Poizvedovanje z jezikom SQL

Definicija. Stavek SELECT v jeziku SQL DML (brez agregacije in grupiranja).

```SQL
5. SELECT [*|ALL|DISTINCT] A, B AS C -- projekcija in preimenovanje
1.      FROM t tabela ... -- preimenovanje (in stiki) tabel
2.      WHERE A = 1 -- selekcija po atributih
        /* GROUP BY ... */
        /* HAVING ... */
3.      ORDER BY A, B [ASC|DESC] -- urejanje izpisa
4.      LIMIT [0,] 10; -- omejevanje izpisa
```

Projekcija in selekcija

Projekcija. Izpiši vrednosti stolpcev A in B v tabeli t (brez podvojenih vrstic).

$$\pi_{A,B}(t)$$

```SQL
SELECT [ALL|DISTINCT] A, B
    FROM t;
```

Izpiši vse vrednosti v tabeli t.

$$t$$

```SQL
SELECT *
    FROM t;
```

Preimenovanje. Izpiši vrednosti stolpcev A in B v tabeli t kot Foo in Bar.

$$\rho_{(Foo,Bar)}(\pi_{A,B}(t))$$

```SQL
SELECT A AS Foo, B AS Bar
    FROM t;
```

Selekcija. Izpiši vse vrstice v tabeli t kjer $A > 100$ in $B \neq 1$.

$$\sigma_{A>100 \wedge B \neq 1}(t)$$

SQL
```sql
SELECT *
    FROM t
    WHERE A > 100 AND B != 1;
```

V logičnih pogojih lahko uporabljamo konjunkcijo $\wedge$, disjunkcijo $\vee$ in negacijo $\neg$…

SQL
```sql
AND, OR, NOT
```

…ter različne operatorje primerjanja.

SQL
```sql
=, !=, <>, <, >, <=, >=, BETWEEN ... AND ..., IS NULL, LIKE "_%..."
```

Domena. *Facebook&Twitter* v datoteki `facebook_twitter.sql`.

facebook (#<u>OID</u>, #<u>PID</u>)
twitter (#<u>OID</u>, #<u>SID</u>)
oseba (<u>ID</u>, Ime, Rojen, #SID)
stan (<u>SID</u>, Stan)

1. naloga. *(Facebook&Twitter)* Z uporabo jezika SQL izpišite identifikatorje, uporabniška imena in rojstne datume vseh polnoletnih oseb.

$$\rho_{(ID, User\ name, Birth)}(\pi_{ID, Ime, Rojen}(\sigma_{Rojen \leq 26.10.2002}(o)))$$

Shema rezultata poizvedbe naj bo enaka spodnji.

ID	User name	Birth
…	…	…

SQL
```sql
[USE facebook_twitter;]
SELECT ID, Ime AS "User name", Rojen AS Birth
    FrOm oseba /* SQL ne razlikuje med velikimi in malimi črkami */
    wHeRe Rojen <= '2002-10-26';
```

2. naloga. *(Facebook&Twitter)* Z uporabo jezika SQL izračunajte starost vseh oseb.

```SQL
[USE facebook_twitter;]
SELECT *, DATEDIFF(CURDATE(), Rojen) / 365 AS Starost /* funkcije in aritmetika */
    FROM oseba;
```

Domena. *WikiLeaks* v datoteki `wikileaks.sql` .

$$\begin{gathered}\text{embassy}\ (\underline{\text{id}}, \text{name})\\ \text{privacy}\ (\underline{\text{id}}, \text{classification})\\ \text{cable}\ (\underline{\text{id}}, \text{date}, \text{header}, \text{content}, \#\text{privacy\_id}, \#\text{embassy\_id})\\ \text{reference}\ (\underline{\#\text{cable\_id}}, \underline{\#\text{ref\_cable\_id}})\end{gathered}$$

3. naloga. *(WikiLeaks)* Z uporabo jezika SQL poiščite identifikatorje, datume, naslove in vsebino zadnjih pet depeš ambasade z identifikatorjem 140.

```SQL
[USE wikileaks;]
SELECT id, DATE(c.date), header, content
    FROM cable c
    WHERE embassy_id = 140
    ORDER BY c.date DESC, id [ASC]
    LIMIT 5;
```

4. naloga. *(WikiLeaks)* Z uporabo jezika SQL izpišite vse podatke o ambasadi v Ljubljani.

```SQL
[USE wikileaks;]
SELECT *
    FROM embassy
    WHERE name LIKE "%Ljubljana%"; -- name = "Embassy Ljubljana"
```

Stične operacije

Kartezični produkt. Izpiši vse pare vrstic v tabelah t_1 in t_2.

$$t_1 \times t_2$$

```SQL
SELECT *
    FROM t1, t2;
```

Naravni stik. Izpiši naravni stik tabel t_1 in t_2.

$$t_1 \bowtie t_2$$

```sql
SELECT *
    FROM t1 NATURAL JOIN t2;
```

Ekvistik. Izpiši ekvistik tabel t_1 in t_2 po stolpcu A.

$$t_1 \bowtie_A t_2$$

```sql
SELECT *
    FROM t1 [INNER] JOIN t2 USING (A);
-- ali
SELECT *
    FROM t1, t2
    WHERE t1.A = t2.A;
```

Pogojni stik. Poišči pare vrstic v tabelah t_1 in t_2, kjer je $t_1.A$ večji ali enak $t_2.B$.

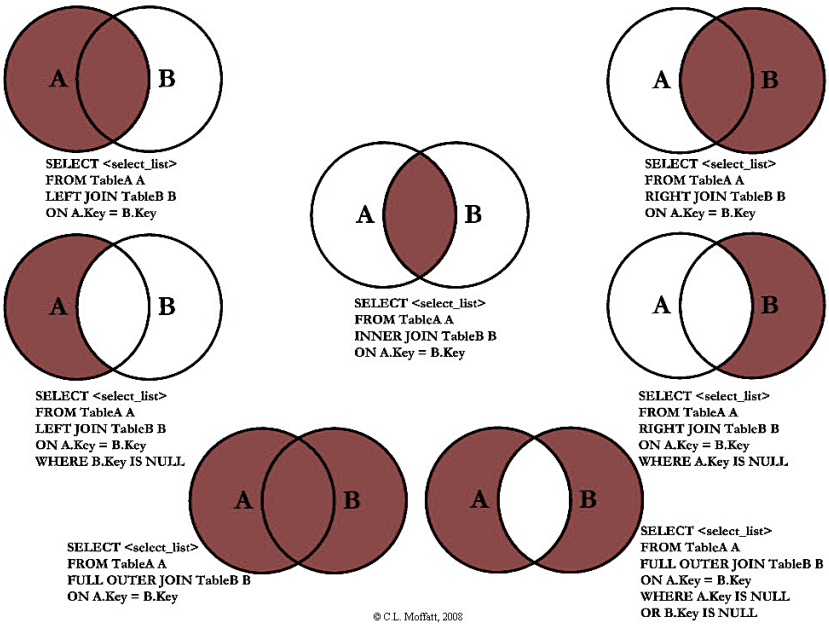
$$t_1 \bowtie_{t_1.A \geq t_2.B} t_2$$

```sql
SELECT *
    FROM t1 [INNER] JOIN t2 ON t1.A >= t2.B;
-- ali
SELECT *
    FROM t1, t2
    WHERE t1.A >= t2.B;
```

Odprti stik. Izpiši vse vrstice v tabeli t_1 in pare vrstic v tabeli t_2, kjer je $t_1.A$ večji ali enak $t_2.B$.

$$t_1 ⟕_{t_1.A \geq t_2.B} t_2$$

```sql
SELECT *
    FROM t1 LEFT [OUTER] JOIN t2 ON t1.A >= t2.B;
```



Domena. *Facebook&Twitter* v datoteki `facebook_twitter.sql` .

facebook (#OID, #PID)
twitter (#OID, #SID)
oseba (ID, Ime, Rojen, #SID)
stan (SID, Stan)

5. naloga. *(Facebook&Twitter)* Z uporabo jezika SQL izpišite vse podatke o osebah na Facebooku.

$$\rho_{(ID,User\ name,Status)}(\pi_{ID,Ime,Stan}(o \bowtie s \bowtie_{ID=OID} f))$$

Shema rezultata poizvedbe naj bo enaka spodnji.

ID	User name	Status
…	…	…

SQL

```
[USE facebook_twitter;]
SELECT DISTINCT ID, Ime AS "User name", Stan AS Status
    FROM oseba NATURAL JOIN stan [INNER] JOIN facebook ON ID = OID;
-- ali
SELECT DISTINCT ID, Ime AS "User name", Stan AS Status
    FROM oseba o, stan s, facebook f
    WHERE o.SID = s.SID AND ID = OID;
```

6. naloga. *(Facebook&Twitter)* Z uporabo jezika SQL poiščite identifikatorje oseb, ki *niso* zgolj na Twitterju.

```SQL
[USE facebook_twitter;]
SELECT DISTINCT f.OID
    FROM facebook f LEFT [OUTER] JOIN twitter t ON f.OID = t.OID OR f.OID = t.SID;
```

Domena. *WikiLeaks* v datoteki `wikileaks.sql`.

$$\begin{gathered}\text{embassy}\ (\underline{\text{id}}, \text{name})\\ \text{privacy}\ (\underline{\text{id}}, \text{classification})\\ \text{cable}\ (\underline{\text{id}}, \text{date}, \text{header}, \text{content}, \#\text{privacy\_id}, \#\text{embassy\_id})\\ \text{reference}\ (\#\underline{\text{cable\_id}}, \#\underline{\text{ref\_cable\_id}})\end{gathered}$$

7. naloga. *(WikiLeaks)* Z uporabo jezika SQL izpišite podatke zaupnih depeš ambasade v Ljubljani leta 2010.

```SQL
[USE wikileaks;]
SELECT c.id, c.date, header
    FROM cable c JOIN privacy p ON c.privacy_id = p.id
        JOIN embassy e ON c.embassy_id = e.id
    WHERE YEAR(c.date) = 2010 AND -- c.date >= '2010-01-01'
        name = "Embassy Ljubljana" AND classification LIKE "%confidential%";
-- ali
SELECT c.id, c.date, header
    FROM cable c, privacy p, embassy e
    WHERE privacy_id = p.id AND embassy_id = e.id AND
        YEAR(c.date) = 2010 AND -- c.date >= '2010-01-01'
        name = "Embassy Ljubljana" AND classification LIKE "%confidential%";
```

8. naloga. *(WikiLeaks)* Z uporabo jezika SQL poiščite dneve, ko je ista ambasada odposlala vsaj dve depeši. Shema rezultata poizvedbe naj bo enaka spodnji.

Embassy	Day
…	…

```SQL
[USE wikileaks;]
SELECT DISTINCT c1.embassy_id AS Embassy, DATE(c1.date) AS Day
    FROM cable c1 JOIN cable c2 USING (date, embassy_id)
    WHERE c1.id != c2.id;
-- ali
SELECT DISTINCT c1.embassy_id AS Embassy, DATE(c1.date) AS Day
    FROM cable c1, cable c2 -- (SELECT * FROM cable ORDER BY date DESC LIMIT 1000) c2
    WHERE c1.id != c2.id AND c1.date = c2.date AND c1.embassy_id = c2.embassy_id;
```

Operacije množic in deljenje

Unija. Izpiši vse vrstice v tabelah t_1 in t_2.

$$t_1 \cup t_2$$

```SQL
SELECT *
    FROM t1
UNION [DISTINCT|ALL]
SELECT *
    FROM t2;
```

Presek. Izpiši vse enake vrstice v tabelah t_1 in t_2,…

$$t_1 \cap t_2$$

```SQL
SELECT *
    FROM t1
INTERSECT -- ni podprto?
SELECT *
    FROM t2;
```

pri čimer je A primarni ključ tabel t_1 in t_2.

```SQL
SELECT *
    FROM t1
    WHERE t1.A IN ( -- ugnezdena poizvedba
        SELECT t2.A FROM t2);
```

Razlika. Izpiši vrstice, ki so zgolj v tabeli t_1, pri čimer je A primarni ključ.

$$t_1 - t_2$$

```SQL
SELECT *
    FROM t1
MINUS -- ni podprto?
SELECT *
    FROM t2;
-- ali
SELECT *
    FROM t1
    WHERE t1.A NOT IN (SELECT t2.A FROM t2);
```

Operatorji. Deljenje lahko implementiramo z dvema operatorjema `NOT EXISTS`.

```SQL
... WHERE [NOT] EXISTS (SELECT...); -- neprazna tabela
... WHERE A [NOT] IN (SELECT A FROM...); -- vsebovanost v tabeli
... WHERE A >= ALL (SELECT A FROM...); -- največja vrednost A
... WHERE A > ANY (SELECT A FROM...); -- ne najmanjša vrednost A
```

Domena. *Facebook&Twitter* v datoteki `facebook_twitter.sql`.

facebook (#OID, #PID)
twitter (#OID, #SID)
oseba (ID, Ime, Rojen, #SID)
stan (SID, Stan)

9. naloga. *(Facebook&Twitter)* Z uporabo jezika SQL poiščite vse podatke najstarejših oseb.

$$o - \rho_{o_1}(o) \triangleright_{o_1.Rojen>o2.Rojen} \rho_{o_2}(o)$$

```SQL
[USE facebook_twitter;]
SELECT *
    FROM oseba
    WHERE Rojen <= ALL (SELECT Rojen FROM oseba WHERE Rojen IS NOT NULL);
```

Domena. *Trgovina* v datoteki `trgovina.sql`.

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)

10. naloga. *(Trgovina)* Z uporabo jezika SQL izpišite imena in priimke strank, ki so naročile izdelek z identifikatorjem 2. Nalogo rešite brez stikov, dočim naj bo shema rezultata poizvedbe enaka spodnji.

Ime in priimek
…

```SQL
[USE trgovina;]
SELECT CONCAT(ime, ' ', priimek) AS "Ime in priimek"
    FROM stranka
    WHERE 2 IN (
        SELECT izdelek_id
            FROM narocilo
            WHERE stranka_id = stranka.id)
```

Domena. *GSM* v datoteki `gsm.sql`.

$$\begin{gathered}\text{prodaja (operater, telefon)}\\ \text{kupuje (stranka, operater)}\\ \text{najraje (stranka, telefon)}\end{gathered}$$

11. naloga. *(GSM)* Z uporabo jezika SQL poiščite stanke, ki kupujejo pri vseh operaterjih (deljenje). Nalogo rešite z uporabo operatorja `NOT EXISTS` in dveh ugnezdenih poizvedb.

```SQL
[USE gsm;]
SELECT DISTINCT stranka
    FROM kupuje k
    WHERE NOT EXISTS (
        SELECT *
            FROM prodaja p
            WHERE NOT EXISTS (
                SELECT *
                    FROM kupuje
                    WHERE stranka = k.stranka AND operater = p.operater));
```

Agregacija in grupiranje

Definicija. Stavek SELECT v jeziku SQL DML z agregacijo (in grupiranjem).

```sql
7. SELECT [A, B,] SUM(C) -- agregacija atributov
1.      FROM t tabela ...
2.      WHERE A != 1 AND C > 0 -- selekcija po atributih
3.      [GROUP BY A, B] -- grupiranje po atributih
4.      HAVING SUM(C) > 100 -- selekcija po agregacijah
5.      ORDER BY [B,] SUM(C);
6.      LIMIT 1;
```

Agregacija. Izračunaj najmanjšo, največjo in povprečno vrednost stolpca A v tabeli t.

$$\tau_{MIN\ A,MAX\ A,AVG\ A}(t)$$

```sql
SELECT MIN(A), MAX(A), AVG(A) -- SUM(A) / COUNT(A)
    FROM t;
```

V agregacijah lahko uporabljamo funkcije za izračun vsote, števila vrednosti brez `NULL`, najmanjše in največje vrednosti, povprečja, standardnega odklona, variance itd.

```sql
SUM, COUNT, MIN, MAX, AVG, STD, VARIANCE ...
```

Grupiranje. Za vsako vrednost stolpca A poiščite število vrstic v tabeli t…

$$_A\tau_{COUNT\ B}(t)$$

```sql
SELECT A, COUNT(*)
    FROM t
    GROUP BY A;
```

Selekcija. …ter izpišite vrednosti A, pri katerih je število vrstic večje od 100.

$$\pi_A(\sigma_{C>100}(\rho_{(A,C)}({}_A\tau_{COUNT\ B}(t))))$$

```sql
SELECT A
    FROM t
    GROUP BY A
    HAVING COUNT(*) > 100;
```

Domena. *Facebook&Twitter* v datoteki `facebook_twitter.sql`.

facebook (#OID, #PID)
twitter (#OID, #SID)
oseba (ID, Ime, Rojen, #SID)
stan (SID, Stan)

12. naloga. *(Facebook&Twitter)* Z uporabo jezika SQL za vsako osebo izpišite identifikator, uporabniško ime in število sledenj na Twitterju.

$$_{ID,Ime}\tau_{COUNT\ OID}(o \ ⟕_{ID=t.SID}\ t)$$

SQL
```
[USE facebook_twitter;]
SELECT ID, Ime, COUNT(OID)
    FROM oseba LEFT [OUTER] JOIN twitter t ON ID = t.SID
    GROUP BY ID; -- ID, Ime
```

13. naloga. *(Facebook&Twitter)* Z uporabo jezika SQL izpišite število prijateljstev na Facebooku in število sledenj na Twitterju.

SQL
```
[USE facebook_twitter;]
SELECT
    (SELECT COUNT(*) FROM facebook WHERE OID < PID), -- AS Prijateljstva
    (SELECT COUNT(*) FROM twitter) -- AS Sledenja
```

Domena. *WikiLeaks* v datoteki `wikileaks.sql`.

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

14. naloga. *(WikiLeaks)* Z uporabo jezika SQL poiščite dneve, ko je bilo odposlanih več kot 300 depeš (oziroma največ depeš v enem dnevu).

```SQL
[USE wikileaks;]
SELECT DATE(c.date), COUNT(*)
    FROM cable c
    GROUP BY c.date
    HAVING COUNT(*) > 300;
-- oziroma
SELECT DATE(c.date), COUNT(*)
    FROM cable c
    GROUP BY c.date
    HAVING COUNT(*) = (
        SELECT MAX(value)
            FROM (
                SELECT COUNT(*) value
                    FROM cable c
                    GROUP BY c.date) alias);
```

15. naloga. *(WikiLeaks)* Z uporabo jezika SQL izpišite število in seznam referenc prvih 100 depeš z največjim številom referenc.

```SQL
[USE wikileaks;]
SELECT cable_id, COUNT(ref_cable_id), GROUP_CONCAT(ref_cable_id SEPARATOR ', ')
    FROM reference
    GROUP BY cable_id
    ORDER BY COUNT(ref_cable_id) DESC
    LIMIT 100;
```