

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**.

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

Deljenje v SQL

- Poišči imena jadralcev, ki so rezervirali **vse** čolne.
- Tipična naloga za deljenje (univerzalna kvantifikacija)

```
SELECT j.ime
FROM jadralec j
WHERE NOT EXISTS
  (SELECT c.cid
   FROM coln c
   MINUS
   SELECT r.cid
   FROM rezervacija r
   WHERE r.jid = j.jid);
```

$$\Pi_{ime}(\Pi_{jid,cid}(Rezervacija)/\Pi_{cid}(Coln) \bowtie_{jid} Jadralec)$$

```
-- Vsi - Rezervirani = prazna mnozica
-- Vsi colni
```

```
-- Rezervirani colni
-- za vsakega jadralca
-- posebej (korelirana)
```

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

Skupinski operatorji v SQL

- Običajni operatorji delujejo nad posameznimi vrsticami kartezičnega produkta
- Skupinski operatorji in funkcije delujejo nad skupinami (množicami), torej nad več vrsticami istočasno
- Rezultat (izračunana vrednost) skupinskega operatorja postane skupinski atribut, ki ga ne smemo mešati z navadnimi atributi

Skupinski operatorji

- Sintaksa:
`OPERATOR ([DISTINCT] ime_atributa)`
- COUNT(): prešteje [različne] vrstice
- SUM(): sešteje [različne] vrednosti
- AVG (): povprečje [različnih]
- MIN(): minimum
- MAX(): maksimum

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

Štetje (COUNT)

- Preštej, koliko je vseh jadralcev!

```
SELECT COUNT(*)  -- prešteje število vrstic
FROM jadralec;   -- v tabeli jadralcev
```

```
+-----+
| COUNT (*) |
+-----+
|          10 |
+-----+
```

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

Štetje (COUNT)

- Preštej, koliko je jadralcev z različnimi imeni!

```
SELECT COUNT(DISTINCT ime) -- prešteje različne vrednosti
FROM jadralec;             -- atributa ime v tabeli jadralcev
```

Tipična uporaba operatorja COUNT:

- COUNT(*)
- COUNT(DISTINCT ime_atributa)

```
+-----+
| COUNT(DISTINCT ime) |
+-----+
|                      9 |
+-----+
```

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

Povprečje (AVG)

- Izračunaj povprečno starost jadralcev!

```
SELECT AVG(starost)
FROM jadralec;
```

```
-- povpreči vrednosti atributa
-- starost v tabeli jadralcev
```

```
+-----+
| AVG(starost) |
+-----+
|           36.9 |
+-----+
```

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

Povprečje (AVG)

- Izračunaj povprečno starost jadralcev z ratingom 10!

```
SELECT AVG(starost)
FROM jadralec
WHERE rating = 10;
```

```
-- povpreči vrednosti atributa
-- starost v tabeli jadralcev,
-- vendar le za tiste z ratingom 10
```

```
+-----+
| AVG(starost) |
+-----+
|           25.5 |
+-----+
```

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

Minimum (MIN) in maksimum (MAX)

- Poišči minimalno in maksimalno starost jadralcev!

```
SELECT MIN(starost)
FROM jadralec;
```

```
+-----+
| MIN(starost) |
+-----+
|           16 |
+-----+
```

```
SELECT MAX(starost)
FROM jadralec;
```

```
+-----+
| MAX(starost) |
+-----+
|          63.5 |
+-----+
```

Skupinski operatorji v osnovnem SELECT stavku

- Lahko nastopajo v SELECT, WHERE ali ORDER BY vrstici
- V SELECT ali WHERE vrstici se v dani poizvedbi lahko nahajajo **samo navadni** ali **samo skupinski** atributi (ne smemo jih mešati)
- V primeru, da potrebujemo obojne attribute, uporabimo gnezdene poizvedbe

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

Skupinski operatorji

- Poišči imena in starost najstarejših jadralcev!

```
SELECT ime, MAX(starost)  
FROM jadralec;
```

```
SELECT ime, starost  
FROM jadralec  
WHERE starost = MAX(starost);
```

```
SELECT ime, starost  
FROM jadralec  
WHERE starost = ( SELECT MAX(starost)  
                  FROM jadralec );
```

+	-----	+	-----	+
	ime		starost	
+	-----	+	-----	+
	Bine		63.5	
+	-----	+	-----	+

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

Skupinski operatorji namesto kvantifikatorjev

- 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)

Skupinski operatorji namesto kvantifikatorjev

- Poišči šifre jadralcev, ki imajo najvišji rating!
Opomba: lahko jih je več.

```
SELECT jid
FROM jadralec
WHERE rating = (SELECT MAX(rating)
                FROM jadralec);
```

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

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

Skupinski operatorji namesto kvantifikatorjev

- 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**.

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

Skupinski operatorji namesto kvantifikatorjev

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

```
SELECT jid
FROM jadralec
WHERE rating > (SELECT MIN(rating)
                FROM jadralec);
```

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

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

Skupinski operatorji

- Poišči imena jadralcev, starejših od najstarejšega jadralca z ratingom 10!

```
SELECT ime
FROM jadralec
WHERE starost > (SELECT MAX(starost)
                 FROM jadralec
                 WHERE rating = 10);
```

```
+-----+
| ime   |
+-----+
| Darko |
| Lojze |
| Bine  |
+-----+
```

Delo nad skupinami

- Skupinski operatorji znotraj ene poizvedbe delujejo le nad eno skupino (množico)
- Če želimo istočasno dobiti rezultate nad več skupinami moramo razširiti SELECT stavek z novimi vrsticami, ki omogočajo uporabo skupinskih operatorjev nad skupinami vrstic
- Primer naloge: za vsak rating v tabeli jadralcev izpiši starost najmlajšega jadralca s tem ratingom.

Razširjeni SELECT stavek

[DISTINCT | ALL]

SELECT A_1, \dots, A_k -- projekcija

FROM T_1, T_2, \dots, T_n -- kartezicni produkt

WHERE P_1 -- selekcija po vrsticah

GROUP BY A_1, A_2, \dots, A_m -- grupiranje po atributih

HAVING P_2 -- selekcija po skupinah

ORDER BY A_i, \dots, A_j -- urejanje po atributih

[ASC|DESC]

Razširjeni SELECT stavek

- GROUP BY x : razdeli množico iz SELECT-FROM-WHERE na podmnožice glede na enake vrednosti atributa x
- GROUP BY x_1, x_2, \dots, x_n : skupine imajo enake vrednosti vseh n atributov (torej je število možnih – vendar ne nujno dejanskih - skupin enako moči kartezičnega produkta vseh n atributov)
- Vsak osnovni atribut, ki se nahaja v SELECT vrstici, se mora nahajati tudi v GROUP BY vrstici
- S pogojem HAVING P ohranimo samo tiste skupine, ki izpolnjujejo pogoj P
- V HAVING vrstici se lahko nahajajo le skupinski atributi in operatorji

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

Delo s skupinami

- Za vsak rating v tabeli jadralcev izpiši starost najmlajšega jadralca s tem ratingom.

```
SELECT MIN(starost)
FROM jadralec
WHERE rating = i;  -- za i = 1, 2, ... 10
```

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

Delo s skupinami

- Za vsak rating v tabeli jadralcev izpiši starost najmlajšega jadralca s tem ratingom.

```
SELECT rating, MIN(starost)
FROM jadralec
GROUP BY rating;
```

rating	MIN(starost)
1	33
3	25.5
7	35
8	25.5
9	35
10	16

- Ali s tem mešamo navadne in skupinske attribute?
- Ne, ker po grupiranju rating postane skupinski atribut!

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

Delo s skupinami

- Za vsak rating v tabeli jadralcev izpiši starost najmlajšega polnoletnega jadralca s tem ratingom, vendar samo za tiste ratinge, ki jih imata vsaj dva jadralca!

```
SELECT rating, MIN(starost)
           AS najmlajsi
FROM jadralec
WHERE starost >= 18
GROUP BY rating
HAVING COUNT(*) > 1;
```

rating	najmlajsi
3	25.5
7	35
8	25.5

Kako deluje ta poizvedba (1)?

```
+-----+-----+-----+-----+
| jid | 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    |
+-----+-----+-----+-----+
```

1. korak:
vsi jadranci.

V splošnem se naredi
kartezični produkt vseh
tabel v FROM delu

Kako deluje ta poizvedba (2)?

jid	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
74	Henrik	9	35
85	Anze	3	25.5
95	Bine	3	63.5

2. korak: selekcija
WHERE starost >= 18

Kako deluje ta poizvedba (3)?

```
+-----+-----+
| rating | starost |
+-----+-----+
|      7 |      45 |
|      1 |      33 |
|      8 |     55.5 |
|      8 |     25.5 |
|     10 |      35 |
|      7 |      35 |
|      9 |      35 |
|      3 |     25.5 |
|      3 |     63.5 |
+-----+-----+
```

3. korak: eliminacija nepotrebnih atributov
samo atributi iz SELECT, GROUP BY in
HAVING vrstic so potrebni za nadaljnje delo

Kako deluje ta poizvedba (4)?

rating starost		
1 33		
3 25.5		
3 63.5		
7 45		
7 35		
8 55.5		
8 25.5		
9 35		
10 35		

4. korak: grupiranje po vrednosti atributa rating

Kako deluje ta poizvedba (5)?

rating		starost	
3		25.5	
3		63.5	
7		45	
7		35	
8		55.5	
8		25.5	

5. korak: eliminacija odvečnih skupin.
Ohranimo samo tiste, za katere velja
HAVING COUNT(*) > 1

Kako deluje ta poizvedba (6)?

rating	starost
3	25.5
7	35
8	25.5

6. korak: izvajanje skupinskega operatorja (v naše primeru MIN) na vsaki posamezni skupini

Opomba: če bi naša SELECT vrstica vsebovala DISTINCT, bi se podvojene vrstice izločile šele po 6. koraku!

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

Delo s skupinami

- Za vsak rdeč čoln izpišite število rezervacij!

```
SELECT c.cid, COUNT(*) AS St_rez
FROM coln c, rezervacija r
WHERE c.cid = r.cid AND barva='rdeca'
GROUP BY c.cid;
```

cid	St_rez
102	3
104	2

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

Delo s skupinami

- Za vsak rdeč čoln izpišite število rezervacij!

```
SELECT c.cid, COUNT(*) AS St_rez  
FROM coln c, rezervacija r  
WHERE c.cid = r.cid  
GROUP BY c.cid  
HAVING c.barva='rdeca';
```

```
SELECT c.cid, COUNT(*) AS St_rez  
FROM coln c, rezervacija r  
WHERE c.cid = r.cid  
GROUP BY c.cid, c.barva  
HAVING c.barva='rdeca';
```

+	-----	+	-----	+
	cid		St_rez	
+	-----	+	-----	+
	102		3	
	104		2	
+	-----	+	-----	+

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

Delo s skupinami

- Za vsak rating z najmanj dvema jadralcema izpišite povprečno starost jadralcev!

```
SELECT j.rating, AVG(j.starost) AS Povp_star
FROM jadralec j
GROUP BY j.rating
HAVING COUNT(*) > 1;
```

rating	Povp_star
3	44.5
7	40
8	40.5
10	25.5

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

Delo s skupinami

- Za vsak rating z najmanj dvema jadralcema izpišite povprečno starost jadralcev!
Alternativna rešitev z gnezdenjem v HAVING vrstici.

```
SELECT j.rating, AVG(j.starost) AS Povp_star
FROM jadralec j
GROUP BY j.rating
HAVING 1 < (SELECT COUNT(*)
            FROM jadralec j2
            WHERE j.rating = j2.rating);
-- korelirano gnezdenje
-- j.rating je skupinski atribut in
-- zato lahko nastopa v HAVING vrstici
-- (tudi v gnezdenem delu)
```

rating	Povp_star
3	44.5
7	40
8	40.5
10	25.5

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

Delo s skupinami

- Za vsak rating z najmanj dvema jadralcema izpišite povprečno starost jadralcev!
Alternativna rešitev z gnezdenjem v FROM vrstici.

```
SELECT t.rating, t.Povp_star
FROM ( SELECT j.rating, AVG(j.starost) AS Povp_star,
              COUNT(*) AS St_ratingov
      FROM jadralec j
      GROUP BY j.rating ) AS t
WHERE t.St_ratingov > 1;
```

```
-- rezultat gnezdene poizvede se
-- uporablja kot začasna tabela t
-- vse nove attribute moramo poimenovati
```

rating	Povp_star
3	44.5
7	40
8	40.5
10	25.5

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

Vaje: gnezdenje, kvantifikatorji, skupinski operatorji

1. Poiščite imena čolnov, ki so jih rezervirali vsi jadralci.
2. Preverite, ali so vsi čolni z dolžino nad 35 čevljev iste barve (s kvantifikatorji).
3. Preverite, ali so vsi čolni z dolžino nad 35 čevljev iste barve (s skupinskimi operatorji).
4. Izpišite šifre, imena čolnov in število njihovih rezervacij urejeno v padajočem vrstnem redu.
5. Izpišite imena, šifre in število rezervacij vsakega jadralca. Kdor ni rezerviral ničesar, bo imel 0 rezervacij. Izpis uredite padajoče po številu rezervacij in naraščajoče po imenu jadralca.
6. Izpišite imena in šifre vseh jadralcev, ki so rezervirali nadpovprečno število čolnov.

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

Vaje: gnezdenje, kvantifikatorji, skupinski operatorji

1. Za vsakega igralca izpišite število njegovih naselji.
2. Za vsako alianso izpišite število njenih igralcev. Razvrstite alianse od tiste z največ do tiste z najmanj igralci.
3. Za vsako alianso izpišite število njenih naselji in skupno populacijo.
4. Katero pleme je najštevilčnejše. (Glede na skupno populacijo).
5. Za vsakega igralca izpišite povprečno velikost njegovega naselja(glede na populacijo), vendar samo za tiste igralce, ki imajo vsaj pet naselji.
6. Za vsak kvadrant preštejte število naselji na tem kvadrantu.*

Employees(emp_no, birth_date, first_name, last_name, gender, hire_date)
Departments(dept_no, dept_name)
Salaries(emp_no, salary, from_date, to_date)
Titles(emp_no, title, from_date, to_date)
Dept_emp(emp_no, dept_no, from_date, to_date)
Dept_manager(emp_no, dept_no, from_date, to_date)

Vaje: gnezdenje, kvantifikatorji, skupinski operatorji

1. Za vsakega zaposlenega izpišite število različnih oddelkov, na katerih je delal.
2. Za vsak oddelek izpišite število zaposlenih. Razvrstite oddelke od tistega z največ do tistega z najmanj zaposlenimi.
3. Poiščite imena zaposlenih, ki so imeli vsaj tri različne nazive.
4. Izpišite številke zaposlenih in njihovih prvih plač urejene po padajočem vrstnem redu zneska prve plače.
5. Ali ima kateri oddelek zaposlenih več žensk, kot moških.