

“Databázové systémy”

Hodina č. 6

Test

- Úterý 3. 11., začátek 9.20 (čili v rámci hodiny) v G02
- Tři snadné příklady
- **Musíte** mít alespoň 2 správně. JINAK!
- Dálkaři (a prezenční studenti s dobrou výmluvou) si domluví termín bokem - možnosti jsme poslali e-mailem
- **Můžete** mít vlastní materiály + dokumentaci SQLite

Opakování

```
SELECT fakulta, SUM(pocet_lidi) as lidi_na_pracovisti  
FROM fakulty  
WHERE fakulta LIKE "f%"  
GROUP BY fakulta  
HAVING lidi_na_pracovisti > 200  
ORDER BY fakulta DESC  
LIMIT 3
```

GROUP_CONCAT()

```
SELECT fakulta, GROUP_CONCAT(katedra, ", "), SUM(rozpocet)
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** ...

TIP:

Vždy si nejdřív zkuste **SELECT se stejnou podmínkou**

DELETE

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

TIP:

Vždy si nejdřív zkuste SELECT se stejnou podmínkou

JOIN

Příspěvky

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

Problémy?

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

Řešení?

Donoři

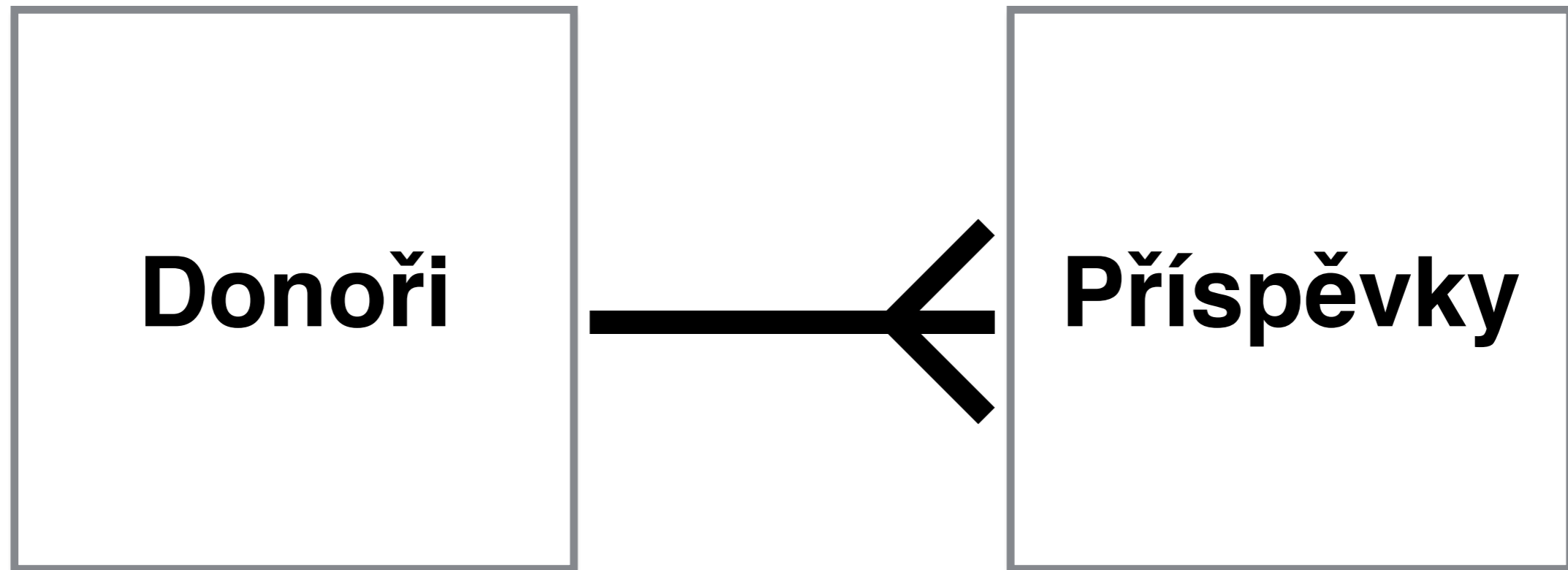
Příspěvky

ID	Jméno	Kontakt	Telefon	Poznámka
1	Honza	jsem@posp.cz	111	Nevolat před 10 dopo
2	Tomáš	fil.hrabal@gmail.com	222	
3	Eliška	eliska@frankbold.org	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://fund2.jdem.cz/>

Řešení?



ID	Jméno	Kontakt	Poznámka
1	Honza	jsem@posp.cz	Nevolat před 10 dopo
2	Tomáš	fil.hrabal@gmail.com	
3	Eliška	eliska@eps.cz	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	konktratní
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

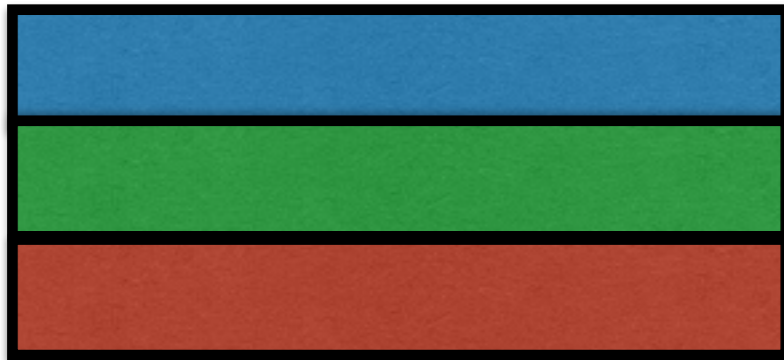
ID	Jméno	Kontakt	Poznámka	Donor	Příspěvek	Kdy	Kampaň
1	Honza	<u>jsem@pos</u> <u>p.cz</u>	Nevolat před 10	1	300	30. 10. 2014	hithit
2	Tomáš	<u>fil.hrabal@</u> <u>gmail.com</u>		1	1000	1. 11. 2014	konktratní
3	Eliška	<u>eliska@eps</u> <u>.cz</u>	Pracuje v EPS	2	500	1. 11. 2014	hithit
				3	300	30. 10.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

Výsledek

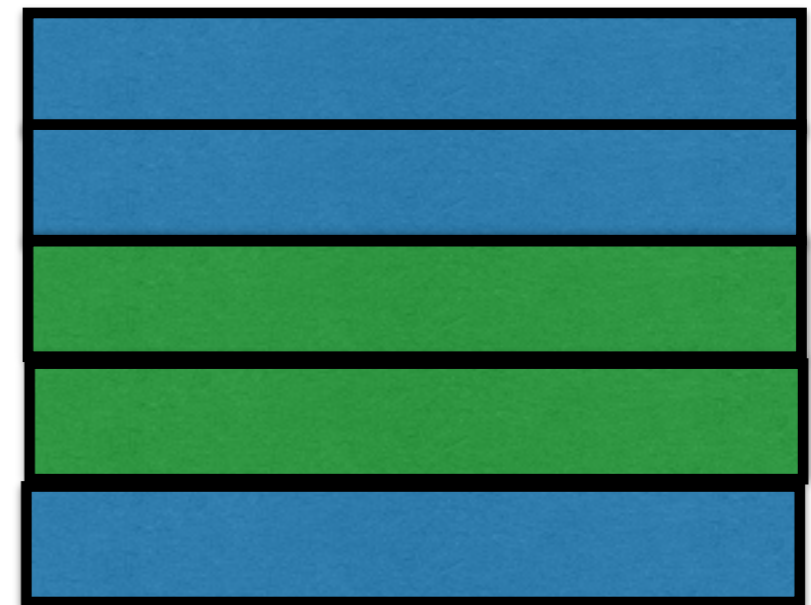
Jméno	Kontakt	Poznámka	Příspěvek	Kampaň
Honza	jsem@posp.cz	Nevolat před 10 dopo	300	hithit
Honza	jsem@posp.cz	Nevolat před 10 dopo	1000	kontaktní
Tomáš	fil.hrabal@gmail.com		500	hithit
Eliška	eliska@eps.cz	Pracuje v EPS	300	kontaktní

JOIN vs. LEFT JOIN

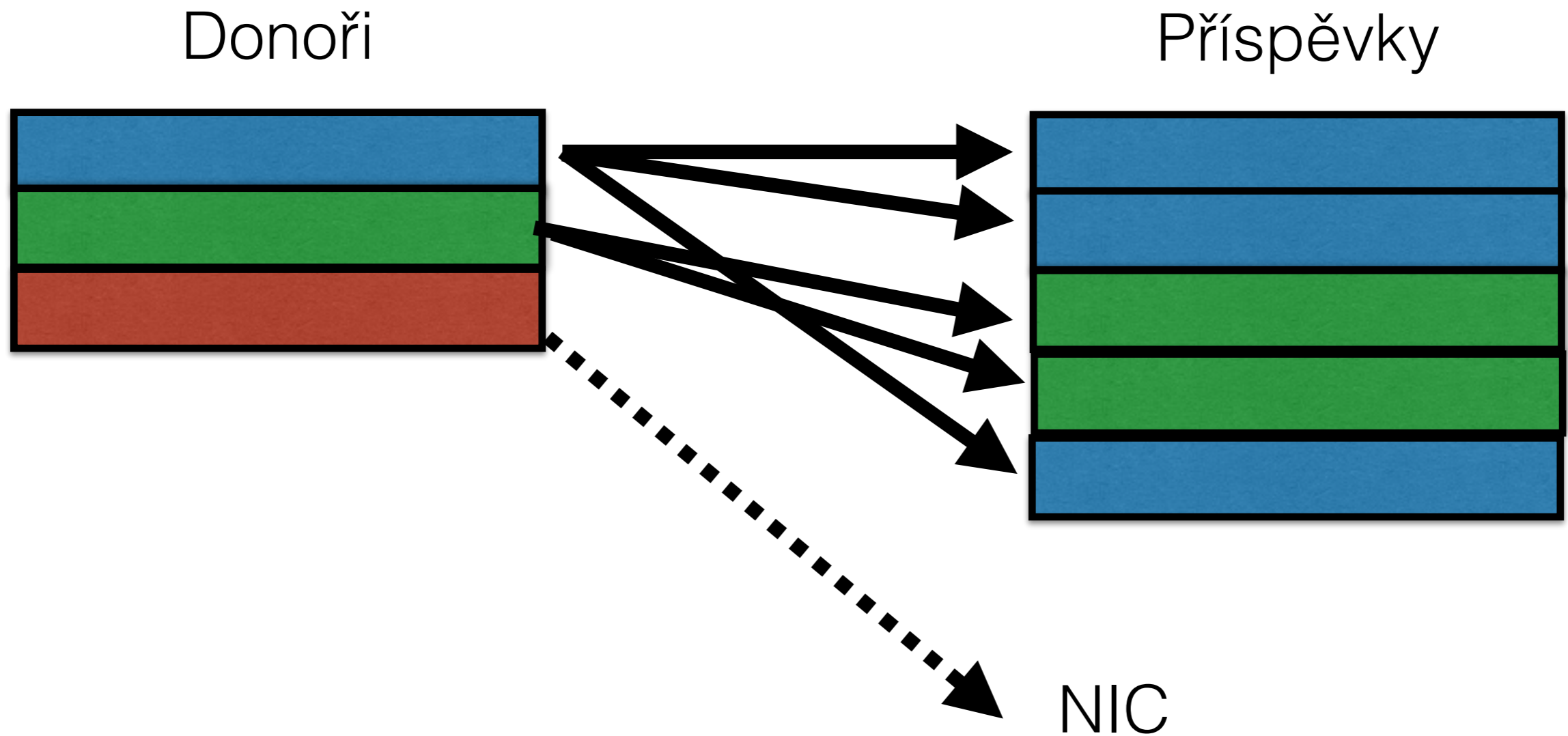
Donoři



Příspěvky



JOIN vs. LEFT JOIN



(INNER) JOIN

Donoři

Blue
Blue
Green
Green
Blue

Příspěvky

Blue
Blue
Green
Green
Blue

POZOR:

Databáze je hloupá!

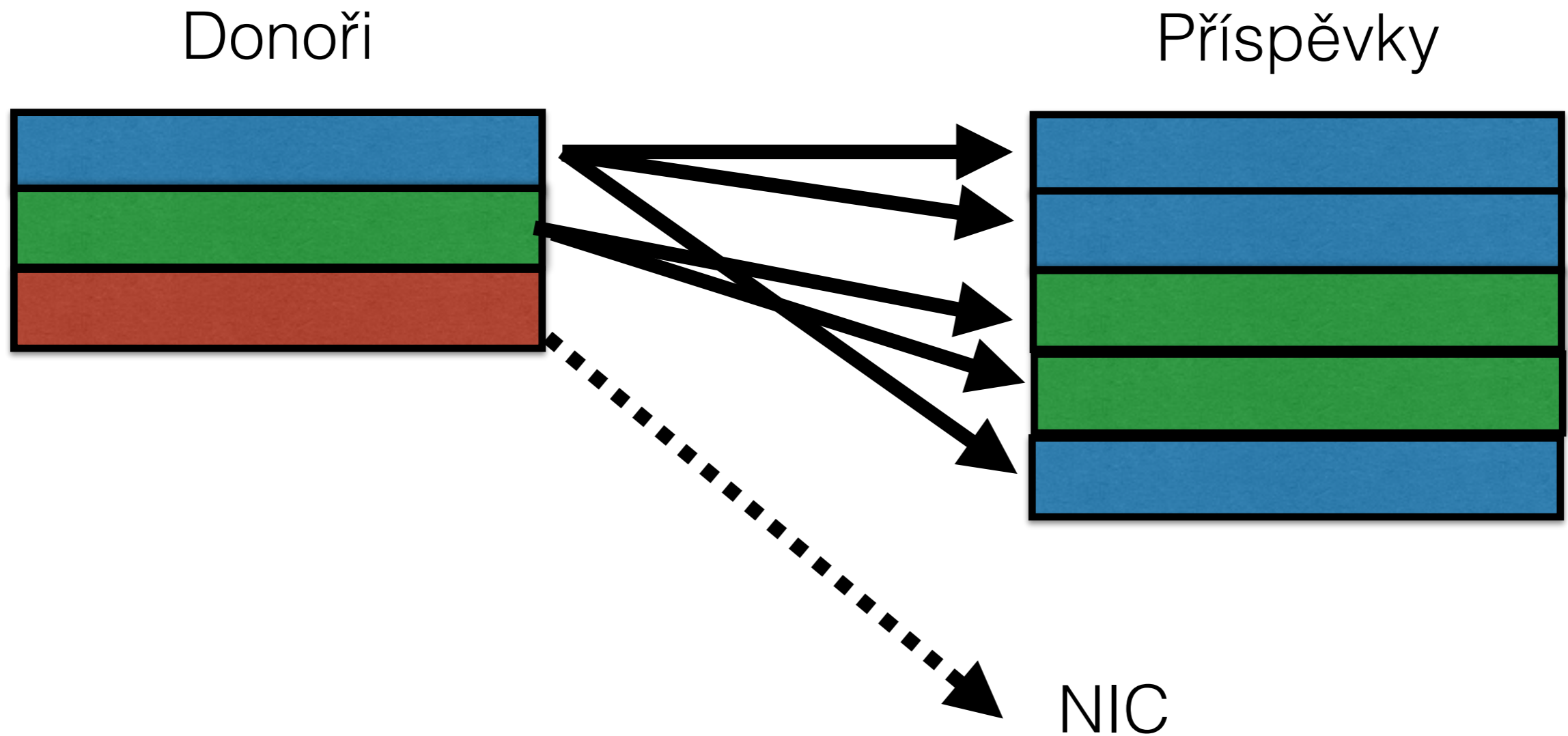
Neví nic o sémantice vašich dat

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

Tento dotaz sice výsledky vrátí, ale nemají žádný smysl

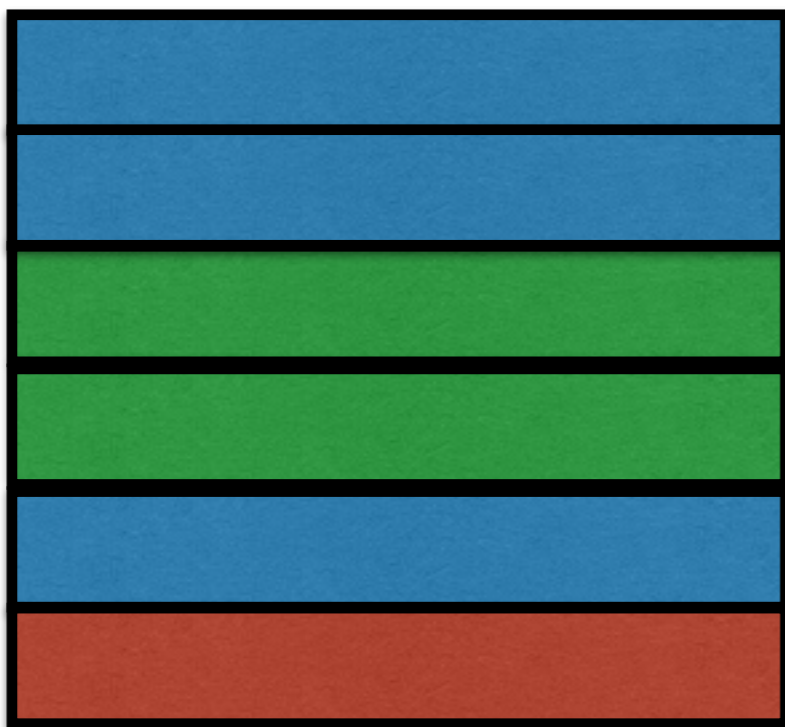
ID	Jméno	Kontakt	Telefon	Ročník	Poznámka
1	Honza	jsem@posp.cz	111	1984	Nevolat před 10 dopo
2	Tomáš	fil.hrabal@gmail.com	222	1985	
3	Eliška	eliska@frankbold.org	333	1983	Pracuje ve FrankBold

JOIN vs. LEFT JOIN

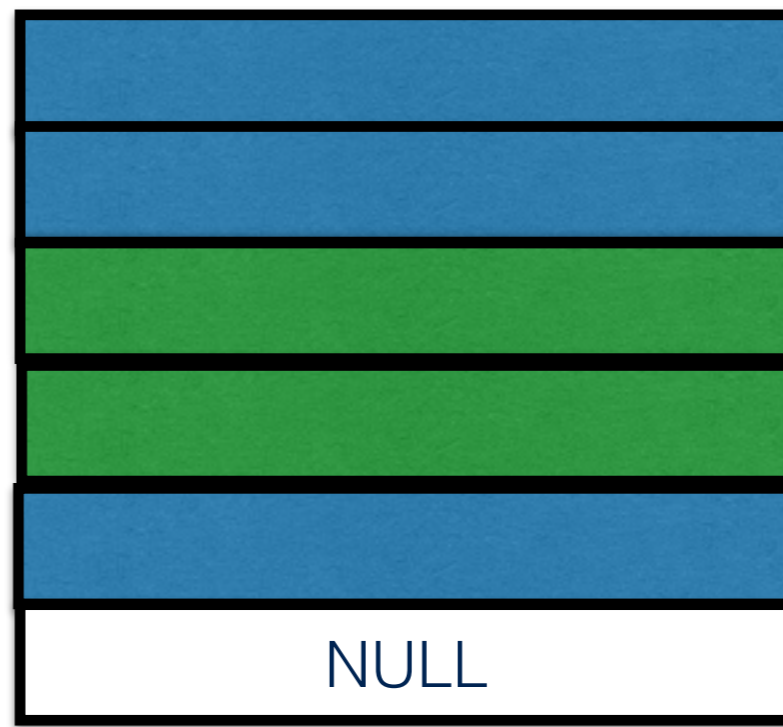


LEFT

Donoři



Příspěvky



Otázka

Kolik kdo z dárců v průběhu roku přispěl?

```
SELECT jmeno, SUM(castka)
FROM donori
LEFT JOIN prispevky ON donori.id = prispevky.donor
GROUP BY donori.id
```

Otázka

Kolik kdo z dárců v průběhu roku přispěl
A zobrazit pouze ty, kteří přispěli alespoň něco.
(Možnost č. 1)

```
SELECT jmeno, SUM(castka)
FROM donori
JOIN prispevky ON donori.id = prispevky.donor
GROUP BY donori.id
```

LEFT JOIN jsme změnili na JOIN - tedy se zobrazují pouze ty
řádky z první tabulky, který mají alespoň jeden odpovídající
řádek z druhé tabulky

Otázka

Kolik kdo z dárců v průběhu roku přispěl
A zobrazit pouze ty, kteří přispěli alespoň něco.
(Možnost č. 2)

```
SELECT jmeno, SUM(castka)
FROM donori
LEFT JOIN priskevky ON donori.id = priskevky.donor
WHERE castka IS NOT NULL
GROUP BY donori.id
```

Pomocí klauzule WHERE můžeme ještě před GROUPOváním vyloučit všechny řádky, které “nic nepřispěly”.

Zkuste si porovnat výsledky dotazu:

```
SELECT jmeno, castka FROM donori LEFT JOIN priskevky ON donori.id = priskevky.donor
a
```

```
SELECT jmeno, castka FROM donori LEFT JOIN priskevky ON donori.id = priskevky.donor
WHERE castka IS NOT NULL
```

V těch prvních se ještě Anna vyskytuje, v těch druhých už ne.

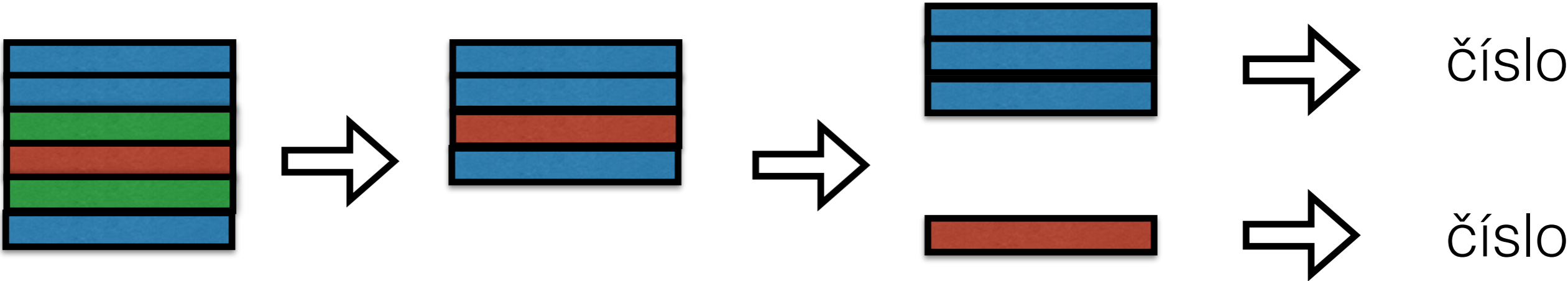
Otázka

Kolik kdo z dárců v průběhu roku přispěl
A zobrazit pouze ty, kteří přispěli alespoň něco.
(Možnost č. 1)

```
SELECT jmeno, SUM(castka)
FROM donori
JOIN prispevky ON donori.id = prispevky.donor
GROUP BY donori.id
HAVING SUM(castka) > 0
```

Pokud chceme v podmínce používat výsledek agregační funkce, nemůžeme ho dát do WHERE, ale do HAVING. Viz diagramy na dalších slajdech

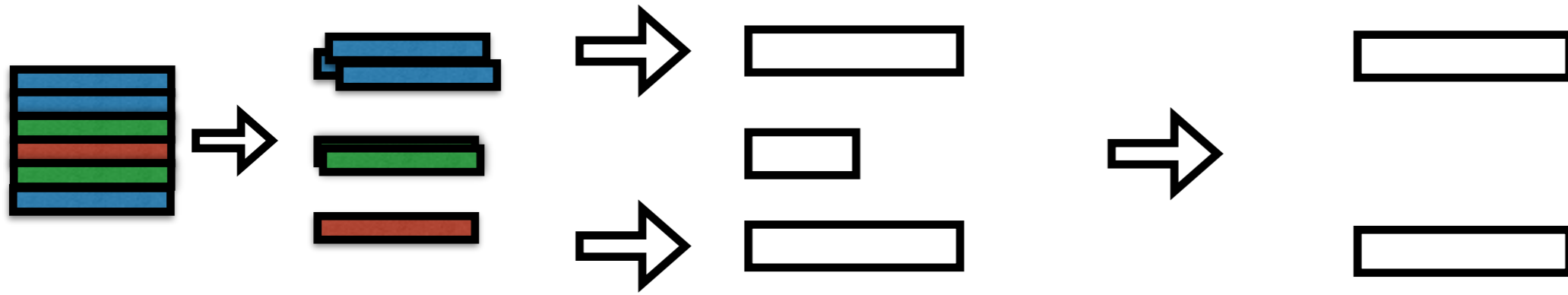
WHERE + GROUP BY + agregace



WHERE

GROUP BY

GROUP BY + agregace + HAVING



GROUP BY

HAVING