

# MariaDB 10.0 / Workbench 6.2

- MariaDB(lastna inštalacija) – **najbolj priporočljivo za vse študente**
  - Strežnik MariaDB 10.0
  - Odjemalec: MySQL Workbench 6.2
  - Povezave na učilnici
- MariaDB in PostgreSQL spletni dostop
  - <http://pb.fri.uni-lj.si> (phpMyAdmin)
  - Uporabniško ime: pb, geslo: pbvaje, baza: vaje

# Primeri osnov poizvedovalnega jezika SQL

```
SELECT *  
FROM coln  
WHERE barva = 'rumena';
```

```
SELECT rating, AVG(starost) AS PovprecnaStarostSkupine  
FROM jadralec  
WHERE starost >= 18 AND EXISTS (SELECT r.cid FROM  
    rezervacija r WHERE r.jid = jadralec.jid)  
GROUP BY rating  
HAVING COUNT(*) > 1;
```

# DML

- Delo nad **obstoječimi** tabelami!
- Povpraševanja
- Dodajanje vrstic
- Brisanje vrstic
- Spreminjanje vrstic

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

# Tabele za vaje

- MariaDB na pb.fri.uni-lj.si jih že ima
  - Shema: vaje (jadralci)
  - Shema: travian (domače naloge)
- MariaDB (samo na lastnih računalnikih!):
  - Z učilnice prenesite datoteko jadralci.sql in travian.sql
  - Odprite jih v MySQL Workbenchu (File->Open SQL Script...)
  - Poženite (samo prvič – torej samo enkrat!!!!)

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

## Primeri tabel

Jadralec:

<u>jid</u>	ime	rating	starost
22	Darko	7	45
29	Borut	1	33
31	Lojze	8	55.5
32	Andrej	8	25.5
58	Rajko	10	35
64	Henrik	7	35
71	Zdravko	10	16
74	Henrik	9	35
85	Anze	3	25.5
95	Bine	3	63.5

Coln:

<u>cid</u>	ime	dolzina	barva
101	Elan	34	modra
102	Elan	34	rdeca
103	Sun Odyssey	37	zelena
104	Bavaria	50	rdeca

Rezervacija:

<u>jid</u>	<u>cid</u>	<u>dan</u>
22	101	2006-10-10
22	102	2006-10-10
22	103	2006-10-08
22	104	2006-10-07
31	102	2006-11-10
31	103	2006-11-06
31	104	2006-11-12
64	101	2006-09-05
64	102	2006-09-08
74	103	2006-09-08

# Osnovni SELECT stavek

**SELECT**  $A_1, A_2, \dots, A_k$

**FROM**  $T_1, T_2, \dots, T_n$

**WHERE**  $P$ ;

- Rezultat SELECT stavka kot začasna tabela!
- SELECT DISTINCT ali ALL:
  - DISTINCT izloči duplikate iz rezultata;
  - privzeta vrednost ALL jih ohrani!

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

# Projekcija

- Poišči (izpiši) šifre in imena vseh jadralcev.

```
SELECT jid, ime  
FROM jadralec;
```

- Poišči barve vseh čolnov.

```
SELECT barva  
FROM coln;
```

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

# Selekcija

- Poišči (izpiši) šifre in imena vseh jadralcev, starejših od 50 let.

```
SELECT jid, ime  
FROM jadralec  
WHERE starost > 50;
```

- Poišči barve vseh čolnov krajših od 40 čevljev.

```
SELECT barva  
FROM coln  
WHERE dolzina < 40;
```



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

## Stik (1)

- Poišči vse pare imen jadralcev in čolnov, kjer je jadralec rezerviral ustrezen čoln.

```
SELECT jadralec.ime, coln.ime
FROM jadralec, rezervacija, coln
WHERE jadralec.jid=rezervacija.jid
      AND rezervacija.cid=coln.cid;
```

ime	ime
Darko	Elan
Darko	Elan
Darko	Sun Odyssey
Darko	Bavaria
Lojze	Elan
Lojze	Sun Odyssey
Lojze	Bavaria
Henrik	Elan
Henrik	Elan
Henrik	Sun Odyssey

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

## Stik (2)

- Poišči vse pare imen jadrancev in čolnov, kjer je jadranec starejši od 50 let rezerviral ustrezen čoln.

```
SELECT jadranec.ime, coln.ime
FROM jadranec, rezervacija, coln
WHERE jadranec.jid=rezervacija.jid AND
      rezervacija.cid=coln.cid AND
      starost > 50;
```

+-----+	+-----+	+-----+
ime	ime	
+-----+	+-----+	+-----+
Lojze	Elan	
Lojze	Sun Odyssey	
Lojze	Bavaria	
+-----+	+-----+	+-----+

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

# SQL

- Poišči vse pare imen jadralcev in čolnov, kjer je jadralec starejši od 50 let rezerviral ustrezen čoln.

Nekatere nerodnosti prejšnje rešitve:

- Imena stolpcev so nejasna - potrebno je preimenovanje
- Nepotrebno pisanje dolgih imen tabel - uporaba aliasov

```
SELECT j.ime AS "Jadralec", c.ime AS "Coln"
FROM jadralec j, rezervacija r, coln c
WHERE j.jid=r.jid AND r.cid=c.cid
      AND j.starost > 50;
```

Jadralec	Coln
Lojze	Elan
Lojze	Sun Odyssey
Lojze	Bavaria

# Komentarji v SQL

- Dve vrsti komentarjev:
  - večvrstični: `/* komentar */`
  - enovrstični:
    - `--` (dva minusa in presledek)
    - `#` (lojtra – samo MariaDB in MySQL)

```
SELECT *           -- Izberi vse
FROM jadralec      /* iz tabele jadralcev */
WHERE starost < 18; # Mladoletni jadralci
```

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

# SQL (eksistenčna kvantifikacija)

- Poišči imena jadralcev, ki so rezervirali **vsaj en** čoln.

```
SELECT DISTINCT j.ime
FROM jadralec j, rezervacija r
WHERE j.jid=r.jid;
```

```
+-----+
| ime    |
+-----+
| Darko  |
| Lojze  |
| Henrik |
+-----+
```

- Univezalna kvantifikacija (rezervirali **vse** čolne) je bistveno bolj zapletena za implementacijo!

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

## SQL (kartezični produkt)

- Poišči imena in ratinge vseh parov jadralcev, ki imajo enak rating.  
 $\rho$ : operator preimenovanja

```
SELECT j1.ime, j2.ime, j1.rating
FROM jadralec j1, jadralec j2
WHERE j1.rating = j2.rating
ORDER BY j1.rating;
```

ime	ime	rating
Borut	Borut	1
Anze	Anze	3
Bine	Anze	3
Anze	Bine	3
Bine	Bine	3
Darko	Henrik	7
Darko	Darko	7
Henrik	Henrik	7
Henrik	Darko	7
Lojze	Lojze	8
Andrej	Lojze	8
Lojze	Andrej	8
Andrej	Andrej	8
Henrik	Henrik	9
Rajko	Zdravko	10
Zdravko	Zdravko	10
Rajko	Rajko	10
Zdravko	Rajko	10

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

## SQL (kartezični produkt)

- Poišči imena in ratinge vseh parov jadralcev, ki imajo enak rating.  
 $\rho$ : operator preimenovanja

```
SELECT DISTINCT j1.ime, j2.ime,  
                j1.rating  
FROM jadralec j1, jadralec j2  
WHERE j1.rating = j2.rating  
ORDER BY j1.rating;
```

ime	ime	rating
Borut	Borut	1
Anze	Anze	3
Bine	Anze	3
Anze	Bine	3
Bine	Bine	3
Darko	Henrik	7
Darko	Darko	7
Henrik	Henrik	7
Henrik	Darko	7
Lojze	Lojze	8
Andrej	Lojze	8
Lojze	Andrej	8
Andrej	Andrej	8
Henrik	Henrik	9
Rajko	Zdravko	10
Zdravko	Zdravko	10
Rajko	Rajko	10
Zdravko	Rajko	10

Zakaj ni razlike od prej?

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

## SQL (kartezični produkt)

- Poišči imena in ratinge vseh parov jadralcev, ki imajo enak rating.  
 $\rho$ : operator preimenovanja

ime	ime	rating
Bine	Anze	3
Anze	Bine	3
Darko	Henrik	7
Henrik	Darko	7
Lojze	Andrej	8
Andrej	Lojze	8
Zdravko	Rajko	10
Rajko	Zdravko	10

Kaj še ni v redu?

```
SELECT DISTINCT j1.ime, j2.ime,  
                j1.rating  
FROM jadralec j1, jadralec j2  
WHERE j1.rating = j2.rating AND  
      j1.ime <> j2.ime -- bolje: jid  
ORDER BY j1.rating;
```



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

## SQL (kartezični produkt)

- Poišči imena in ratinge vseh parov jadralcev, ki imajo enak rating.  
 $\rho$ : operator preimenovanja

ime	ime	rating
Anze	Bine	3
Darko	Henrik	7
Lojze	Andrej	8
Rajko	Zdravko	10

```
SELECT DISTINCT j1.ime, j2.ime,  
                j1.rating  
FROM jadralec j1, jadralec j2  
WHERE j1.rating = j2.rating AND  
      j1.jid < j2.jid  
ORDER BY j1.rating;
```

# Operatorji v SQL (WHERE vrstica)

- =
- != ali <>
- <=, >=, <, >
- BETWEEN x AND y:  $x \leq \text{vrednost} \leq y$
- AND, OR, NOT
- LIKE: približna primerjava nizov znakov
- SIMILAR TO vzorec [ESCAPE znak]:  
regularni izrazi (SQL:1999)
- IS [NOT] NULL (atribut označen kot nedefiniran)

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

# SQL: operator LIKE

- Poišči starost jadralcev, katerih imena se začnejo na B in imajo najmanj 5 črk.

```
+-----+
| starost|
+-----+
|      33 |
+-----+
```

```
SELECT j.starost
FROM jadralec j
WHERE j.ime LIKE 'B____%'; /* 4 podcrtaji */
```

- Znak '\_' ustreza natanko eni poljubni črki
- Znak '%' ustreza nič ali več poljubnim črkam

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

# SQL: regularni izrazi

- Poišči starost jadralcev, katerih imena se začnejo na B in imajo najmanj 5 črk.

```
SELECT j.starost
FROM jadralec j
WHERE j.ime REGEXP '^b.....*$';           -- MariaDB, MySQL
-- ali (REGEXP = RLIKE)
WHERE j.ime RLIKE '^b[a-z]{4}[a-z]*$';       -- MariaDB, MySQL
-- ali
WHERE REGEXP_LIKE (j.ime, '^B[a-z]{4}[a-z]*$'); -- Oracle
```

- Znak '^' označuje začetek, '\$' pa konec niza (sicer se išče poljuben podniz)
- Znak '.' (pika) ustreza natanko enemu poljubnemu znaku
- [a-z] je katera koli črka med 'a' in 'z'
- Znak '\*' pomeni nič ali več ponovitev predhodnega znaka
- SQL:1999: operator SIMILAR TO z regularnimi izrazi (redko implementirano)

# Operatorji za delo z množicami

- UNION: unija  $\cup$   
UNION ALL: unija  $\cup$  s ponavljanjem elementov
- INTERSECT: presek  $\cap$
- MINUS ali EXCEPT: razlika  $-$
- IN, NOT IN (tabela): pripadnost  $\in$  in  $\notin$
- ALL, ANY: kvantifikatorja  $\forall$  in  $\exists$
- EXISTS, NOT EXISTS (tabela): (ne)praznost množice
- UNIQUE (tabela): enoličnost elementov v tabeli
- Operatorji IN, NOT IN in EXISTS so osnova za gnezdenje poizvedb

# SQL: operatorji za delo z množicami

- Unija: UNION, UNION ALL (ohrani duplikate)
- Presek: INTERSECT (MariaDB ne podpira)
- Razlika: MINUS ali EXCEPT (MariaDB ne podpira)
- Sintaksa:  
SELECT ...  
<OPERATOR>  
SELECT ...;
- Kompatibilnost tabel (ali rezultatov SELECT stavka):  
isto število stolpcev, istoležni stolpci istega tipa

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

## SQL: vsebovanost elementa

- Poišči imena jadralcev z ratingi 1, 3 ali 7.

```
SELECT ime
FROM jadralec
WHERE (rating = 1) OR
      (rating = 3) OR
      (rating = 7);
```

```
+-----+
| ime    |
+-----+
| Darko  |
| Borut  |
| Henrik |
| Anze   |
| Bine   |
+-----+
```

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

## SQL: vsebovanost elementa

- Poišči imena jadralcev, z ratingi 1, 3 ali 7.

```
SELECT ime
FROM jadralec
WHERE rating IN (1,3,7);
```

Množica v SQL. Namesto seznama imamo lahko tudi rezultat neke poizvedbe.

```
+-----+
| ime    |
+-----+
| Darko  |
| Borut  |
| Henrik |
| Anze   |
| Bine   |
+-----+
```



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

## SQL: unija množic

- Poišči imena jadralcev, ki so rezervirali rdeč ali zelen čoln.

$$\pi_{jadralec.ime}(\sigma_{barva=rdeca}(\text{Coln}) \mid \times_{cid} \mid \text{Rezervacija} \mid \times_{jid} \mid \text{Jadralec}) \cup$$

$$\pi_{jadralec.ime}(\sigma_{barva=zelen}(\text{Coln}) \mid \times_{cid} \mid \text{Rezervacija} \mid \times_{jid} \mid \text{Jadralec})$$

```
SELECT DISTINCT j.ime
FROM jadralec j, rezervacija r, coln c
WHERE j.jid=r.jid AND r.cid=c.cid AND
      c.barva='rdeca'
```

**UNION**

```
SELECT DISTINCT j.ime
FROM jadralec j, rezervacija r, coln c
WHERE j.jid=r.jid AND r.cid=c.cid AND
      c.barva='zelen';
```

```
+-----+
| ime    |
+-----+
| Darko  |
| Lojze  |
| Henrik |
+-----+
```

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

## SQL: presek množic

- Poišči imena jadralcev, ki so rezervirali rdeč IN zelen čoln.

$$\pi_{jadralec.ime}(\sigma_{barva=rdeca}(\text{Coln}) \mid \times_{cid} \mid \text{Rezervacija} \mid \times_{jid} \mid \text{Jadralec}) \cap$$

$$\pi_{jadralec.ime}(\sigma_{barva=zelen}(\text{Coln}) \mid \times_{cid} \mid \text{Rezervacija} \mid \times_{jid} \mid \text{Jadralec})$$

```
SELECT DISTINCT j.ime
FROM jadralec j, rezervacija r, coln c
WHERE j.jid=r.jid AND r.cid=c.cid AND
      c.barva='rdeca'
```

**INTERSECT**

```
SELECT DISTINCT j.ime
FROM jadralec j, rezervacija r, coln c
WHERE j.jid=r.jid AND r.cid=c.cid AND
      c.barva='zelen';
```

```
+-----+
| ime    |
+-----+
| Darko  |
| Lojze  |
| Henrik |
+-----+
```

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

## Presek z uporabo gnezdenja

- Poišči imena jadralcev, ki so rezervirali rdeč IN zelen čoln.

$$\pi_{jadralec.ime}(\sigma_{barva=rdeca}(Coln) \mid \times_{cid} \mid Rezervacija \mid \times_{jid} \mid Jadralec) \cap$$

$$\pi_{jadralec.ime}(\sigma_{barva=zelen}(Coln) \mid \times_{cid} \mid Rezervacija \mid \times_{jid} \mid Jadralec)$$

```

SELECT DISTINCT j.ime          -- Prva mnozica
FROM jadralec j, rezervacija r, coln c
WHERE j.jid=r.jid AND r.cid=c.cid AND
      c.barva='rdeca'
AND j.ime IN (                 -- Druga mnozica
  SELECT DISTINCT j.ime
  FROM jadralec j, rezervacija r, coln c
  WHERE j.jid=r.jid AND r.cid=c.cid
  AND c.barva='zelena');

```

ime
Darko
Lojze
Henrik

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

## SQL: razlika množic

- Poišči imena jadralcev, ki so rezervirali rdeč čoln vendar nikoli zelenega.

$$\pi_{jadralec.ime}(\sigma_{barva=rdeca}(\text{Coln}) \mid \times_{cid} \mid \text{Rezervacija} \mid \times_{jid} \mid \text{Jadralec}) -$$

$$\pi_{jadralec.ime}(\sigma_{barva=zeleni}(\text{Coln}) \mid \times_{cid} \mid \text{Rezervacija} \mid \times_{jid} \mid \text{Jadralec})$$

```
SELECT DISTINCT j.ime
FROM jadralec j, rezervacija r, coln c
WHERE j.jid=r.jid AND r.cid=c.cid AND
      c.barva='rdeca'
```

**MINUS**                    -- ali EXCEPT

```
SELECT DISTINCT j.ime
FROM jadralec j, rezervacija r, coln c
WHERE j.jid=r.jid AND r.cid=c.cid AND
      c.barva='zelena';
```

```
+-----+
| ime   |
+-----+
+-----+
```

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

## Razlika množic z gnezdenjem

- Poišči imena jadralcev, ki so rezervirali rdeč čoln vendar nikoli zelenega.

$$\pi_{jadralec.ime}(\sigma_{barva=rdeca}(\text{Coln}) \mid \times_{cid} \mid \text{Rezervacija} \mid \times_{jid} \mid \text{Jadralec}) -$$

$$\pi_{jadralec.ime}(\sigma_{barva=zelenca}(\text{Coln}) \mid \times_{cid} \mid \text{Rezervacija} \mid \times_{jid} \mid \text{Jadralec})$$

```
SELECT DISTINCT j.ime
FROM jadralec j, rezervacija r, coln c
WHERE j.jid=r.jid AND r.cid=c.cid AND
      c.barva='rdeca'
```

```
+-----+
|  ime  |
+-----+
+-----+
```

```
AND j.ime NOT IN (
  SELECT DISTINCT j.ime
  FROM jadralec j, rezervacija r, coln c
  WHERE j.jid=r.jid AND r.cid=c.cid AND
        c.barva='zelenca');
```

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

# SQL (gnezdene poizvedbe)

- Poišči imena jadralcev, ki so rezervirali čoln s šifro 103.

$$\pi_{ime} \left( \sigma_{jid \in \{ \pi_{jid} (\sigma_{cid=103} (\text{Rezervacija}) \}} (\text{Jadralec}) \right)$$

```
SELECT ime
FROM jadralec
WHERE jid IN
  (SELECT jid
   FROM rezervacija -- Mnozica rezervacij
   WHERE cid=103); -- colna 103
```

```
+-----+
| ime    |
+-----+
| Darko  |
| Lojze  |
| Henrik |
+-----+
```

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

# Alternativna sintaksa za stik: operator JOIN

```
SELECT jadralec.ime
FROM jadralec, rezervacija
WHERE jadralec.jid=rezervacija.jid AND rezervacija.cid = 103;
```

```
SELECT jadralec.ime      -- Naravni stik
FROM jadralec NATURAL JOIN rezervacija
WHERE rezervacija.cid = 103;
```

```
SELECT jadralec.ime      -- Ekvistik
FROM jadralec JOIN rezervacija USING(jid)
WHERE rezervacija.cid = 103;
```

```
SELECT jadralec.ime      -- Pogojni stik
FROM jadralec JOIN rezervacija ON (rezervacija.jid = jadralec.jid)
WHERE rezervacija.cid = 103;
```

```
SELECT jadralec.ime      -- Pogojni stik s sestavljenim pogojem
FROM jadralec JOIN rezervacija ON (rezervacija.jid = jadralec.jid AND
rezervacija.cid = 103);
```

**Obstaja še več  
različic, niso pa  
vedno vse  
implementirane,  
zato pozor!**

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

## SQL (Korelirane gnezdene poizvedbe)

- Poišči imena jadralcev, ki so rezervirali čoln številka 103.

```
SELECT j.ime
FROM jadralec j
WHERE EXISTS
    (SELECT *
     FROM rezervacija r
     WHERE r.cid = 103 AND
           r.jid = j.jid);
```

-- Neprazna mnozica  
-- rezervacij colna  
-- 103 za vsakega  
-- jadralca  
-- posebej

```
+-----+
| ime    |
+-----+
| Darko  |
| Lojze  |
| Henrik |
+-----+
```

- Problem: neučinkovitost, zato se jim izognemo, kadar je le mogoče.



# Gnezdenje v FROM vrstici

- Pozvedbe lahko gnezdimo tudi v FROM vrstici, pri čemer se rezultat poizvedbe naprej obravnava kot (začasna) tabela in ga je zato potrebno ustrezno poimenovati.
- Vsi atributi v SELECT vrstici gnezdene poizvedbe morajo imeti eksplicitno določena imena.

```
SELECT stik.*
FROM (SELECT j.jid, r.cid, j.starost
      FROM jadralec j, rezervacija r
      WHERE j.jid = r.jid) AS stik
WHERE stik.starost > 40;
```

**AS** opcijsko, zaradi preglednosti

Začasna tabela stik ----->

Rdeče vrstice bodo izločene

+-----+-----+-----+			
jid   cid   starost			
+-----+-----+-----+			
22   101	45		
22   102	45		
22   103	45		
22   104	45		
31   102	55.5		
31   103	55.5		
31   104	55.5		
64   101	35		
64   102	35		
74   103	35		
+-----+-----+-----+			

# Urejanje izpisa SELECT stavka

- SELECT stavku dodamo vrstico:  
`ORDER BY ime_atributa [ASC ali DESC]`  
ali  
`ORDER BY številka_atributa [ASC ali DESC]`  
ali (za več atributov)  
`ORDER BY ime1 ASC, ime2 DESC, ...`
- Lahko urejamo tudi po izrazu ali na novo izračunanem atributu, ki ga ustrezno poimenujemo.
- Urejanje pri množiških operacijah ali gnezdenju ni smiselno, zato ni dovoljeno.

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

## Urejanje izpisa

- Izpiši imena jadrancev urejena po količniku med ratingom in starostjo.

ROUND(stevilo, mest) zaokroži rezultat na dano število mest.

```
SELECT ime, ROUND(rating/starost,2)
FROM jadralec;
```

ime	ROUND(rating/starost,2)
Darko	0.16
Borut	0.03
Lojze	0.14
Andrej	0.31
Rajko	0.29
Henrik	0.20
Zdravko	0.62
Henrik	0.26
Anze	0.12
Bine	0.05

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

## Urejanje izpisa

- Izpiši imena jadrancev urejena po količniku med ratingom in starostjo.

ROUND(stevilo, mest) zaokroži rezultat na dano število mest.

```
SELECT ime, ROUND(rating/starost,2)
FROM jadralec
ORDER BY 2 DESC;
```

ime	ROUND(rating/starost,2)
Zdravko	0.62
Andrej	0.31
Rajko	0.29
Henrik	0.26
Henrik	0.20
Darko	0.16
Lojze	0.14
Anze	0.12
Bine	0.05
Borut	0.03

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

## Urejanje izpisa

- Izpiši imena jadrancev urejena po količniku med ratingom in starostjo.

```
SELECT ime, ROUND(rating/starost,2)
        AS kolicnik
FROM jadralec
ORDER BY kolicnik DESC;
```

ime	kolicnik
Zdravko	0.62
Andrej	0.31
Rajko	0.29
Henrik	0.26
Henrik	0.20
Darko	0.16
Lojze	0.14
Anze	0.12
Bine	0.05
Borut	0.03

# Kvantifikatorji v SQL

- Preveri veljavnost kvantificiranega ( $\exists, \forall$ ) logičnega pogoja nad celotno množico skalarnih (posameznih) vrednosti atributa
- Kvantifikatorja:
  - ANY (ali SOME): eksistenčni
  - ALL: univerzalni
- Sintaksa (v WHERE vrstici):  
`WHERE atribut operator ANY ali ALL (množica)`  
npr.  
`WHERE x < ANY (SELECT ...);`

# Pomen kvantifikatorjev

**WHERE  $x < \text{ANY}(\text{SELECT } y \dots)$ ;**  $\exists y : x < y$   
**WHERE  $x = \text{ANY}(\text{SELECT } y \dots)$ ;**  $x \in \{y \mid \dots\}$  Isto kot IN  
**WHERE  $x <> \text{ANY}(\text{SELECT } y \dots)$ ;**  $\exists y : x \neq y$

**WHERE  $x < \text{ALL}(\text{SELECT } y \dots)$ ;**  $\forall y : x < y$   
**WHERE  $x = \text{ALL}(\text{SELECT } y \dots)$ ;**  $\forall y : x = y$   
**WHERE  $x <> \text{ALL}(\text{SELECT } y \dots)$ ;**  $\forall y : x \neq y$  Isto kot NOT IN

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

## Kvantifikatorji v SQL

- Poišči šifre jadralcev, ki imajo ratinge višje kot (vsaj en) jadralec z imenom Henrik. Opomba: Henrika sta dva!

```
SELECT j.jid
FROM jadralec j
WHERE j.rating > ANY
  (SELECT j2.rating
   FROM jadralec j2
   WHERE j2.ime='Henrik');
```

```
+-----+
|  jid  |
+-----+
|   31  |
|   32  |
|   58  |
|   71  |
|   74  |
+-----+
```

- Rating mora biti višji od vsaj enega Henrika!



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

## Kvantifikatorji v SQL

- Poišči šifre jadralcev, ki imajo ratinge višje kot (vsi) jadralci z imenom Henrik.

```
SELECT j.jid
FROM jadralec j
WHERE j.rating > ALL
  (SELECT j2.rating
   FROM jadralec j2
   WHERE j2.ime='Henrik');
```

```
+-----+
|  jid  |
+-----+
|   58  |
|   71  |
+-----+
```

- Rating mora biti višji od **vseh** Henrikov!

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

## Kvantifikatorji v SQL

- Poišči šifre jadralcev, ki imajo najvišji rating!

Opomba: lahko jih je več.

```
SELECT j.jid
FROM jadralec j
WHERE j.rating >= ALL
      (SELECT j2.rating
       FROM jadralec j2);
```

```
+-----+
|  jid  |
+-----+
|   58  |
|   71  |
+-----+
```

- Rating mora biti višji ali enak od **vseh ratingov**, torej tudi od lastnega!

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

## Kvantifikatorji v SQL

- Poišči šifre jadralcev, ki nimajo najnižjega ratinga!  
Opomba: lahko jih je več.

```
SELECT j.jid
FROM jadralec j
WHERE j.rating > ANY
      (SELECT j2.rating
       FROM jadralec j2);
```

```
+-----+
|  jid  |
+-----+
|   22  |
|   31  |
|   32  |
|   58  |
|   64  |
|   71  |
|   74  |
|   85  |
|   95  |
+-----+
```

- Rating mora biti strogo višji od **vsaj enega ratinga**.

# Vaje

Napišite poizvedbe, ki rešijo spodnje naloge:

1. Izpišite imena jadrancev s sodimi ratingi.
2. Ugotovite, ali imata kakšna dva čolna enako ime.
3. Izpišite imena čolnov daljših od 35 čevljev, ki so jih rezervirali jadranci mlajši od 35 let.
4. Ugotovite, ali imajo vsi jadranci različna imena.
5. Izpišite imena jadrancev, ki so v koledarskem poletju 2006 rezervirali čoln, katerega ime vsebuje sonce (sun).
6. Za vse jadrance izpišite podatke o njihovih rezervacijah. Za tiste, ki še niso nič rezervirali, naj bodo polja o rezervacijah prazna.

# Vaje

Napišite poizvedbe, ki rešijo spodnje naloge:

1. Izpišite vse podatke naselij alianse „Nwm“.
2. Poiščite vsa imena alians, ki imajo naselja samo v severno-vzhodem kvadrantu ( $x > 0$ ,  $y > 0$ ).
3. Poiščite imena vseh naselji, ki imajo v imenu številko naselja v obliki „[xy]“ ali „[x.y]“, kjer sta x in y števki/cifri.
4. Poiščite vsa naselja, ki so od koordinate (30, 20) oddaljena manj kot 100 enot.
5. Izpiši tiste igralce plemena „Galci“, ki imajo naselje z več kot 1000 prebivalci.

# Vaje: množice in gnezdenje v SQL

Z uporabo gnezdenja in množičkih operatorjev napišite poizvedbe, ki rešijo spodnje naloge:

1. Izpišite šifre čolnov daljših od 35 čevljev, ki so jih rezervirali jadralci mlajši od 40 let.
2. Za vse jadralce izpišite podatke o njihovih rezervacijah. Za tiste, ki še niso nič rezervirali, naj bodo polja o rezervacijah prazna. Rešite z uporabo zunanega stika in izpis uredite po `jid` !
3. Za vse jadralce izpišite podatke o njihovih rezervacijah. Za tiste, ki še niso nič rezervirali, naj bodo polja o rezervacijah prazna. Rešite brez uporabe zunanega stika in izpis uredite po `jid` !
4. Preverite, ali pri prejšnji nalogi dobite enak rezultat, kot pri uporabi zunanega stika.