

Slaven Sarić

Visoka poslovna škola Minerva
Studij za informatički menadžment

SQL

Riješeni primjeri

Sadržaj

Naredba SELECT	4
Naredba WHERE	7
Naredba ORDER BY	13
Funkcije	18
Naredbe GROUP BY i HAVING	25
Naredba JOIN.....	34
Naredbe INSERT, UPDATE i DELETE.....	46
Naredbe CREATE TABLE, ALTER TABLE i DROP TABLE.....	49

SQL (Structured Query Language) je deklarativni programski jezik namijenjen radu s relacijskim bazama podataka. Upravo je njegova deklarativnost ono što ga razlikuje od "pravih", imperativnih programskih jezika kao što su npr. C++, C# ili Java. Za razliku od ovih drugih, u SQL se ne navodi kako uraditi određenu operaciju nad podacima nego se samo navodi što se želi dobiti izvršenjem takve jedne operacije. U imperativnom programskom jeziku moramo točno naredbama navesti kako i po kojem ključu dohvatiti određene podatke iz tablice relacijske baze podataka, dok u SQL samo navedemo što želimo, a RDBMS (Relational Database Management System) sustav će se pobrinuti kako to izvesti i usput eventualno optimirati.

Naravno, često nije jednostavno opisati što se želi dobiti, osobito jer za opis možemo koristiti samo relativno malen skup ključnih riječi (red veličine desetak). Malen broj naredbi, međutim nipošto ne ograničava složenost upita koji se mogu generirati. Standardno kao i u svemu, jednostavne SQL upite je jednostavno razumjeti, dok je složene upite složeno kako razumjeti tako i kreirati. Popularnost SQL jezika dokazuju i uspješni pokušaji integracije SQL naredbi unutar sintakse imperativnih jezika kao što je slučaj s LINQ tehnologijom unutar C# programskog jezika na .Net platformi.

U nastavku ćemo kroz veći broj primjera pokazati SQL naredbe na primjeru tablica narudžbi i kupaca iz standardne i dobro poznate baze podataka *Northwind*. Tablice *Narudzbe* i *Kupci* nalaze se u relacijskoj vezi jedan prema više (one to many) i kupac koji je izvršio narudžbu zapisan je u tablici *Narudzbe* kao polje *KupacID*.

Poglavlje je organizirano na način da se pokažu primjeri s određenom SQL naredbom, ali tako da se u svakom od primjera koriste samo do tog trenutka obrađene SQL naredbe. Primjeri su izrađeni u MS Access i SQL Server RDBMS sustavu, a iako se sintaksa i skup SQL naredbi donekle razlikuju kod različitih proizvođača, velika većina primjera bi se trebala izvršiti bez greške bez obzira o koje se RDBMS sustavu radi.

SQL nije osjetljiv na velika i mala slova, ali bez obzira na to mi ćemo u našim primjerima razumljivo, biti konzistentni te ćemo sve ključne riječi SQL jezika pisati velikim slovima, dok ćemo za nazive tablica i polja koristiti tzv. Pascal notaciju po kojoj svaka pojedina riječ u identifikatoru (riječi ne smiju biti odvojene razmakom!) počinje velikim slovom kao npr: Ime, Prezime, Narudzbe, DatumNarudzbe, JakostStruje, SnagaMotora, itd...

Naredba SELECT

Naredba SELECT je najupotrebljavanija SQL naredba kojom se dohvaća određeni skup podataka iz tablice. Sintaksa naredbe SELECT je sljedeća:

SELECT polje1, polje2, ... **FROM** Naziv_Tablice

gdje su:

- o polje1, polje2, ... – nazivi polja koja se dohvaćaju
- o Naziv_Tablice – naziv tablice koja sadrži podatke koji se izdvajaju

Nakon ključne riječi SELECT navode se, odvojena zarezom, polja iz sloga koja se žele dohvatiti. Ako se žele dohvatiti sva polja iz sloga nije ih potrebno sve navesti, dovoljno je navesti znak '*'. Nakon ključne riječi FROM navodi se tablica baze podataka koja je izvor podataka. Kasnije ćemo vidjeti da to ne mora biti stvarna tablica iz baze podataka nego npr. neka druga SELECT naredba.

Zadaci

1. Napišite SQL naredbu koja će izdvojiti sve narudžbe iz tablice **Kupci**. Prikažite sva polja o pojedinom kupcu.

Napomena: ako u zadacima u nastavku nije naglašeno koja polja treba izdvojiti, potrebno je izdvojiti sva polja iz tablice.

SELECT * FROM Kupci

ID	Ime	Prezime	Adresa
1	Maria	Anders	Obere Str. 57, Bogota
2	Antonio	Moreno	Mataderos 2312, San Jose
3	Hanna	Moos	Forsterstr. 57, Amsterdam
4	Elizabeth	Lincoln	Fauntleroy Circus, London
5	Yang	Wang	87 Polk St, Singapur
6	Patricio	Simpson	Moncada 8585, Buenos Aires
7	Horst	Kloss	Taucherstrasse 10, Dortmund
8	Elizabeth	Brown	2743 Bering St., Boston
9	James	Ashworth	89 Jefferson Way, Portland
10	Pascale	Cartrain	Boulevard Tirou, 255, Lyon

Prvi primjer nema veliku upotrebnu vrijednost jer će kao rezultat dati sva polja svih slogova tablice *Kupci*, a to isto možemo dobiti ako tablicu jednostavno otvorimo u *view* modu.

Ipak, ovo je primjer koji predstavlja potpunu SQL naredbu i obično je prvi primjer u svakom opisu SQL jezika.

2. Napišite SQL naredbu koja će izdvojiti sve kupce iz tablice **Kupci**. Prikažite sljedeća polja: **Ime kupca** i **Prezime kupca**.

SELECT Ime, Prezime FROM Kupci

Ime	Prezime
Maria	Anders
Antonio	Moreno

Često nema potrebe za izdvajanjem svih polja. Zato naredba SELECT dozvoljava da joj se kao argumenti navedu imena polja čije se vrijednosti trebaju izdvojiti. Redoslijed polja ne mora biti onakav kako je zapisan u tablici, a navedena polja razdvajaju se zarezom.

Ime	Prezime
Hanna	Moos
Elizabeth	Lincoln
Yang	Wang
Patricio	Simpson
Horst	Kloss
Elizabeth	Brown
James	Ashworth
Pascale	Cartrain

3. Napišite SQL naredbu koja će iz tablice **Kupci** izdvojiti sve vrijednosti polja **Ime** ne ponavljajući ga pritom.

`SELECT DISTINCT Ime FROM Kupci`

Ime
Antonio
Elizabeth
Hanna
Horst
James
Maria
Pascale
Patricio
Yang

Primjer uvodi novu ključnu riječ DISTINCT koja omogućuje izdvajanje samo slogova sa različitim vrijednostima polja. Ključna riječ DISTINCT se odnosi na sva navedena polja tako da bi:

```
SELECT DISTINCT Ime, Prezime FROM Kupci
```

izdvojio sve kupce osim onih koji imaju isto ime i prezime.

4. Napišite SQL naredbu koja će iz tablice **Narudžbe** izdvojiti sve mjesece u kojima je roba isporučivana ne ponavljajući ih pritom.

`SELECT DISTINCT MONTHNAME(MONTH(DatumIsporuke)) AS [Mjesec] FROM Narudžbe`

Mjesec
kolovoz
lipanj
ožujak
siječanj
srpanj
svibanj
travanj
veljača

Ključna riječ tzv. alias AS nije dio SQL jezika, nego samo naznačuje pod kojim naslovom će se prikazati izdvojena polja.

Primjer pokazuje i tzv. ugnježđenje poziva funkcija prisutno u svim programskim jezicima. Argument od DISTINCT je naziv mjeseca koji se dobije pozivom funkcije MONTHNAME s funkcijom MONTH kao argumentom.

5. Napišite SQL naredbu koja će iz tablice **Kupci** izdvojiti sva različita imena i prezimena kupaca.

`SELECT DISTINCT (Ime + ' ' + Prezime) AS [Ime i prezime] FROM Kupci`

Ime i prezime
Antonio Moreno
Elizabeth Brown
Elizabeth Lincoln
Hanna Moos
Horst Kloss
James Ashworth
Maria Anders
Pascale Cartrain
Patricio Simpson
Yang Wang

6. Napišite SQL naredbu koja će iz tablice **Kupci** izdvojiti sva slova s kojim započinje prezime nekog kupca.

`SELECT DISTINCT LEFT(Prezime, 1) AS [Prvo slovo prezimena] FROM Kupci`

Prvo slovo prezimena
A
B
C
K
L
M
S
W

Funkcije LEFT i RIGHT imaju dva argumenta:

- polje iz tablice
- broj znakova polja koji će se izdvojiti (za funkciju LEFT slijeva, a za funkciju RIGHT zdesna)

Kako je u primjeru drugi argument 1, izdvojit će se sva različita prva slova prezimena kupaca.

7. Napišite SQL naredbu koja će iz tablice **Kupci** izdvojiti sva slova s kojim završava prezime nekog kupca.

`SELECT DISTINCT RIGHT(Prezime, 1) AS [Prvo slovo prezimena] FROM Kupci`

Zadnje slovo prezimena
g
h
n
o
s

Naredba WHERE

Naredba SELECT sama za sebe nema pretjeranog smisla ako se ne navedu uvjeti koje slogovi moraju udovoljiti da bi bili izdvojeni. U tu svrhu služi SQL naredba WHERE kojoj se kao argumenti navode uvjeti koje slogovi trebaju udovoljavati.

Pritom uvjeta može biti i više i tada se razdvajaju logičkim operatorima AND i/ili OR. Kao i u svim programskim jezicima operator AND zahtijeva da svi navedeni uvjeti budu istiniti, dok operator OR zahtijeva da barem jedan od uvjeta bude istinit.

Sintaksa naredbe WHERE je sljedeća:

SELECT polje1, polje2, ... **FROM** Naziv_Tablice

WHERE polje operator vrijednost

gdje su:

- o operator – predstavlja operaciju uspoređivanja (<, >, =, <>, ...)
- o vrijednost – vrijednost s kojom se uspoređuje vrijednost polja

Zadaci

8. Napišite SQL naredbu koja će izdvojiti sve iznose narudžbi manje od 100 kn ne ponavljajući ih pritom.

SELECT DISTINCT Iznos FROM Narudzbe
WHERE Iznos < 100

Iznos
88,73
97,56
99,80

Izdvajaju se svi iznosi manji od 100. Za uspoređivanje WHERE naredba koristi standardne operatore uspoređivanja:

- < manje od
- > veće od
- <= manje ili jednako od
- >= veće ili jednako od
- = jednako
- <> nije jednako

9. Napišite SQL naredbu koja će izdvojiti sve plaćene narudžbe veće od 500 kn.

SELECT * FROM Narudzbe
WHERE Placeno = True AND Iznos > 500

ID	Iznos	DatumNarudzbe	DatumIsporuke	Placeno	KupacID
4	590,78	1.2.2010	4.2.2010	True	8
13	612,00	3.4.2010	7.4.2010	True	4
20	1025,77	5.5.2010	8.5.2010	True	7
22	879,60	12.5.2010	16.5.2010	True	5
25	532,90	29.5.2010	31.5.2010	True	2
33	1300,10	14.7.2010	16.7.2010	True	10
34	578,45	20.7.2010	22.7.2010	True	4

10. Napišite SQL naredbu koja će izdvojiti sve neplaćene narudžbe kupca Hanna Moos iz Engleske. Prikažite sljedeća polja: **Iznos narudžbe, Datum Narudžbe i Datum Isporuke.**

```
SELECT Iznos, DatumNarudzbe, DatumIsporuke FROM Narudzbe  
WHERE KupacID = 3 AND Placeno = False
```

Iznos	DatumNarudzbe	DatumIsporuke
800,00	11.3.2010	15.3.2010
290,78	29.4.2010	2.5.2010
209,80	18.6.2010	22.6.2010

U ovom primjeru operatorom AND uvjetuje se da se izdvoje samo računi koji u polju *KupacID* imaju vrijednost 3 te koji za polje *Placeno* imaju vrijednost *False*.

Ako bismo kao argument naredbe WHERE naveli izraz u kojem bismo imali kombinaciju AND i OR operatora treba obratiti pozornost da operator AND ima veći prioritet. To znači da će se u izrazu:

WHERE izraz1 OR izraz2 AND izraz3

najprije izračunati *izraz2 AND izraz3*, a dobiveni rezultat dovesti u OR vezu s *izraz1*.

Naravno da se sa zagradama može promijeniti takav redoslijed uspoređivanja.

11. Napišite SQL naredbu koja će izdvojiti sve kupce čije prezime počinje sa slovom 'A' ili 'B'.

```
SELECT * FROM Kupci  
WHERE Prezime LIKE "A*" OR Prezime LIKE "B*"
```

ID	Ime	Prezime	Adresa
1	Maria	Anders	Obere Str. 57, Bogota
8	Elizabeth	Brown	2743 Bering St., Boston
9	James	Ashworth	89 Jefferson Way, Portland

Operator LIKE izdvaja slogove čija polja udovoljavaju navedenom usklađenju. Pritom se mogu koristiti i zamjenski znakovi (wildcards):

* ili % označava bilo koliko bilo kojih znakova
? ili _ označava bilo koji, ali samo jedan znak

Naprimjer:

Ime LIKE MAR* će izdvojiti Marija, Marijana, Marina i Martin, ali ne Mate niti Anamarija.

Ime LIKE *MAR* izdvojit će se i Marija i Anamarija, ali ne i Maja.

Ime LIKE MARI?A izdvaja Marija, Marina i Marica, ali ne i Marijana.

Ime LIKE ?INKO izdvaja Vinko i Dinko, ali ne i Marinko.

12. Napišite SQL naredbu koja će izdvojiti sve kupce čija adresa sadrži niz "STR". Prikažite samo polja: **Ime, Prezime i Adresa.**

```
SELECT Ime, Prezime, Adresa FROM Kupci  
WHERE Adresa LIKE "*STR*"
```


Ime	Prezime	Adresa
Maria	Anders	Obere Str. 57, Bogota
Hanna	Moos	Forsterstr. 57, Amsterdam
Horst	Kloss	Taucherstrasse 10, Dortmund

13. Napišite SQL naredbu koja će izdvojiti sve narudžbe veće od 700 kn, te koje nisu od kupaca Horst Kloss i Pascale Cartrain.

```
SELECT * FROM Narudzbe
WHERE Iznos > 700 AND KupacID <> 1 AND KupacID <> 7
```

ID	Iznos	DatumNarudzbe	DatumIsporuke	Placeno	KupacID
2	1000,89	27.1.2010	31.1.2010	False	2
9	800,00	11.3.2010	15.3.2010	False	3
19	765,00	3.5.2010	6.5.2010	False	5
22	879,60	12.5.2010	16.5.2010	True	5
33	1300,10	14.7.2010	16.7.2010	True	10
38	900,80	11.8.2010	13.8.2010	False	9

SQL operator "nije jednako" navodi se kao <>, za razliku od svih jezika s C sintaksom gdje se taj operator piše kao !=

14. Napišite SQL naredbu koja će izdvojiti sve narudžbe u 3. mjesecu 2010.

```
SELECT * FROM Narudzbe
WHERE DatumNarudzbe >= #2010-3-1# AND DatumNarudzbe <= #2010-3-31#
```

ID	Iznos	DatumNarudzbe	DatumIsporuke	Placeno	KupacID
8	200,56	4.3.2010	8.3.2010	Yes	1
9	800,00	11.3.2010	15.3.2010	No	3
10	412,30	21.3.2010	22.3.2010	No	2
11	290,80	24.3.2010	29.3.2010	Yes	4
12	578,90	29.3.2010	1.4.2010	No	2

Iako se formati datuma mogu različito navoditi ovaj je najuobičajeniji:
#gggg-mm-dd#

15. Napišite SQL naredbu koja će izdvojiti sve narudžbe od Elisabeth Brown. Prikažite samo polja: **Datum narudžbe** i **Iznos bez PDV**.

```
SELECT DatumNarudzbe, Iznos FROM Narudzbe
WHERE KupacID = 8
```

DatumNarudzbe	Iznos
1.2.2010	590,78
18.5.2010	350,00
7.6.2010	645,50

Brojčane vrijednosti navode se bez navodnika, dok se nizovi znakova navode u navodnicima.

100 je broj
"100" je niz znakova

16. Napišite SQL naredbu koja će izdvojiti sve narudžbe isporučene 21.04.2010.

```
SELECT * FROM Narudzbe
WHERE DatumNarudzbe = #2010-4-21#
```

ID	Iznos	DatumNarudzbe	DatumIsporuke	Placeno	KupacID
16	909,45	21.4.2010	25.4.2010	No	7

17. Napišite SQL naredbu koja će izdvojiti sve narudžbe koje nisu isporučene u mjesecu narudžbe.

```
SELECT * FROM Narudzbe  
WHERE MONTH(DatumNarudzbe) <> MONTH(DatumIsporuke)
```

ID	Iznos	DatumNarudzbe	DatumIsporuke	Placeno	KupacID
12	578,90	29.3.2010	1.4.2010	No	2
18	290,78	29.4.2010	2.5.2010	No	3

18. Napišite SQL naredbu koja će izdvojiti sve narudžbe u rasponu od 800 do 1000 kn.

```
SELECT * FROM Narudzbe  
WHERE Iznos >= 800 AND Iznos <= 1000
```

ID	Iznos	DatumNarudzbe	DatumIsporuke	Placeno	KupacID
9	800,00	11.3.2010	15.3.2010	No	3
16	909,45	21.4.2010	25.4.2010	No	7
22	879,60	12.5.2010	16.5.2010	Yes	5
38	900,80	11.8.2010	13.8.2010	No	9

Ako bismo umjesto operatora AND naveli operator OR bile bi izdvojene sve narudžbe jer bi svaka udovoljavala barem jednom navedenom uvjetu.

19. Napišite SQL naredbu koja će izdvojiti sve neplaćene narudžbe veće od 800 kn. Prikažite sva polja osim **ID** i **Datum isporuke**.

```
SELECT Iznos, DatumNarudzbe, Placeno, KupacID FROM Narudzbe  
WHERE Placeno = False AND Iznos > 800
```

Iznos	DatumNarudzbe	Placeno	KupacID
1000,89	27.1.2010	No	2
909,45	21.4.2010	No	7
900,80	11.8.2010	No	9

Za razliku od prethodnog primjera, ovdje zamjena operatora AND s OR ne bi rezultirala izdvajanjem svih slogova!

20. Napišite SQL naredbu koja će izdvojiti sve plaćene narudžbe Elizabeth Lincoln i Elizabeth Brown.

```
SELECT * FROM Narudzbe  
WHERE Placeno = True AND (KupacID = 4 OR KupacID = 8)
```

Bez zagrada rezultati bi bili drukčiji. Zbog većeg prioriteta operatora AND izdvojili bi se svi plaćeni računi Elizabeth Lincoln plus svi računi Elizabeth Brown.

21. Napišite SQL naredbu koja će kreirati pogled (view) **Placeni** nad tablicom **Narudzbe** s poljima **Iznos**, **Datum narudžbe** i **KupacID**. Nakon toga iz pogleda izdvojite sve račune koji su stariji od 50 dana.

```
SELECT Iznos, DatumNarudzbe, KupacID INTO Placeni  
FROM Narudzbe  
WHERE Placeno = True
```

```
SELECT * FROM Placeni  
WHERE NOW() - DatumNarudzbe > 50
```

Iznos	DatumNarudzbe	KupacID
356,23	24.1.2010	3
290,78	27.1.2010	5
590,78	1.2.2010	8
99,80	17.2.2010	7
200,56	4.3.2010	1
290,80	24.3.2010	4
612,00	3.4.2010	4
208,90	17.4.2010	3
467,70	26.4.2010	1
1025,77	5.5.2010	7

Pogled (View) je virtualna tablica kreirana od rezultata izvršenja SQL izraza. U ovom primjeru Placeni je virtualna tablica nad kojom se može izvršiti SQL upit kao i nad stvarnom tablicom.

Trenutni datum i vrijeme dobije se pozivom funkcije NOW().

22. Napišite SQL naredbu koja će izdvojiti sve plaćene račune koji su veći od svakog neplaćenog računa. Prikažite samo polja: **DatumNarudzbe** i **Iznos**.

```
SELECT DatumNarudzbe, Iznos FROM Narudzbe  
WHERE Placeno = True AND  
Iznos > ALL (SELECT Iznos FROM Narudzbe WHERE Placeno = False)
```

23. Napišite SQL naredbu koja će izdvojiti sve račune kupaca Horst Kloss i Pascale Cartrain čiji su iznosi veći od svakog neplaćenog računa.

```
SELECT * FROM Narudzbe  
WHERE (KupacID = 7 OR KupacID = 10) AND  
Iznos > ALL (SELECT Iznos FROM Narudzbe WHERE Placeno = False)
```

ID	Iznos	DatumNarudzbe	DatumIsporuke	Placeno	KupacID
20	1025,77	5.5.2010	8.5.2010	True	7
33	1300,10	14.7.2010	16.7.2010	True	10

Najprije se s posebnom SQL naredbom izdvajaju svi iznosi neplaćenih računa, a onda s operatorom ALL naznačujemo DBMS sustavu da Iznos mora biti veći od svakog tako dobivenog iznosa.

Naravno da s lijeve i desne strane operatora > mora biti isti tip podatka (u ovom slučaju to je brojčana vrijednost *Iznos*).

24. Napišite SQL naredbu koja će izdvojiti sve plaćene račune koji su barem 10 % veći od svakog neplaćenog računa. Ispišite polja **Iznos** i **Datum narudžbe**.

```
SELECT Iznos, DatumNarudzbe FROM Narudzbe  
WHERE Placeno = True AND  
Iznos > ALL(SELECT Iznos * 1.1 FROM Narudzbe WHERE Placeno = False)
```

Iznos	DatumNarudzbe
1300,10	14.7.2010

25. Napišite SQL naredbu koja će izdvojiti sve račune kupaca Elizabeth Lincoln i Elizabeth Brown koji su stariji od barem jednog plaćenog računa kupca Hanna Moos.

```
SELECT * FROM Narudzbe  
WHERE (KupacID = 4 OR KupacID = 8) AND  
DatumNarudzbe < ANY(SELECT DatumNarudzbe FROM Narudzbe  
WHERE Placeno = True AND KupacID = 3)
```

ID	Iznos	DatumNarudzbe	DatumIsporuke	Placeno	KupacID
4	590,78	1.2.2010	4.2.2010	True	8
11	290,80	24.3.2010	29.3.2010	True	4
13	612,00	3.4.2010	7.4.2010	True	4

U ovom primjeru koristi se operator ANY i s njim naznačujemo DBMS sustavu da lijeva strana operatora mora biti manja od barem jedne vrijednosti dobivene izvršenjem SQL naredbe navedene na desnoj strani.

Opet je i s lijeve i s desne strane operatora < isti tip podatka (u ovom slučaju to je datumska vrijednost *DatumNarudzbe*).

Naredba ORDER BY

Poredak slogova izdvojenih naredbom SELECT je neodređen i to bez obzira kakav uvjet naveli s naredbom WHERE. Često je međutim neophodno rezultate izvršenja dobiti poredane po određenom polju ili po više njih. Upravo taj zadatak obavlja naredba ORDER BY sa sljedećom sintaksom:

SELECT polje1, polje2, ... **FROM** Naziv_Tablice
ORDER BY polje ASC | DESC

gdje su:

- ASC i DESC – načini sortiranja (ASC je uzlazno, DESC je silazno. ASC je podrazumijevano.)

Zadaci:

26. Napišite SQL naredbu koja će izdvojiti sve plaćene račune sortirane po iznosu.

SELECT * FROM Narudzbe
WHERE Placeno = True
ORDER BY Iznos

ID	Iznos	DatumNarudzbe	DatumIsporuke	Placeno	KupacID
30	88,73	24.6.2010	29.6.2010	Yes	6
6	99,80	17.2.2010	20.2.2010	Yes	7
28	109,71	10.6.2010	11.6.2010	Yes	2
8	200,56	4.3.2010	8.3.2010	Yes	1
15	208,90	17.4.2010	21.4.2010	Yes	3
3	290,78	27.1.2010	31.1.2010	Yes	5
11	290,80	24.3.2010	29.3.2010	Yes	4
1	356,23	24.1.2010	27.1.2010	Yes	3
17	467,70	26.4.2010	27.4.2010	Yes	1
25	532,90	29.5.2010	31.5.2010	Yes	2
4	590,78	1.2.2010	4.2.2010	Yes	8
13	612,00	3.4.2010	7.4.2010	Yes	4
22	879,60	12.5.2010	16.5.2010	Yes	5
20	1025,77	5.5.2010	8.5.2010	Yes	7

Naredba ORDER BY navodi se na kraju SQL izraza i ona sortira rezultate, u ovom slučaju po polju *Iznos* (točnije, po vrijednosti zapisanoj u polju *Iznos*).

27. Napišite SQL naredbu koja će izdvojiti sve neplaćene račune sortirane po datumu narudžbe.

SELECT * FROM Narudzbe
WHERE Placeno = False
ORDER BY DatumNarudzbe

ID	Iznos	DatumNarudzbe	DatumIsporuke	Placeno	KupacID
2	1000,89	27.1.2010	31.1.2010	No	2
5	100,50	13.2.2010	17.2.2010	No	2
7	190,00	25.2.2010	28.2.2010	No	9
9	800,00	11.3.2010	15.3.2010	No	3

ID	Iznos	DatumNarudzbe	DatumIsporuke	Placeno	KupacID
10	412,30	21.3.2010	22.3.2010	No	2
12	578,90	29.3.2010	1.4.2010	No	2
14	367,51	12.4.2010	15.4.2010	No	5
16	909,45	21.4.2010	25.4.2010	No	7
18	290,78	29.4.2010	2.5.2010	No	3
19	765,00	3.5.2010	6.5.2010	No	5
21	97,56	10.5.2010	15.5.2010	No	4
23	350,00	18.5.2010	20.5.2010	No	8
24	489,75	21.5.2010	24.5.2010	No	5
26	210,05	1.6.2010	4.6.2010	No	6
27	645,50	7.6.2010	12.6.2010	No	8
29	209,80	18.6.2010	22.6.2010	No	3

28. Napišite SQL naredbu koja će izdvojiti sve račune od Pascale Cartrain sortirane po datumu isporuke silazno. Prikažite sljedeća polja: **Datum narudžbe, Iznos i Je li plaćen.**

```
SELECT DatumNarudzbe, Iznos, Placeno FROM Narudzbe
WHERE KupacID = 10
ORDER BY DatumIsporuke DESC
```

DatumNarudzbe	Iznos	Placeno
14.7.2010	1300,10	Yes
3.7.2010	280,94	No

Sortiranje može biti uzlazno ili silazno. Uzlazno sortiranje naznačuje se ključnom riječju ASC, dok se silazno naznačuje ključnom riječju DESC.

Uzlazno sortiranje je podrazumijevano te ključnu riječ ASC nije neophodno navoditi.

29. Napišite SQL naredbu koja će izdvojiti sve račune od kupaca Maria Anders, Elizabeth Lincoln i Horst Kloss iz mjeseca siječnja veljače i travnja. Račune sortirajte po polju KupacID, zatim po iznosu silazno, te po datumu narudžbe.

```
SELECT * FROM Narudzbe
WHERE KupacID IN (1,4,7) AND MONTH(DatumNarudzbe) IN (1,2,4)
ORDER BY KupacID, Iznos DESC, DatumNarudzbe
```

ID	Iznos	DatumNarudzbe	DatumIsporuke	Placeno	KupacID
17	467,70	26.4.2010	27.4.2010	Yes	1
13	612,00	3.4.2010	7.4.2010	Yes	4
16	909,45	21.4.2010	25.4.2010	No	7
6	99,80	17.2.2010	20.2.2010	Yes	7

Ovaj primjer će sortirati narudžbe po kupcu kao prvi uvjet sortiranja, a zatim će sve narudžbe pojedinog kupca sortirati po iznosu silazno te ako kupac ima dvije ili više narudžbi istog iznosa njih će sortirati po datumu narudžbe.

Primijetite da za sortiranje po kupcu i po datumu narudžbe nije navedena ključna riječ ASC pa se za njih koristi podrazumijevano uzlazno sortiranje.

Ovaj primjer pokazuje i upotrebu ključne riječi IN koju će DBMS sustav implementirati tako da vrati istinu ako je vrijednost pronađena unutar navedenog skupa vrijednosti. Ako skup vrijednosti ima malen broj članova može se zamijeniti s operatorom uspoređivanja.

KupacID IN (1,4)
je isto kao
KupacID=1 OR KupacID=4

30. Napišite SQL naredbu koja će izdvojiti sve račune od kupca Antonio Moreno, te će ih sortirati po broju proteklih dana između isporuke i narudžbe. Prikažite samo polja **Iznos** i **Razlika dana**.

```
SELECT Iznos, DatumIsporuke - DatumNarudzbe AS [Razlika dana] FROM Narudzbe  
WHERE KupacID = 2  
ORDER BY DatumIsporuke - DatumNarudzbe
```

Iznos	Razlika dana
109,71	1
412,30	1
532,90	2
578,90	3
100,50	4
1000,89	4

Argument naredbe ORDER BY može biti i izraz koji treba izračunati. U tom slučaju možemo reći kao da naredba ORDER BY sortira po dinamički generiranom polju čija će vrijednost biti izračunati izraz.

31. Napišite SQL naredbu koja će izdvojiti prva tri plaćena računa po veličini.

```
SELECT TOP 3 * FROM Narudzbe  
WHERE Placeno = True  
ORDER BY Iznos DESC
```

ID	Iznos	DatumNarudzbe	DatumIsporuke	Placeno	KupacID
33	1300,10	14.7.2010	16.7.2010	True	10
20	1025,77	5.5.2010	8.5.2010	True	7
22	879,60	12.5.2010	16.5.2010	True	5

Naredba TOP izdvaja samo određeni broj slogova i taj broj je prosljeđen kao argument (u ovom primjeru to je broj 3).

Za ovaj primjer to možemo ovako reći: izdvajaju se svi plaćeni računi sortirani po iznosu silazno, a nakon toga se uzimaju prva 3. Budući da su računi sortirani silazno to će biti 3 najveća plaćena računa.

32. Napišite SQL naredbu koja će izdvojiti najmanjih 5 neplaćenih računa po veličini, ali tako da budu poredani od većeg prema manjem.

```
SELECT * FROM  
(SELECT TOP 5 * FROM Narudzbe  
WHERE Placeno = False  
ORDER BY Iznos)  
ORDER BY Iznos DESC
```

ID	Iznos	DatumNarudzbe	DatumIsporuke	Placeno	KupacID
26	210,05	1.6.2010	4.6.2010	False	6
29	209,80	18.6.2010	22.6.2010	False	3
7	190,00	25.2.2010	28.2.2010	False	9
5	100,50	13.2.2010	17.2.2010	False	2
21	97,56	10.5.2010	15.5.2010	False	4

Izvor podataka ne mora nužno biti tablica nego može biti i SQL upit koji će vratiti određeni broj slogova. Treba obratiti pozornost na zagrade jer se inače upit neće izvršiti.

33. Napišite SQL naredbu koja će ispisati Datum narudžbe i Iznos narudžbe desete po veličini.

```
SELECT LAST(DatumNarudzbe) AS [Deseti po veličini], LAST(Iznos) AS [Iznos narudžbe] FROM
```

(SELECT TOP 10 * FROM Narudzbe
ORDER BY Iznos DESC)

ILI

SELECT TOP 1 DatumNarudzbe AS [Deseti po veličini], Iznos AS [Iznos narudžbe] FROM
(SELECT TOP 10 * FROM Narudzbe
ORDER BY Iznos DESC)
ORDER BY Iznos

Deseti po veličini	Iznos narudžbe
3.4.2010	612,00

Standardni SQL nema naredbu za dohvat n-tog sloga (istina, ni naredbe TOP i LAST nisu dio standardnog SQL jezika) pa se pomoću naredbe LAST uzima posljednji slog od prvih 10 sortiranih po iznosu silazno, a to je upravo deseti račun po veličini. Isti zadatak riješen je i bez naredbe LAST tako da se uzme prvi slog od 10 najvećih računa sortiranih uzlazno, a to je opet deseti račun po veličini.

34. Napišite SQL naredbu koja će izdvojiti prve tri i zadnje tri narudžbe po iznosu.

(SELECT TOP 3 * FROM Narudzbe ORDER BY Iznos DESC)
UNION
(SELECT TOP 3 * FROM Narudzbe ORDER BY Iznos)

ID	Iznos	DatumNarudzbe	DatumIsporuke	Placeno	KupacID
30	88,73	24.6.2010	29.6.2010	True	6
21	97,56	10.5.2010	15.5.2010	False	4
6	99,80	17.2.2010	20.2.2010	True	7
2	1000,89	27.1.2010	31.1.2010	False	2
20	1025,77	5.5.2010	8.5.2010	True	7
33	1300,10	14.7.2010	16.7.2010	True	10

Naredba UNION spaja rezultate izvršenja dviju SELECT naredbi. Obje SELECT naredbe moraju imati isti broj i redoslijed polja koja se izdvajaju.

35. Napišite SQL naredbu koja će izdvojiti najveći neplaćeni i najmanji plaćeni račun.

(SELECT TOP 1 * FROM Narudzbe WHERE Placeno = False ORDER BY Iznos DESC)
UNION
(SELECT TOP 1 * FROM Narudzbe WHERE Placeno = True ORDER BY Iznos)

ID	Iznos	DatumNarudzbe	DatumIsporuke	Placeno	KupacID
2	1000,89	27.1.2010	31.1.2010	False	2
30	88,73	24.6.2010	29.6.2010	True	6

36. Napišite SQL naredbu koja će izdvojiti sve narudžbe kupca Maria Anders te one koje su veće od 1000 kn. Ako pojedina narudžba zadovoljava oba uvjeta treba je prikazati samo jednom.

(SELECT * FROM Narudzbe WHERE KupacID = 1)
UNION

(SELECT * FROM Narudzbe WHERE Iznos > 1000)

ID	Iznos	DatumNarudzbe	DatumIsporuke	Placeno	KupacID
2	1000,89	27.1.2010	31.1.2010	False	2
8	200,56	4.3.2010	8.3.2010	True	1
17	467,70	26.4.2010	27.4.2010	True	1
20	1025,77	5.5.2010	8.5.2010	True	7
33	1300,10	14.7.2010	16.7.2010	True	10
37	260,67	7.8.2010	10.8.2010	False	1

Za naredbu UNION vrijedi da ako pojedini slog udovoljava objema SELECT naredbama slog je izdvojen samo jednom.

37. Napišite SQL naredbu koja će izdvojiti sve narudžbe kupca Maria Anders te one koje su manje od 100 kn. Ako pojedina narudžba zadovoljava oba uvjeta treba je prikazati svaki put. Narudžbe sortirajte po iznosu.

(SELECT * FROM Narudzbe WHERE KupacID = 1)

UNION ALL

(SELECT * FROM Narudzbe WHERE Iznos < 100)

ORDER BY Iznos

ID	Iznos	DatumNarudzbe	DatumIsporuke	Placeno	KupacID
30	88,73	24.6.2010	29.6.2010	True	6
21	97,56	10.5.2010	15.5.2010	False	4
6	99,80	17.2.2010	20.2.2010	True	7
8	200,56	4.3.2010	8.3.2010	True	1
37	260,67	7.8.2010	10.8.2010	False	1
17	467,70	26.4.2010	27.4.2010	True	1

Za UNION ALL vrijedi da ako pojedini slog udovoljava objema SELECT naredbama slog je izdvojen oba puta.

38. Napišite SQL naredbu koja će od 10 najvećih računa izdvojiti samo plaćene. Izdvojene račune sortirajte po datumu narudžbe.

SELECT * FROM (SELECT TOP 10 * FROM Narudzbe ORDER BY Iznos DESC)

WHERE Placeno = True

ORDER BY DatumNarudzbe

ID	Iznos	DatumNarudzbe	DatumIsporuke	Placeno	KupacID
13	612,00	3.4.2010	7.4.2010	True	4
20	1025,77	5.5.2010	8.5.2010	True	7
22	879,60	12.5.2010	16.5.2010	True	5
33	1300,10	14.7.2010	16.7.2010	True	10

Izvor podataka je SELECT naredba koja izdvaja 10 najvećih računa. Nakon toga se od tih 10 uz pomoć WHERE uvjeta izdvajaju samo plaćeni.

39. Napišite SQL naredbu koja će 5 najstarijih računa sortirati po veličini od većeg prema manjem.

SELECT * FROM (SELECT TOP 5 * FROM Narudzbe ORDER BY DatumNarudzbe)

ORDER BY Iznos DESC

ID	Iznos	DatumNarudzbe	DatumIsporuke	Placeno	KupacID
2	1000,89	27.1.2010	31.1.2010	False	2

Opet se primjer izvršava u dva koraka, najprije se uzima 5 najstarijih računa koji se zatim sortiraju po iznosu silazno.

ID	Iznos	DatumNarudzbe	DatumIsporuke	Placeno	KupacID
4	590,78	1.2.2010	4.2.2010	True	8
1	356,23	24.1.2010	27.1.2010	True	3
3	290,78	27.1.2010	31.1.2010	True	5
5	100,50	13.2.2010	17.2.2010	False	2

40. Napišite SQL naredbu koja će od 5 najvećih računa najprije prikazati plaćene, pa nakon toga neplaćene. Prikažite polja **Iznos**, **Datum narudžbe** i **Je li plaćen**.

```
SELECT * FROM (SELECT TOP 5 Iznos, DatumNarudzbe, Placeno FROM Narudzbe ORDER BY Iznos DESC)  
ORDER BY Placeno
```

Iznos	DatumNarudzbe	Placeno
1025,77	5.5.2010	True
1300,10	14.7.2010	True
900,80	11.8.2010	False
909,45	21.4.2010	False
1000,89	27.1.2010	False

41. Napišite SQL naredbu koja će izdvojiti najveći od svih računa koji su manji od svih računa kupca Hanna Moss.

```
SELECT TOP 1 * FROM Narudzbe  
WHERE Iznos < ALL (SELECT Iznos FROM Narudzbe WHERE KupacID=3)  
ORDER BY Iznos DESC
```

Naredba TOP 1 će izdvojiti samo jedan račun, a kako su računi sortirani od većeg prema manjem to će biti najveći od svih koji udovoljavaju uvjetu da su manji od svih računa navedenog kupca.

Funkcije

Funkcije su standardni dio svake implementacije SQL jezika i koriste se za izvršenje najčešćih izračuna kao što su zbroj, prosjek, maksimalna ili minimalna vrijedenost, zaokruživanje, operacije s nizovima znakova... Postoje dvije grupe funkcija, agregatne i skalarne funkcije. Agregatne funkcije vrše izračune nad skupom vrijednosti polja iz slogova tablice koji udovoljavaju postojećim uvjetima i vraćaju jednu vrijednost. Ovoj grupi pripadaju:

- SUM - izračun zbroja vrijednosti
- AVG - izračun prosjeka vrijednosti
- COUNT - izračun ukupnog broja vrijednosti
- MAX i MIN - izračun maksimalne i minimalne vrijednosti.

Za razliku od agregatnih, skalarne funkcije vrše izračun nad jednom vrijednosti (obično je to vrijednost nekog polja) i takve su npr:

- ROUND - zaokružuje vrijednosti na određeni broj decimala
- LEN - vraća duljinu niza znakova
- LCASE i UCASE - pretvara niz znakova u mala, odnosno velika slova
- NOW - vraća trenutni datum i vrijeme

Sintaksa upotrebe funkcija je sljedeća:

SELECT Naziv_Funkcije(polje) **FROM** Naziv_Tablice

gdje je:

- Naziv_Funkcije – naziv funkcije (SUM, MAX, MIN, AVG, COUNT, ...)

Zadaci:

42. Napišite SQL naredbu koja će izračunati koliko je Antonio Moreno izvršio narudžbi.

```
SELECT COUNT(Iznos) AS [Broj narudžbi Antonio Moreno] FROM Narudzbe  
WHERE KupacID = 2
```

Broj narudžbi Antonio Moreno
6

Funkcija COUNT broji ukupan broj iznosa koji udovoljavaju uvjetu, a to je da je račun od kupca Antonio Moreno.

Kao argument funkcije COUNT mogli smo navesti *DatumNarudzbe* ili neko drugo polje jer će izbrojeni broj biti isti, međutim problem može nastati ako za neko od tih polja nije definirana vrijednost, odnosno ako ima vrijednost *null*. Tada taj slog ne bi bio prebrojen pa je zato najjednostavnije navesti COUNT(*) koji će i u tom slučaju prebrojiti sve slogove.

43. Napišite SQL naredbu koja će izračunati koliko Yang Wang ima neplaćenih narudžbi.

```
SELECT COUNT(Iznos) AS [Broj neplaćenih narudžbi] FROM Narudzbe  
WHERE KupacID = 5 AND Placeno = False
```

Koristimo COUNT(Iznos) jer vrijednost polja *Iznos* teško da može biti nedefinirana, a račun bez iznosa ako i postoji ne treba ga brojiti.

Broj neplaćenih narudžbi
4

44. Napišite SQL naredbu koja će izračunati koliko su ukupno narudžbi naručili Elisabeth Lincoln i Elisabeth Brown.

```
SELECT COUNT(Iznos) AS [Broj narudžbi od Elisabeth] FROM Narudzbe  
WHERE KupacID = 4 OR KupacID = 8
```

Broj narudžbi od Elisabeth
7

45. Napišite SQL naredbu koja će izračunati koliko je narudžbi naručeno u 5. mjesecu 2010.

```
SELECT COUNT(Iznos) AS [Ukupno u svibnju] FROM Narudzbe  
WHERE DatumNarudzbe >= #2010-5-1# AND DatumNarudzbe <= #2010-5-31#
```

Ukupno u svibnju
7

46. Napišite SQL naredbu koja će izračunati koliko se u tablici nalazi različitih imena kupaca.

```
SELECT COUNT(*) AS [Ukupno različitih imena] FROM  
(SELECT DISTINCT(Ime) FROM Kupci)
```

Ukupno različitih imena
9

Opet je izvor podataka virtualna tablica koja će biti kreirana izvršenjem SELECT naredbe, a naredba DISTINCT osigurava neponavljanje imena kupaca.

47. Napišite SQL naredbu koja će izračunati koliki je ukupni iznos svih nenaplaćenih računa.

```
SELECT SUM(Iznos) AS [Ukupna potraživanja] FROM Narudzbe  
WHERE Placeno = False
```

Ukupna potraživanja
8638,93

Funkcija SUM je jedna od najvažnijih funkcija, a sintaksa poziva je kao i kod ostalih agregatnih funkcija. Argument funkcije je naziv polja čija će se ukupna vrijednost izračunati. Naravno, uzimaju se u obzir samo polja slogova koji udovoljavaju uvjetu navedenom u WHERE naredbi.

48. Napišite SQL naredbu koja će izračunati koliki je ukupni iznos naplaćenog PDV-a.

```
SELECT ROUND(SUM(Iznos) / 1.23 * 0.23, 2) AS [Ukupni PDV] FROM Narudzbe  
WHERE Placeno = True
```

Ukupni PDV
1494,67

Funkcija SUM vraća jednu vrijednost te je s tom vrijednošću moguće vršiti razne izračune kao što je ovaj o izračunu PDV-a.

Primjer pokazuje i upotrebu skalarne funkcije ROUND čiji je prvi argument prethodno izračunati izraz. Drugi argument je broj decimala na koju se vrijednost zaokružuje.

49. Napišite SQL naredbu koja će izračunati koliko je ukupno dugovanje kupca Horst Kloss.

```
SELECT SUM(Iznos) AS [Potraživanje Horst Kloss] FROM Narudzbe  
WHERE KupacID = 7 AND Placeno = False
```

Potraživanje Horst Kloss
909,45

50. Napišite SQL naredbu koja će izračunati prosječan iznos neplaćenog računa.

```
SELECT ROUND(AVG(Iznos), 2) AS [Prosječno nenaplaćeno] FROM Narudzbe  
WHERE Placeno = False
```

Prosječno nenaplaćeno
454,68

Agregatna funkcija AVG izračunava prosjek vrijednosti polja koja udovoljavaju uvjetu. Kako ne bismo ostali zatrpani decimalama koje će funkcija AVG vratiti, vrijednost se na kraju zaokružuje na dvije decimale.

51. Napišite SQL naredbu koja će izračunati prosječan iznos naplaćenog računa od kupca James Ashworth.

```
SELECT ROUND(AVG(Iznos), 2) AS [Prosječno naplaćeno od James Ashworth]  
FROM Narudzbe  
WHERE KupacID = 9 AND Placeno = True
```

Prosječno naplaćeno od James Ashworth
360,45

52. Napišite SQL naredbu koja će izračunati iznos najvećeg neplaćenog računa.

```
SELECT MAX(Iznos) AS [Najveći nenaplaćeni račun] FROM Narudzbe  
WHERE Placeno = False
```

Najveći nenaplaćeni račun
1000,89

Vjerujemo da funkcije MAX i MIN nije potrebno dodatno objašnjavati jer ne samo da je način korištenja isti kao i kod ostalih agregatnih funkcija nego i njihovi nazivi govore kakvu vrijednost vraćaju.

53. Napišite SQL naredbu koja će izračunati iznos najmanjeg plaćenog računa.

```
SELECT MIN(Iznos) AS [Najmanji naplaćeni račun] FROM Narudzbe  
WHERE Placeno = True
```

Najmanji naplaćeni račun
88,73

54. Napišite SQL naredbu koja će izračunati iznos najvećeg plaćenog računa u mjesecu ožujku.

```
SELECT ROUND(MAX(Iznos), 2) AS [Najveći naplaćeni u ožujku] FROM Narudzbe
```

WHERE Placeno = True AND MONTH(DatumNarudzbe) = 3

Najveći naplaćeni u ožujku
290,80

55. Napišite SQL naredbu koja će izračunati iznos najvećeg plaćenog računa u mjesecima ožujku ili travnju.

SELECT ROUND(MAX(Iznos), 2) AS [Najveći naplaćeni u ožujku ili travnju] FROM Narudzbe
WHERE Placeno = True AND (MONTH(DatumNarudzbe) = 3 OR MONTH(DatumNarudzbe) = 4)

Najveći naplaćeni u ožujku ili travnju
612,00

Primijetite zagrade oko argumenata OR uvjeta jer inače primjer ne bi ispravno radio zbog većeg prioriteta operatora AND.
Kad ne bi bilo zagrada primjer bi računao maksimalni iznos plaćenih računa u ožujku i svih računa u travnju.

56. Napišite SQL naredbu koja će izračunati koliko se ukupno znakova nalazi u svim imenima i prezimenima kupaca.

SELECT SUM(LEN(Ime) + LEN(Prezime)) AS [Ukupno znakova] FROM Kupci

Ukupno znakova
124

Skalarna funkcija LEN vraća dužinu polja u znakovima. Iako je to najčešći slučaj, argument funkcije ne mora biti naziv polja.

LEN("hello, world") će vratiti vrijednost 12.

57. Napišite SQL naredbu koja će izračunati od koliko se znakova sastoji najduže ime i prezime kupca.

SELECT MAX(LEN(Ime) + LEN(Prezime)) AS [Maksimalna dužina] FROM Kupci

Maksimalna dužina
16

58. Napišite SQL naredbu koja će izdvojiti sve narudžbe manje od 150 kn ako u tablici ima ijedna narudžba iznosa 800 kn. Ispišite sljedeća polja: **Datum narudžbe** i **Iznos**.

SELECT DatumNarudzbe, Iznos FROM Narudzbe
WHERE Iznos < 150 AND EXISTS (SELECT * FROM Narudzbe WHERE Iznos = 800)

DatumNarudzbe	Iznos
13.2.2010	100,50
17.2.2010	99,80
10.5.2010	97,56
10.6.2010	109,71
24.6.2010	88,73

Operator EXISTS vraća *true* ako SQL naredba kao proslijeđeni argument vrati barem jedan slog. Jednako tako ako niti jedan slog nije pronađen, EXISTS vraća *false*.

59. Napišite SQL naredbu koja će izdvojiti sve račune od kupca Hanna Moos ako taj kupac ima barem jedan neplaćeni račun. Račune sortirajte po iznosu. Ispišite sljedeća polja: **KupacID, Iznos i Datum narudžbe.**

```
SELECT KupacID, Iznos, DatumNarudzbe FROM Narudzbe
WHERE KupacID = 3 AND EXISTS
    (SELECT * FROM Narudzbe WHERE KupacID = 3 AND Placeno = False)
ORDER BY Iznos
```

KupacID	Iznos	DatumNarudzbe
3	208,90	17.4.2010
3	209,80	18.6.2010
3	290,78	29.4.2010
3	356,23	24.1.2010
3	800,00	11.3.2010

60. Napišite SQL naredbu koja će izračunati ukupnu kamatu za sve neplaćene račune uz kamatnu stopu od 7.5 %. Za godinu pretpostavite da nije prijestupna.

```
SELECT ROUND(SUM(ROUND(NOW() - DatumNarudzbe, 0) / 365 * 0.075 * Iznos), 2)
    AS [Ukupna kamata] FROM Narudzbe
WHERE Placeno = False
```

Ukupna kamata
392,03

Razlika u danima dobije se pomoću funkcije NOW koja vraća trenutni datum i vrijeme. S obzirom da ta razlika nije cijeli broj s funkcijom ROUND zaokružujemo dobivenu razliku na cijeli broj. Primjer bi trebalo doraditi tako da vodi računa je li godina prijestupna jer je tada i izračunata kamata nešto drukčija.

61. Napišite SQL naredbu koja će izračunati prosječan iznos svih narudžbi i prosječan iznos najveće i najmanje narudžbe.

```
SELECT ROUND(AVG(Iznos), 2) AS [Prosječni iznos],
    ROUND((MAX(Iznos) + MIN(Iznos)) / 2, 2)
    AS [Srednja vrijednost najveće i najmanje narudžbe]
FROM Narudzbe
```

Broj funkcija / izraza koji se računaju s jednom SELECT naredbom može biti i više od jedan i tada ih je potrebno razdvojiti zarezom.

Prosječni iznos	Srednja vrijednost najveće i najmanje narudžbe
468,11	694,42

62. Napišite SQL naredbu koja će izračunati koliko je ukupno dana proteklo od najstarije do najnovije narudžbe.

```
SELECT MAX(DatumNarudzbe) - MIN(DatumNarudzbe) AS [Protekli dani]
FROM Narudzbe
```

Protekli dani
214

63. Napišite SQL naredbu koja će izračunati kolika je u danima prosječna starost neplaćenih narudžbi.

```
SELECT ROUND(AVG(NOW() - DatumNarudzbe), 2) AS [Prosječna starost]
FROM Narudzbe
WHERE Placeno = False
```

Prosječna starost
631,78

64. Napišite SQL naredbu koja će izračunati ukupni iznos tri najveća plaćena računa.

```
SELECT SUM(Iznos) AS [Zbroj tri najveća] FROM
(SELECT TOP 3 Iznos FROM Narudzbe
WHERE Placeno = True
ORDER BY Iznos DESC)
```

Zbroj tri najveća
3205,47

Tri najveća plaćena računa se spremaju u virtualnu tablicu uz pomoć operatora TOP i silaznog sortiranja, a zatim se funkcijom SUM računa ukupni iznos ta tri računa.

65. Napišite SQL naredbu koja će izračunati prosječni iznos tri najstarija neplaćena računa.

```
SELECT ROUND(AVG(Iznos), 2) AS [Prosjek tri najstarija neplaćena] FROM
(SELECT TOP 3 Iznos FROM Narudzbe
WHERE Placeno = False
ORDER BY DatumNarudzbe)
```

Prosjek tri najstarija neplaćena
430,46

66. Napišite SQL naredbu koja će izračunati koliko je od 10 najvećih računa plaćenih, a koliko neplaćenih.

```
SELECT COUNT(*) AS [Plaćeni], 10-COUNT(*) AS [Neplaćeni] FROM
(
SELECT TOP 10 * FROM Narudzbe ORDER BY Iznos DESC
)
WHERE Placeno = True
```

Najprije se uzima 10 najvećih od kojih se nakon toga broje samo plaćeni uz pomoć uvjeta u WHERE operatoru. Broj neplaćenih je tada 10 – broj plaćenih.

67. Napišite SQL naredbu koja će izračunati koliko je različitih kupaca naručilo 10 najvećih narudžbi.

```
SELECT COUNT(*) FROM
(
SELECT DISTINCT(KupacID) FROM
(SELECT TOP 10 * FROM Narudzbe ORDER BY Iznos DESC)
)
```

Naredba koliko god bila složena, ako se podijeli na tri dijela postaje dosta jednostavnija. Prva naredba izdvaja 10 najvećih narudžbi, druga od njih 10 uzima samo one od različitih kupaca, a treća naredba ih prebrojava.

Naredbe GROUP BY i HAVING

GROUP BY naredba, kao što i samo ime kaže koristi se za grupiranje po određenom polju ili više njih. Najčešće dolazi u kombinaciji s funkcijama tako da se izračuna skupna vrijednost po određenom polju kao npr. ukupni iznos računa grupirano po kupcima, prosječni iznos po mjesecima itd...

Naredba HAVING dolazi uvijek u kombinaciji s GROUP BY i ona omogućuje da se grupiraju samo podaci koji zadovoljavaju određeni uvjet. Upravo taj uvjet se navodi kao argument naredbe HAVING. Naime, bez upotrebe naredbe HAVING nemoguće je npr. grupirati kupce po ukupnom iznosu njegovih računa uzevši u obzir samo kupce čiji je ukupni iznos računa veći od određenog iznosa. Naredba WHERE se ne može koristiti u tu svrhu kao npr.

```
SELECT KupacID, SUM(Iznos) FROM Narudzbe  
GROUP BY KupacID  
WHERE SUM(Iznos) > 1000
```

Razlog nemogućnosti je što se naredba WHERE izvršava u tijeku dohvata podataka kada se još ne zna koji od kupaca ima ukupni iznos računa veći od 1000. Zato je rješenje upotreba naredbe HAVING:

```
SELECT KupacID, SUM(Iznos) FROM Narudzbe  
GROUP BY KupacID  
HAVING SUM(Iznos) > 1000
```

Naredba HAVING izvodi se na već izračunatim i grupiranim podacima kada se za svakog kupca zna koliki je njegov ukupni iznos računa.

Sintaksa upotrebe naredbi GROUP BY i HAVING je sljedeća:

```
SELECT polje, Agregacijska_Funkcija(polje) FROM Naziv_Tablice  
WHERE polje operator vrijednost  
GROUP BY polje
```

gdje je:

- Agregacijska_Funkcija – funkcija koja se izvodi nad grupiranim slogovima

```
SELECT polje, Agregacijska_Funkcija(polje) FROM Naziv_Tablice  
WHERE polje operator vrijednost  
GROUP BY polje  
HAVING Agregacijska_Funkcija(polje) operator vrijednost
```

Zadaci:

68. Napišite SQL naredbu koja će izračunati ukupni iznos narudžbi grupirano po kupcima.

```
SELECT KupacID, SUM(Iznos) AS [Ukupni iznos] FROM Narudzbe  
GROUP BY KupacID
```

KupacID	Ukupni iznos
1	928,93
2	2735,20
3	1865,71

KupacID	Ukupni iznos
4	1578,81
5	3132,64
6	898,78
7	2545,02
8	1586,28
9	1871,81
10	1581,04

Ovo je klasični primjer upotrebe naredbe GROUP BY u kojem se za svakog kupca računa koliki je njegov ukupni iznos računa. Sintaksa naredbe nalaže da se naziv polja po kojem se grupira navodi kao argument naredbe SELECT i kao argument naredbe GROUP BY.

69. Napišite SQL naredbu koja će izračunati prosječni iznos narudžbe grupirano po kupcima.

```
SELECT KupacID, ROUND(AVG(Iznos), 2) AS [Prosječni iznos] FROM Narudzbe
GROUP BY KupacID
```

KupacID	Prosječni iznos
1	309,64
2	455,87
3	373,14
4	394,70
5	522,11
6	299,59
7	636,26
8	528,76
9	467,95
10	790,52

Opet isti primjer, ali ovaj put uz izračun prosjeka. Naredba GROUP BY je dovoljno sposobna da grupira po jednom kupcu i više izračuna kao npr. da za svakog kupca prikaže i njegov ukupni i prosječni iznos:

```
SELECT KupacID, SUM (Iznos), AVG(Iznos) FROM Narudzbe
GROUP BY KupacID
```

70. Napišite SQL naredbu koja će izračunati ukupan broj narudžbi grupirano po kupcima, te po tome jesu li narudžbe plaćene.

```
SELECT KupacID, Placeno, COUNT(*) AS [Ukupno narudžbi] FROM Narudzbe
GROUP BY KupacID, Placeno
```

KupacID	Placeno	Ukupno narudžbi
1	True	2
1	False	1
2	True	2
2	False	4
3	True	2
3	False	3
4	True	3
4	False	1
5	True	2
5	False	4
6	True	1
6	False	2

KupacID	Placeno	Ukupno narudžbi
7	True	2
7	False	2
8	True	1
8	False	2
9	True	1
9	False	3
10	True	1
10	False	1

Ovo je primjer grupiranja po više polja. Osnovni uvjet grupiranja je ID kupca odnosno polje *KupacID*, a sve narudžbe određenog kupca (s istom vrijednosti *KupacID*) grupiraju se nadalje po tome koliko je od njih plaćenih, a koliko neplaćenih. Ako se grupiranje vrši po više polja tada ih je sve potrebno navesti, ne samo u naredbi GROUP BY nego i kao argument naredbe SELECT.

71. Napišite SQL naredbu koja će izračunati broj narudžbi pojedinog kupca.

```
SELECT KupacID, COUNT(Iznos) AS [Broj narudžbi] FROM Narudzbe
GROUP BY KupacID
```

KupacID	Broj narudžbi
1	3
2	6
3	5
4	4
5	6
6	3
7	4
8	3
9	4
10	2

72. Napišite SQL naredbu koja će izdvojiti najveću narudžbu za pojedinog kupca.

```
SELECT KupacID, ROUND(MAX(Iznos), 2) AS [Najveća narudžba] FROM Narudzbe
GROUP BY KupacID
```

KupacID	Najveća narudžba
1	467,70
2	1000,89
3	800,00
4	612,00
5	879,60
6	600,00
7	1025,77
8	645,50
9	900,80
10	1300,10

Funkcijom ROUND poravnamo iznose na dvije decimale jer bi inače ispis bio vrlo nepregledan:

```

467.7
1000.89
800
612
```

73. Napišite SQL naredbu koja će izračunati ukupnu naplaćenu narudžbu grupirano po kupcima.

```
SELECT KupacID, ROUND(SUM(Iznos), 2) AS [Ukupna plaćena narudžba] FROM Narudzbe  
WHERE Placeno = True  
GROUP BY KupacID
```

KupacID	Ukupna plaćena narudžba
1	668,26
2	642,61
3	565,13
4	1481,25
5	1170,38
6	88,73
7	1125,57
8	590,78
9	360,45
10	1300,10

74. Napišite SQL naredbu koja će izračunati razliku između najvećeg i najmanjeg računa grupirano po kupcima. Slogove sortirajte po veličini izračunate razlike.

```
SELECT KupacID, MAX(Iznos) - MIN(Iznos) AS [Razlika] FROM Narudzbe  
GROUP BY KupacID  
ORDER BY MAX(Iznos) - MIN(Iznos)
```

KupacID	Razlika
1	267,14
8	295,50
6	511,27
4	514,44
5	588,82
3	591,10
9	710,80
2	900,39
7	925,97
10	1019,16

Ovaj put grupiranje se vrši prema izračunatom izrazu što u sintaksi ništa ne mijenja. Umjesto funkcije koja vrši izračun imamo izraz koji to isto radi.

75. Napišite SQL naredbu koja će izračunati ukupnu narudžbu za pojedini mjesec.

```
SELECT MONTHNAME(MONTH(DatumNarudzbe)) AS [Mjesec], ROUND(SUM(Iznos), 2)  
AS [Ukupno] FROM Narudzbe  
GROUP BY MONTH(DatumNarudzbe)
```

Mjesec	Ukupno
siječanj	1647,90
veljača	981,08
ožujak	2282,56
travanj	2856,34

Argument naredbe SELECT je naziv mjeseca, a argument naredbe GROUP BY je brojčana oznaka mjeseca što je i sasvim razumljivo za ovaj primjer jer je računanje lakše raditi s brojevima (GROUP BY), a prikaz je čitljiviji sa slovima (SELECT).

Mjesec	Ukupno
svibanj	4140,58
lipanj	1263,79
srpanj	3099,49
kolovoz	2452,48

76. Napišite SQL naredbu koja će izračunati broj isporučenih narudžbi po mjesecima.

```
SELECT MONTHNAME(MONTH(DatumIsporuke)) AS Mjesec, COUNT(Iznos) AS [Ukupno isporučeno]
FROM Narudzbe
GROUP BY MONTH(DatumIsporuke)
```

Mjesec	Ukupno isporučeno
siječanj	3
veljača	4
ožujak	4
travanj	6
svibanj	8
lipanj	5
srpanj	5
kolovoz	5

77. Napišite SQL naredbu koja će izračunati ukupnu narudžbu za pojedinog kupca, ali samo za kupce čiji je ukupni naručeni iznos veći od 2500.

```
SELECT KupacID, ROUND(SUM(Iznos), 2) AS [Ukupna narudžba] FROM Narudzbe
GROUP BY KupacID
HAVING SUM(Iznos) > 2500
```

KupacID	Ukupna narudžba
2	2735,20
5	3132,64
7	2545,02

Nismo mogli navesti
WHERE SUM(Iznos) > 2500
zato jer se naredba WHERE izvršava u tijeku dohvata podataka kada se još ne zna koji od kupaca ima ukupni iznos računa veći od 2500.

Naredba HAVING izvodi se pak na već izračunatim i grupiranim podacima kada se za svakog kupca zna koliki je njegov ukupni iznos računa.

78. Napišite SQL naredbu koja će izračunati ukupnu narudžbu za pojedinog kupca, ali samo za kupce čiji je prosječni naručeni iznos veći od 500.

```
SELECT KupacID, ROUND(SUM(Iznos), 2) AS [Ukupna narudžba],
ROUND(AVG(Iznos), 2) AS [Prosječna narudžba]
FROM Narudzbe
GROUP BY KupacID
HAVING AVG(Iznos) > 500
```

KupacID	Ukupna narudžba	Prosječna narudžba
5	3132,64	522,11

Iako to u zadatku nije traženo, u naredbi SELECT navodimo i prosječnu narudžbu tek da se u prikazu uvjerimo da naredba ispravno izdvaja zahtijevane kupce.

KupacID	Ukupna narudžba	Prosječna narudžba
7	2545,02	636,26
8	1586,28	528,76
10	1581,04	790,52

79. Napišite SQL naredbu koja će izračunati prosječni iznos narudžbe za pojedinog kupca, ali samo za kupce koji imaju više od 3 narudžbe.

```
SELECT KupacID, ROUND(AVG(Iznos), 2) AS [Prosječna narudžba] FROM Narudzbe
GROUP BY KupacID
HAVING COUNT(Iznos) > 3
```

KupacID	Prosječna narudžba
2	455,87
3	373,14
4	394,70
5	522,11
7	636,26
9	467,95

80. Napišite SQL naredbu koja će izdvojiti sve kupce čiji je maksimalni račun veći od prosjeka 10 najvećih računa. Kupce sortirajte po iznosu maksimalnog računa silazno.

```
SELECT KupacID, MAX(Iznos) AS [Maksimalni iznos] FROM Narudzbe
GROUP BY KupacID
HAVING MAX(Iznos) >
    (SELECT AVG(Iznos) FROM (SELECT TOP 10 Iznos FROM Narudzbe
    ORDER BY Iznos DESC))
ORDER BY MAX(Iznos) DESC
```

KupacID	Maksimalni iznos
10	1300,10
7	1025,77
2	1000,89
9	900,80

Primjer je složeniji utoliko što se u naredbi HAVING granična vrijednost ne navodi, nego se računa, a računa se kao prosjek 10 najvećih računa. Prosjek 10 najvećih dobije se kao TOP 10 od svih računa sortiranih po iznosu silazno.

Drugi ORDER BY nema veze s grupiranjem, nego na kraju samo sortira već grupirane podatke po maksimalnom iznosu silazno.

81. Napišite SQL naredbu koja će izdvojiti sve kupce čiji je prosječni iznos računa veći od prosjeka 10 najvećih plaćenih računa.

```
SELECT KupacID, AVG(Iznos) AS [Prosječni iznos] FROM Narudzbe
GROUP BY KupacID
HAVING AVG(Iznos) >
    (SELECT AVG(Iznos) FROM (SELECT TOP 10 Iznos FROM Narudzbe
    WHERE Placeno = True
```

ORDER BY Iznos DESC))

KupacID	Prosječni iznos
10	790,52

Usporedba se vrši s prosjekom 10 najvećih plaćenih računa. Ako bi se vršila usporedba s prosjekom samo plaćenih od 10 najvećih računa naredba bi bila dosta složenija jer bismo morali ugnijezditi još jednu SELECT naredbu:

```
SELECT KupacID, AVG(Iznos) AS [Prosječni iznos] FROM Narudzbe
GROUP BY KupacID
HAVING AVG(Iznos) > (SELECT AVG(Iznos) FROM
                     (SELECT Iznos FROM
                      (SELECT TOP 10 * FROM Narudzbe
                       ORDER BY Iznos DESC)
                     WHERE Placeno = TRUE))
```

82. Napišite SQL naredbu koja će izdvojiti sve kupce čiji je minimalni iznos neplaćenog računa manji od od prosjeka 3 najmanja neplaćena računa.

```
SELECT KupacID, ROUND(MIN(Iznos),2) AS [Minimalni iznos] FROM Narudzbe
WHERE Placeno = False
GROUP BY KupacID
HAVING MIN(Iznos) <
      (SELECT AVG(Iznos) FROM (SELECT TOP 3 Iznos FROM Narudzbe
                              WHERE Placeno = False
                              ORDER BY Iznos))
```

KupacID	Minimalni iznos
2	100,5
4	97,56

83. Napišite SQL naredbu koja će izdvojiti sve narudžbe kupca koja ima najviše narudžbi.

```
SELECT * FROM Narudzbe
WHERE KupacID IN (SELECT TOP 1 KupacID FROM Narudzbe
                  GROUP BY KupacID
                  ORDER BY SUM(Iznos) DESC)
```

ID	Iznos	DatumNarudzbe	DatumIsporuke	Placeno	KupacID
2	1000,89	27.1.2010	31.1.2010	False	2
3	290,78	27.1.2010	31.1.2010	True	5
5	100,50	13.2.2010	17.2.2010	False	2
10	412,30	21.3.2010	22.3.2010	False	2
12	578,90	29.3.2010	1.4.2010	False	2
14	367,51	12.4.2010	15.4.2010	False	5
19	765,00	3.5.2010	6.5.2010	False	5
22	879,60	12.5.2010	16.5.2010	True	5
24	489,75	21.5.2010	24.5.2010	False	5

U WHERE uvjetu mora biti operator IN a ne '=' jer drugi SELECT može vratiti više od jednog *KupacID* (ako dva kupca ili više njih ima isti (najveći) broj narudžbi).

Ovo vrijedi za Access, u SQL Serveru TOP 1 će uvijek vratiti 1 slog.

ID	Iznos	DatumNarudzbe	DatumIsporuke	Placeno	KupacID
25	532,90	29.5.2010	31.5.2010	True	2
28	109,71	10.6.2010	11.6.2010	True	2
32	340,00	10.7.2010	13.7.2010	False	5

U WHERE uvjetu mora biti operator IN a ne '=' jer drugi SELECT može vratiti više od jednog *KupacID* (ako dva kupca ili više njih ima isti (najveći) broj narudžbi).

Ovo vrijedi za Access, u SQL Serveru TOP 1 će uvijek vratiti 1 slog.

84. Napišite SQL naredbu koja će izdvojiti sve neplaćene narudžbe

```
SELECT * FROM Narudzbe
WHERE KupacID IN (SELECT TOP 1 KupacID FROM Narudzbe
                  WHERE Placeno = False
                  GROUP BY KupacID
                  ORDER BY SUM(Iznos) DESC) AND
```

Placeno = False

ID	Iznos	DatumNarudzbe	DatumIsporuke	Placeno	KupacID
2	1000,89	27.1.2010	31.1.2010	No	2
5	100,50	13.2.2010	17.2.2010	No	2
10	412,30	21.3.2010	22.3.2010	No	2
12	578,90	29.3.2010	1.4.2010	No	2

85. Napišite SQL naredbu koja će izdvojiti sve narudžbe kupca koji ima najviše neplaćenih narudžbi.

```
SELECT * FROM Narudzbe
WHERE KupacID IN (SELECT TOP 1 KupacID FROM Narudzbe
                  WHERE Placeno = False
                  GROUP BY KupacID
                  ORDER BY COUNT(Iznos) DESC)
ORDER BY KupacID, DatumNarudzbe
```

ID	Iznos	DatumNarudzbe	DatumIsporuke	Placeno	KupacID
2	1000,89	27.1.2010	31.1.2010	No	2
5	100,50	13.2.2010	17.2.2010	No	2
10	412,30	21.3.2010	22.3.2010	No	2
12	578,90	29.3.2010	1.4.2010	No	2
25	532,90	29.5.2010	31.5.2010	Yes	2
28	109,71	10.6.2010	11.6.2010	Yes	2
3	290,78	27.1.2010	31.1.2010	Yes	5
14	367,51	12.4.2010	15.4.2010	No	5
19	765,00	3.5.2010	6.5.2010	No	5
22	879,60	12.5.2010	16.5.2010	Yes	5
24	489,75	21.5.2010	24.5.2010	No	5
32	340,00	10.7.2010	13.7.2010	No	5

86. Napišite SQL naredbu koja će izdvojiti sve neplaćene narudžbe kupca koji ima najviše dugovanja u mjesecu travnju 2010.


```
SELECT * FROM Narudzbe
WHERE KupacID IN (SELECT TOP 1 KupacID FROM Narudzbe
WHERE Placeno = False AND DatumNarudzbe >= #2010-4-1#
AND DatumNarudzbe <= #2010-4-30#
GROUP BY KupacID
ORDER BY SUM(Iznos) DESC) AND Placeno = False
```

ID	Iznos	DatumNarudzbe	DatumIsporuke	Placeno	KupacID
16	909,45	21.4.2010	25.4.2010	No	7
39	510,00	20.8.2010	22.8.2010	No	7

Posao je olakšan utoliko što nema kombinacije AND i OR operatora, pa ne moramo paziti na prioritete i zagrade.

87. Napišite SQL naredbu koja će izračunati koja dva kupca imaju najveću ukupnu kamatu za neplaćene račune. Kamate računajte po kamatnoj stopi 6.55 %. Za godinu pretpostavite da nije prijestupna.

```
SELECT TOP 2 KupacID, ROUND(SUM((NOW() - DatumNarudzbe) / 365 * 0.0655 * Iznos), 2)
AS [Ukupna kamata] FROM Narudzbe
WHERE Placeno = False
GROUP BY KupacID
ORDER BY SUM((NOW() - DatumNarudzbe) / 365 * 0.0655 * Iznos) DESC
```

KupacID	Ukupna kamata
2	116,14
5	80,48

Broj proteklih dana zaokružujemo na cijeli broj samo kod izračuna (SELECT) jer to nalažu pravila izračuna kamata.
Kod sortiranja po toj istoj vrijednosti to nije neophodno jer će slogovi svakako biti ispravno sortirani.

Naredba JOIN

U svim dosadašnjim primjerima podaci su se izdvajali iz jedne tablice i iako su se rezultirajući podaci odnosili na dvije tablice (Kupci i Narudzbe) to je bilo samo zbog relacijske veze između tablica preko polja *KupacID*.

Stvarnost je dakle bila da je određeni upit mogao dati vrijednosti polja samo jedne tablice. To se može promijeniti uvođenjem naredbe JOIN koja omogućuje da se dohvate podaci iz više različitih tablica.

Postoje različite vrste JOIN operacija, a uz pretpostavku da su *Kupci* lijeva a *Narudzbe* desna tablica to bi ovako izgledalo:

- INNER JOIN – najčešći oblik, izdvajaju se samo kupci koji imaju barem jednu narudžbu, odnosno narudžbe koje imaju kupca.
- LEFT JOIN – izdvajaju se svi kupci bez obzira imaju li neku narudžbu plus sve narudžbe koje imaju definiranog kupca.
- RIGHT JOIN – izdvajaju se samo kupci koji imaju narudžbu plus sve narudžbe bez obzira imaju li definiranog kupca (iako narudžba bez kupca ne bi smjela postojati).

Sintaksa upotrebe naredbi JOIN je sljedeća:

SELECT polje **FROM** Naziv_Tablice1

INNER JOIN Naziv_Tablice2 **ON** Naziv_Tablice1.polje = Naziv_Tablice2.polje

gdje su:

- Naziv_Tablice1 i Naziv_Tablice2 – nazivi tablica koje su u relacijskoj vezi

Zadaci:

88. Napišite SQL naredbu koja će izdvojiti sve neplaćene račune veće od 500 kn. Prikažite sljedeće podatke: **Ime i prezime kupca, Datum narudžbe i Iznos**. Račune sortirajte po iznosu silazno.

```
SELECT Ime, Prezime, DatumNarudzbe, Iznos FROM Kupci  
INNER JOIN Narudzbe ON Narudzbe.KupacID = Kupci.ID  
WHERE Placeno = False AND Iznos > 500  
ORDER BY Iznos DESC
```

ILI

```
SELECT Ime, Prezime, DatumNarudzbe, Iznos FROM Narudzbe, Kupci  
WHERE Narudzbe.KupacID = Kupci.ID AND  
Placeno = False AND Iznos > 500  
ORDER BY Iznos DESC
```

Ime	Prezime	DatumNarudzbe	Iznos
Antonio	Moreno	27.1.2010	1000,89
Horst	Kloss	21.4.2010	909,45
James	Ashworth	11.8.2010	900,80
Hanna	Moos	11.3.2010	800,00
Yang	Wang	3.5.2010	765,00
Elizabeth	Brown	7.6.2010	645,50
Patricio	Simpson	22.7.2010	600,00
Antonio	Moreno	29.3.2010	578,90

Sve primjere u ovom poglavlju riješit ćemo na dva načina s naredbom JOIN i bez nje. Iako je ovaj drugi način "programerskiji", prvi je nekako standardniji.

Najvažnija novost je što sad koristimo polja iz dviju tablica koje su povezane preko zajedničkog polja. Tu poveznicu zadajemo iza ključne riječi ON koja je sastavni dio naredbe JOIN (prvi način) ili jednostavno kao uvjet u WHERE naredbi (drugi način).

Ime	Prezime	DatumNarudzbe	Iznos
Horst	Kloss	20.8.2010	510,00

89. Napišite SQL naredbu koja će izdvojiti sve kupce neplaćenih računa starijih od 180 dana. Račune sortirajte po iznosu silazno, pa po datumu narudžbe.

```
SELECT Ime, Prezime, DatumNarudzbe, Iznos FROM Kupci  
INNER JOIN Narudzbe ON Narudzbe.KupacID = Kupci.ID  
WHERE NOW() - DatumNarudzbe > 180 AND Placeno = False  
ORDER BY Iznos DESC, DatumNarudzbe
```

ILI

```
SELECT Ime, Prezime, DatumNarudzbe, Iznos FROM Narudzbe, Kupci  
WHERE Narudzbe.KupacID = Kupci.ID AND  
NOW() - DatumNarudzbe > 180 AND Placeno = False  
ORDER BY Iznos DESC, DatumNarudzbe
```

Ime	Prezime	DatumNarudzbe	Iznos
Antonio	Moreno	27.1.2010	1000,89
Hanna	Moos	11.3.2010	800,00
Antonio	Moreno	29.3.2010	578,90
Antonio	Moreno	21.3.2010	412,30
James	Ashworth	25.2.2010	190,00
Antonio	Moreno	13.2.2010	100,50

Mnogi će se složiti da je drugi način razumljiviji i čitljiviji jer se s operatorima AND i/ili OR navodi obvezna poveznica između zajedničkog polja (*KupacID* u tablici *Narudzbe* i *ID* u tablici *Kupci*) zajedno s ostalim uvjetima koji moraju biti zadovoljeni, a to su u ovom slučaju zadana starost i neplaćenost računa.

90. Napišite SQL naredbu koja će izdvojiti imena i prezimena svih dužnika ne ponavljajući pritom dužnike.

```
SELECT DISTINCT (Ime + " " + Prezime) AS [Ime i prezime] FROM Kupci  
INNER JOIN Narudzbe ON Narudzbe.KupacID = Kupci.ID  
WHERE Placeno = False
```

ILI

```
SELECT DISTINCT (Ime + " " + Prezime) AS [Ime i prezime] FROM Narudzbe, Kupci  
WHERE Narudzbe.KupacID = Kupci.ID AND  
Placeno = False
```

Ime i prezime
Antonio Moreno
Elizabeth Brown
Elizabeth Lincoln
Hanna Moos
Horst Kloss
James Ashworth
Maria Anders
Pascale Cartrain
Patricio Simpson

Umjesto naredbe `DISTINCT`, za isti smo rezultat mogli upotrijebiti naredbu `GROUP BY`:

```
SELECT Ime, Prezime FROM Narudzbe, Kupci  
WHERE Narudzbe.KupacID = Kupci.ID AND Placeno = False  
GROUP BY Ime, Prezime
```

Ime i prezime
Yang Wang

91. Napišite SQL naredbu koja će ispisati sve podatke o tri dužnika koji imaju najviše ukupnog duga. Ispišite i njihov ukupni dug.

```
SELECT TOP 3 Kupci.ID, Ime, Prezime, ROUND(SUM(Iznos),2) AS [Iznos duga] FROM Kupci  
INNER JOIN Narudzbe ON Kupci.ID = Narudzbe.KupacID  
WHERE Placeno = False  
GROUP BY Kupci.ID, Ime, Prezime  
ORDER BY SUM(Iznos) DESC
```

ILI

```
SELECT TOP 3 Kupci.ID, Ime, Prezime, ROUND(SUM(Iznos),2) AS [Iznos duga]  
FROM Narudzbe, Kupci  
WHERE Kupci.ID = Narudzbe.KupacID  
AND Placeno = False  
GROUP BY Kupci.ID, Ime, Prezime  
ORDER BY SUM(Iznos) DESC
```

ID	Ime	Prezime	Iznos duga
2	Antonio	Moreno	2092,59
5	Yang	Wang	1962,26
9	James	Ashworth	1511,36

Novo u ovom primjeru je jedino što su sada objedinjene naredbe JOIN i GROUP BY tako da ovaj primjer za razliku od dosadašnjih u grupiranju podataka prikazuje imena i prezimena kupaca.

92. Napišite SQL naredbu koja će ispisati ime i prezime dužnika koji ima najveći broj neplaćenih računa. Ispišite i broj njegovih neplaćenih računa.

```
SELECT TOP 1 Kupci.ID, Ime, Prezime, COUNT(Iznos) AS Ukupno FROM Kupci  
INNER JOIN Narudzbe ON Kupci.ID = Narudzbe.KupacID  
WHERE Placeno = False  
GROUP BY Kupci.ID, Ime, Prezime  
ORDER BY COUNT(Iznos) DESC
```

ILI

```
SELECT TOP 1 Kupci.ID, Ime, Prezime, COUNT(Iznos) AS Ukupno FROM Narudzbe, Kupci  
WHERE Narudzbe.KupacID = Kupci.ID AND Placeno = False  
GROUP BY Kupci.ID, Ime, Prezime  
ORDER BY COUNT(Iznos) DESC
```

ID	Ime	Prezime	Ukupno
5	Yang	Wang	4
2	Antonio	Moreno	4

Kao što smo već naveli, TOP 1 u Access-u vraća više slogova ako ih više od jednog ima najveću vrijednost po kojoj se sortira (u ovom slučaju broj narudžbi), dok će SQL Server uvijek vratiti samo jedan slog. Također, u SQL Serveru *False* i *True* treba navesti u navodnicima:
Placeno = 'False'

93. Napišite SQL naredbu koja će ispisati imena i prezimena svih dužnika koji imaju ukupan dug veći od 1500. Ispišite i iznose njihovih neplaćenih računa sortirane silazno.

```
SELECT Kupci.ID, Ime, Prezime, ROUND(SUM(Iznos),2) AS [Iznos duga] FROM Kupci  
INNER JOIN Narudzbe ON Kupci.ID = Narudzbe.KupacID  
WHERE Placeno = False  
GROUP BY Kupci.ID, Ime, Prezime  
HAVING SUM(Iznos) > 1500  
ORDER BY SUM(Iznos) DESC
```

ILI

```
SELECT Kupci.ID, Ime, Prezime, ROUND(SUM(Iznos),2) AS [Iznos duga] FROM Narudzbe, Kupci  
WHERE Kupci.ID = Narudzbe.KupacID AND  
Placeno = False  
GROUP BY Kupci.ID, Ime, Prezime  
HAVING SUM(Iznos) > 1500  
ORDER BY SUM(Iznos) DESC
```

ID	Ime	Prezime	Iznos duga
2	Antonio	Moreno	2092,59
5	Yang	Wang	1962,26
9	James	Ashworth	1511,36

U odnosu na prethodnih nekoliko primjera novo je uvođenje naredbe HAVING, ali opet na isti, dobro poznati način. Kao što znamo, naredba HAVING za razliku od naredbe WHERE izvodi se na već izračunatim i grupiranim podacima.

94. Napišite SQL naredbu koja će ispisati ime i prezime kupca koji ima najmanje ukupno dugovanje, ali uzmite u obzir samo kupce koji imaju više od jednog neplaćenog računa.

```
SELECT TOP 1 Kupci.ID, Ime, Prezime, SUM(Iznos) AS [Iznos duga] FROM Kupci  
INNER JOIN Narudzbe ON Kupci.ID = Narudzbe.KupacID  
WHERE Placeno = False  
GROUP BY Kupci.ID, Ime, Prezime  
HAVING COUNT(Iznos) > 1  
ORDER BY SUM(Iznos)
```

ILI

```
SELECT TOP 1 Kupci.ID, Ime, Prezime, SUM(Iznos) AS [Iznos duga] FROM Narudzbe, Kupci  
WHERE Kupci.ID = Narudzbe.KupacID AND Placeno = False  
GROUP BY Kupci.ID, Ime, Prezime  
HAVING COUNT(Iznos) > 1  
ORDER BY SUM(Iznos)
```

ID	Ime	Prezime	Iznos duga
6	Patricio	Simpson	810,05

95. Napišite SQL naredbu koja će izračunati u kojem mjesecu je ukupno najveće dugovanje, ali uzmite u obzir samo mjesece s barem 4 neplaćena računa.

```
SELECT TOP 1 MONTHNAME(MONTH(DatumNarudzbe)) AS Mjesec, YEAR(DatumNarudzbe) AS Godina,  
SUM(Iznos) AS [Ukupno dugovanje] FROM Narudzbe
```

WHERE Placeno = False
GROUP BY MONTH(DatumNarudzbe), YEAR(DatumNarudzbe)
HAVING COUNT(Iznos) > 3
ORDER BY SUM(Iznos) DESC

Mjesec	Ukupno dugovanje
kolovoz	2092,03

Funkciju MONTH koja vraća brojčanu vrijednost mjeseca iz proslijeđenog datuma kao argumenta već smo upoznali. Sukladno tome funkcija YEAR vraća brojčanu vrijednost godine iz proslijeđenog datuma kao argumenta.

Postoje još i sljedeće funkcije:

- DAY – vraća dan u mjesecu
- WEEKDAY – vraća dan u tjednu (1-ned, 2-pon, ...)
- HOUR, MINUTE, SECOND – vraća sate, minute i sekunde iz vremena (polje DatumNarudzbe ima definiran i datum i vrijeme).

SQL Server za dohvat ovih podataka koristi funkciju DATEPART kojoj se kao drugi argument navodi vrijeme, a kao prvi argument ono što se želi dobiti: dw (dan u tjednu), hour, minute, second, ...

96. Napišite SQL naredbu koja će izdvojiti sve račune čiji je iznos barem 50 % veći od prosječnog iznosa računa. Prikažite sljedeće podatke: **Ime i prezime kupca, Datum narudžbe i Iznos.**

SELECT Ime, Prezime, DatumNarudzbe, Iznos FROM Kupci
INNER JOIN Narudzbe ON Narudzbe.KupacID = Kupci.ID
WHERE Iznos > (SELECT AVG(Iznos) FROM Narudzbe) * 1.5

ILI

SELECT Ime, Prezime, DatumNarudzbe, Iznos FROM Narudzbe, Kupci
WHERE Narudzbe.KupacID = Kupci.ID AND
Iznos > (SELECT AVG(Iznos) FROM Narudzbe) * 1.5

Ime	Prezime	DatumNarudzbe	Iznos
Antonio	Moreno	27.1.2010	1000,89
Hanna	Moos	11.3.2010	800,00
Horst	Kloss	21.4.2010	909,45
Yang	Wang	3.5.2010	765,00
Horst	Kloss	5.5.2010	1025,77
Yang	Wang	12.5.2010	879,60
Pascale	Cartrain	14.7.2010	1300,10
James	Ashworth	11.8.2010	900,80

97. Napišite SQL naredbu koja će ispisati imena i prezimena kupaca čiji je prosječni iznos računa veći od ukupnog prosječnog iznosa. Ispišite i prosječne iznose njihovih računa.

SELECT Kupci.ID, Ime, Prezime, ROUND(AVG(Iznos),2) AS [Prosječni iznos] FROM Kupci
INNER JOIN Narudzbe ON Kupci.ID = Narudzbe.KupacID
GROUP BY Kupci.ID, Ime, Prezime
HAVING AVG(Iznos) > (SELECT AVG(Iznos) FROM Narudzbe)

ILI

```
SELECT Kupci.ID, Ime, Prezime, ROUND(AVG(Iznos),2) AS [Prosječni iznos] FROM Narudzbe, Kupci
WHERE Kupci.ID = Narudzbe.KupacID
GROUP BY Kupci.ID, Ime, Prezime
HAVING AVG(Iznos) > (SELECT AVG(Iznos) FROM Narudzbe)
```

ID	Ime	Prezime	Prosječni iznos
8	Elizabeth	Brown	528,76
7	Horst	Kloss	636,26
10	Pascale	Cartrain	790,52
5	Yang	Wang	522,11

98. Napišite SQL naredbu koja će ispisati imena i prezimena kupaca čiji je prosječni iznos plaćenog računa veći od ukupnog prosječnog iznosa plaćenih računa. Ispišite i prosječne iznose njihovih plaćenih računa.

```
SELECT Kupci.ID, Ime, Prezime, ROUND(AVG(Iznos),2) AS [Prosječni iznos] FROM Kupci
INNER JOIN Narudzbe ON Kupci.ID = Narudzbe.KupacID
WHERE Placeno = True
GROUP BY Kupci.ID, Ime, Prezime
HAVING AVG(Iznos) > (SELECT AVG(Iznos) FROM Narudzbe
WHERE Placeno = True)
```

ILI

```
SELECT Kupci.ID, Ime, Prezime, ROUND(AVG(Iznos),2) AS [Prosječni iznos] FROM Narudzbe, Kupci
WHERE Kupci.ID = Narudzbe.KupacID AND
Placeno = True
GROUP BY Kupci.ID, Ime, Prezime
HAVING AVG(Iznos) > (SELECT AVG(Iznos) FROM Narudzbe
WHERE Placeno = True)
```

ID	Ime	Prezime	Prosječni iznos
8	Elizabeth	Brown	590,78
4	Elizabeth	Lincoln	493,75
7	Horst	Kloss	562,78
10	Pascale	Cartrain	1300,10
5	Yang	Wang	585,19

Razlikujte ovaj primjer od prethodnog. U prethodnom su se tražili svi računi kupca koji su udovoljavali uvjetu, dok se u ovom traže kupci čiji skupni iznos treba udovoljavati uvjetu. Zato se u prvom primjeru uvjet mogao izraziti WHERE naredbom jer se u tijeku prolaska kroz slogove odmah zna udovoljava li slog uvjetu ili ne. U ovom primjeru to se zna tek na kraju prolaza svih slogova i zato se koristi naredba HAVING.

99. Napišite SQL naredbu koja će ispisati imena i prezimena kupaca čiji je maksimalni iznos računa manji od ukupnog prosječnog iznosa računa.

```
SELECT Kupci.ID, Ime, Prezime FROM Kupci
INNER JOIN Narudzbe ON Kupci.ID = Narudzbe.KupacID
GROUP BY Kupci.ID, Ime, Prezime
HAVING MAX(Iznos) < (SELECT AVG(Iznos) FROM Narudzbe)
```

ILI

```
SELECT Kupci.ID, Ime, Prezime FROM Narudzbe, Kupci
WHERE Kupci.ID = Narudzbe.KupacID
GROUP BY Kupci.ID, Ime, Prezime
HAVING MAX(Iznos) < (SELECT AVG(Iznos) FROM Narudzbe)
```

ID	Ime	Prezime
1	Maria	Anders

100. Napišite SQL naredbu koja će ispisati imena i prezimena kupaca čija je prosječni iznos narudžbi veći od prosječnog iznosa narudžbe kupca Antonio Moreno. Ispišite i prosječne iznose narudžbi.

```
SELECT Kupci.ID, Kupci.Ime, Kupci.Prezime, ROUND(AVG(Iznos),2) AS [Prosječni iznos] FROM Kupci
INNER JOIN Narudzbe ON Kupci.ID = Narudzbe.KupacID
GROUP BY Kupci.ID, Kupci.Ime, Kupci.Prezime
HAVING AVG(Iznos) > (SELECT AVG(Iznos) FROM Narudzbe
WHERE KupacID = 2)
```

ILI

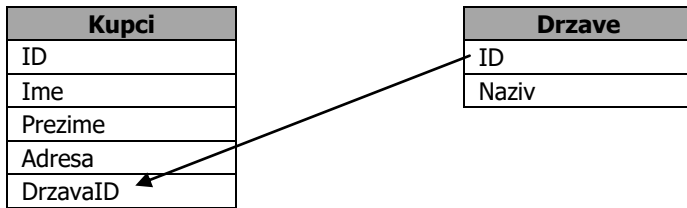
```
SELECT Kupci.ID, Kupci.Ime, Kupci.Prezime, ROUND(AVG(Iznos),2) AS [Prosječni iznos]
FROM Narudzbe, Kupci
WHERE Kupci.ID = Narudzbe.KupacID
GROUP BY Kupci.ID, Kupci.Ime, Kupci.Prezime
HAVING AVG(Iznos) > (SELECT AVG(Iznos) FROM Narudzbe
WHERE KupacID = 2)
```

ID	Ime	Prezime	Prosječni iznos
5	Yang	Wang	522,11
7	Horst	Kloss	636,26
8	Elizabeth	Brown	528,76
9	James	Ashworth	467,95
10	Pascale	Cartrain	790,52

U ovom primjeru navodili smo puno ime polja navođenjem naziva tablice i naziva polja odvojene točkom (*Kupci.Ime, Kupci.Prezime*, itd...). Ako se polje s istim nazivom ne nalazi u obje tablice to nije neophodno, ali bez obzira na veći kôd ovaj način je možda čitljiviji jer se odmah vidi kojoj tablici pripada polje. Polje ID se međutim mora navesti s nazivom tablice jer se nalazi i u tablici *Kupci* i u tablici *Narudzbe*.

JOIN s tri tablice

Za izvršavanje sljedećih primjera dodajte treću tablicu **Drzave** s poljima **ID** i **Naziv**, te je dovedite u relacijsku vezu **One to Many** s tablicom **Kupci** na sljedeći način:



te unesite sljedeće slogove u tablicu **Drzave**:

ID	Naziv
1	Hrvatska
2	Njemačka
3	Kolumbija
4	SAD
5	Engleska
6	Nizozemska
7	Italija
8	Argentina
9	Francuska
10	Singapur

dok u tablicu **Kupci** dodajte polje **DrzavaID**:

ID	Ime	Prezime	Adresa	DrzavaID
1	Maria	Anders	Obere Str. 57, Bogota	3
2	Antonio	Moreno	Mataderos 2312, San Jose	4
3	Hanna	Moos	Forsterstr. 57, Amsterdam	6
4	Elizabeth	Lincoln	Fauntleroy Circus, London	5
5	Yang	Wang	87 Polk St, Singapur	10
6	Patricio	Simpson	Moncada 8585, Buenos Aires	8
7	Horst	Kloss	Taucherstrasse 10, Dortmund	2
8	Elizabeth	Brown	2743 Bering St., Boston	4
9	James	Ashworth	89 Jefferson Way, Portland	4
10	Pascale	Cartrain	Boulevard Tirou, 255, Lyon	9

101. Napišite SQL naredbu koja će ispisati sve iznose neplaćenih računa većih od 500 kn, te će ispisati imena i prezimena kupaca i država kojima kupci pripadaju.

```
SELECT Narudzbe.Iznos, Kupci.Ime, Kupci.Prezime, Drzave.Naziv
FROM (Narudzbe INNER JOIN Kupci ON Narudzbe.KupacID=Kupci.ID)
INNER JOIN Drzave ON Kupci.DrzavaID=Drzave.ID
WHERE Placeno = False AND Narudzbe.Iznos > 500
```

ILI

```
SELECT Narudzbe.Iznos, Kupci.Ime, Kupci.Prezime, Drzave.Naziv  
FROM Narudzbe, Kupci, Drzave  
WHERE Narudzbe.KupacID=Kupci.ID AND Kupci.DrzavaID=Drzave.ID AND  
Placeno = False AND Narudzbe.Iznos > 500
```

Iznos	Ime	Prezime	Naziv
1000,89	Antonio	Moreno	SAD
800,00	Hanna	Moos	Nizozemska
578,90	Antonio	Moreno	SAD
909,45	Horst	Kloss	Njemačka
765,00	Yang	Wang	Singapur
645,50	Elizabeth	Brown	SAD
600,00	Patricio	Simpson	Argentina
900,80	James	Ashworth	SAD
510,00	Horst	Kloss	Njemačka

Sad su povezana dva para tablica tako da dvaput uvjetujemo jednakost polja: kupca u tablici narudžbi i državu u tablici kupaca. S obzirom da se podaci dohvaćaju iz tri tablice zbog jasnijeg prikaza navodimo za svako polje i tablicu kojoj polje pripada.

102. Napišite SQL naredbu koja će izračunati ukupni iznos računa grupirano po državama.

```
SELECT Drzave.ID, Drzave.Naziv, SUM(Narudzbe.Iznos) AS [Ukupni iznos]  
FROM (Narudzbe INNER JOIN Kupci ON Narudzbe.KupacID=Kupci.ID)  
INNER JOIN Drzave ON Kupci.DrzavaID=Drzave.ID  
GROUP BY Drzave.ID, Drzave.Naziv
```

ILI

```
SELECT Drzave.ID, Drzave.Naziv, SUM(Narudzbe.Iznos) AS [Ukupni iznos]  
FROM Narudzbe, Kupci, Drzave  
WHERE Narudzbe.KupacID=Kupci.ID AND Kupci.DrzavaID=Drzave.ID  
GROUP BY Drzave.ID, Drzave.Naziv
```

ID	Naziv	Ukupni iznos
2	Njemačka	2545,02
3	Kolumbija	928,93
4	SAD	6193,29
5	Engleska	1578,81
6	Nizozemska	1865,71
8	Argentina	898,78
9	Francuska	1581,04
10	Singapur	3132,64

Ako bismo u primjeru koristili RIGHT JOIN dobili bismo u popisu i Hrvatsku i Italiju, države koje nemaju kupaca s narudžbama.

103. Napišite SQL naredbu koja će izračunati prosječan iznos računa grupirano po državama, uzimajući u obzir samo države koje imaju manje od 4 računa.

```
SELECT Drzave.ID, Drzave.Naziv, ROUND(AVG(Narudzbe.Iznos), 2) AS [Prosječni iznos]  
FROM (Narudzbe INNER JOIN Kupci ON Narudzbe.KupacID= Kupci.ID)  
INNER JOIN Drzave ON Kupci.DrzavaID=Drzave.ID  
GROUP BY Drzave.ID, Drzave.Naziv  
HAVING COUNT(*) < 4
```

ILI

```
SELECT Drzave.ID, Drzave.Naziv, ROUND(AVG(Narudzbe.Iznos), 2) AS [Prosječni iznos]
FROM Narudzbe, Kupci, Drzave
WHERE Narudzbe.KupacID= Kupci.ID AND Kupci.DrzavaID=Drzave.ID
GROUP BY Drzave.ID, Drzave.Naziv
HAVING COUNT(*) < 4
```

ID	Naziv	Prosječni iznos
3	Kolumbija	309,64
8	Argentina	299,59
9	Francuska	790,52

Primjer je vrlo sličan prethodnom uz dodatni uvjet o broju računa koji se mora riješiti uz pomoć naredbe HAVING.

104. Napišite SQL naredbu koja će izračunati ukupni iznos računa grupirano po državama, uzimajući u obzir samo države koje imaju više računa nego što ih ima Maria Anders.

```
SELECT Drzave.ID, Drzave.Naziv, SUM (Narudzbe.Iznos) AS [Prosječni iznos]
FROM (Narudzbe INNER JOIN Kupci ON Narudzbe.KupacID=Kupci.ID)
INNER JOIN Drzave ON Kupci.DrzavaID= Drzave.ID
GROUP BY Drzave.ID, Drzave.Naziv
HAVING COUNT(*) > (SELECT COUNT(*) FROM Narudzbe WHERE KupacID = 1)
```

ILI

```
SELECT Drzave.ID, Drzave.Naziv, SUM (Narudzbe.Iznos) AS [Prosječni iznos]
FROM Narudzbe, Kupci, Drzave
WHERE Narudzbe.KupacID=Kupci.ID AND Kupci.DrzavaID=Drzave.ID
GROUP BY Drzave.ID, Drzave.Naziv
HAVING COUNT(*) > (SELECT COUNT(*) FROM Narudzbe WHERE KupacID = 1)
```

ID	Naziv	Prosječni iznos
2	Njemačka	2545,02
4	SAD	6193,29
5	Engleska	1578,81
6	Nizozemska	1865,71
10	Singapur	3132,64

U SQL naredbi koja vraća vrijednost koja predstavlja ograničenje (broj računa od Marie Anders) navodimo njen ID iz tablice *Narudzbe*.
Ako bismo željeli umjesto njenog ID-a navesti ime i prezime morali bismo uraditi JOIN tablica *Kupci* i *Narudzbe*:

```
HAVING COUNT(*) >
SELECT COUNT(*) FROM Narudzbe, Kupci WHERE Kupci.ID
= Narudzbe.KupacID AND
Kupci.Ime = 'Maria' AND Kupci.Prezime='Anders'
```

105. Napišite SQL naredbu koja će izračunati koja država ne računajući SAD ima najviše računa.

```
SELECT TOP 1 Drzave.ID, Drzave.Naziv, COUNT(*) AS [Broj računa]
FROM (Narudzbe INNER JOIN Kupci ON Narudzbe.KupacID=Kupci.ID)
INNER JOIN Drzave ON Kupci.DrzavaID=Drzave.ID
WHERE DrzavaID <> 4
GROUP BY Drzave.ID, Drzave.Naziv
ORDER BY COUNT(Narudzbe.Iznos) DESC
```

ILI

```
SELECT TOP 1 Drzave.ID, Drzave.Naziv, COUNT(*) AS [Broj računa]
FROM Narudzbe, Kupci, Drzave
WHERE Narudzbe.KupacID= Kupci.ID AND Kupci.DrzavaID=Drzave.ID AND DrzavaID <> 4
GROUP BY Drzave.ID, Drzave.Naziv
ORDER BY COUNT(Narudzbe.Iznos) DESC
```

ID	Naziv	Broj računa
10	Singapur	6

Ako bismo, slično kao i u prošlom primjeru, htjeli navesti usporedbu s nazivom države (SAD), a ne s njenim ID-om iz tablice *Kupci*, ne bismo trebali kreirati nikakav novi JOIN jer je on već urađen.

Jednostavno bismo naveli:

```
WHERE Drzave.Naziv <> 'SAD'
ili s obzirom da jedino tablica Drzave ima polje Naziv
WHERE Naziv <> 'SAD'
```

106. Napišite SQL naredbu koja će izdvojiti sve države čiji kupci koji imaju barem jednu narudžbu.

```
SELECT DISTINCT Drzave.Naziv
FROM (Narudzbe INNER JOIN Kupci ON Narudzbe.KupacID= Kupci.ID)
INNER JOIN Drzave ON Kupci.DrzavaID= Drzave.ID
```

ILI

```
SELECT DISTINCT Drzave.Naziv
FROM Narudzbe, Kupci, Drzave
WHERE Narudzbe.KupacID= Kupci.ID AND Kupci.DrzavaID = Drzave.ID
```

Naziv
Argentina
Engleska
Francuska
Kolumbija
Nizozemska
Njemačka
SAD
Singapur

Zadatak nam rješava sama INNER JOIN definicija, a kako Hrvatska i Italija nemaju niti jednog kupca nema ih u popisu. Ako bi i imale definiranog kupca, ali bez izvršene narudžbe, opet ne bi bile u rezultatima jer bi veza *Kupci <-> Narudzbe* vratila *false*.

107. Napišite SQL naredbu koja će izdvojiti sve države koje nemaju niti jednu narudžbu.

```
SELECT Drzave.Naziv
FROM (Drzave LEFT JOIN Kupci ON Drzave.ID=Kupci.DrzavaID) LEFT JOIN Narudzbe ON
Narudzbe.KupacID = Kupci.ID
WHERE Drzave.ID NOT IN
(
SELECT Drzave.ID
FROM Narudzbe, Kupci, Drzave
WHERE Narudzbe.KupacID= Kupci.ID AND Kupci.DrzavaID = Drzave.ID
);
```

Naziv
Hrvatska
Italija

Zadatak je inverzan u odnosu na prethodni. S obzirom da je naveden LEFT JOIN, Hrvatska i Italija će biti u rezultatima izvršenja prve SELECT naredbe. Kako će druga SELECT naredba izdvojiti samo države koje imaju po neku narudžbu razlika između prvih i drugih rezultata daje tražene države.

Naredbe INSERT, UPDATE i DELETE

Sve SQL naredbe koje smo do sada upoznali nisu radile nikakve promjene nad podacima tablice baze podataka. Naredbe koje mogu mijenjati podatke u tablici upoznat ćemo u ovom poglavlju preko tri različite operacije koje se mogu izvršiti nad podacima, a to su umetanje novih slogova, izmjena postojećih te brisanje slogova iz tablice.

Sintaksa naredbi za izmjenu podataka u tablici je sljedeća:

INSERT INTO Naziv_Tablice
VALUES (vrijednost1, vrijednost2, vrijednost3, ...)

gdje je:

- o vrijednost1, vrijednost2, vrijednost3 – vrijednosti polja novododanog sloga

UPDATE Naziv_Tablice
SET polje1 = vrijednost1, polje2 = vrijednost2, ...
WHERE polje operator vrijednost

DELETE FROM Naziv_Tablice
WHERE polje operator vrijednost

Zadaci:

108. Napišite SQL naredbu koja će dodati sljedećeg kupca u tablicu:

- Ime: John
- Prezime: Smith
- Adresa: Washington way 45, NewYork

INSERT INTO Kupci (Ime, Prezime, Adresa)
VALUES ('John', 'Smith', 'Washington way 45, NewYork')

Vrijednosti polja tipa *text* i *datetime* navode se unutar jednostrukih navodnika, dok se brojčane i *true/false* vrijednosti navode bez njih. Redoslijed navedenih polja mora biti podudaran.

Tako bi se novi slog u tablicu *Narudzbe* dodao na sljedeći način:
INSERT INTO Narudzbe (Iznos, DatumNarudzbe, DatumIsporuke, Placeno, KupacID)
VALUES (500, '2010-08-27', '2010-08-28', True, 5)

109. Napišite SQL naredbu koja će za 1 kn povećati iznos svih neplaćenih računa koji su manji od 100 kn.

UPDATE Narudzbe
SET Iznos = Iznos + 1
WHERE Iznos < 100 AND Placeno = False

SQL nema ugrađene operatore +=, -=, *= itd...

110. Napišite SQL naredbu koja će svim neplaćenim računima promijeniti vrijednost polja **Placeno**.

UPDATE Narudzbe
SET Placeno = True

WHERE Placeno = False

111. Napišite SQL naredbu koja će sve neplaćene račune od kupca Hanna Moos pretvoriti u plaćene.

```
UPDATE Narudzbe
SET Placeno = True
WHERE KupacID = 3 AND Placeno = False
```

112. Napišite SQL naredbu koja će sve neplaćene račune prije 1.3.2010 manje od 100 kn pretvoriti u plaćene.

```
UPDATE Narudzbe
SET Placeno = True
WHERE DatumNarudzbe < #2010-3-1# AND Placeno = False AND Iznos < 100
```

113. Napišite SQL naredbu koja će iznose svih neplaćenih računa uvećati za 10 %, te će im Datum narudžbe postaviti na 31.12.2010.

```
UPDATE Narudzbe
SET Iznos = Iznos*1.1, DatumNarudzbe = #2010-12-31#
WHERE Placeno = False
```

U SET dijelu naredbe UPDATE može se navesti i više izraza koji će promijeniti vrijednosti više različitih polja u slogu.

114. Napišite SQL naredbu koja će svim računima osim onim od kupaca Maria Anders i Horst Kloss postaviti datum isporuke jednak datumu narudžbe.

```
UPDATE Narudzbe
SET DatumIsporuke = DatumNarudzbe
WHERE KupacID <> 1 AND KupacID <> 7
```

U ovom primjeru vrijednosti jednog polja dodjeljujemo vrijednost drugog polja iz istog sloga.

115. Napišite SQL naredbu koja će svim računima čija roba nije isporučena unutar četiri dana smanjiti vrijednost za 3 %.

```
UPDATE Narudzbe
SET Iznos = Iznos*0.97
WHERE DatumIsporuke - DatumNarudzbe > 4
```

116. Napišite SQL naredbu koja će kupcu Elizabeth Brown promijeniti prezime u White.

```
UPDATE Kupci
SET Prezime = 'White'
WHERE ID = 8
```

117. Napišite SQL naredbu koja će sve neplaćene račune uvećati za iznos kamate uz kamatnu stopu od 11 %. Za godinu pretpostavite da nije prijestupna.

```
UPDATE Narudzbe
```

```
SET Iznos = Iznos * ( 1 + (ROUND(NOW() - DatumNarudzbe, 0)) / 365 * 0.11)  
WHERE Placeno = False
```

118. Napišite SQL naredbu koja će izbrisati sve neplaćene račune.

```
DELETE * FROM Narudzbe  
WHERE Placeno = False
```

Sintaksa naredbe DELETE je ista kao i kod naredbe SELECT.

119. Napišite SQL naredbu koja će izbrisati sve neplaćene račune starije od 100 dana.

```
DELETE * FROM Narudzbe  
WHERE NOW() - DatumNarudzbe > 100
```

120. Napišite SQL naredbe koje će izbrisati kupca Elizabeth Lincoln.

```
DELETE * FROM Narudzbe  
WHERE KupacID = 4
```

```
DELETE * FROM Kupci  
WHERE ID = 4
```

Tablice su u relacijskoj vezi s nametnutim referencijalnim integritetom koji onemogućuju brisanje kupca koji ima narudžbu. Zato se najprije brišu sve narudžbe kupca, a tek nakon toga i taj kupac iz tablice *Kupci*.

Naredbe CREATE TABLE, ALTER TABLE i DROP TABLE

Naredbe koje slijede ne spadaju u obvezno znanje SQL jezika, ne samo zato što ih je puno lakše uraditi u grafičkom okruženju koje svaki DBMS sustav osigurava, nego i zato što njihovo izvršenje mijenja temelje baze podataka i odnose među tablicama, pa se ne koriste tako često.

Sintaksa naredbi za kreiranje i izmjenu strukture tablice je sljedeća:

CREATE TABLE Naziv_Tablice

```
(  
    polje1 tip,  
    polje2 tip,  
    polje3 tip,  
    ...  
)
```

ALTER TABLE Naziv_Tablice

DROP TABLE Naziv_Tablice

Zadaci:

121. Napišite SQL naredbu koja će kreirati tablicu **Narudzbe2** sa sljedećim poljima:

- ID - Primary Key
- Iznos - Number
- Datum narudžbe - Datum
- Datum isporuke – Datum
- Je li račun plaćen – Boolean
- ID kupca - Number

CREATE TABLE Narudzbe2

```
(  
    ID Integer PRIMARY KEY,  
    Iznos Double,  
    DatumNarudzbe DateTime,  
    DatumIsporuke DateTime,  
    Placeno bit,  
    KupacID Integer  
)
```

Svaki DBMS sustav za svoje baze podataka ima neke posebnosti u tipovima podataka polja, ali ovo bi bili neki standardni tipovi koje svaka baza podataka mora podržavati:

- *int* – cjelobrojna vrijednost
- *float, double, decimal* – realna vrijednost
- *bit* – true/false vrijednost
- *datetime* – datumska vrijednost
- *varchar* – niz znakova varijabilne dužine
- *nvarchar* – niz znakova fiksne dužine

122. Napišite SQL naredbu koja će u tablicu **Narudzbe2** dodati sljedeće slogove:

- ID: 1
- Iznos: 100
- Datum narudzbe: 30.12.2010
- Datum isporuke: 30.12.2010
- Plaćeno: False
- KupacID: 4

- ID: 2

- Iznos: 200
- Datum narudzbe: 31.12.2010
- Datum isporuke: 31.12.2010
- Plaćeno: True
- KupacID: 5

```
INSERT INTO Narudzbe2 (ID, Iznos, DatumNarudzbe, DatumIsporuke, Placeno, KupacID)  
VALUES (1, 100, '2010-12-30', '2010-12-30', False, 4)
```

```
INSERT INTO Narudzbe2 (ID, Iznos, DatumNarudzbe, DatumIsporuke, Placeno, KupacID)  
VALUES (1, 100, '2010-12-30', '2010-12-30', True, 5)
```

123. Napišite SQL naredbu koja će u tablici **Narudzbe2** dodati polje **DatumPlacanja**.

```
ALTER TABLE Narudzbe2  
ADD DatumPlacanja DateTime
```

Dodavanje novog polja u strukturi sloga tablice zaista nije radnja koju ćemo često raditi, a i tada je puno vjerojatnije da će to biti urađeno u grafičkom okruženju DBMS sustava. Svejedno, ni u SQL jeziku naredba nije nimalo složena, navodi se ključnom riječi ADD nakon koje dolaze naziv i tip polja.

124. Napišite SQL naredbu koja će iz tablice **Narudzbe2** izbrisati polje **DatumIsporuke**.

```
ALTER TABLE Narudzbe2  
DROP COLUMN DatumIsporuke
```

125. Napišite SQL naredbu koja će izbrisati cijelu tablicu **Narudzbe2** zajedno sa svim slogovima.

```
DROP TABLE Narudzbe2
```

Brisanje tablice naravno, znači i brisanje svih slogova sadržanih u tablici.

Narudzbe					
ID	Iznos	DatumNarudzbe	DatumIsporuke	Placeno	KupacID
1	356,23	24.1.2010	27.1.2010	Yes	3
2	1000,89	27.1.2010	31.1.2010	No	2
3	290,78	27.1.2010	31.1.2010	Yes	5
4	590,78	1.2.2010	4.2.2010	Yes	8
5	100,50	13.2.2010	17.2.2010	No	2
6	99,80	17.2.2010	20.2.2010	Yes	7
7	190,00	25.2.2010	28.2.2010	No	9
8	200,56	4.3.2010	8.3.2010	Yes	1
9	800,00	11.3.2010	15.3.2010	No	3
10	412,30	21.3.2010	22.3.2010	No	2
11	290,80	24.3.2010	29.3.2010	Yes	4
12	578,90	29.3.2010	1.4.2010	No	2
13	612,00	3.4.2010	7.4.2010	Yes	4
14	367,51	12.4.2010	15.4.2010	No	5
15	208,90	17.4.2010	21.4.2010	Yes	3
16	909,45	21.4.2010	25.4.2010	No	7
17	467,70	26.4.2010	27.4.2010	Yes	1
18	290,78	29.4.2010	2.5.2010	No	3
19	765,00	3.5.2010	6.5.2010	No	5
20	1025,77	5.5.2010	8.5.2010	Yes	7
21	97,56	10.5.2010	15.5.2010	No	4
22	879,60	12.5.2010	16.5.2010	Yes	5
23	350,00	18.5.2010	20.5.2010	No	8
24	489,75	21.5.2010	24.5.2010	No	5
25	532,90	29.5.2010	31.5.2010	Yes	2
26	210,05	1.6.2010	4.6.2010	No	6
27	645,50	7.6.2010	12.6.2010	No	8
28	109,71	10.6.2010	11.6.2010	Yes	2
29	209,80	18.6.2010	22.6.2010	No	3
30	88,73	24.6.2010	29.6.2010	Yes	6
31	280,94	3.7.2010	4.7.2010	No	10
32	340,00	10.7.2010	13.7.2010	No	5
33	1300,10	14.7.2010	16.7.2010	Yes	10
34	578,45	20.7.2010	22.7.2010	Yes	4
35	600,00	22.7.2010	22.7.2010	No	6
36	360,45	4.8.2010	5.8.2010	Yes	9
37	260,67	7.8.2010	10.8.2010	No	1
38	900,80	11.8.2010	13.8.2010	No	9
39	510,00	20.8.2010	22.8.2010	No	7
40	420,56	25.8.2010	27.8.2010	No	9

Kupci			
ID	Ime	Prezime	Adresa
1	Maria	Anders	Obere Str. 57, Bogota
2	Antonio	Moreno	Mataderos 2312, San Jose
3	Hanna	Moos	Forsterstr. 57, Amsterdam
4	Elizabeth	Lincoln	Fauntleroy Circus, London
5	Yang	Wang	87 Polk St, Singapur
6	Patricio	Simpson	Moncada 8585, Buenos Aires
7	Horst	Kloss	Taucherstrasse 10, Dortmund
8	Elizabeth	Brown	2743 Bering St., Boston
9	James	Ashworth	89 Jefferson Way, Portland
10	Pascale	Cartrain	Boulevard Tirou, 255, Lyon