different select statement>;**
        - **drop view <view name>;**

příklad: vytvoření pohledu pro přehled zaměstnanců ve všech odděleních

```
create view zam_odd_view as

select zam.id, zam.prijmeni, zam.jmeno, odd.nazev

from zamestnanec as zam left join oddeleni as odd on odd.id = zam.odd_id ;
```