

Poizvedovalni jezik SQL

Poizvedovanje po relacijski podatkovni bazi.

1. del

Uvod v SQL (1)

- **SQL** (**S**tructured **Q**uery **L**anguage) je jezik za obvladovanje podatkov v relacijskih podatkovnih bazah.
- Čez čas je doživel številne revizije:
 - **SQL-86** (SQL-87)
 - **SQL:1999** (SQL3) – regularni izrazi, prožilniki, objekti
 - **SQL:2003** – osnovne lastnosti XML, sekvence
 - **SQL:2006** – napredne lastnosti XML, XQuery
 - **SQL:2008** – napredne funkcije uokvirjenja

Skupine SQL ukazov

- **DML** (Data Manipulation Language)
 - **SELECT**, INSERT, UPDATE, DELETE
- **DDL** (Data Definition Language)
 - CREATE, ALTER, DROP
- **DCL** (Data Control Language)
 - GRANT, REVOKE
- **TPO** (Transaction Processing Option)
 - COMMIT, ROLLBACK

Uvod v SQL (2)

- Vsak proizvajalec (ponudnik sistema za obvladovanje podatkovnih baz) podpre določeno množico funkcionalnosti standarda SQL.
- Dodatno gradivo glede SQL stavkov najdete na spletni učilnici.

Oblika klasičnega SQL stavka

```
SELECT A1, A2, ..., Ak  
FROM T1, T2, ..., Tn  
WHERE P;
```

izberi atribute (A₁, A₂, ..., A_k)
iz tabel (T₁, T₂, ..., T_n)
pri pogoju (P)

Enostavni primeri SQL stavkov

- Projekcija

```
SELECT    A1, A2, ..., An  
FROM      T;
```

- Selekcija

```
SELECT    A1, A2, ..., An  
FROM      T  
WHERE    P;
```

- Ekvistik

```
SELECT    T1.A1, T2.A1  
FROM      T1, T2  
WHERE    T1.A3 = T2.A3;
```



Welcome to phpMyAdmin

Language

English

Log in

Username:

Password:

Go

MySQL - phpMyAdmin

- Dostopni podatki
 - URL
 - <http://todo.fri.uni-lj.si/phpmyadmin/>
 - **Uporabniško ime:** digitalna identiteta UL
Uporabniško ime je enako sistemu za dostop do eUčilnice (primer: ab1234)
Uporabniško ime zapišite brez domene
 - **Geslo:** začetno geslo je enako uporabniškemu imenu



MySQL - MyWebSQL

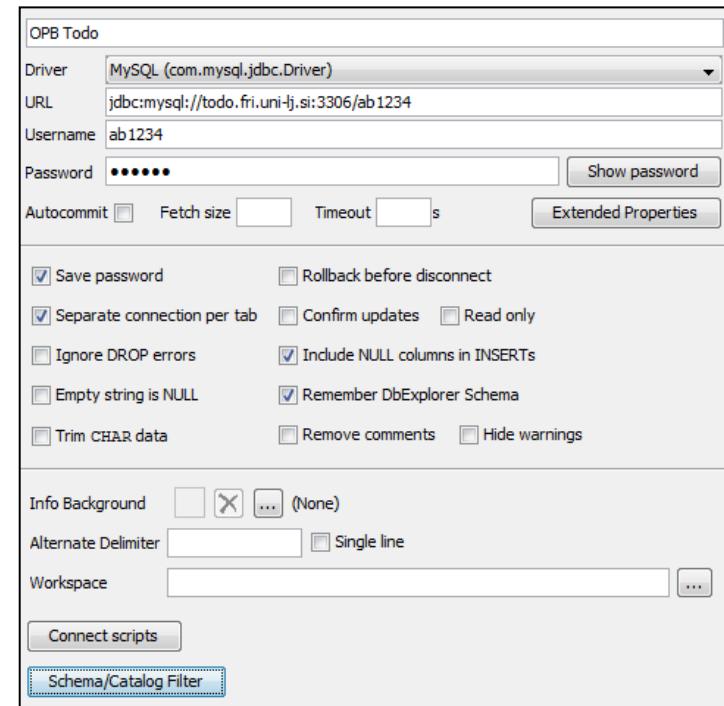
Login form for MyWebSQL:

User ID:	<input type="text"/>
Password:	<input type="password"/>
MySQL Server:	<input type="text" value="localhost MySQL"/>
Language:	<input type="text" value="English"/>
<input type="button" value="Login"/>	

- Dostopni podatki
 - URL
 - <http://todo.fri.uni-lj.si/mywebsql/>
 - **Uporabniško ime:** digitalna identiteta UL
Uporabniško ime je enako sistemu za dostop do eUčilnice (primer: ab1234)
Uporabniško ime zapišite brez domene
 - **Geslo:** začetno geslo je enako uporabniškemu imenu

SQL Workbench

- Driver: **MySQL**
- URL: **jdbc:mysql://todo.fri.uni-lj.si:3306/ab1234**
 - Hostname: **todo.fri.uni-lj.si**
 - Port: **3306**
 - Database: **<uporabniško ime>**
- Username: **digitalna identiteta UL**
(uporabniško ime zapišite brez domene)
- Password: **začetno geslo je enako uporabniškemu imenu**





Welcome to phpMyAdmin

Language

English

Log in

Username:

Password:

Go

MySQL - phpMyAdmin

- Dostopni podatki
 - URL
 - <http://todo.fri.uni-lj.si/phpmyadmin/>
 - **Uporabniško ime:** opb + vpisna številka
primer: opb24930359
 - **Geslo:** začetno geslo je enako uporabniškemu imenu



MySQL - MyWebSQL

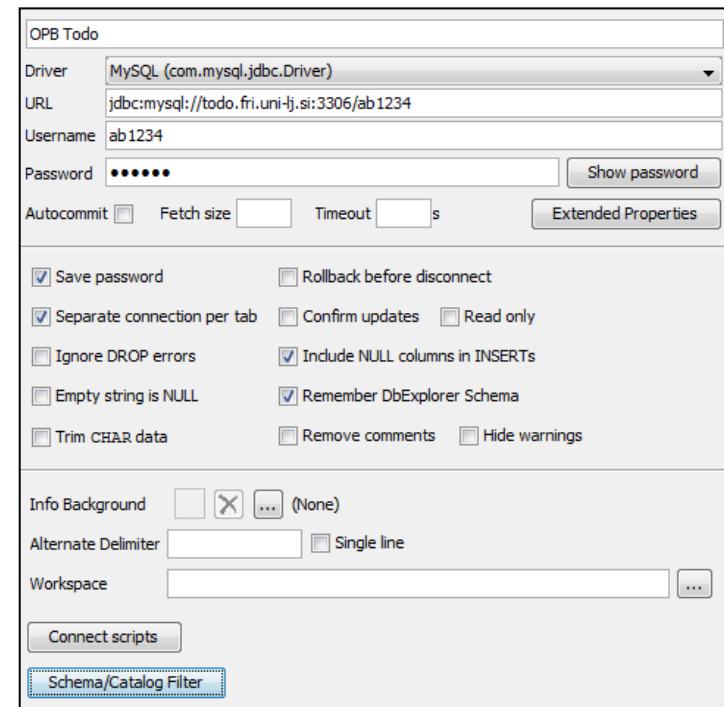
Login form for MyWebSQL:

User ID:	<input type="text"/>
Password:	<input type="password"/>
MySQL Server:	<input type="text" value="Localhost MySQL"/>
Language:	<input type="text" value="English"/>
<input type="button" value="Login"/>	

- Dostopni podatki
 - URL
 - <http://todo.fri.uni-lj.si/mywebsql/>
 - **Uporabniško ime:** opb + vpisna številka
primer: opb24930359
 - **Geslo:** začetno geslo je enako uporabniškemu imenu

SQL Workbench

- Driver: **MySQL**
- URL: **jdbc:mysql://todo.fri.uni-lj.si:3306/opb24930359**
 - Hostname: **todo.fri.uni-lj.si**
 - Port: **3306**
 - Database: <uporabniško ime>
- Username: **opb + vpisna številka** (primer: opb24930359)
- Password: **začetno geslo je enako uporabniškemu imenu**



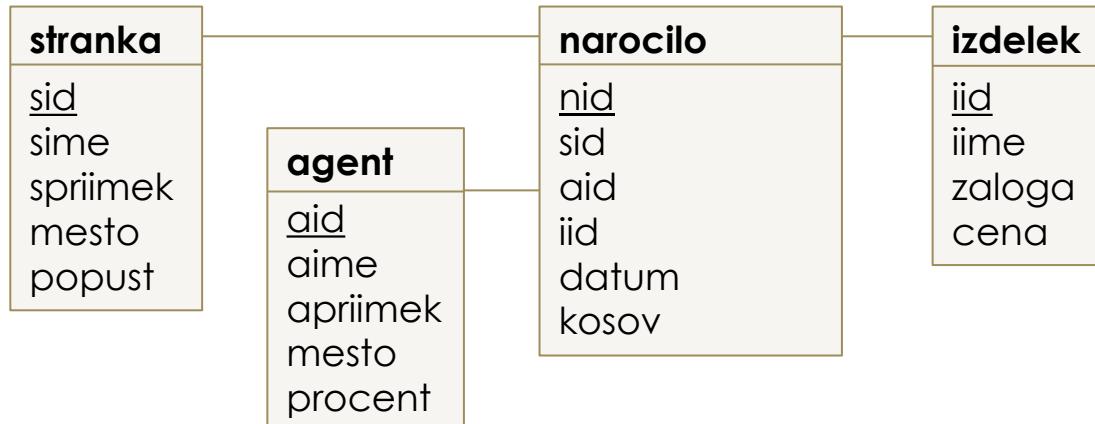
Primeri (1)

- Imamo naslednje tabele nad katerimi bomo izvajali poizvedbe.

phpMyAdmin



stranka	(sid, sime, spriimek, mesto, popust)
agent	(aid, aime, apriimek, mesto, procent)
izdelek	(iid, iime, zaloga, cena)
narocilo	(nid, sid, aid, iid, datum, kosov)



Primeri

(2) (1. vprašanje)

- **stranka** (sid, sime, spriimek, mesto, popust)
- agent** (aid, aime, apriimek, mesto, procent)
- izdelek** (iid, iime, zaloga, cena)
- narocilo** (nid, sid, aid, iid, datum, kosov)

The screenshot shows the phpMyAdmin interface. At the top, it says "phpMyAdmin". Below that is a toolbar with icons for home, export, import, etc. A dropdown menu shows "trgovina" selected. On the left, there's a sidebar with icons for each table: agent (highlighted with a red box), izdelek, narocilo, and stranka.

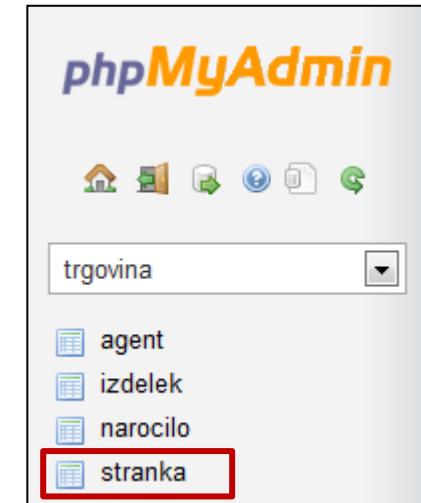
- Izpiši imena in idente kranjskih agentov!

```
SELECT aid, aime
FROM agent
WHERE mesto = 'Kranj';
```

Primeri

(3) (2. vprašanje)

- **stranka** (sid, sime, spriimek, mesto, popust)
agent (aid, aime, apriimek, mesto, procent)
izdelek (iid, iime, zaloga, cena)
narocilo (nid, sid, aid, iid, datum, kosov)



- Izpiši podatke vseh strank!

SELECT sid, sime, mesto, popust
FROM stranka;

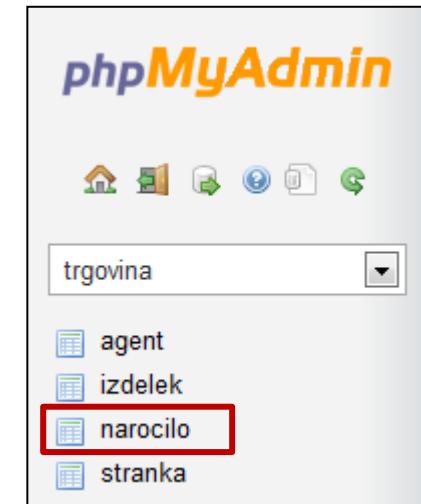
=

SELECT *
FROM stranka;

Primeri

(4) (3. vprašanje)

→ **stranka** (sid, sime, spriimek, mesto, popust)
agent (aid, aime, apriimek, mesto, procent)
izdelek (iid, iime, zaloga, cena)
narocilo (nid, sid, aid, iid, datum, kosov)



- Izpiši idente izdelkov, ki so bili kdaj naročeni!

```
SELECT DISTINCT iid  
FROM narocilo;
```

Primeri

(5) (4. vprašanje)

- **stranka** (sid, sime, spriimek, mesto, popust)
- **agent** (aid, aime, apriimek, mesto, procent)
- **izdelek** (iid, iime, zaloga, cena)
- **narocilo** (nid, sid, aid, iid, datum, kosov)

○ Izpiši vse pare (sime, aime), kjer je sime naročil nekaj pri aime!

```
SELECT stranka.sime, agent.aime  
FROM stranka, narocilo, agent  
WHERE stranka.sid = narocilo.sid AND  
narocilo.aid = agent.aid;
```

stranka
<u>sid</u>
sime
spriimek
mesto
popust

narocilo
<u>nid</u>
sid
aid
iid
datum
kosov

agent
<u>aid</u>
aime
apriimek
mesto
procent

Primeri

(6) (5. vprašanje)

- **stranka** (sid, sime, spriimek, mesto, popust)
- **agent** (aid, aime, apriimek, mesto, procent)
- **izdelek** (iid, iime, zaloga, cena)
- **naročilo** (nid, sid, aid, iid, datum, kosov)

○ Za vsako naročilo izračunaj čisti dobiček po naslednji formuli:

- dobiček = $0,4 \times \text{kosov} \times \text{cena enega kosa} - (\text{popust} + \text{marza})/100 \times \text{kosov} \times \text{cena enega kosa}$

```
SELECT n.nid, n.sid, n.aid, n.iid, 0.4*(n.kosov * i.cena) -  
0.01*(s.popust + a.procent) * (n.kosov * i.cena) AS dobicek  
FROM narocilo n, stranka s, agent a, izdelek i  
WHERE s.sid = n.sid AND n.aid = a.aid AND n.iid = i.iid;
```

Primeri

(7) (6. vprašanje)

- **stranka** (sid, sime, spriimek, mesto, popust)
- agent** (aid, aime, apriimek, mesto, procent)
- izdelek** (iid, iime, zaloga, cena)
- narocilo** (nid, sid, aid, iid, datum, kosov)

- Izpiši vse pare strank, ki imajo sedež v istem mestu:

stranka s1	
sid	mesto
A	LJ
B	NM
C	LJ
D	LJ

stranka s2	
sid	mesto
A	LJ
B	NM
C	LJ
D	LJ

pričakujemo
in nočemo še tega

s1	s2
A	C
A	D
C	D
A	A
C	C
D	D
C	A
D	A
D	C



Primeri

(8) (6. vprašanje)

→	stranka	(sid, sime, spriimek, mesto, popust)
	agent	(aid, aime, apriimek, mesto, procent)
	izdelek	(iid, iime, zaloga, cena)
	narocilo	(nid, sid, aid, iid, datum, kosov)

- Izpiši vse pare strank, ki imajo sedež v istem mestu:

```
SELECT      s1.sid, s2.sid ←
FROM        stranka s1, stranka s2
WHERE       s1.mesto = s2.mesto AND s1.sid < s2.sid;
```

Preimenujemo obe tabeli,
da se lahko sklicujemo na
vsako posebej

< je namesto <> zato, da ne izpisujemo
pomenskih duplikatov (npr. (A,C) in (C,A)).

Primeri

(9) (7. vprašanje)

stranka	(sid, sime, spriimek, mesto, popust)
agent	(aid, aime, apriimek, mesto, procent)
izdelek	(iid, iime, zaloga, cena)
narocilo	(nid, sid, aid, iid, datum, kosov)



- Izpiši idente izdelkov, ki sta jih naročili vsaj 2 stranki:

```
SELECT DISTINCT n1.iid
FROM narocilo n1, narocilo n2
WHERE n1.iid = n2.iid AND n1.sid <> n2.sid;
```



Zaradi tega primerjalnega operatorja se sicer pojavijo duplikati, ampak jih DISTINCT odstrani. Namesto $<>$, je lahko tudi $<$ ali $>$.

Primeri

(10) (8. vprašanje)

stranka	(sid, sime, spriimek, mesto, popust)
agent	(aid, aime, apriimek, mesto, procent)
izdelek	(iid, iime, zaloga, cena)
→ narocilo	(nid, sid, aid, iid, datum, kosov)

- Izpiši idente strank, ki so naročile vsaj 1 izdelek, ki ga je prodal tudi agent A6:

narocilo n1		narocilo n2	
sid	iid	aid	sid
?	x		

sid	iid	aid
	x	A6

```
SELECT DISTINCT n1.sid
FROM narocilo n1, narocilo n2
WHERE n1.iid = n2.iid AND n2.aid = 6;
```

Primerjalni operatorji (1)

Operator	Opis
=	je enako
!= ali <>	različno
> in >=	večje in večje ali enako
< in <=	manjše in manjše ali enako
IN (...)	ustreza poljubnemu elementu množice
BETWEEN ... AND ...	med dvema vrednostima
LIKE '...'	ustreza vzorcu niza
IS NULL	neznana vrednost
NOT ...	negacija predhodnega operatorja

Primerjalni operatorji

(2) (BETWEEN)

○ **BETWEEN**

- namesto takšnega zapisa

```
SELECT      a1, a2, ...
FROM        t1, t2, ...
WHERE       t1.a1 >= vrednost1 AND t1.a1 <= vrednost2;
```

- lahko uporabimo naslednjega

```
SELECT      a1, a2, ...
FROM        t1, t2, ...
WHERE       t1.a1 BETWEEN vrednost1 AND vrednost2;
```

Primerjalni operatorji

(3) (LIKE)

○ LIKE

```
SELECT    a1, a2, ...
FROM      t1, t2, ...
WHERE     t1.a1 LIKE '43%' OR t1.a1 LIKE '4_';
```

Poljubno nadaljevanje niza,
ki se začne s predpono "43"
(npr. **43**, **43xy**, **4321**, **43-1** itd.)

Ujemanje enega znaka v nizu, ki se
začne s "4" (npr. **4x**, **42**, **4-** itd.)

○ Kako v nizu poiščemo znaka % in _?

- Uporabimo „ubežni znak“ (escape character), npr.
želimo izpisati nize, ki vsebujejo podniz "SA_".

```
SELECT    employee_id, last_name, job_id
FROM      employees
WHERE     job_id LIKE '%SA\_%' ESCAPE '\';
```

Vaje

Priprava podatkovne baze in enostavna
povpraševanja.

Priprava podatkovne baze

- Kreirajte tabele s pomočjo SQL skripte **kreiraj_GSM.sql**.

```
CREATE TABLE prodaja (operator VARCHAR(10), telefon VARCHAR(10));  
CREATE TABLE kupuje (stranka VARCHAR(10), operator VARCHAR(10));  
CREATE TABLE najraje (stranka VARCHAR(10), telefon VARCHAR(10));
```

```
INSERT INTO kupuje (stranka, operator) VALUES ('Marko', 'Mobitel');  
INSERT INTO kupuje (stranka, operator) VALUES ('Marko', 'Simobil');
```

...

```
INSERT INTO najraje (stranka, telefon) VALUES ('Petra', 'Nokia');  
INSERT INTO najraje (stranka, telefon) VALUES ('Janez', 'Siemens');
```

...

```
INSERT INTO prodaja (operator, telefon) VALUES ('Simobil', 'SE');  
INSERT INTO prodaja (operator, telefon) VALUES ('Simobil', 'Siemens');
```

Vprašanja (1)

1. Katere stranke kupujejo pri operaterju Mobitel?
2. Pri katerih mobilnih operaterjih kupuje Petra?
3. Katere mobilne telefone lahko kupuje Petra, glede na to, da kupuje le pri nekaterih operaterjih, pri katerih prodajajo le nekatere vrste mobilnih telefonov?

Vprašanja (2)

4. Pri katerih operaterjih prodajajo Janezov najljubši telefon?
5. Izpišite katere telefone prodaja posamezni operater. Izpis naj ima naslednjo obliko:

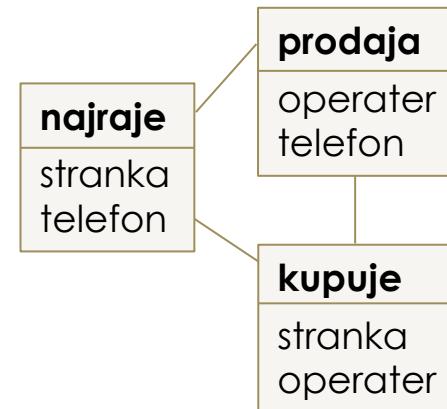
Operater prodaja telefon

Mobitel prodaja telefone znamke Nokia

...

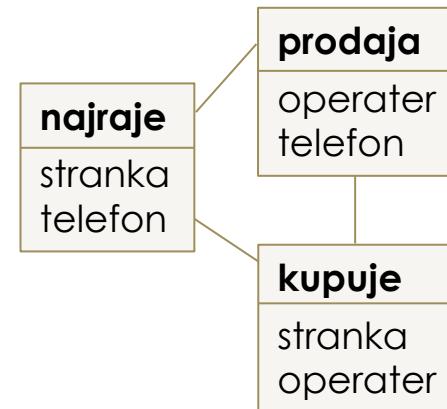
Rešitve (1) (1. vprašanje)

- Katere stranke kupujejo pri operaterju Mobitel?



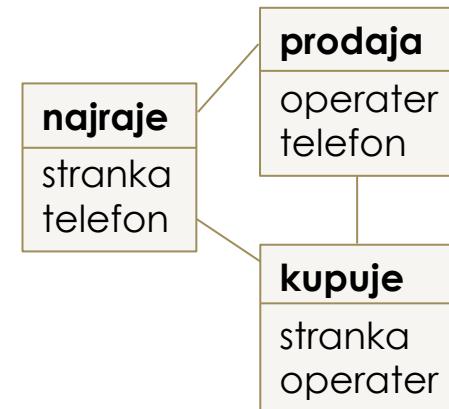
Rešitve (2) (2. vprašanje)

- Pri katerih mobilnih operaterjih kupuje Petra?



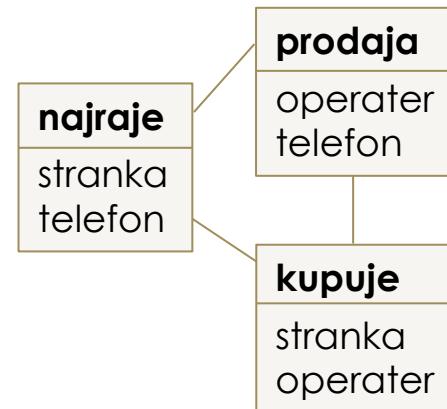
Rešitve (3) (3. vprašanje)

- Katere mobilne telefone lahko kupuje Petra, glede na to, da kupuje le pri nekaterih operaterjih, pri katerih prodajajo le nekatere vrste mobilnih telefonov?



Rešitve (4) (4. vprašanje)

- Pri katerih operaterjih prodajajo Janezov najljubši telefon?



Rešitve

(5) (5. vprašanje)

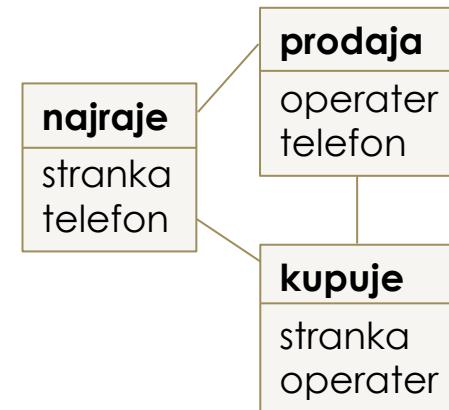
- Izpišite katere telefone prodaja posamezni operater. Izpis naj ima naslednjo obliko:

Operator prodaja telefon

Mobitel prodaja telefone znamke Nokia

...

- Za združevanje nizov v selekcijskem pogoju uporabite funkcijo **CONCAT**



Poizvedovalni jezik SQL

Poizvedovanje po relacijski podatkovni bazi.

2. del

Množice

(1) (9. vprašanje)

- **stranka** (sid, sime, spriimek, mesto, popust)
- **agent** (aid, aime, apriimek, mesto, procent)
- **izdelek** (iid, iime, zaloga, cena)
- **narocilo** (nid, sid, aid, iid, datum, kosov)

Ime in popust strank,
ki so kaj naročile pri
agentu iz Kranja oz. Kopra.

- Izpiši imena
in popuste
strank, ki so
naročile kaj
pri agentu
iz Kranja ali
Kopra:

```
SELECT  
FROM  
WHERE
```

```
);
```

```
sime, popust  
stranka  
sid IN (
```

```
SELECT  
FROM  
WHERE
```

```
)
```

ID strank, ki so naročile pri
agentih iz Kranja oz. Kopra.

```
SELECT  
FROM  
WHERE
```

aid
agent
mesto IN ('Kranj', 'Koper')

ID agentov, ki so iz Kranja oz. Kopra.

Množice

(2) (10. vprašanje)

- **stranka** (sid, sime, spriimek, mesto, popust)
- **agent** (aid, aime, apriimek, mesto, procent)
- **izdelek** (iid, iime, zaloga, cena)
- **narocilo** (nid, sid, aid, iid, datum, kosov)

○ Izpiši imena strank, ki so naročile izdelek, ki ima iid = 2:

```
SELECT      sime
FROM        stranka
WHERE       2 IN (
    SELECT      iid
    FROM        narocilo
    WHERE       sid = stranka.sid
);
```

sid se nanaša na
najbližjo tabelo
(narocilo).

Izdelek z iid=2 je med izdelki, ki jih je naročila stranka.

Množice

(3) (10. vprašanje)

- **stranka** (sid, sime, spriimek, mesto, popust)
- **agent** (aid, aime, apriimek, mesto, procent)
- **izdelek** (iid, iime, zaloga, cena)
- **narocilo** (nid, sid, aid, iid, datum, kosov)

- Izpiši imena strank, ki so naročile izdelek, ki ima iid = 2
(2. način)

```
SELECT      DISTINCT sime
FROM        stranka, narocilo
WHERE       stranka.sid = narocilo.sid AND narocilo.iid = 2;
```

Operatorji nad množicami (1)

- Operatorji nad množicami so:

Operator	Opis
UNION	vse vrstice iz vseh množic (brez duplikatov)
UNION ALL	vse vrstice iz vseh množic (z duplikati)
MINUS	vrstice, ki so samo v prvi množici
INTERSECT	presek množic

(SELECT ...)
UNION
(SELECT ...)

- MySQL ne pozna **MINUS** in **INTERSECT!**
- **IN**, **ALL** in **ANY** so „multiple row“ operaterji
 - vrednost primerjajo z vsemi vrednostmi iz podane poizvedbe

Operatorji nad množicami (2)

- Primeri za ANY:

```
SELECT ...  
FROM ...  
WHERE x < ANY (  
    SELECT y  
    FROM ...  
    WHERE ...  
);
```

manj kot
maksimum

```
SELECT ...  
FROM ...  
WHERE x <> ANY (  
    SELECT y  
    FROM ...  
    WHERE ...  
);
```

```
SELECT ...  
FROM ...  
WHERE x = ANY (  
    SELECT y  
    FROM ...  
    WHERE ...  
);
```

vsebovanost
(IN)

Operaterji nad množicami (3)

Primeri za ALL:

```
SELECT ...  
FROM ...  
WHERE x < ALL (  
    SELECT y  
    FROM ...  
    WHERE ...  
);
```

manj kot
minimum

```
SELECT ...  
FROM ...  
WHERE x > ALL (  
    SELECT y  
    FROM ...  
    WHERE ...  
);
```

več kot
maksimum

```
SELECT ...  
FROM ...  
WHERE x <> ALL (  
    SELECT y  
    FROM ...  
    WHERE ...  
);
```

ni vsebovanosti
(NOT IN)

Množice

(4) (11. vprašanje)

- **stranka** (sid, sime, spriimek, mesto, popust)
- **agent** (aid, aime, apriimek, mesto, procent)
- **izdelek** (iid, iime, zaloga, cena)
- **narocilo** (nid, sid, aid, iid, datum, kosov)

- Izpiši idente tistih agentov, ki imajo najnižji odstotek provizije:

```
SELECT      aid
FROM        agent
WHERE       procent <= ALL (
    SELECT      procent
    FROM        agent
);

```

Procent mora biti **manjši ali enak vsem procentom**,
t.j. manjši od vseh ostalih in enak najnižjemu.

Množice

(5) (12. vprašanje)

- **stranka** (sid, sime, spriimek, mesto, popust)
- **agent** (aid, aime, apriimek, mesto, procent)
- **izdelek** (iid, iime, zaloga, cena)
- **narocilo** (nid, sid, aid, iid, datum, kosov)

- Izpiši ident agentov, ki nimajo najvišjega odstotka provizije:

```
SELECT      aid
FROM        agent
WHERE       procent < ANY (
SELECT      procent
FROM        agent
);
```

Procent mora biti **manjši od določenega procenta**,
t.j. vsi procenti, ki niso najvišji.

Razširitev osnovne oblike SQL stavka (1)

```
SELECT      A1, A2, ..., Ak
FROM        T1, T2, ..., Tn
WHERE       P1
GROUP BY    A1, ..., Ag
HAVING      P2
ORDER BY    A2, ..., Ao [DESC | ASC];
```

urejanje izpisa

← grupiranje po atributih
← omejevanje grup

- Vrstni red operacij je pomemben in sicer: **GROUP BY, HAVING** in **ORDER BY** (in po potrebi **LIMIT**).

Razširitev osnovne oblike SQL stavka (2)

- **GROUP BY** omogoča operacije nad več vrsticami

Funkcija	Opis
AVG	povprečje
COUNT	število vrednosti v stolpcu, različne od NULL
MAX	največja vrednost (štetila, nizi, datumi)
MIN	najmanjša vrednost (štetila, nizi, datumi)
SUM	vsota
STDDEV	standardna deviacija
VARIANCE	varianca

Razširitev osnovne oblike SQL stavka (3)

- **GROUP BY** primeri
 - duplikati

```
SELECT COUNT(iid)  
FROM narocila;
```

```
SELECT COUNT(DISTINCT iid)  
FROM narocila;
```

- atributi

```
SELECT major, SUM(tuition_paid)  
FROM student  
GROUP BY major;
```

atribut po katerem
grupiramo
(lahko ga izpustimo)

Razširitev osnovne oblike SQL stavka (4)

- **HAVING** za omejevanje grup

```
SELECT major, SUM(tuition_paid)  
FROM student  
GROUP BY major  
HAVING SUM(tuition_paid) > 2500;
```

- **WHERE** deluje nad vrsticami,
HAVING deluje nad skupinami!

Zanimajo nas samo tisti predmeti, kjer skupna plačana šolnina večja od 2.500 €.

Razširitev osnovne oblike SQL stavka (5)

○ ORDER BY

- za izpis vrstic po (abecednem) vrstnem redu
- urejamo lahko po več stolpcih

...

ORDER BY atribut1, atribut2, ...

- privzeto urejanje je naraščajoče
- če želimo padajoče uporabimo

...

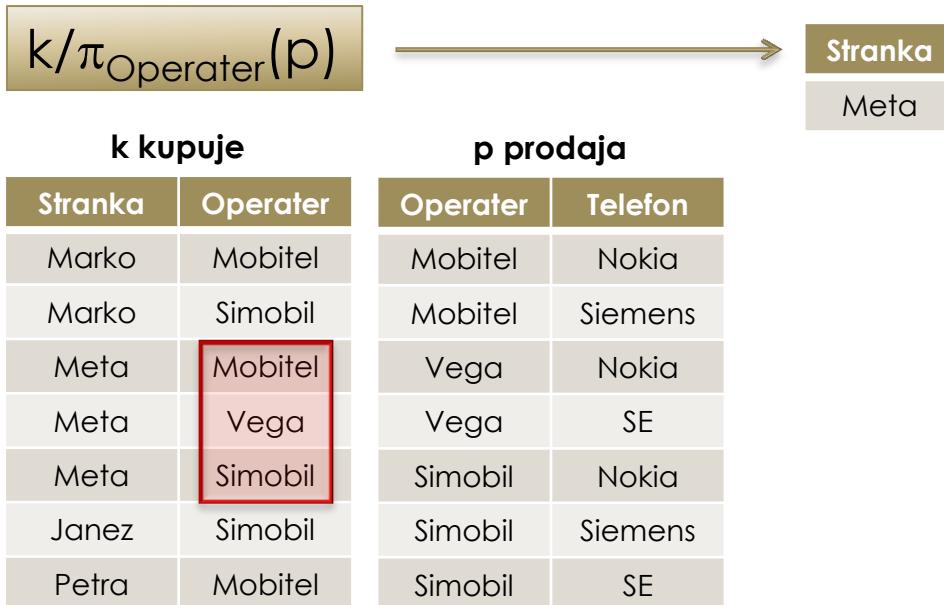
ORDER BY atribut1 **DESC**

EXISTS in NOT EXISTS

- Uporaba rezerviranih besed **EXISTS**
 - preverja obstoj vrstic v rezultatu ugnezdrne poizvedbe
 - dovolj je že **obstoj ene vrstice** in je logična vrednost **true**.
- in **NOT EXISTS**
 - preverja ali je rezultat ugnezdrne poizvedbe prazna množica

Izvedba operacije deljenja (1)

- Primer poizvedbe iz relacijske algebре:
Katere stranke kupujejo pri vseh operaterjih, ki so navedeni v relaciji p?



$k / \pi_{\text{Operater}}(p)$

k kupuje

Stranka	Operater
Marko	Mobitel
Marko	Simobil
Meta	Mobitel
Meta	Vega
Meta	Simobil
Janez	Simobil
Petra	Mobitel

p prodaja

Operater	Telefon
Mobitel	Nokia
Mobitel	Siemens
Vega	Nokia
Vega	SE
Simobil	Nokia
Simobil	Siemens
Simobil	SE

Izvedba operacije deljenja (2)

- Primer poizvedbe iz relacijske algebре:
Katere stranke kupujejo pri vseh operaterjih, ki so navedeni v relaciji p?

```
SELECT DISTINCT stranka
  FROM kupuje k1
 WHERE NOT EXISTS (
    SELECT * 
      FROM prodaja
     WHERE NOT EXISTS (
        SELECT * 
          FROM kupuje k2
         WHERE k1.stranka = k2.stranka AND
               k2.operator = prodaja.operator ));
```

Operaterji, kateri izbrani stranki ne prodajajo telefonov.

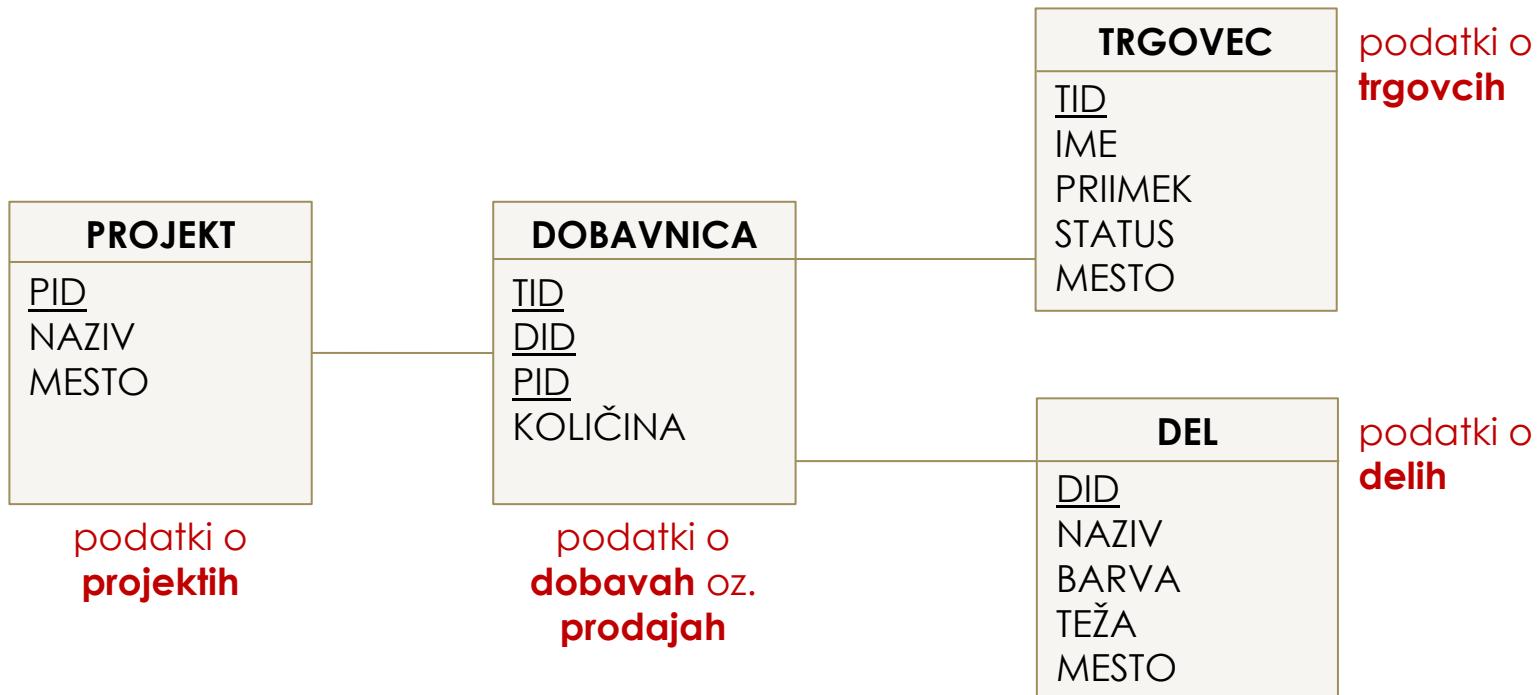
Operaterji pri katerih kupuje izbrana stranka.

Ne sme obstajati operater, pri katerem izbrana stranka ne kupuje.

Vaje

Povpraševanje po podatkovni bazi.

Problemska domena DOBAVITELJ



Vprašanja (1)

1. Poščite imena in priimke tistih trgovcev, ki imajo status večji od 20 in so iz Kranja?
2. Poščite nazine delov, katerih teža znaša vsaj 17kg in so rdeči.
3. Poščite imena in priimke trgovcev, ki so v okviru ene dobave prodali največ kosov določenega dela.
4. Poščite imena in priimke trgovcev, ki so prodajali projektom iz Novega mesta.

Vprašanja (2)

5. Poишčite nazine delov, ki se do sedaj še niso prodajali.
6. Poишčite nazine delov, ki sta jih prodajala vsaj dva različna trgovca.
7. Poишčite nazine delov, ki so jih kupovali hkrati projekti iz Kranja in Maribora.

Rešitve

(1) (1. vprašanje)

- Poишčite imena in priimke tistih trgovcev, ki imajo status večji od 20 in so iz Kranja?



TRGOVEC (TID, IME, PRIIMEK, STATUS, MESTO)
DEL (DID, NAZIV, BARVA, TEŽA, MESTO)
PROJEKT (PID, NAZIV, MESTO)
DOBAVNICA (TID, DID, PID, KOLIČINA)

Rešitve

(2) (2. vprašanje)

- Poишčite nazive delov, katerih teža znaša vsaj 17kg in so rdeči.



TRGOVEC (TID, IME, PRIIMEK, STATUS, MESTO)
DEL (DID, NAZIV, BARVA, TEŽA, MESTO)
PROJEKT (PID, NAZIV, MESTO)
DOBAVNICA (TID, DID, PID, KOLIČINA)

Rešitve (3) (3. vprašanje)

- Poишčite imena in priimke trgovcev, ki so v okviru ene dobave prodali največ kosov določenega dela.



TRGOVEC (TID, IME, PRIIMEK, STATUS, MESTO)
DEL (DID, NAZIV, BARVA, TEŽA, MESTO)
PROJEKT (PID, NAZIV, MESTO)
DOBAVNICA (TID, DID, PID, KOLIČINA)

Rešitve (4) (4. vprašanje)

- Poишčite imena in priimke trgovcev, ki so prodajali projektom iz Novega mesta.



TRGOVEC (TID, IME, PRIIMEK, STATUS, MESTO)
DEL (DID, NAZIV, BARVA, TEŽA, MESTO)
PROJEKT (PID, NAZIV, MESTO)
DOBAVNICA (TID, DID, PID, KOLIČINA)

Rešitve

(5) (5. vprašanje)

- Poишčite nazine delov, ki se do sedaj še niso prodajali.



TRGOVEC (TID, IME, PRIIMEK, STATUS, MESTO)
DEL (DID, NAZIV, BARVA, TEŽA, MESTO)
PROJEKT (PID, NAZIV, MESTO)
DOBAVNICA (TID, DID, PID, KOLIČINA)

Rešitve

(6) (6. vprašanje)

- Poишčite identifikatorje in nazine delov, ki sta jih prodajala vsaj dva različna trgovca.



TRGOVEC (TID, IME, PRIIMEK, STATUS, MESTO)
DEL (DID, NAZIV, BARVA, TEŽA, MESTO)
PROJEKT (PID, NAZIV, MESTO)
DOBAVNICA (TID, DID, PID, KOLIČINA)

Rešitve (7) (7. vprašanje)

- Poишčite nazine delov, ki so jih kupovali hkrati projekti iz Kranja in Maribora.



TRGOVEC (TID, IME, PRIIMEK, STATUS, MESTO)
DEL (DID, NAZIV, BARVA, TEŽA, MESTO)
PROJEKT (PID, NAZIV, MESTO)
DOBAVNICA (TID, DID, PID, KOLIČINA)

Poizvedovalni jezik SQL

Skupine SQL ukazov

3. del

Skupine SQL ukazov

- **DDL** (Data Definition Language)
 - CREATE, ALTER, DROP
- **DCL** (Data Control Language)
 - GRANT, REVOKE
- **DML** (Data Manipulation Language)
 - SELECT, INSERT, UPDATE, DELETE
- **TPO** (Transaction Processing Option)
 - COMMIT, ROLLBACK

Skupine SQL ukazov

(1) (DDL)

- Ukazi skupine DDL so:
 - CREATE** za kreiranje objektov (TABLE, USER, VIEW itd.)

```
CREATE TABLE novaTabela (
    id INTEGER NOT NULL,
    ime VARCHAR(30) NOT NULL,
    priimek VARCHAR(30) NOT NULL,
    starost INTEGER NULL);
```

ime tabele

ime atributa

podatkovni tip atributa

obveznost atributa

The diagram shows a CREATE TABLE statement with three annotations: 'ime tabele' with a red arrow pointing to the table name 'novaTabela'; 'ime atributa' with a red arrow pointing to the first column 'id'; and 'obveznost atributa' with a red arrow pointing to the fourth column 'starost' which has a 'NULL' constraint.

Skupine SQL ukazov

(2) (DDL)

- Ukazi skupine DDL so:
 - ALTER** za spreminjanje objektov in njihovih lastnosti

Dodajanje stolpcev

```
ALTER TABLE novaTabela  
ADD (naslov VARCHAR(50));
```

Spreminjanje stolpcev

```
ALTER TABLE novaTabela  
MODIFY (naslov VARCHAR(70));
```

Brisanje stolpcev

```
ALTER TABLE novaTabela  
DROP COLUMN naslov;
```

Skupine SQL ukazov

(3) (DDL)

- Ukazi skupine DDL so:
 - DROP** za uničenje objektov in njihovih lastnosti

```
DROP TABLE novaTabela;
```

Kopiranje tabel

- Kopiranje vsebine tabele v novo tabelo.
 - Prenese se relacijska shema in podatki.
 - Integritetne omejitve se ne prenesejo.

```
CREATE TABLE novaTabela AS  
SELECT *  
FROM staraTabela;
```

Skupine SQL ukazov

(4) (DDL)

- Manipulacija z uporabniki:
 - kreiranje

```
CREATE      USER      uporabnik  
IDENTIFIED BY      kolme      geslo;
```

- brisanje

```
DROP      USER      uporabnik  
kolme;
```

- spreminjanje gesla

```
ALTER      USER      uporabnik  
IDENTIFIED BY      kolme      novoGeslo;
```

Skupine SQL ukazov

(5) (DCL)

- Ukazi skupine DCL so:
 - GRANT** za podeljevanje pravic.

```
GRANT imePrivilegija ON imeObjekta TO uporabniskolme;
```

```
GRANT ALL ON ab1234.* TO 'uporbnik'@'localhost';
```

- REVOKE** za odvzem pravic.

```
REVOKE imePrivilegija ON imeObjekt FROM uporabniskolme;
```

```
REVOKE ALL PRIVILEGES FROM uporabniskolme
```

Skupine SQL ukazov

(6) (DML)

- Ukazi skupine DML so:
 - **SELECT** za poizvedovanje.
 - **INSERT** za vnašanje (najmanjša enota je vrstica).
 - **UPDATE** za posodabljanje (najmanjša enota je stolpec v vrstici).
 - **DELETE** za brisanje (najmanjša enota je vrstica).

Skupine SQL ukazov

(7) (DML)

- Ukazi skupine DML so:
 - INSERT** za vnašanje (najmanjša enota je vrstica).

```
INSERT INTO Projekt (pid, naziv, mesto)  
VALUES (3010, 'Tiskalnik', 'Ljubljana');
```

- Tabelo lahko napolnimo tudi na podlagi vrednosti iz druge tabele.

```
INSERT      INTO novaTabela  
          SELECT      *  
          FROM        staraTabela;
```

Skupine SQL ukazov

(8) (DML)

- Ukazi skupine DML so:
 - UPDATE** za posodabljanje (najmanjša enota je stolpec v vrstici).

```
UPDATE projekt
SET naziv = 'Sortirnik'
WHERE naziv = 'Sortirni stroj';
```

← Pomembno, da ne izpustimo pogoja, sicer se posodobijo vse vrstice v tabeli!

Skupine SQL ukazov

(9) (DML)

- Ukazi skupine DML so:
 - DELETE** za brisanje (najmanjša enota je vrstica).

```
DELETE      FROM projekt  
WHERE        naziv = 'Sortirni stroj';
```

← Pomembno, da ne izpustimo pogoja, sicer se izbrišejo vse vrstice v tabeli!
Če iz tabele izbrišemo vse vrstice, tabela še vedno obstaja!

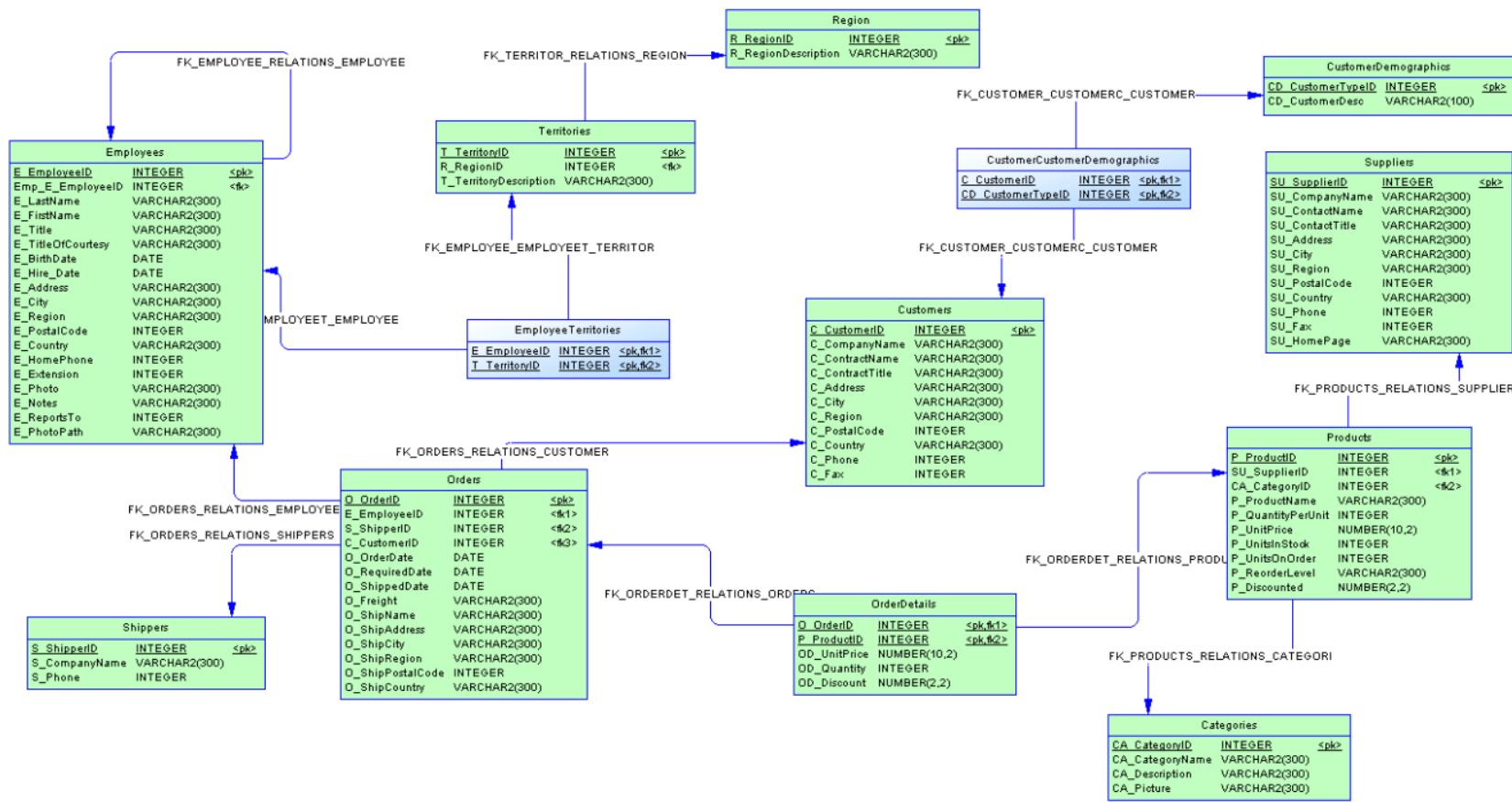
Skupine SQL ukazov (10) (TPO)

- Ukazi skupine TPO so:
 - **COMMIT** za uveljavljanje transakcije in postanejo vidni drugim uporabnikom.
 - **ROLLBACK** za razveljavitev transakcije od zadnje uveljavitve naprej.
 - S pomočjo teh ukazov zagotavljamo **konsistentnost** podatkovne baze!

Vaje

Povpraševanje po podatkovni bazi.

Problemska domena NORTHWIND



Vprašanja (1)

1. Poisci imena in priimke vseh zaposlenih v regiji z identifikacijsko številko 1, katerih ime se začne s črko M.
2. Za vsako kategorijo poiščite število različnih strank, ki so kdajkoli naročile kaj iz te kategorije. Izpišite ime kategorije in število različnih strank, vendar zgolj tiste, za katere obstaja več kot 80 takih strank.

Vprašanja (2)

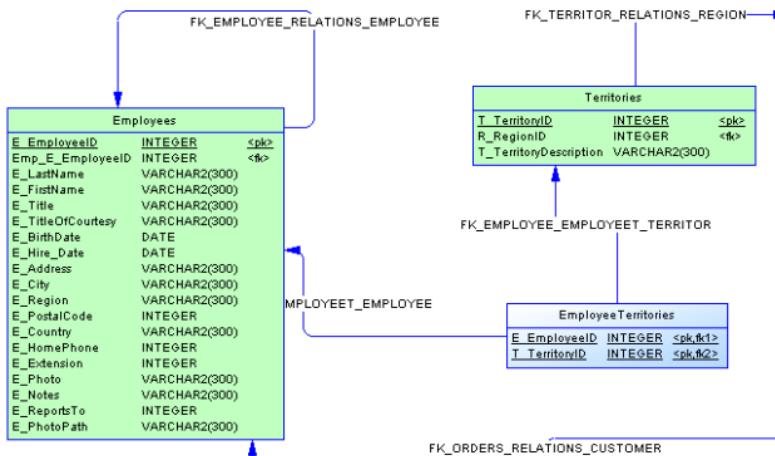
3. Poisci identifikacijske številke strank, ki so že kdaj naročile produkte iz vsaj tretjine vseh kategorij (ni nujno da v enem naročilu).
4. Poisci identifikacijske številke zaposlenih, ki so obravnavali največje število naročil.
5. Izpiši imena in priimke zaposlenih, ki niso nikoli obravnavali naročila za stranko, ki bi bila iz istega mesta kot zaposleni.

Vprašanja (3)

6. Izpišite identifikacijsko številko ter skupno ceno naročila, za vsa naročila, ki vsebujejo vsaj en produkt, ki spada v kategorijo z identifikacijsko številko 6.
- Skupno ceno izračunate kot vsoto $\sum_i k_i c_i (1 - p_i)$, kjer je k_i količina produkta i , c_i cena posameznega kosa ter p_i odobren popust za ta produkt.
 - Skupno ceno naročila zaokrožite na celo število s pomočjo funkcije ROUND(X).

Rešitve (1) (1. vprašanje)

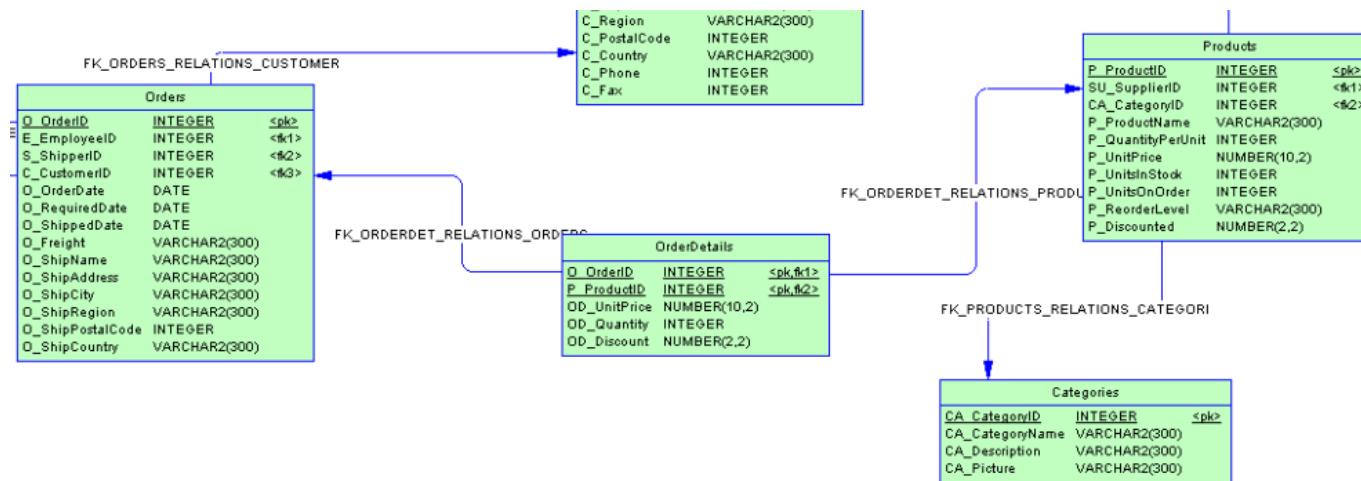
- Poisci imena in priimke vseh zaposlenih v regiji z identifikacijsko številko 1, katerih ime se začne s črko M.



Rešitve

(2) (2. vprašanje)

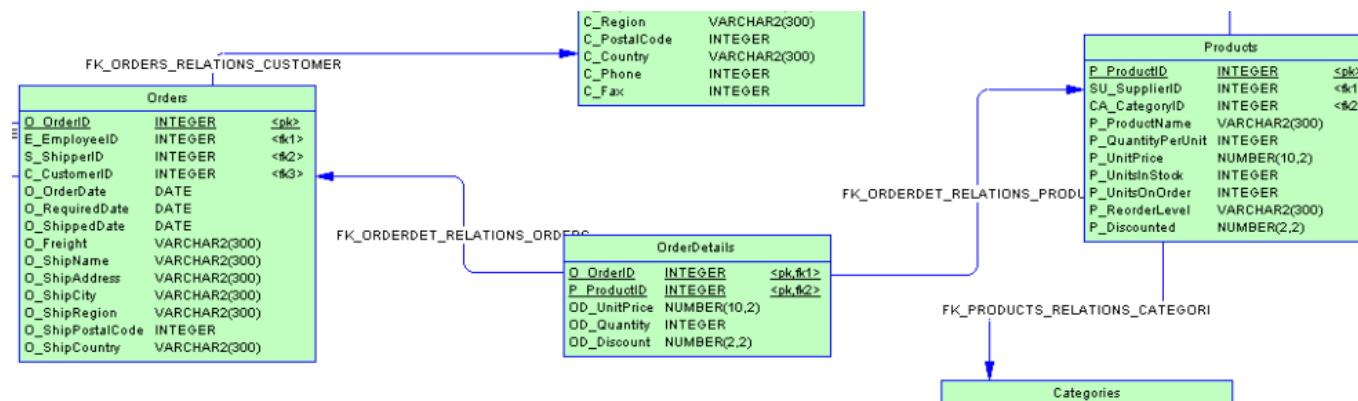
- Za vsako kategorijo poiščite število različnih strank, ki so kdajkoli naročile kaj iz te kategorije. Izpišite ime kategorije in število različnih strank, vendar zgolj tiste, za katere obstaja več kot 80 takih strank.



Rešitve

(3) (3. vprašanje)

- Poisci identifikacijske številke strank, ki so že kdaj naročile produkte iz vsaj tretjine vseh kategorij (ni nujno da v enem naročilu).



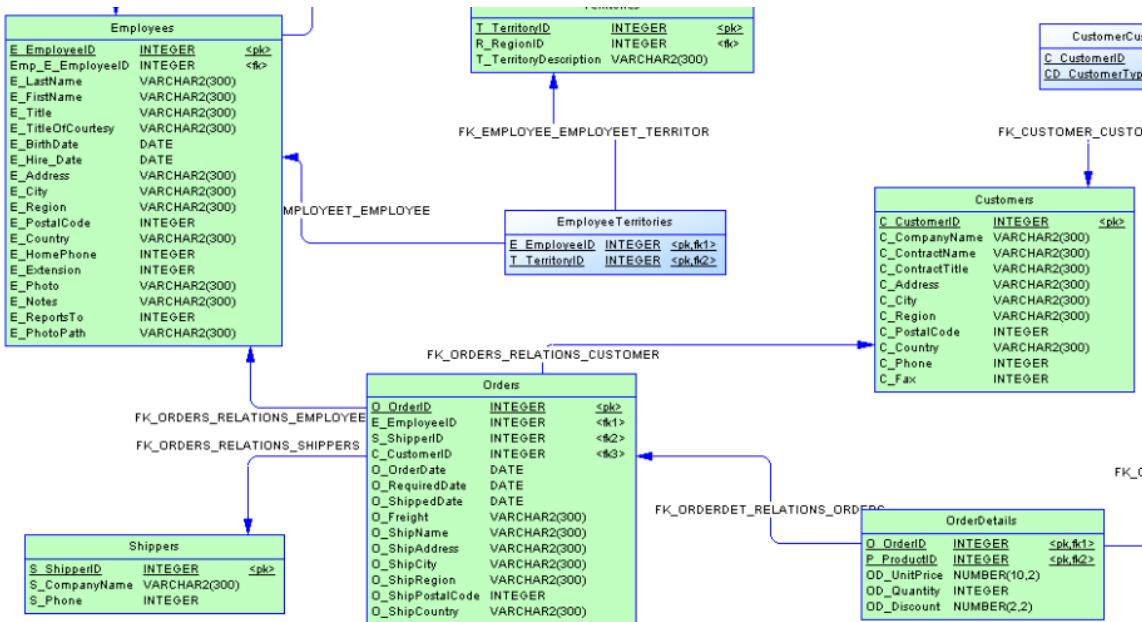
Rešitve (4) (4. vprašanje)

- Poisci identifikacijske številke zaposlenih, ki so obravnavali največje število naročil.

Orders		
O_OrderID	INTEGER	<pk>
E_EmployeeID	INTEGER	<%1>
S_ShipperID	INTEGER	<%2>
C_CustomerID	INTEGER	<%3>
O_OrderDate	DATE	
O_RequiredDate	DATE	
O_ShippedDate	DATE	
O_Freight	VARCHAR2(300)	
O_ShipName	VARCHAR2(300)	
O_ShipAddress	VARCHAR2(300)	
O_ShipCity	VARCHAR2(300)	
O_ShipRegion	VARCHAR2(300)	
O_ShipPostalCode	INTEGER	
O_ShipCountry	VARCHAR2(300)	

Rešitve (5) (5. vprašanje)

- Izpišite imena in priimke zaposlenih, ki niso nikoli obravnavali naročila za stranko, ki bi bila iz istega mesta kot zaposleni.



Rešitve (6) (6. vprašanje)

- Izpišite identifikacijsko številko ter skupno ceno naročila, za vsa naročila, ki vsebujejo vsaj en produkt, ki spada v kategorijo z identifikacijsko številko 6.
 - Skupno ceno izračunate kot vsoto $\sum_i k_i c_i(1 - p_i)$, kjer je k_i količina produkta i , c_i cena posameznega kosa ter p_i odobren popust za ta produkt.
 - Skupno ceno naročila zaokrožite na celo število s pomočjo funkcije ROUND(X).

Rešitve

(7) (6. vprašanje)

