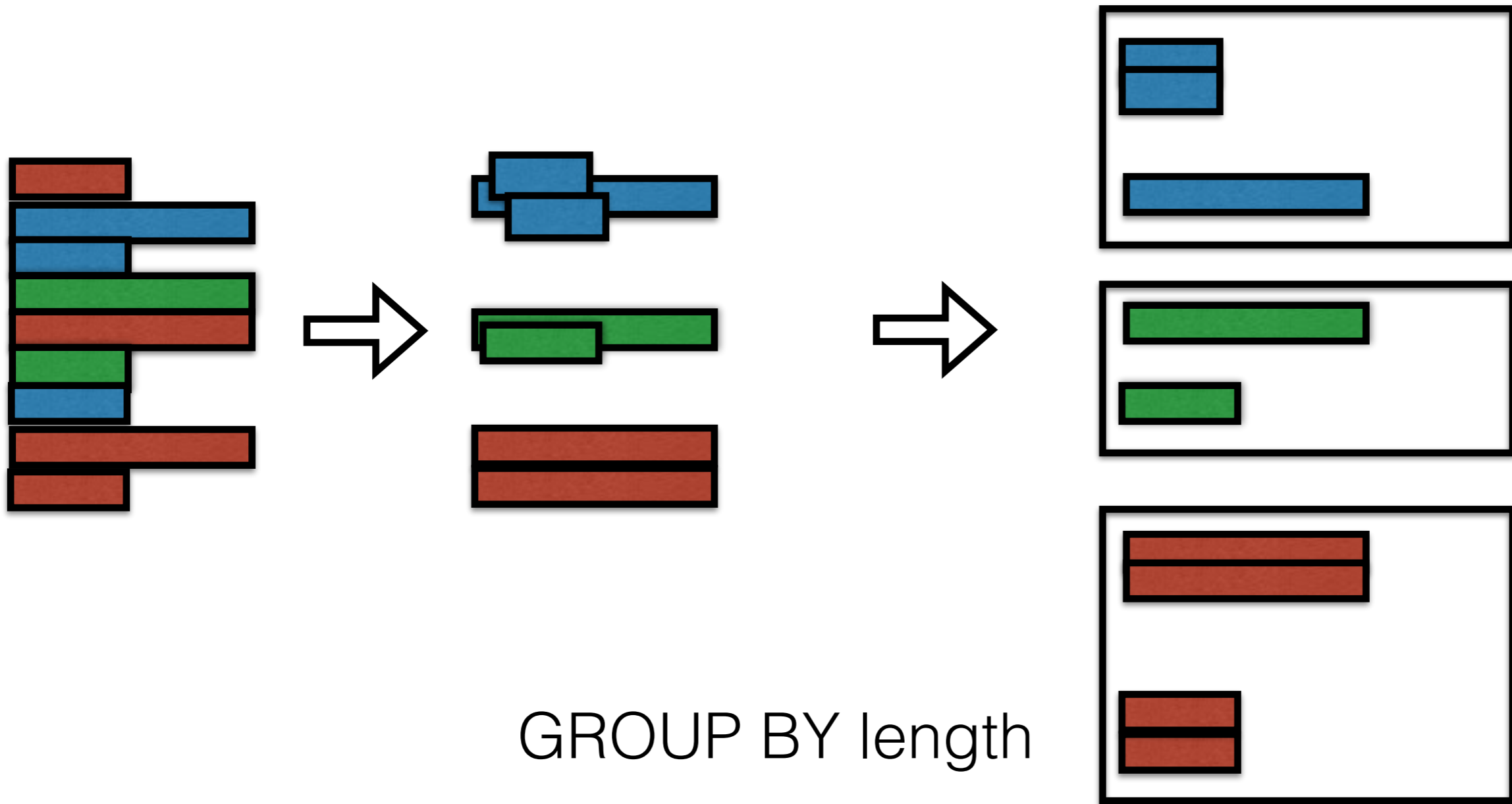


# Úvod do jazyka SQL

Hodina č. 6

# dvojitý GROUP BY



<http://munidb.jdem.cz/>

Počet lidí, které mají jednotlivé fakulty v jednotlivých budovách

```
SELECT fakulta, budova, SUM(pocet_lidi)  
FROM fakulty GROUP BY fakulta, budova
```

**COUNT(\*)**

**VS**

**COUNT(sloupec)**

	A	B	
1	<b>Kniha</b>	<b>Pujceno</b>	
2	A	Osoba 1	
3	A	Osoba 2	
4	A		
5	B	Osoba 1	
6	B	Osoba 2	
7	C		
8	D		
9	E	Osoba 1	
10	E		
11	F	Osoba 2	
12	G	Osoba 3	
13			

## Databáze “minidb”

Vypište počet výtisků jednotlivých knih spolu s počtem těch, které jsou zrovna vypůjčené

```
SELECT kniha, COUNT(*) as vytisku FROM  
minidb GROUP BY kniha
```

VS.

```
SELECT kniha, COUNT(Pujceno) as  
vypujceno FROM minidb GROUP BY kniha
```

## Databáze “minidb”

Vypište počet výtisků jednotlivých knih spolu s počtem těch, které jsou zrovna vypůjčené

```
SELECT kniha,  
COUNT(*) AS vytisku,  
COUNT(Pujceno) AS vypujceno,  
COUNT(*) - COUNT(Pujceno) AS volnych  
FROM minidb GROUP BY kniha
```



**DISTINCT**

## Databáze “fakulty”

Vypište jednotlivé fakulty.

```
SELECT fakulta FROM fakulty GROUP BY  
fakulta
```

nebo

```
SELECT DISTINCT fakulta FROM fakulty
```

# Databáze “fakulty”

Vypište počet budov jednotlivých fakult

## Databáze “fakulty”

Vypište počet budov jednotlivých fakult

```
SELECT fakulta, COUNT(budova),  
COUNT(katedra) FROM fakulty GROUP BY  
fakulta;
```

**^ TOHLE NEFUNGUJE!**

**oba COUNTy prostě spočítají “počet řádků”**

**Budov je však méně, jen se opakují**

## Databáze “fakulty”

Vypište počet budov jednotlivých fakult

```
SELECT fakulta, COUNT(DISTINCT budova),  
COUNT(katedra) FROM fakulty GROUP BY  
fakulta;
```

**^ TOHLE FUNGUJE!**

**DISTINCT způsobí, že každá z budov je  
započítána jen jednou**

# GROUP\_CONCAT

*The `group_concat()` function returns a string which is the concatenation of all non-NULL values of X. If parameter Y is present then it is used as the separator between instances of X. A comma (",") is used as the separator if Y is omitted. The order of the concatenated elements is arbitrary.*

## Databáze “fakulty”

Vypište seznam budov pro jednotlivé fakulty

```
SELECT fakulta, GROUP_CONCAT(DISTINCT  
budova) FROM fakulty GROUP BY fakulta
```

# Slovesa SQL

- SELECT ...
- INSERT
- UPDATE
- DELETE



# INSERT

```
INSERT INTO table (column1, column2,...)  
VALUES (value1, value2,...);
```

# UPDATE

**UPDATE** table **SET** column1 = value1,  
column2 = value2 **WHERE** ... **AND** ...

Doporučení: Při UPDATE si vždy prvně pomocí SELECT vyzkoušejte, jestli WHERE podmínka vrací opravdu to co chcete.

# DELETE

**DELETE FROM** table **WHERE** ... **AND** ...

Doporučení: Při DELETE si vždy prvně pomocí SELECT vyzkoušejte, jestli WHERE podmínka vrací opravdu to co chcete.

# DELETE

**DELETE FROM** table **WHERE** ... **AND** ...

**JOIN**

**Příspěvky**

Jméno	E-mail	Příspěvek	Kampaň
Honza	<u><a href="mailto:jsem@posp.cz">jsem@posp.cz</a></u>	300	hithit
Honza	<u><a href="mailto:jsem@posp.cz">jsem@posp.cz</a></u>	1000	kontaktní
Tomáš	<u><a href="mailto:fil.hrabal@gmail.com">fil.hrabal@gmail.com</a></u>	500	hithit
Eliška	<u><a href="mailto:eliska@eps.cz">eliska@eps.cz</a></u>	300	kontaktní

# Problémy?

(například)

- Změna kontaktů u dárce - musíte změnit všude
- Pokud byste chtěli více kontaktů (telefon, adresa, ...), tak se tabulka nafukuje, ač vám původně šlo jen o evidenci příspěvků
- Problematické evidování poznámek (např. “nevolat”)
- Evidence potenciálních dárců je problematická
- Kontakty nemůžete snadno využít v jiné interní agendě



# Řešení?

## Oddělené tabulky

**Donoři**

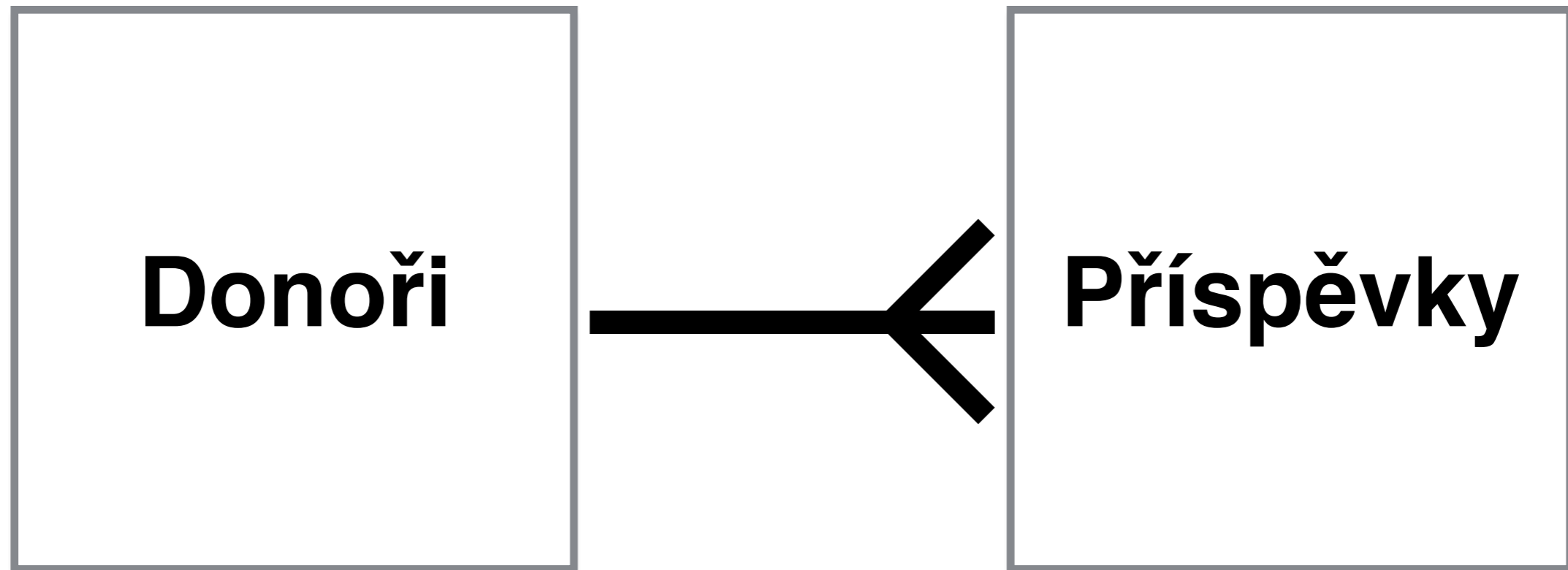
**Příspěvky**

ID	Jméno	Kontakt	Telefon	Poznámka
1	Honza	<a href="mailto:jsem@posp.cz">jsem@posp.cz</a>	111	Nevolat před 10 dopo
2	Tomáš	<a href="mailto:fil.hrabal@gmail.com">fil.hrabal@gmail.com</a>	222	
3	Eliška	<a href="mailto:eliska@frankbold.org">eliska@frankbold.org</a>	333	Pracuje ve FrankBold

Donor	Příspěvek	Kdy	Kampaň
1	300	30. 10. 2014	hithit
1	1000	1. 11. 2014	konktní
2	500	1. 11. 2014	hithit
3	300	30. 10. 2014	kontaktní

**<http://fund3.jdem.cz/>**

# Řešení?



ID	Jméno	Kontakt	Poznámka
1	Honza	<a href="mailto:jsem@posp.cz">jsem@posp.cz</a>	Nevolat před 10 dopo
2	Tomáš	<a href="mailto:fil.hrabal@gmail.com">fil.hrabal@gmail.com</a>	
3	Eliška	<a href="mailto:eliska@eps.cz">eliska@eps.cz</a>	Pracuje v EPS

Donor	Příspěvek	Kdy	Kampaň
1	300	30. 10. 2014	hithit
1	1000	1. 11. 2014	konktratní
2	500	1. 11. 2014	hithit
3	300	30. 10. 2014	kontaktní

# JOIN

ID	Jméno	Kontakt	Poznámka	Donor	Příspěvek	Kdy	Kampaň
1	Honza	<u>jsem@pos</u> <u>p.cz</u>	Nevolat před 10	1	300	30. 10. 2014	hithit
2	Tomáš	<u>fil.hrabal@</u> <u>gmail.com</u>		1	1000	1. 11. 2014	konktatní
3	Eliška	<u>eliska@eps</u> <u>.cz</u>	Pracuje v EPS	2	500	1. 11. 2014	hithit
				3	300	30. 10. 2014	kontaktní

# JOIN

**SELECT** \* **FROM** prispevky

**JOIN** donori **ON** donori.ID = prispevky.donor

ID	Jméno	Kontakt	Poznámka		Donor	Příspěvek	Kdy	Kampaň
1	Honza	<u>jsem@pos</u> p.cz	Nevolat před 10	→	1	300	30. 10. 2014	hithit
2	Tomáš	<u>fil.hrabal@</u> gmail.com		→	1	1000	1. 11. 2014	konktratní
3	Eliška	<u>eliska@eps</u> .cz	Pracuje v EPS	→	2	500	1. 11. 2014	hithit
				→	3	300	30. 10. 2014	kontaktní



# Výsledek

Jméno	Kontakt	Poznámka	Příspěvek	Kampaň
Honza	<a href="mailto:jsem@posp.cz">jsem@posp.cz</a>	Nevolat před 10 dopo	300	hithit
Honza	<a href="mailto:jsem@posp.cz">jsem@posp.cz</a>	Nevolat před 10 dopo	1000	kontaktní
Tomáš	<a href="mailto:fil.hrabal@gmail.com">fil.hrabal@gmail.com</a>		500	hithit
Eliška	<a href="mailto:eliska@eps.cz">eliska@eps.cz</a>	Pracuje v EPS	300	kontaktní

**Se "spojenou" tabulkou  
se dá pracovat úplně  
stejně jako s "klasickou"  
tabulkou**

# **POZOR:**

# **Databáze je hloupá!**

**Neví nic o sémantice vašich dat**

```
SELECT * FROM donori  
JOIN prispevky  
ON donori.rocnik = prispevky.castka
```