

# Kolokvij 2

1. V PB so shranjene naslednje tabele, ki opisujejo delovanje jadranskega kluba:

Jadralec(jid, ime, rating, starost)  
 Coln(cid, ime, dolzina, barva)  
 Rezervacija(#jid, #cid, dan)

Z uporabo a) relacijske algebre in b) jezika SQL poiščite imena jadralcev, mlajših od 20 let, ki so doslej rezervirali vsaj enega izmed modrih čolnov.

ekvistik → morda se ujemaat jid in cid (10+10) \_\_\_\_ točk

a)  $\pi_{\text{jadralec.ime}} \left( \sigma_{\text{starost} < 20} (\text{jadralec}) \bowtie_{\text{jid}} \text{rezervacija} \bowtie_{\text{cid}} \sigma_{\text{barva} = \text{'modra'}} (\text{coln}) \right)$

↳ lahko tudi:  $\pi_{\text{jadralec.ime}} \left( \sigma_{\text{starost} < 20 \wedge \text{barva} = \text{'modra'}} (\text{jadralec} \bowtie_{\text{jid}} \text{rezervacija} \bowtie_{\text{cid}} \text{coln}) \right)$

b) `SELECT DISTINCT j.ime  
 FROM jadralec j JOIN rezervacija r USING(jid) JOIN coln c USING(cid)  
 WHERE starost < 20 AND barva = 'modra';`

2. Imejmo relacije, kot so podane kot v prvi nalogi. (a) v eni vrstici opišite pomen naslednjega izraza v relacijski algebri, (b) poenostavite izraz, če je mogoče in (c) zapišite funkcionalno ekvivalentno poizvedbo v jeziku SQL.

$$\pi_{\text{ime}} \left( \left( \sigma_{\text{rating} < 5} (\text{Jadralec}) \bowtie_{\text{jid}} \text{Rezervacija} \right) \cap \left( \sigma_{\text{starost} > 20} (\text{Jadralec}) \bowtie_{\text{jid}} \text{Rezervacija} \right) \right)$$

(5+5+5) \_\_\_\_ točk

a) Izbere vsa imena jadralcev, ki imajo rating manjši od 5 in starost večjo od 20.

b)  $\pi_{\text{ime}} \left( \sigma_{\text{rating} < 5 \wedge \text{starost} > 20} (\text{Jadralec}) \right)$

c) `SELECT DISTINCT ime  
 FROM jadralec  
 WHERE rating < 5 AND starost > 20;`

3. Relacijska shema iz prve naloge omogoča, da imamo lahko vse čolne poimenovane z istim imenom.
- (a) V eni vrstici pojasnite, kateri koncept relacijskega podatkovnega modela nam to omogoča.
- (b) Minimalno spremenite relacijske sheme iz prve naloge, tako da bo zagotavljala, da so imena vseh čolnov različna in zapišite spremenjene relacijske sheme.
- (c) (Dodatna, neobvezna naloga) Ali bi v SQL lahko isti problem rešili še enostavneje? Kako?

↳ ker imajo id (primarni ključ), pol ni varen a ma drug ime (5+5+5) \_\_\_ točk

a) Zaradi omejitve poimovnega ključa, ki ni čez ime

b) jadralec (jid, ime, rating, starost)

coln (cid, ime, dolzina, barva) ⇒ coln (~~cid~~, ime, dolzina, barva)

rezervacija (#jid, #cid, dan)

rezervacijo (#jid, #ime, dan)

↳ zamenas da je ime primary key

c) Modificiramo ime z UNIQUE

ime VARCHAR(10) UNIQUE

↳ niz z 0-10 znaki

CHAR(10) → niz z samo/hujno 10 znaki

4. Napišite SQL poizvedbo, ki za vsak dan, ko je bil rezerviran vsaj en čoln, izpiše število jadrancev, ki so ta/te čolne rezervirali!

↳ Dan prvi stolpec, prešteto kd ljudi drugi

(10) \_\_\_ točk

SELECT dan, COUNT(DISTINCT(jid))

FROM rezervacija

GROUP BY dan

↳ gledamo iz rezervacije

↳ grupiramo po dnevih, da nam izpiše za vsak dan

↳ število id je, ker to je kol folk je rezerviral

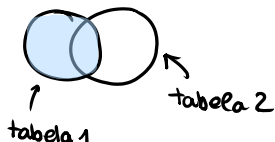
↳ ker je lahko ko rezerviramo več kot 1x na dan in ga mesmemo večkrat šteti

5. V eni vrstici opišite, a) kaj počnemo s standardnim SQL operatorjem LEFT JOIN ... USING (...). Denimo, da uporabljate verzijo SUPB, ki tega operatorja ne implementira. Kako bi (v sintaksi za MariaDB/MySQL) zapisali spodnjo poizvedbo brez operatorja LEFT JOIN?

```
SELECT j.jid, COUNT(r.cid) AS stevilo
FROM jadralec j LEFT JOIN rezervacija r USING (jid)
WHERE j.rating IN (1,3,5,7,9)
GROUP BY j.jid
ORDER BY jid;
```

(15) \_\_\_\_ točk

a) **LEFT JOIN** vrne vse podatke iz leve tabele in vse ki imajo stik v desni, kjer nimajo stika se doda NULL.



**USING** → pove stik

```
b) SELECT j.jid, COUNT(r.cid) AS stevilo
FROM jadralec j JOIN rezervacija r USING(jid)
WHERE j.rating IN (1,3,5,7,9)
GROUP BY j.jid
UNION
```

kjer jadralci imajo stik z rezervacijo

```
SELECT j.jid, 0 AS stevilo
FROM jadralec j
WHERE j.jid NOT IN (SELECT rezervacija.jid FROM rezervacija AND j.rating IN(1,3,5,7,9))
ORDER BY jid
```

kjer jadralci nimajo stika → tem oblogi stevilo 0

natural join → združi in odstrani ponovitve

6. Obkrožite, katere izmed naslednjih vrstic implementirajo naslednjo operacijo: jadralec  $\bowtie$  rezervacija

PAZI! da 2 stolpca zadostujeta

- a) jadralec LEFT JOIN rezervacija **ON(jid=cid)**
- b) jadralec JOIN rezervacija **ON(jid)** → po ON mora biti pogoj → poveži stolpec / ne odstrani ponovitev
- c) jadralec JOIN rezervacija **ON(jadralec.jid=rezervacija.jid)**
- d**) jadralec NATURAL JOIN rezervacija
- e) jadralec LEFT JOIN rezervacija **USING(jid)**
- f**) jadralec JOIN rezervacija USING(jid) → uporabna se za združitve dveh stolpcev (ponavadi z natural join kadar se upenuata več kot 2)
- g**) jadralec INNER JOIN rezervacija USING(jid)
- h) jadralec NATURAL JOIN rezervacija USING(jid)
- i) jadralec NATURAL JOIN rezervacija ON(jid)

INNER JOIN = JOIN



nebi spleh SQL pustiv, natural joinu ne moraš dat pogoj!

(10) \_\_\_\_ točk

7. Izmislite si dve preprosti, smiselno povezani relaciji s skupno najmanj šestimi atributi in vsako z najmanj eno smiselno funkcionalno odvisnostjo. V **eni vrstici** zapišite relaciji ter jima ustrezno označite primarne in tuje ključe. Ne kopirajte relacij s predavanj ali vaj!

→ ker ma stranka lahko samo en hotel, hotel pa več strank  
 (10) \_\_\_ točk  
 stranka (#hid, ime, emso, datum)  
 Hotel (hid, soba, drjava)  
 emso je edinstven, zato je primarni ključ stranke  
 primarni ključ vedno podčrtan

8. Imejmo relacijske sheme jadralnega kluba iz prve naloge, ter poizvedbo:

SELECT j.jid, r.cid, r.dan → kar nima stika, ima vrednost NULL / če bi bil INNER BI BIU samo h) prav!  
 FROM jadralec j LEFT OUTER JOIN rezervacija r USING(jid);

Katere izmed spodnjih vrstic so možen (veljaven) odgovor na zgornjo poizvedbo?

	jid	cid	dan
a)	NULL	NULL	NULL
b)	NULL	NULL	23-3-2017
c)	NULL	102	NULL
d)	NULL	102	23-3-2017
e)	33	NULL	NULL
f)	33	NULL	23-3-2017
g)	44	102	NULL
h)	44	102	23-3-2017

e) 33 NULL NULL : ali jadralec ni nič rezerviral  
 h) 44 102 23-3-2017: ali pa je imel rezervacijo člena na določen dan

al je vse null v teh dveh ali ima oboje vrednost

(10) \_\_\_ točk

nesme bit null zaradi left join (za right je obratno)