

VJEŽBA 5 –NAREDBE JOIN, UNION, EXCEPT, INTERSECT

NAREDBA JOIN (POVEZIVANJE)

Kako riješiti sljedeće upite?

- Od kojeg dobavljača se dobavlja pojedini artikl?
- Koje su narudžbe pojedinih kupaca?
- Koji su artikli koje kupuju određeni kupci?

Ovakvi upiti zahtijevaju korištenje podataka iz više tablica. Za potrebe takvih upita koristimo naredbu JOIN.

SQL naredba JOIN povezuje zapise iz dvije tablice pomoću vrijednosti atributa koje su im zajedničke. Rezultat tog kombiniranja je nova, privremena tablica, tzv. *joined table*.

Osnovni tipovi JOIN naredbe su:

- INNER JOIN (kraće JOIN)
- OUTER JOIN
 - LEFT OUTER JOIN (kraće LEFT JOIN)
 - RIGHT OUTER JOIN (kraće RIGHT JOIN).
- CROSS JOIN

Zajedničke vrijednosti atributa pomoću kojih povezujemo dvije tablice su obično primarni ključ jedne tablice koji je strani ključ u drugoj tablici.

SINTAKSA NAREDBE JOIN

```
SELECT tablica1.atribut, tablica2.atribut
FROM tablica1 INNER JOIN tablica2
  ON tablica1.atribut = tablica2.atribut
WHERE uvjeti
```

Kreiranje upita koji prikazuje sve narudžbe zaposlenika je jednostavan zadatak korištenjem naredbe JOIN.

```
-- Pronađite EmployeeID i OrderID za sve narudžbe

SELECT EmployeeID, OrderID
FROM Orders;

-- Izradite upit koji prikazuje sve narudžbe zaposlenika

SELECT Employees.EmployeeID, Employees.FirstName,
  Employees.LastName, Orders.OrderID, Orders.OrderDate
FROM Employees INNER JOIN Orders ON
  Employees.EmployeeID = Orders.EmployeeID
ORDER BY Orders.OrderDate;
```

	EmployeeID	FirstName	LastName	OrderID	OrderDate
1	5	Steven	Buchanan	10248	1996-07-04 00:00:00.000
2	6	Michael	Suyama	10249	1996-07-05 00:00:00.000
3	4	Margaret	Peacock	10250	1996-07-08 00:00:00.000
4	3	Janet	Leverling	10251	1996-07-08 00:00:00.000
5	4	Margaret	Peacock	10252	1996-07-09 00:00:00.000
6	3	Janet	Leverling	10253	1996-07-10 00:00:00.000
7	5	Steven	Buchanan	10254	1996-07-11 00:00:00.000
8	9	Anne	Dodsworth	10255	1996-07-12 00:00:00.000
9	3	Janet	Leverling	10256	1996-07-15 00:00:00.000
10	4	Margaret	Peacock	10257	1996-07-16 00:00:00.000
11	1	Nancy	Davolio	10258	1996-07-17 00:00:00.000
12	4	Margaret	Peacock	10259	1996-07-18 00:00:00.000
13	4	Margaret	Peacock	10260	1996-07-19 00:00:00.000
14	4	Margaret	Peacock	10261	1996-07-19 00:00:00.000
15	8	Laura	Callahan	10262	1996-07-22 00:00:00.000
16	9	Anne	Dodsworth	10263	1996-07-23 00:00:00.000
17	6	Michael	Suyama	10264	1996-07-24 00:00:00.000
18	2	Andrew	Fuller	10265	1996-07-25 00:00:00.000
19	3	Janet	Leverling	10266	1996-07-26 00:00:00.000
20	4	Margaret	Peacock	10267	1996-07-29 00:00:00.000
21	8	Laura	Callahan	10268	1996-07-30 00:00:00.000
22	5	Steven	Buchanan	10269	1996-07-31 00:00:00.000

Imena tablica koriste se kao prefiksi u nazivima atributa za identificiranje tablice iz koje želimo odabrati atribut. Iako je to potrebna samo onda kada atribut s istim nazivom postoji u obje tablice, uvijek je dobra ideja uključiti prefiks jer takav pristup čini kod učinkovitim i čitljivim.

ALIASI NAZIVA TABLICA

Korištenje punog naziva tablice kao prefiksa naziva atributa može nepotrebno povećati duljinu SQL upita. Stoga koristimo ALIASE naziva tablica koji skraćuju kod.

Sljedeći primjer, identičan po funkcionalnosti prethodnom, ilustrira korištenje aliasa naziva tablice.

```
-- Izradite upit koji prikazuje sve narudžbe zaposlenika koristeći aliase
SELECT e.EmployeeID, e.FirstName, e.LastName,
       o.OrderID, o.OrderDate
FROM Employees e INNER JOIN Orders o ON
       e.EmployeeID = o.EmployeeID
ORDER BY o.OrderDate;
```

JOIN VIŠE TABLICA

Povezivanje više tablica može biti vrlo složeno, a vrijeme obrade potrajati dugo, ali sama sintaksa je relativno jednostavna.

SINTAKSA

```
SELECT tablica1.atribut, tablica2.atribut, tablica3.atribut
FROM tablica1
  INNER JOIN tablica2
    ON tablica1.atribut = tablica2.atribut
  INNER JOIN tablica3
    ON tablica2.atribut = tablica3.atribut
WHERE uvjeti
```

Dakle, kako bi se tablica dodala u vezu (JOIN) više tablica, mora postojati u FROM izrazu, ili mora već biti uključena u vezu (JOIN) s nekom drugom tablicom u FROM izrazu.

Pogledajmo sljedeći primjer:

```
SELECT tablica1.atribut, tablica2.atribut, tablica3.atribut
FROM tablica1
  INNER JOIN tablica3 ON tablica2.atribut = tablica3.atribut
  INNER JOIN tablica2 ON tablica1.atribut = tablica2.atribut
WHERE uvjeti
```

Ovaj upit nije korektan i neće se izvršiti zbog toga što pokušava povezati tablicu3 s tablicom2 prije nego je tablica2 povezana s tablicom1.

```
/*
Izraditi upit koji prikazuje OrderID, naziv tvrtke koja je ostvarila
narudžbu, te ime i prezime zaposlenika povezanog s narudžbom.
Prikazati samo narudžbe ostvarene nakon 1.1.1998 koje su otpremljene nakon
datuma dostave.
Sortirati prema nazivu tvrtke.
*/

SELECT o.OrderID, c.CompanyName, e.FirstName, e.LastName
FROM Orders o
  JOIN Employees e ON e.EmployeeID = o.EmployeeID
  JOIN Customers c ON c.CustomerID = o.CustomerID
WHERE o.ShippedDate > o.RequiredDate AND o.OrderDate > '1998-01-01'
ORDER BY c.CompanyName;
```

	OrderID	CompanyName	FirstName	LastName
1	10924	Berglunds snabbköp	Janet	Leverling
2	10970	Bólido Comidas preparadas	Anne	Dodsworth
3	10827	Bon app'	Nancy	Davolio
4	10816	Great Lakes Food Market	Margaret	Peacock
5	10960	HILARION-Abastos	Janet	Leverling
6	10927	La corne d'abondance	Margaret	Peacock
7	10828	Rancho grande	Anne	Dodsworth
8	10847	Save-a-lot Markets	Margaret	Peacock

OUTER JOIN (VANJSKI JOIN)

Svi tipovi do sada obrađenih veza tablica (JOIN) bili su tipa INNER JOIN, što znači da su kao rezultat upita bili vraćeni samo oni zapisi koji imaju istovjetne vrijednosti atributa preko kojih se vrši JOIN u obje tablice. Npr. kada se radi INNER JOIN vezi između tablica Employees i Orders, dobiveni su samo zaposlenici koji imaju odgovarajuće narudžbe, te narudžbe koje pripadaju određenim zaposlenicima.

Za usporedbu, pogledajmo još neke primjere INNER JOIN naredbe.

```
/*
  Napravite upit koje prikazuje one kupce koji se nalaze u istim gradovima
  kao i zaposlenici. Prikažite ime zaposlenika.
*/

SELECT DISTINCT c.CustomerId, c.CompanyName, c.ContactName, e.EmployeeID,
e.LastName, e.FirstName, c.City
FROM Customers c INNER JOIN Employees e ON
c.City = e.City
ORDER BY c.CustomerId
```

	CustomerId	CompanyName	ContactName	City
1	AROUT	Around the Horn	Thomas Hardy	London
2	BSBEV	B's Beverages	Victoria Ashworth	London
3	CONSH	Consolidated Holdings	Elizabeth Brown	London
4	EASTC	Eastern Connection	Ann Devon	London
5	NORTS	North/South	Simon Crowther	London
6	SEVES	Seven Seas Imports	Hari Kumar	London
7	TRAIH	Trail's Head Gourmet Provisioners	Helvetius Nagy	Kirkland
8	WHITC	White Clover Markets	Karl Jablonski	Seattle

```
/*
  Napravite upit koje prikazuje broj zaposlenika i kupaca iz svakog grada u
  kojem postoje zaposlenici.
*/

SELECT COUNT(DISTINCT e.EmployeeID) AS numEmployees,
COUNT(DISTINCT c.CustomerID) AS numCompanies,
e.City, c.City
FROM Employees e INNER JOIN Customers c ON
e.City = c.City
GROUP BY e.City, c.City
ORDER BY numEmployees DESC;
```

	numEmployees	numCompanies	City	City
1	4	6	London	London
2	2	1	Seattle	Seattle
3	1	1	Kirkland	Kirkland

LEFT JOIN

LEFT JOIN (također LEFT OUTER JOIN) vraća sve zapise iz prvo navedene tablice u JOIN izrazu čak i ako nema podudaranja s vrijednostima u drugoj tablici.

SINTAKSA

```
SELECT tablica1.atribut, tablica2.atribut
FROM tablica1
LEFT OUTER JOIN tablica2 ON (tablica1.atribut = tablica2.atribut)
WHERE uvjeti
```

Svi redci iz Tablice1 bit će vraćeni bez obzira imaju li postoje li iste vrijednosti odgovarajućeg atributa u Tablici2.

```
/*
    Napravite upit koje prikazuje one kupce koji se nalaze u istim gradovima
    kao i zaposlenici. Prikažite ime zaposlenika.
*/

SELECT DISTINCT c.CustomerId, c.CompanyName, c.ContactName, e.EmployeeID,
e.LastName, e.FirstName,
           c.City as CityCustomer, e.City as CityEmployee
FROM Customers c LEFT JOIN Employees e ON
c.City = e.City
ORDER BY c.CustomerId
```

	CustomerId	CompanyName	ContactName	EmployeeID	LastName	FirstName	CityCustomer	CityEmployee
1	ALFKI	Alfreds Futterkiste	Maria Anders	NULL	NULL	NULL	Berlin	NULL
2	ANATR	Ana Trujillo Emparedados y helados	Ana Trujillo	NULL	NULL	NULL	México D.F.	NULL
3	ANTON	Antonio Moreno Taquería	Antonio Moreno	NULL	NULL	NULL	México D.F.	NULL
4	AROUT	Around the Horn	Thomas Hardy	5	Buchanan	Steven	London	London
5	AROUT	Around the Horn	Thomas Hardy	6	Suyama	Michael	London	London
6	AROUT	Around the Horn	Thomas Hardy	7	King	Robert	London	London
7	AROUT	Around the Horn	Thomas Hardy	9	Dodsworth	Anne	London	London
8	BERGS	Berglunds snabbköp	Christina Berglund	NULL	NULL	NULL	Luleå	NULL
9	BLAUS	Blauer See Delikatessen	Hanna Moos	NULL	NULL	NULL	Mannheim	NULL
10	BLONP	Blondesddsl père et fils	Frédérique Citeaux	NULL	NULL	NULL	Strasbourg	NULL
11	BOLID	Bólido Comidas preparadas	Martín Sommer	NULL	NULL	NULL	Madrid	NULL
12	BONAP	Bon app'	Laurence Lebihan	NULL	NULL	NULL	Marseille	NULL
13	BOTTM	Bottom-Dollar Markets	Elizabeth Lincoln	NULL	NULL	NULL	Tsawassen	NULL
14	BSBEV	B's Beverages	Victoria Ashworth	5	Buchanan	Steven	London	London
15	BSBEV	B's Beverages	Victoria Ashworth	6	Suyama	Michael	London	London
16	BSBEV	B's Beverages	Victoria Ashworth	7	King	Robert	London	London
17	BSBEV	B's Beverages	Victoria Ashworth	9	Dodsworth	Anne	London	London
18	CACTU	Cactus Comidas para llevar	Patricio Simpson	NULL	NULL	NULL	Buenos Aires	NULL
19	CENTC	Centro comercial Moctezuma	Francisco Chang	NULL	NULL	NULL	México D.F.	NULL
20	CHOPS	Chop-suey Chinese	Yang Wang	NULL	NULL	NULL	Bem	NULL
21	COMMI	Comércio Mineiro	Pedro Afonso	NULL	NULL	NULL	Sao Paulo	NULL

Upit vraća sve kupce bez obzira ima li u njihovim gradovima zaposlenika ili ne. Ukoliko u gradu kupca nema zaposlenika vrijednosti koje nedostaju ispisuje se kao NULL.

```
/*
    Napravite upit koje prikazuje broj zaposlenika i kupaca iz svakog grada u
    kojem postoje zaposlenici.
*/

SELECT COUNT(DISTINCT e.EmployeeID) AS numEmployees,
       COUNT(DISTINCT c.CustomerID) AS numCompanies,
       e.City as CityEmployees, c.City as CityCustomers
FROM Employees e LEFT JOIN Customers c ON
    (e.City = c.City)
GROUP BY e.City, c.City
ORDER BY numEmployees DESC;
```

	numEmployees	numCompanies	CityEmployees	CityCustomers
1	4	6	London	London
2	2	1	Seattle	Seattle
3	1	0	Tacoma	NULL
4	1	0	Redmond	NULL
5	1	1	Kirkland	Kirkland

Upit vraća sve gradove u kojima žive zaposlenici bez obzira ima li u njima kupaca ili ne. Ukoliko u gradu nema kupca vrijednosti koje nedostaju ispisuje se kao NULL.

RIGHT JOIN

RIGHT JOIN (također RIGHT OUTER JOIN) vraća sve zapise iz druge navedene tablice u JOIN izrazu čak iako nema podudaranja s vrijednostima u prvoj tablici.

SINTAKSA

```
SELECT tablica1.atribut, tablica2.atribut
FROM tablica1
    RIGHT OUTER JOIN tablica2 ON tablica1.atribut = tablica2.atribut
WHERE uvjeti
```

Svi redci iz Tablice2 bit će vraćeni bez obzira postoje li iste vrijednosti odgovarajućeg atributa u Tablici1.

```
/*
    Napravite upit koje prikazuje one zaposlenike koji se nalaze u istim
    gradovima kao i kupci. Prikažite ime zaposlenika.
*/

SELECT DISTINCT c.CustomerId, c.CompanyName, c.ContactName, e.EmployeeID,
               e.LastName, e.FirstName,
               c.City as CityCustomer, e.City as CityEmployee
```



```
FROM Customers c RIGHT JOIN Employees e ON
c.City = e.City
ORDER BY e.employeeId
```

	CustomerId	CompanyName	ContactName	EmployeeID	LastName	FirstName	CityCustomer	CityEmployee
1	WHITC	White Clover Markets	Karl Jablonski	1	Davolio	Nancy	Seattle	Seattle
2	NULL	NULL	NULL	2	Fuller	Andrew	NULL	Tacoma
3	TRAIH	Trail's Head Gourmet Provisioners	Helvetius Nagy	3	Leverling	Janet	Kirkland	Kirkland
4	NULL	NULL	NULL	4	Peacock	Margaret	NULL	Redmond
5	AROUT	Around the Horn	Thomas Hardy	5	Buchanan	Steven	London	London
6	BSBEV	B's Beverages	Victoria Ashworth	5	Buchanan	Steven	London	London
7	CONSH	Consolidated Holdings	Elizabeth Brown	5	Buchanan	Steven	London	London
8	EASTC	Eastern Connection	Ann Devon	5	Buchanan	Steven	London	London
9	NORTS	North/South	Simon Crowther	5	Buchanan	Steven	London	London
10	SEVES	Seven Seas Imports	Hari Kumar	5	Buchanan	Steven	London	London
11	AROUT	Around the Horn	Thomas Hardy	6	Suyama	Michael	London	London
12	BSBEV	B's Beverages	Victoria Ashworth	6	Suyama	Michael	London	London
13	CONSH	Consolidated Holdings	Elizabeth Brown	6	Suyama	Michael	London	London
14	EASTC	Eastern Connection	Ann Devon	6	Suyama	Michael	London	London
15	NORTS	North/South	Simon Crowther	6	Suyama	Michael	London	London
16	SEVES	Seven Seas Imports	Hari Kumar	6	Suyama	Michael	London	London
17	AROUT	Around the Horn	Thomas Hardy	7	King	Robert	London	London
18	BSBEV	B's Beverages	Victoria Ashworth	7	King	Robert	London	London
19	CONSH	Consolidated Holdings	Elizabeth Brown	7	King	Robert	London	London
20	EASTC	Eastern Connection	Ann Devon	7	King	Robert	London	London
21	NORTS	North/South	Simon Crowther	7	King	Robert	London	London

```
/*
Napravite upit koje prikazuje broj zaposlenika i kupaca iz svakog grada u
kojem postoje kupci.
*/

SELECT COUNT(DISTINCT e.EmployeeID) AS numEmployees,
COUNT(DISTINCT c.CustomerID) AS numCompanies,
e.City as CityEmployees, c.City as CityCustomers
FROM Employees e RIGHT JOIN Customers c ON
(e.City = c.City)
GROUP BY e.City, c.City
ORDER BY numEmployees DESC;
```

Svi redci iz tablice Customers bit će prebrojani bez obzira postoje li odgovarajući gradovi u tablici Employees.

	numEmployees	numCompanies	City	City
1	4	6	London	London
2	2	1	Seattle	Seattle
3	1	1	Kirkland	Kirkland
4	0	1	NULL	Walla Walla
5	0	1	NULL	Warszawa
6	0	1	NULL	Aachen
7	0	1	NULL	Albuquerque
8	0	1	NULL	Anchorage
9	0	1	NULL	Århus
10	0	1	NULL	Barcelona
11	0	1	NULL	Barquisimeto
12	0	1	NULL	Bergamo
13	0	1	NULL	Berlin
14	0	1	NULL	Bern
15	0	1	NULL	Boise

FULL OUTER JOIN

FULL JOIN (također FULL OUTER JOIN) vraća sve retke iz obje tablice uključene u vezu čak i ako ne postoji podudaranje vrijednosti atributa.

SINTAKSA

```
SELECT tablica1.atribut, tablica2.atribut
FROM tablica1
FULL [OUTER] JOIN tablica2 ON tablica1.atribut = tablica2.atribut
WHERE uvjeti
```

Naredbom će biti vraćeni svi redci iz Tablice1 i Tablice2.

```
/*
  Napravite upit koje prikazuje sve zaposlenike i kupce te gradove u kojima
  se nalaze.
*/

SELECT DISTINCT c.CustomerId, c.CompanyName, c.ContactName, e.EmployeeID,
e.LastName, e.FirstName,
       c.City as CityCustomer, e.City as CityEmployee
FROM Customers c FULL JOIN Employees e ON
c.City = e.City
ORDER BY c.CustomerId
```


	CustomerId	CompanyName	ContactName	EmployeeID	LastName	FirstName	CityCustomer	CityEmployee
1	NULL	NULL	NULL	4	Peacock	Margaret	NULL	Redmond
2	NULL	NULL	NULL	2	Fuller	Andrew	NULL	Tacoma
3	ALFKI	Alfreds Futterkiste	Maria Anders	NULL	NULL	NULL	Berlin	NULL
4	ANATR	Ana Trujillo Emparedados y helados	Ana Trujillo	NULL	NULL	NULL	México D.F.	NULL
5	ANTON	Antonio Moreno Taquería	Antonio Moreno	NULL	NULL	NULL	México D.F.	NULL
6	AROUT	Around the Hom	Thomas Hardy	5	Buchanan	Steven	London	London
7	AROUT	Around the Hom	Thomas Hardy	6	Suyama	Michael	London	London
8	AROUT	Around the Hom	Thomas Hardy	7	King	Robert	London	London
9	AROUT	Around the Hom	Thomas Hardy	9	Dodsworth	Anne	London	London
10	BERGS	Berglunds snabbköp	Christina Berglund	NULL	NULL	NULL	Luleå	NULL
11	BLAUS	Blauer See Delikatessen	Hanna Moos	NULL	NULL	NULL	Mannheim	NULL
12	BLONP	Blondesd's père et fils	Frédérique Citeaux	NULL	NULL	NULL	Strasbourg	NULL
13	BOLID	Bólido Comidas preparadas	Martín Sommer	NULL	NULL	NULL	Madrid	NULL
14	BONAP	Bon app'	Laurence Leblan	NULL	NULL	NULL	Marseille	NULL
15	BOTTM	Bottom-Dollar Markets	Elizabeth Lincoln	NULL	NULL	NULL	Tsawassen	NULL
16	BSBEV	B's Beverages	Victoria Ashworth	5	Buchanan	Steven	London	London
17	BSBEV	B's Beverages	Victoria Ashworth	6	Suyama	Michael	London	London
18	BSBEV	B's Beverages	Victoria Ashworth	7	King	Robert	London	London
19	BSBEV	B's Beverages	Victoria Ashworth	9	Dodsworth	Anne	London	London
20	CACTU	Cactus Comidas para llevar	Patricio Simpson	NULL	NULL	NULL	Buenos Aires	NULL
21	CENTC	Centro comercial Moctezuma	Francisco Chang	NULL	NULL	NULL	México D.F.	NULL
22	CHOPS	Chop-suey Chinese	Yang Wang	NULL	NULL	NULL	Bem	NULL
23	COMMI	Comércio Mineiro	Pedro Afonso	NULL	NULL	NULL	Sao Paulo	NULL
24	CONSH	Consolidated Holdings	Elizabeth Brown	5	Buchanan	Steven	London	London
25	CONSH	Consolidated Holdings	Elizabeth Brown	6	Suyama	Michael	London	London

```

/*
Izradite upit koji prikazuje broj zaposlenika i kupaca iz istog grada.
*/

SELECT COUNT(DISTINCT e.EmployeeID) AS numEmployees,
COUNT(DISTINCT c.CustomerID) AS numCompanies,
e.City, c.City
FROM Employees e FULL JOIN Customers c ON
(e.City = c.City)
GROUP BY e.City, c.City
ORDER BY numEmployees DESC;

```

Svi zapisi iz svake tablice su prebrojani bez obzira postoje li odgovarajuće vrijednosti u drugoj tablici.

	numEmployees	numCompanies	City	City
1	4	6	London	London
2	2	1	Seattle	Seattle
3	1	0	Redmond	NULL
4	1	0	Tacoma	NULL
5	1	1	Kirkland	Kirkland
6	0	1	NULL	Walla Walla
7	0	1	NULL	Warszawa
8	0	1	NULL	Aachen
9	0	1	NULL	Albuquerque
10	0	1	NULL	Anchorage
11	0	1	NULL	Århus
12	0	1	NULL	Barcelona
13	0	1	NULL	Barquisimeto
14	0	1	NULL	Bergamo
15	0	1	NULL	Berlin

CROSS JOIN

CROSS JOIN vraća KARTEZIJEV PRODUKT vrijednosti zapisa iz obje tablice, dakle skup zapisa kod kojeg je svaki zapis iz prve tablice povezan sa svakim zapisom iz druge tablice.

Kod naredbe CROSS JOIN ne navode se atributi preko kojih se povezivanje vrši, jer oni ne postoje.

SINTAKSA

```
SELECT tablica1.atribut, tablica2.atribut
FROM tablica1 CROSS JOIN tablica2

/*
Prikažite sve moguće kombinacije dobavljača i proizvođa.
*/

SELECT * FROM Suppliers CROSS Join Products
```

UNION (UNIJA)

UNION se koristi za dohvaćanje zapisa iz više tablica ili više skupova zapisa iz jedne tablice.

Operator unije (UNION) definira relacijsku operaciju unije, tj. povezuje rezultate više upita u jedinstveni skup rezultata.

SINTAKSA:

```
SELECT tablica1.atribut, tablica1.atribut ... FROM tablica1
UNION [ALL]
SELECT tablica2.atribut, tablica2.atribut ... FROM tablica2
.
```

```
ORDER BY sort
```

Pri formiranju unije mora vrijediti:

- upiti nad čijim se rezultati provodi unija moraju imati jednaki broj izlaznih atributa
- atributi se moraju ispisivati istim redoslijedom
- odgovarajući izlazni atributi u upitima koji se povezuju unijom moraju biti istog tipa podataka.

```
/*  
Upit koji daje brojeve telefona svih prijevoznika, kupaca i  
dobavljača.  
*/
```

```
SELECT CompanyName, Phone FROM Shippers  
UNION  
SELECT CompanyName, Phone FROM Customers  
UNION  
SELECT CompanyName, Phone FROM Suppliers  
ORDER BY CompanyName;
```

UNION ALL

UNION operator uklanja duplicirane zapise rezultata obaju upita.

Ukoliko želimo prikazati sve zapise, bez eliminacije duplikata, potrebno je navesti opciju ALL.

EXCEPT (RAZLIKA) I INTERSECT (PRESJEK)

Operatori EXCEPT (razlika) i INTERSECT (presjek) vraćaju jedinstvene (DISTINCT) vrijednosti usporedbom rezultata dvaju upita.

Operator EXCEPT (razlika) vraća sve jedinstvene (DISTINCT) zapise iz lijevog (prvog) upita koji nisu nađeni u desnom (drugom) upitu.

Operator INTERSECT vraća sve jedinstvene (DISTINCT) zapise iz koji se nalaze u oba upita.

Osnovna pravila za kombiniranje rezultata upita korištenjem operatora EXCEPT i INTERSECT:

- Broj i redoslijed atributa mora biti isti u oba upita.
- Tipovi podataka odgovarajućih atributa moraju biti kompatibilni.

```
/*  
Ispišite samo one gradove u kojima postoje i kupci i zaposlenici.  
*/
```

```
SELECT city, country, region FROM employees  
INTERSECT  
SELECT city, country, region FROM customers
```

```
/*
Ispišite samo one gradove u kojima postoje kupci ali ne i zaposlenici.
*/

SELECT city, country, region FROM customers
EXCEPT
SELECT city, country, region FROM employees

/*
Ispišite samo one gradove u kojima postoje zaposlenici ali ne i kupci.
*/

SELECT city, country, region FROM employees
EXCEPT
SELECT city, country, region FROM customers

/* Ispišite one kupce koji kupuju kod zaposlenika EmployeeID = 5
i zaposlenika EmployeeID = 6 */

SELECT CustomerID FROM orders
WHERE EmployeeID = 5
INTERSECT
SELECT CustomerID FROM orders
WHERE EmployeeID = 6
```