

# Structured Query Language - SQL

- Rezultat projektov v IBM (1974-77)
- Vsak proizvajalec ga po svoje razširja
- Standardi:
  - SQL-87 (1986, 1987): ANSI SQL
  - SQL-89 (1989): ANSI SQL, FIPS popravki
  - **SQL-92 (1992): SQL2, ANSI/ISO SQL**
  - SQL:1999 (1999): SQL3, objekti, rekurzija, dogodki, regularni izrazi
  - SQL:2003 (2003): podpora XML, avtomatsko generiranje polj, delo s sekvencami
  - SQL:2006, SQL:2008: dodatna podpora delu z XML, integracija XQuery, drugi manjši popravki
- V praksi: **ni 100% podpore standardom**

# Structured Query Language - SQL

- SQL
  - Beginning SQL. Paul Wilton and John W. Colby. Wrox, 2005.
- SQL in relacijska algebra, teorija o PB
  - R. Ramakrishnan, J. Gehrke: Database Management Systems, 3. izdaja, McGraw-Hill, 2002

# SQL 92

- Data definition language (DDL)
- Data manipulation language (DML)
- Varnost
- Transakcije
- Client / server podpora
- Embedded/dynamic SQL

# DML

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

# MariaDB 10.2 / Workbench 6.3

- MariaDB(lastna inštalacija) – **najbolj priporočljivo za vse študente**
  - Strežnik MariaDB 10.3
  - Odjemalec: MySQL Workbench 8.0
  - Povezave na učilnici (snov prejšnjega tedna)
- MariaDB in PostgreSQL spletni dostop
  - <http://pb.fri.uni-lj.si> (phpMyAdmin)
  - Uporabniško ime: **pb**, Geslo: **pbvaje**, Baza: **vaje**

```
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...)
  - Glej snov prejšnjega tedna.

# 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;
```

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



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

## Primeri tabel

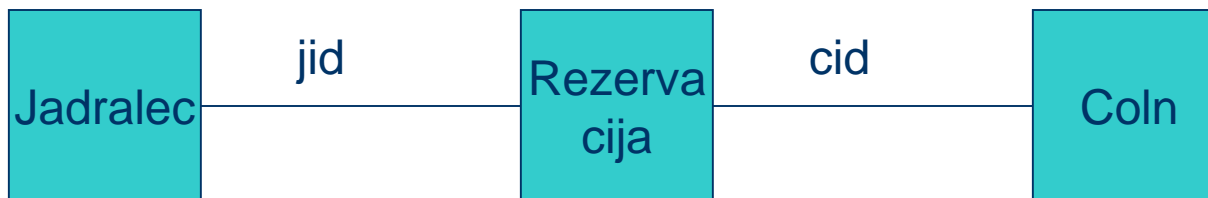
Tabele lahko opišemo tudi z relacijsko shemo:

```
Jadralec(jid, ime, rating, starost)
```

```
Coln(cid, ime, dolzina, barva)
```

```
Rezervacija(#jid, #cid, dan)
```

- Pomen in povezava relacij:



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

## Primeri tabel

V relacijsko shemo lahko dodamo tudi domene:

```
Jadralec(jid:int, ime:varchar(20),  
         rating:int, starost:real)
```

```
Coln(cid:int, ime:varchar(20),  
     dolzina:real, barva:varchar(20))
```

```
Rezervacija(#jid:int, #cid:int,  
            dan:date)
```

Na vajah si bomo pomagali z obliko zapisa, ki je zgornjem desnem kotu.

# Osnovni SELECT stavek

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

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

**WHERE**  $P;$

- Rezultat SELECT stavka je 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 DISTINCT 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 dolžine med 35 in 40 čevljev.

```
SELECT DISTINCT barva  
FROM coln  
WHERE dolzina >= 35  
AND dolzina <= 40;
```

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

## Stik (1)

- Poišči vse čolne, ki jih je rezerviral Lojze.

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

ime	ime
Lojze	Elan
Lojze	Sun Odyssey
Lojze	Bavaria

Pogoj `jadralec.ime = 'Lojze'` je primer **slabe** prakse!  
Težava, ko imamo več jadralcev z imenom Lojze.

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

## Stik (2)

- 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 (3)

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

```
SELECT jadralec.ime, coln.ime
FROM jadralec, rezervacija, coln
WHERE jadralec.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 – Preimenovanja in aliasi

- Poišči vse pare imen jadrancev 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  
s katerim  
lahko komentirate  
blok skripte `*/`
  - enovrstični: `--` ali `#`
    - `--` (dva minusa in presledek)
    - `#` (lojtra – samo MariaDB in MySQL)

# Komentarji v SQL - Primer

# Vaja 1

```
SELECT *           -- Izberi vse stolpce
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

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

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

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

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

# SQL

- Poišči imena jadralcev, ki so rezervirali rdeč čoln.

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

- Pisanje znakovnih nizov v narekovajih.
- Zakaj je potreben DISTINCT?

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.

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

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.

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

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.

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 ASC;
```

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 ASC;
```

# SQL: osnovni SELECT stavek (ponovitve)

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

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

**WHERE**  $P$ ;

- Rezultat **SELECT** stavka si lahko predstavljamo kot začasno tabelo, ki jo izpišemo, ali z njo počnemo kaj drugega
- **SELECT DISTINCT**: izloči duplikate iz rezultata

# 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)

# SQL

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

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

```
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' OR c.barva='zelena');
```

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

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

# SQL

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

```
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' AND c.barva='zelena');
```

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

- Rezultat – prazna množica???
- Kje je napaka?
- Reševanje s pomočjo množic (prihodnjič)



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

## Vaje: osnovni SQL

Napišite poizvedbe, ki rešijo spodnje naloge:

1. Izpišite imena jadrancev s sodimi ratingi.
2. Ugotovite, ali imata dva čolna enako ime.
3. Izpišite imena jadrancev z lihimi ratingi.
4. Izpišite imena čolnov daljših od 35 čevljev, ki so jih rezervirali jadranci stari 35 let ali manj.
5. Za vse jadrance, ki so rezervirali čoln Bavaria ali Sun Odyssey, izpišite jadrалčevo ime in datum rezervacije.
6. Ugotovite, ali imajo vsi jadranci različna imena.
7. Izpišite imena jadrancev, ki so v koledarskem poletju 2006 rezervirali čoln, katerega ime vsebuje sonce (sun).

```
Aliansa(aid, alliance),           Pleme(tid, tribe)
Igralec(pid, player, #tid, #aid)
Naselje(id, vid, village, x, y, population, #pid)
```

## Vaje: osnovni SQL

Napišite poizvedbe, ki rešijo spodnje naloge:

1. Izpišite vse alianse, ki vsebujejo 'mgp'.
2. Izpišite vsa možna plemena;
3. Izpišite vsa imena igralcev, ki imajo vsaj eno naselje z populacijo 1000 ali več.
4. Izpišite imena vseh naselji igralca Ronin.
5. Izpišite vsa imena igralcev, ki se začnejo z nizom 'moj'.
6. Za igralce iz naloge 5. izpišite še ime pripadajočega plemena.
7. Izpišite vse podatke naselji, ki so v I. Kvadrantu ( $x \geq 0$ ,  $y \geq 0$ ) in se njihova imena končajo na samoglasnik ter imajo populacijo 750 ali več.