

Poizvedovanje z jezikom SQL

Osnove podatkovnih baz

2. letnik univerzitetnega študija na FRI

Poizvedovalni jezik SQL (MySQL 5.7)

Data Manipulation Language (DML):

CALL, **DELETE**, DO, HANDLER, **INSERT**, LOAD DATA INFILE, LOAD XML, REPLACE, **SELECT**, **UPDATE**

Data Definition Language (DDL):

ALTER DATABASE, ALTER EVENT, ALTER FUNCTION, ALTER PROCEDURE, ALTER SERVER, **ALTER TABLE**, ALTER VIEW, **CREATE DATABASE**, CREATE EVENT, **CREATE INDEX**, CREATE PROCEDURE, CREATE FUNCTION, CREATE SERVER, **CREATE TABLE**, CREATE TABLESPACE, CREATE TRIGGER, **CREATE VIEW**, DROP DATABASE, DROP EVENT, DROP FUNCTION, DROP INDEX, DROP PROCEDURE, DROP FUNCTION, DROP SERVER, **DROP TABLE**, DROP TABLESPACE, DROP TRIGGER, DROP VIEW, RENAME TABLE, TRUNCATE TABLE

DESCRIBE, EXPLAIN, HELP, USE

Poizvedovalni jezik SQL (MySQL 5.7)

Data Control Language (DCL):

SET, SHOW, **ALTER USER**, **CREATE USER**, **DROP USER**, **GRANT**, RENAME USER, **REVOKE**, SET PASSWORD, ANALYZE TABLE, CHECK TABLE, CHECKSUM TABLE, OPTIMIZE TABLE, REPAIR TABLE, CREATE FUNCTION, DROP FUNCTION, INSTALL PLUGIN, UNINSTALL PLUGIN, BINLOG, CACHE INDEX, FLUSH, KILL, LOAD INDEX INTO CACHE, RESET

Transaction Processing Option (TPO):

START TRANSACTION, **COMMIT**, **ROLLBACK**, SAVEPOINT, ROLLBACK TO SAVEPOINT, RELEASE SAVEPOINT, LOCK TABLES, UNLOCK TABLES, SET TRANSACTION

PURGE BINARY LOGS, RESET MASTER, SET sql_log_bin, CHANGE MASTER TO, CHANGE REPLICATION FILTER, MASTER_POS_WAIT, RESET SLAVE, SET GLOBAL sql_slave_skip_counter, START SLAVE, STOP SLAVE, START GROUP_REPLICATION, STOP GROUP_REPLICATION

Oblika stavka **SELECT** (MySQL 5.7)

```
SELECT [ALL | DISTINCT]  
    izraz [, izraz ...]  
    [FROM stiki_tabel [, stiki_tabel ...]  
    [WHERE pogoji_nad_stolpci]  
    [GROUP BY {stolpci | izrazi}]  
    [HAVING pogoji_nad_skupinami]  
    [ORDER BY {stolpci | izrazi} [ASC | DESC]]  
    [LIMIT [preskoci_vrstic ,] stevilo_vrstic ]]
```

Oblike stavka SELECT (MySQL 5.7)

Projekcija $\pi_{A_1, \dots, A_n}(T)$: **SELECT** A_1, \dots, A_n **FROM** T

Selekcija $\pi_{A_1, \dots, A_n}(\sigma_P(T))$: **SELECT** A_1, \dots, A_n **FROM** T **WHERE** P

Kartezični produkt $\pi_{A_1, \dots, A_n, B_1, \dots, B_m}(T_1 \times T_2)$:
SELECT $A_1, \dots, A_n, B_1, \dots, B_m$ **FROM** T_1, T_2

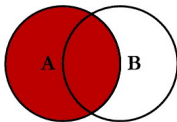
Pogojni stik $\pi_{A_1, \dots, A_n, B_1, \dots, B_m}(T_1 \bowtie_{T_1.A_i = T_2.B_j} T_2)$:
SELECT $A_1, \dots, A_n, B_1, \dots, B_m$ **FROM** T_1 **JOIN** T_2 **ON** $T_1.A_i = T_2.B_j$

Pol-odprti stik $\pi_{A_1, \dots, A_n, B_1, \dots, B_m}(T_1 \ltimes_{T_1.A_i = T_2.B_j} T_2)$:
SELECT $A_1, \dots, A_n, B_1, \dots, B_m$ **FROM** T_1 **LEFT JOIN** T_2 **ON** $T_1.A_i = T_2.B_j$

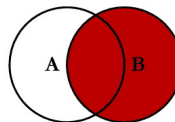
Naravni stik $T_1 \bowtie T_2$: **SELECT** $*$ **FROM** T_1 **NATURAL JOIN** T_2

Stične operacije SELECT (MySQL 5.7)

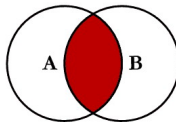
SQL JOINS



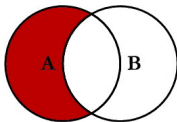
```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
```



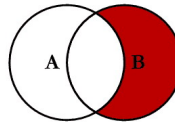
```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
```



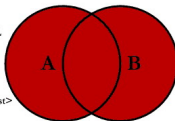
```
SELECT <select_list>
FROM TableA A
INNER JOIN TableB B
ON A.Key = B.Key
```



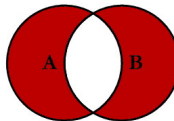
```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
WHERE B.Key IS NULL
```



```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
```



```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
```



```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
OR B.Key IS NULL
```

© C.L. Moffatt, 2008

Primerjalni operatorji SELECT (MySQL 5.7)

=

je enako

! = ali <>

je različno

> in >=

je večje in je večje ali enako

< in <=

je manjše in je manjše ali enako

IN (...)

ustreza poljubnemu elementu množice

BETWEEN ... AND ...

je med podanima vrednostima

LIKE '...%...'

ustreza vzorcu niza znakov

IS NULL

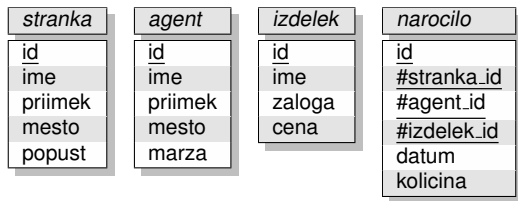
je neznana vrednost

NOT ...

negacija operatorja

Domena Trgovina

V podatkovni bazi *trgovina.sql* vodimo podatke manjše potujoče trgovine, ki deluje v slovenskih ruralnih področjih. Stranke, ki trgovino osebno obiščejo, lahko naročijo več enakih ali različnih izdelkov. Zaradi osebnega pristopa strankam o njih hranimo tudi nekaj osebnih podatkov, vključno s pridobljenim popustom zvestobe. Lastnik trgovine želi zagotoviti rigorozno sledljivost naročil, zato pri vsakem naročilu zabeležimo, kateri agent je določeni stranki prodal izdelke.



1. naloga Trgovina

stranka (id, ime, priimek, mesto, popust)

agent (id, ime, priimek, mesto, marza)

izdelek (id, ime, zaloga, cena)

narocilo (id, #stranka_id, #agent_id, #izdelek_id, datum, kolicina)

Z uporabo jezika SQL poiščite:

(a) vse podatke o vseh strankah

(b) identifikatorje in imena kranjskih agentov

(c) identifikatorje vsaj enkrat naročenih izdelkov

(d) pare imen strank in agentov, ki so sklenili naročilo

2. naloga Trgovina

stranka (id, ime, priimek, mesto, popust)

agent (id, ime, priimek, mesto, marza)

izdelek (id, ime, zaloga, cena)

```
narocilo (id, #stranka_id, #agent_id, #izdelek_id, datum, kolicina)
```

Z uporabo jezika SQL poiščite:

(a) dobiček vsakega naročila po spodnji formuli

$$0,4 \times \text{količina} \times \text{cena kosa} - \frac{\text{popust} + \text{marža}}{100} \times \text{količina} \times \text{cena kosa}$$

(b) identifikatorje parov strank s sedežem v istem mestu

(c) identifikatorje izdelkov, ki sta jih kupili vsaj dve stranki

(d) stranke, ki so kupile izdelek, ki ga je prodal tudi agent 6

1. domača naloga GSM

prodaja (operater, telefon)

kupuje (stranka, operater)

najraje (stranka, telefon)

Z uporabo jezika SQL poiščite:

- (a) stranke, ki kupujejo pri Mobitelu
- (b) operaterje pri katerih kupuje Petra
- (c) telefone, ki jih lahko kupi Petra z upoštevanjem (b)
- (d) operaterje, ki prodajajo Janezov najljubši telefon
- (e) telefone, ki jih prodaja posamezni operater

Operater prodaja telefon
<i>Mobitel prodaja telefone znamke Apple</i>
<i>Simobil prodaja telefone znamke HTC</i>
...

Operacije množic SELECT (MySQL 5.7)

UNION	vse vrstice dveh tabel (brez duplikatov)
UNION ALL	vse vrstice dveh tabel (z duplikati)
INTERSECT	le vrstice v obeh tabelah (ni podprta)
MINUS	vrstice le v prvi tabeli (ni podprta)

EXISTS (...)	neprazna množica ali tabela
... IN (...)	vsebovanost v množici ali tabeli
... > ALL (...)	ujemanje z vsemi v množici ali tabeli
... < ANY (...)	obstoj ujemanja v množici ali tabeli

SELECT * FROM T_1 UNION SELECT * FROM T_2

SELECT * FROM T_1 WHERE A IN (SELECT A FROM T_2)

SELECT * FROM T WHERE $A \geq$ ALL (SELECT A FROM T)

SELECT * FROM T WHERE $A <$ ANY (SELECT A FROM T)

3. naloga Trgovina

stranka (id, ime, priimek, mesto, popust)

agent (id, ime, priimek, mesto, marža)

izdelek (id, ime, zaloga, cena)

narocilo (id, #stranka_id, #agent_id, #izdelek_id, datum, kolicina)

Z uporabo jezika SQL poiščite:

(a) imena strank, ki so naročile izdelek 2 (s stikom)

(b) imena strank, ki so naročile izdelek 2 (brez stikov)

(c) stranke, ki so že naročile pri agentu iz Kranja ali Kopra (b)

4. naloga Trgovina

stranka (id, ime, priimek, mesto, popust)

agent (id, ime, priimek, mesto, marža)

izdelek (id, ime, zaloga, cena)

narocilo (id, #stranka_id, #agent_id, #izdelek_id, datum, kolicina)

Z uporabo jezika SQL poiščite:

(a) identifikatorje agentov z najnižjim odstotkom marže

(b) identifikatorje agentov, ki nimajo najvišjega odstotka marže

(c) stranke, ki so kupovale le izdelke dražje od 50 EUR

2. domača naloga GSM

prodaja (operater, telefon)

kupuje (stranka, operater)

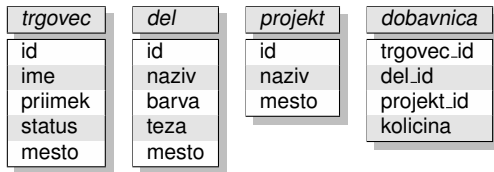
najraje (stranka, telefon)

Z uporabo jezika SQL poiščite:

- (a) stranke, ki kupujejo pri vseh operaterjih
(implementirajte deljenje z dvema NOT EXISTS)

Domena Dobavitelj

V podatkovni bazi *dobavitelj.sql* vodimo podatke o dobavljanju delov za namene izvajanja projektov. Dele zagotavljajo različni trgovci na podlagi izpolnjene dobavnice. Dobavnice pripadajo izbranim projektom, ki določene dele potrebujejo.



5. naloga Dobavitelj

trgovec (id, ime, priimek, status, mesto)

del (id, naziv, barva, teza, mesto)

projekt (id, naziv, mesto)

dobavnica (trgovec_id, del_id, projekt_id, kolicina)

Z uporabo jezika SQL poiščite:

(a) imena in priimke kranjskih trgovcev s statusom večjim od 20

(b) nazive rdečih delov, ki so težki vsaj 17kg

(c) imena in priimke trgovcev, ki so v okviru ene dobave prodali največ kosov poljubnega dela

6. naloga Dobavitelj

trgovec (id, ime, priimek, status, mesto)

del (id, naziv, barva, teza, mesto)

projekt (id, naziv, mesto)

dobavnica (trgovec_id, del_id, projekt_id, kolicina)

Z uporabo jezika SQL poiščite:

(a) trgovce, ki so prodajali novomeščanskim projektom

(b) nazive še nikoli prodanih delov

(c) identifikatorje delov, ki sta jih prodala vsaj dva trgovca

(d) nazive delov, ki so jih kupovali kranjski in mariborski projekti

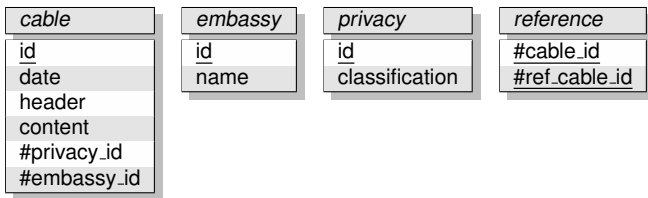
Agregacija z grupiranjem SELECT (MySQL 5.7)

COUNT	število vrednosti brez NULL
MIN	najmanjša vrednost
MAX	največja vrednost
SUM	vsota vseh vrednosti
STDDEV	deviacija vrednosti
VARIANCE	varianca vrednosti

```
SELECT  $A_1, \dots, A_k, \text{SUM}(A_n) / \text{COUNT}(A_n)$   
FROM  $T_1$  NATURAL JOIN  $T_2$   
WHERE  $A_i < \dots$   
GROUP BY  $A_1, \dots, A_k$   
HAVING  $\text{COUNT}(A_n) > \dots$   
...
```

Normalizirana domena WikiLeaks

V podatkovni bazi *wikileaks.sql* hranimo podatke o depešah, ki so si jih med seboj pošiljala veleposlaništva.



Zaradi omejitev velikosti datoteka ne vsebuje tekstovnih polj. Polno različico primerno za uvoz v podatkovno bazo PostgreSQL si lahko prenesete iz:

http://file.wikileaks.org/torrent/cable_db_full.7z.torrent

7. naloga WikiLeaks

```
cable (id, date, header, content, #privacy_id, #embassy_id)
```

embassy (id, name)

privacy (id, classification)

```
reference (#cable_id, #ref_cable_id)
```

Z uporabo jezika SQL poiščite:

(a) število zaupnih depeš ambasade v Ljubljani po letu 2000

(b) število depeš za vsak nivo zaupanja z imeni atributov *Nivo* in *Število*

(c) dneve, ko je bilo odposlanih več kot 30 tajnih depeš

(d) dan, ko je bilo odposlanih največ tajnih depeš z ORDER BY in LIMIT

(e) dneve, ko je bilo odposlanih največ tajnih depeš *

8. naloga WikiLeaks

cable (id, date, header, content, #privacy_id, #embassy_id)

embassy (id, name)

privacy (id, classification)

```
reference (#cable_id, #ref_cable_id)
```

Z uporabo jezika SQL poiščite:

- (a) dneve z največjim številom poslanih depeš, urejene padajoče, od 11. do 20. zadetka, in seznam veleposlaništev, ki so te depeše pošiljala

(b) depeše, ki se sklicujejo le na depeše ljubljanske ambasade

- (c) veleposlaništvo, na čigar depeše se sklicuje največ depeš

- (d) veleposlaništvo, na čigar depeše se sklicujejo depeše največ različnih veleposlaništev

3. domača naloga WikiLeaks

cable (id, date, header, content, #privacy_id, #embassy_id)
embassy (id, name)
privacy (id, classification)
reference (#cable_id, #ref_cable_id)

Z uporabo jezika SQL poiščite:

(a) veleposlaništvo, ki se sklicuje na depeše največ različnih veleposlaništev
z ORDER BY in LIMIT

(b) veleposlaništvo, ki se je že sklicevalo na depeše vseh veleposlaništev

(c) poiščite število veleposlaništev, ki so se sklicevala na svoje depeše, in
število depeš, ki so se sklicevala same nase

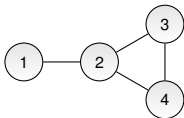
Normalizirana domena Facebook & Twitter

oseba(id, ime, rojstni_dan)

<u>id</u>	ime	rojstni_dan
1	Jill	9.3.1990
2	Jack	2.6.1950
3	Joe	1.8.1989
4	Jenn	7.1.2001

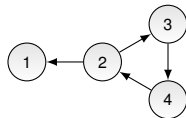
facebook(#oseba_id, #prijatelj_id)

<u>#oseba_id</u>	<u>#prijatelj_id</u>
1	2
2	1
2	3
3	2
3	4
...	...



twitter(#oseba_id, #sledilec_id)

<u>#oseba_id</u>	<u>#sledilec_id</u>
1	2
3	2
4	3
2	4



9. naloga Facebook & Twitter

oseba (id, ime, rojstni_dan)

facebook (#oseba_id, #prijatelj_id)

twitter (#oseba_id, #sledilec_id)

Z uporabo jezika SQL:

(a) ustvarite podatkovno bazo *social*

(b) ustvarite vse tabele s primarnimi in tujimi ključi

(c) napolnite tabelo *oseba* z ustreznimi vrednostmi

(d) napolnite tabelo *facebook* s prošnjami za prijateljstvo

10. naloga Facebook & Twitter

oseba (id, ime, rojstni_dan)

facebook (#oseba_id, #prijatelj_id)

twitter (#oseba_id, #sledilec_id)

Z uporabo jezika SQL:

(a) izbrišite Facebook prošnjo za prijateljstvo med osebama 3 in 4

(b) dodajte uporabnika *misko/Kranjec.01* s polnim dostopom do baze *social*

(c) dodajte uporabnika *pavle/Sedmak.01* z bralnim dostopom do baze *social* in mu omogočite dostop le iz izbranega IP naslova

(d) preskusite delovanje ustvarjenih uporabniških računov

4. domača naloga Facebook & Twitter

oseba (id, ime, rojstni_dan)

facebook (#oseba_id, #prijatelj_id)

twitter (#oseba_id, #sledilec_id)

Z uporabo jezika SQL:

(a) napolnite tabelo *twitter* z ustreznimi vrednostmi

(b) izdelajte proceduro, ki poskrbi za obojestranska prijateljstva na Facebook
