

# Agregácia v SQL

Databázové praktikum, 2019/2020

# SQL a agregácia

- Niekedy chceme namiesto vypísania zoznamu riadkov radšej zistiť ich počet / súčet a pod.
  - Na to slúžia tzv. agregáčné funkcie
  - Sum, Min, Max, Avg, Stdev, Count, ...
  - <https://www.postgresql.org/docs/current/static/functions-aggregate.html>
- Niekedy chceme riadky zoskupiť podľa nejakého atribútu
  - Napr. chceme zoskupiť zamestnancov podľa ich oddelenia a pod.
  - Prípadne by sme chceli zistiť počet zamestnancov v rámci jednotlivých oddelení

# GROUP BY

- **SELECT** <zoznam\_atributov>  
**FROM** <zoznam\_tabuliek>  
**WHERE** <podmienka>  
**GROUP BY** <zoznam\_atributov>  
**HAVING** <podmienka>  
[**ORDER BY** ... **LIMIT** ... **OFFSET** ...]
- Príkaz GROUP BY zoskupí riadky s rovnakou hodnotou v uvedených atribútoch do jedného riadku
- Zoznam atribútov v SELECT časti môže obsahovať len atribúty uvedené v GROUP BY časti a agregáčn é funkcie
  - Toto nie je celkom pravda pre všetky databázové systémy (napr. MySQL takúto reštrikciu nemá)
- Podmienka v HAVING môže obsahovať agregáčn é funkcie, zatiaľ čo WHERE nemôže

# GROUP BY príklad

- **SELECT** deptno, count(\*)  
**FROM** emp  
**WHERE** sal>3000  
**GROUP BY** deptno  
**HAVING** count(\*)>=2

# GROUP BY príklad

name	deptno	sal
John	10	1000
Thomas	10	3100
George	10	3200
Lucas	20	3100
Bob	20	2050
Joe	30	1000
Francis	30	3050
Hugo	40	1000
Mike	40	5000
Robert	40	2900
Anna	50	8000

WHERE



name	deptno	sal
George	10	3200
Thomas	10	3100
Lucas	20	3100
Francis	30	3050
Mike	40	5000
Anna	50	8000

```
SELECT deptno, count(*)  
FROM emp  
WHERE sal>3000  
GROUP BY deptno  
HAVING count(*)>=2
```

# GROUP BY príklad

name	deptno	sal
George	10	3200
Thomas	10	3100
Lucas	20	3100
Francis	30	3050
Mike	40	5000
Anna	50	8000

GROUP BY



deptno	count(*)	Name	deptno	sal
10	2			
		George	10	3200
		Thomas	10	3100
20	1			
		Lucas	20	3100
30	1			
		Francis	30	3050
40	1			
		Mike	40	5000
50	1			
		Anna	50	8000

```
SELECT deptno, count(*)  
FROM emp  
WHERE sal>3000  
GROUP BY deptno  
HAVING count(*)>=2
```

# GROUP BY příklad

deptno	count(*)	Name	deptn	sal
10	2			
		George	10	3200
		Thomas	10	3100
20	1			
		Lucas	20	3100
30	1			
		Francis	30	3050
40	1			
		Mike	40	5000
50	1			
		Anna	50	8000

HAVING



deptno	count(*)
10	2

```
SELECT deptno, count(*)  
FROM emp  
WHERE sal>3000  
GROUP BY deptno  
HAVING count(*)>=2
```

# Drobnosti

- Zoznam atribútov v SELECT časti môže obsahovať len atribúty uvedené v GROUP BY časti a agregáčné funkcie
- Toto je pre programátorov trochu otrava:
  - Student(StudentID, Meno, Priezvisko, TriedaID)  
Trieda(TriedaID, Nazov)
  - SELECT s.triedaid, t.**nazov**, count(\*) FROM student as s, trieda as t  
WHERE s.triedaid = t.triedaid GROUP BY s.triedaid, t.**nazov**
  - TriedaID jednoznačne určuje názov triedy, no programátor to musí zbytočne zapísať 2x



# Drobnosti

- MySQL:

V SELECT časti môžem použiť akýkoľvek atribút. Ak je z množiny atribútov, ktoré nie sú v GROUP BY, vyberie sa náhodný prvok zo skupiny

- PostgreSQL:

When GROUP BY is present, or any aggregate functions are present, it is not valid for the SELECT list expressions to refer to ungrouped columns except within aggregate functions or when the ungrouped column is functionally dependent on the grouped columns, since there would otherwise be more than one possible value to return for an ungrouped column. A functional dependency exists if the grouped columns (or a subset thereof) are the primary key of the table containing the ungrouped column.

# Drobnosti

- Ak SELECT obsahuje agregáčnú funkciu, ale bez GROUP BY
  - potom všetky riadky akoby boli zaradené do jednej skupiny – výstupom je riadok
    - Napr. `SELECT count(*) FROM emp;` `SELECT max(sal) FROM emp;`
  - podobne ak uvedieme HAVING bez GROUP BY