

Unutrašnje i spoljašnje spajanje tabela

Unutrašnje spajanje tabela (Inner Join)

Primer:

Prikazati spisak klijenata sa događajima koje se organizuju za njih iz studijskog primera DJs on Demand.

Struktura tabele D_CLIENTS (KLIJENTI):

Table	Column	Data Type	Length	Precision	Scale	Primary Key	
D_CLIENTS	CLIENT_NUMBER	Number	-	5	0	1	
	FIRST_NAME	Varchar2	25	-	-	-	
	LAST_NAME	Varchar2	30	-	-	-	
	PHONE	Number	-	15	0	-	
	EMAIL	Varchar2	50	-	-	-	

Unutrašnje spajanje tabela (Inner Join)

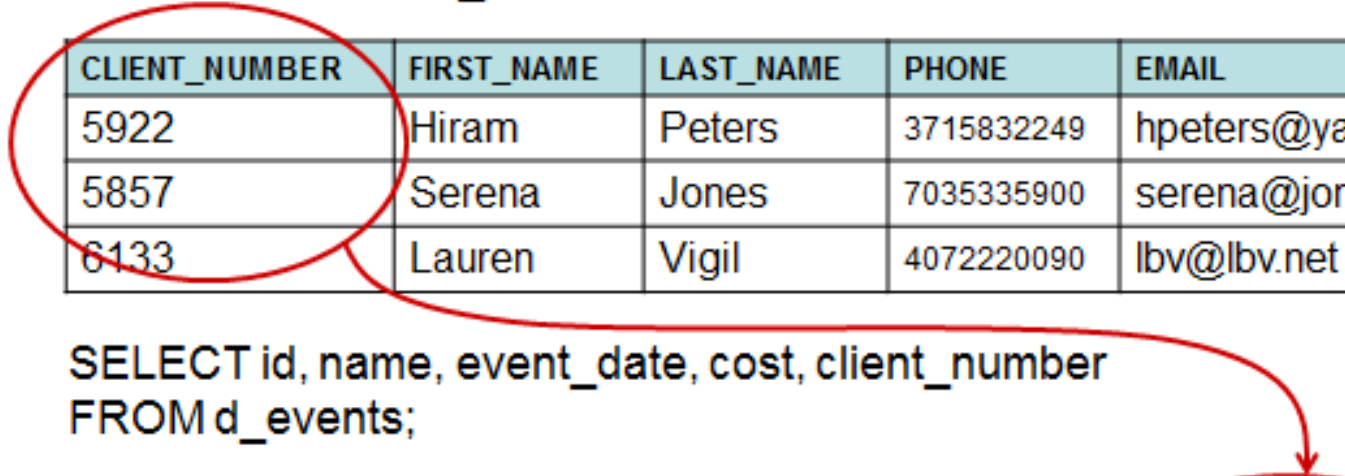
Struktura tabele D_EVENTS (DOGAĐAJI):

Table	Column	Data Type	Length	Precision	Scale	Primary Key	...
D_EVENTS	ID	Number	-	5	0	1	
	NAME	Varchar2	50	-	-	-	
	EVENT_DATE	Date	7	-	-	-	
	DESCRIPTION	Varchar2	50	-	-	-	
	COST	Number	-	8	2	-	
	VENUE_ID	Number	-	5	0	-	
	PACKAGE_CODE	Number	-	5	0	-	
	THEME_CODE	Number	-	10	0	-	
	CLIENT_NUMBER	Number	-	5	0	-	

Unutrašnje spajanje tabela (Inner Join)

Da bi se dobila tražena informacija potrebno je spojiti tabele D_CLIENTS i D_EVENTS. Uslov spajanja je jednakost stranog ključa *client_number* u tabeli D_EVENTS sa primarnim ključem *client_number* u tabeli D_CLIENTS.

```
SELECT * FROM d_clients;
```



CLIENT_NUMBER	FIRST_NAME	LAST_NAME	PHONE	EMAIL
5922	Hiram	Peters	3715832249	hpeters@yahoo.com
5857	Serena	Jones	7035335900	serena@jones.com
6133	Lauren	Vigil	4072220090	lbv@lbv.net

```
SELECT id, name, event_date, cost, client_number  
FROM d_events;
```

ID	NAME	EVENT_DATE	COST	CLIENT_NUMBER
100	Peters Graduation	04-MAY-14	8000	5922
105	Vigil wedding	04-APR-28	10000	6133

Unutrašnje spajanje tabela (Inner Join)

Rešenje:

```
SELECT first_name, last_name, name, event_date  
FROM d_clients, d_events  
WHERE d_clients.client_number=d_events.client_number;
```

ILI

```
SELECT first_name, last_name, name, event_date  
FROM d_clients JOIN d_events  
ON d_clients.client_number=d_events.client_number;
```

Kvalifikovanje kolona

- Kada se pojavi isto ime kolone u obe tabele, onda se ispred imena kolone mora dodati ime tabele kojoj pripada. To se zove **kvalifikovanje kolona**.
- Kombinacijom imena tabele i imena kolone se eliminišu dvosmisleni nazivi kada dve tabele sadrže kolone sa istim imenom. Tako, u ovom primeru, uslov spajanja glasi:

d_clients.client_number=d_events.client_number

Alijasi tabela

- Kvalifikovani nazivi kolona mogu da budu suviše dugi i nepregledni. To može da se reši uvođenjem alijasa tabela.
- Alijasi tabela predstavljaju druge nazive tabela.
- Uvode se u klauzuli FROM, iza originalnog naziva tabele i bez znakova navoda.
- Ako su uvedeni alijasi tabela, onda se oni moraju koristiti u svim kvalifikovanim nazivima kolona u svim klauzulama SELECT-rečenice u kojoj su uvedeni.

Alijasi tabela

Primer SELECT-rečenice sa alijasima tabela:

```
SELECT c.first_name, c.last_name, e.name, e.event_date  
FROM d_clients c, d_events e  
WHERE c.client_number=e.client_number;
```

ILI

```
SELECT c.first_name, c.last_name, e.name, e.event_date  
FROM d_clients c JOIN d_events e  
ON c.client_number=e.client_number;
```

Primenom bilo kojeg od navedena četiri formata SELECT-rečenice (sa ključnom reči JOIN ili bez, sa alijasima tabela ili bez) dobija se isti rezultat:

FIRST_NAME	LAST_NAME	NAME	EVENT_DATE
Hiram	Peters	Peters Graduation	04-MAY-14
Lauren	Vigil	Vigil wedding	04-APR-28

Alijasi tabela

Izlazni format primenjenog SQL upita može se preurediti tako da nazivi kolona u izlaznoj tabeli budu na srpskom jeziku, da datumi budu prikazani u međunarodnom formatu datuma *YYYY-MM-DD* po standardu ISO 8601 i da vraćeni redovi budu sortirani po datumu događaja.

```
SELECT c.first_name||' '||UPPER(c.last_name) AS "Klijent", e.name  
AS "Događaj", TO_CHAR(e.event_date, 'YYYY-MM-DD') AS "Datum"  
FROM d_clients c JOIN d_events e  
ON c.client_number=e.client_number  
ORDER BY "Datum";
```

Klijent	Događaj	Datum
Hiram PETERS	Peters Graduation	2014-05-04
Lauren VIGIL	Vigil wedding	2028-04-04

Alijasi tabela

Komentar:

U izlaznoj tabeli može se uočiti neobičan datum jednog događaja, zapravo, Lauren ili njeni roditelji planiraju tačan datum njenog venčanja mnogo godina unapred. Može se pretpostaviti da je reč o grešci pri unosu podataka. Verovatno je osoba koja je unosila podatke obrnula redosled u podrazumevanom Oraklovom formatu datuma DD-MON-YY, pa je prvo unela dve cifre za godinu, zatim skraćenicu za mesec i na kraju dve cifre za dan u mesecu. Tako je dobijen datum 4. april 2028. godine umesto datuma 28. april 2004. godine.

Ovo pokazuje da je, **pri projektovanju i programiranju baze podataka, neophodno obezbediti da korisniku bude jasno u kom formatu treba da unosi podatke.**

Unutrašnje spajanje tabela (Inner Join)

U razmatranom primeru korišćena su dva formata spajanja tabela:

Format spajanja svojstven Oracle-u:

```
SELECT kolone  
FROM tabela_1, tabela_2  
WHERE uslov_spajanja;
```

Standardni format spajanja po standardu ANSI/ISO SQL 99:

```
SELECT kolone  
FROM tabela_1 JOIN tabela_2  
ON uslov_spajanja;
```

Unutrašnje spajanje tabela (Inner Join)

- Oba formata spajanja tabela daju isti rezultat, koji obuhvata samo one redove iz obe tabele koji se mogu upariti po datom uslovu spajanja. Takvo spajanje se naziva unutrašnje spajanje (Inner Join).
- Inner Join izostavlja one redove iz tabele koji se po uslovu spajanja ne mogu spojiti ni sa jednim redom druge tabele koja učestvuje u spajanju.

Spajanje tabele sa samom sobom (Selfjoin)

- Pri prevođenju konceptualnog modela u relacioni model rekurzivna veza se transformiše u kolonu-strani ključ koji ukazuje na primarni ključ iste tabele.
- Iz takve tabele se mogu dobiti podaci o instancama entiteta povezanih rekurzivnom vezom. Za to je potrebno spojiti tabelu samu sa sobom.
- Spajanje tabele sa samom sobom se naziva Selfjoin.

Spajanje tabele sa samom sobom (Selfjoin)

Ako se koristi Selfjoin, onda će se isti naziv tabele pojaviti dva puta u istoj klauzuli FROM. Zato je kod Selfjoin-a neophodno koristiti dva različita alijasa za istu tabelu.

Na primer, ako je potrebno prikazati spisak svih zaposlenih iz tabele EMPLOYEES sa imenima njihovih rukovodilaca, onda ta tabela ima dvostuku ulogu:

- S jedne strane, ona sadrži podatke o svakom zaposlenom. Za tu ulogu koristiće se alijas, npr. **emp**;
- S druge strane, ona sadrži podatke o rukovodiocima referenciranim vrednostima stranog ključa iz iste tabele. Za tu ulogu koristiće se drugi alijas iste tabele, npr. **mng**.

Spajanje tabele sa samom sobom (Selfjoin)

Primer:

```
SELECT emp.first_name||' '||emp.last_name "Zaposleni",  
       mng.first_name||' '||mng.last_name "Rukovodilac"  
FROM employees emp JOIN employees mng  
ON mng.employee_id = emp.manager_id  
ORDER BY "Zaposleni";
```

Zaposleni	Rukovodilac
Alexander Hunold	Lex De Haan
Bruce Ernst	Alexander Hunold
Curtis Davies	Kevin Mourgous
Diana Lorentz	Alexander Hunold
Eleni Zlotkey	Steven King
Ellen Abel	Eleni Zlotkey
...	...

Spajanje po jednakosti (Equijoin)

Spajanje tabela gde je uslov spajanja jednakost kolona (npr. **tabela_1.kolona_m= tabela_2.kolona_n**) naziva se ekvi-spajanje (spajanje po jednakosti) odnosno Equijoin.

Prirodno spajanje (Natural Join)

Ako kolone koje se izjednačavaju pri spajanju tabela, imaju isti naziv u obe tabele, takvo ekvi-spajanje se naziva prirodno spajanje, odnosno Natural Join. Na primer:

```
d_clients.client_number = d_events.client_number
```

Ako se u klauzuli FROM koristi ključna reč NATURAL JOIN, onda je **klauzula sa uslovom spajanja suvišna** jer sistem sam traži kolone sa istim nazivima u obe tabele i spaja tabele po jednakosti tih kolona.

Prirodno spajanje (Natural Join)

Primer:

```
SELECT first_name||' '||last_name "Klijent", phone "Telefon",  
name "Dogadjaj", event_date "Datum"  
FROM d_clients NATURAL JOIN d_events;
```

Klijent	Telefon	Dogadjaj	Datum
Hiram Peters	3715832249	Peters Graduation	14-MAY-04
Lauren Vigil	4072220090	Vigil wedding	28-APR-04

Prirodno spajanje i klauzula USING

Ako više kolona iz jedne tabele ima iste nazive kao kolone iz druge tabele, onda NATURAL JOIN poredi vrednosti svih parova kolona sa istim nazivom i spaja one redove iz te dve tabele kojima svi parovi kolona sa istim nazivom imaju jednake vrednosti.

Ponekad to nije dobro zato što:

- moguće je da prirodna upita koji se kreira ne zahteva izjednačavanje svih tih kolona;
- može da se desi da kolone sa istim nazivom nemaju isti tip podataka pa se nad njima ne može izvršiti operacija poređenja po jednakosti.

Prirodno spajanje i klauzula USING

Da bi se to izbeglo, koristi se običan JOIN sa klauzulom USING.

```
SELECT kolone  
FROM tabela_1 JOIN tabela_2  
USING (kolona ili kolone);
```

Kolone koje se koriste u klauzuli USING ne treba da imaju kvalifikator (ime tabele ili alias tabele) bilo gde u SQL rečenici.

Prirodno spajanje i klauzula USING

Primer:

```
SELECT first_name||' '||last_name "Klijent", phone "Telefon",  
name "Dogadjaj", event_date "Datum"  
FROM d_clients JOIN d_events  
USING (client_number);
```

Klijent	Telefon	Dogadjaj	Datum
Hiram Peters	3715832249	Peters Graduation	14-MAY-04
Lauren Vigil	4072220090	Vigil wedding	28-APR-04

Prirodno spajanje i klauzula USING

Primer:

Prikazati imena i prezimena zaposlenih, kao i nazive i identifikacione brojeve odeljenja u kojima rade. Koristiti tabele EMPLOYEES i DEPARTMENTS.

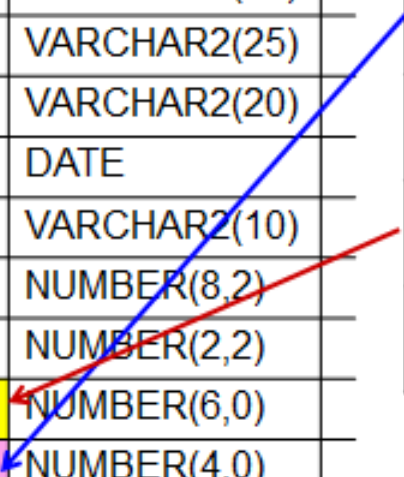
Strukture tabela EMPLOYEES i DEPARTMENTS:

EMPLOYEES

Column Name	Data Type
EMPLOYEE_ID	NUMBER(6,0)
FIRST_NAME	VARCHAR2(20)
LAST_NAME	VARCHAR2(25)
EMAIL	VARCHAR2(25)
PHONE_NUMBER	VARCHAR2(20)
HIRE_DATE	DATE
JOB_ID	VARCHAR2(10)
SALARY	NUMBER(8,2)
COMMISSION_PCT	NUMBER(2,2)
MANAGER_ID	NUMBER(6,0)
DEPARTMENT_ID	NUMBER(4,0)

DEPARTMENTS

Column Name	Data Type
DEPARTMENT_ID	NUMBER(4,0)
DEPARTMENT_NAME	VARCHAR2(30)
MANAGER_ID	NUMBER(6,0)
LOCATION_ID	NUMBER(4,0)



Prirodno spajanje i klauzula USING

```
SELECT last_name, first_name, department_id,  
       department_name  
FROM employees JOIN departments  
USING (department_id);
```

LAST_NAME	FIRST_NAME	DEPARTMENT_ID	DEPARTMENT_NAME
Whalen	Jennifer	10	Administration
Hartstein	Michael	20	Marketing
Fay	Pat	20	Marketing
Davies	Curtis	50	Shipping
...			

Prirodno spajanje i klauzula USING

ILI

```
SELECT last_name||', '||first_name AS "Zaposleni",  
department_id ||' - '|| department_name AS "Organizaciona  
jedinica"  
FROM employees JOIN departments  
USING (department_id);
```

Zaposleni	Organizaciona jedinica
Whalen, Jennifer	10 - Administration
Hartstein, Michael	20 - Marketing
Fay, Pat	20 - Marketing
Davies, Curtis	50 - Shipping
...	

Prirodno spajanje i klauzula USING

Primer:

Prikazati spisak zaposlenih koji rade u računovodstvu. Urediti spisak po abecednom redosledu.

```
SELECT last_name||', '||first_name AS "Zaposleni", department_id  
||' - '|| department_name AS "Organizaciona jedinica"  
FROM employees JOIN departments  
USING (department_id)  
WHERE department_name='Accounting'  
ORDER BY "Zaposleni";
```

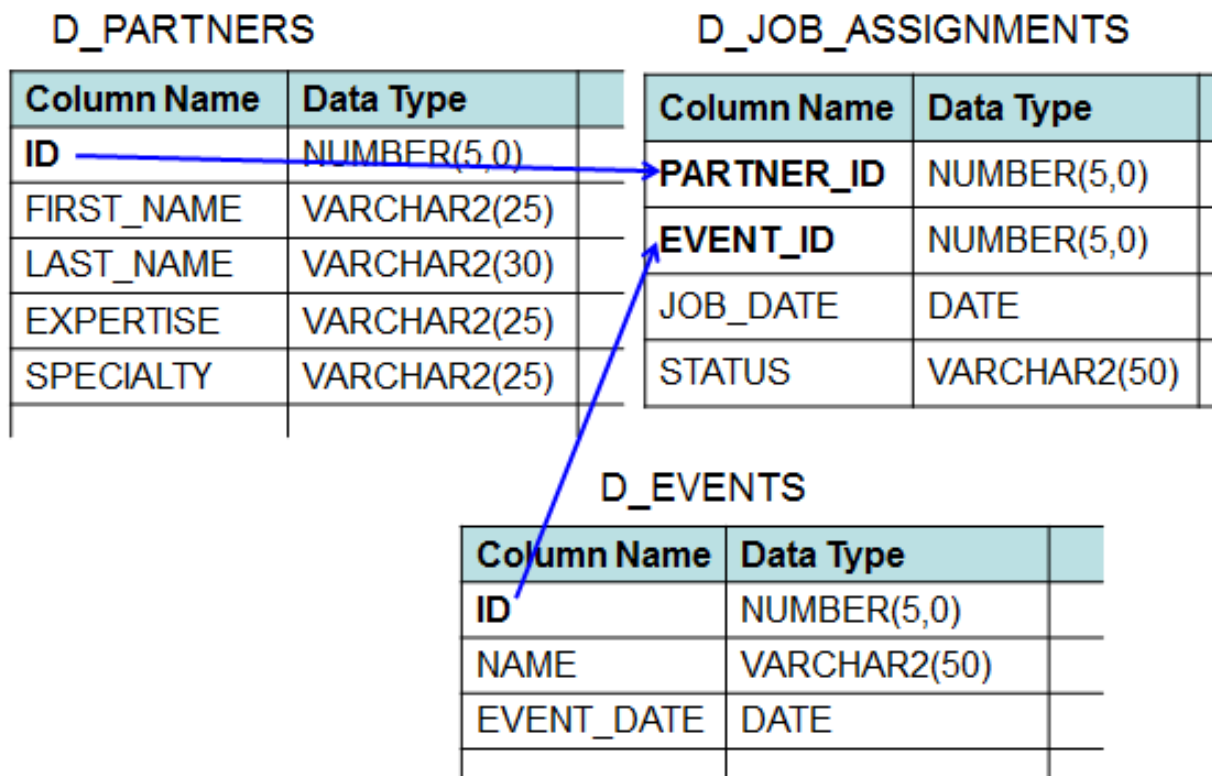
Zaposleni	Organizaciona jedinica
Gietz, William	110 - Accounting
Higgins, Shelley	110 - Accounting

Prirodno spajanje i klauzula USING

Primer:

Prikazati spisak DJ-partnera i događaja na kojima rade.

Strukture tabela D_PARTNERS, D_EVENTS i vezne tabele D_JOB_ASSIGNMENTS:



Prirodno spajanje i klauzula USING

Rešenje:

```
SELECT first_name, last_name, name, event_date  
FROM (d_partners JOIN d_job_assignments ON d_partners.id  
= d_job_assignments.partner_id) JOIN d_events  
ON d_job_assignments.event_id = d_events.id;
```

FIRST_NAME	LAST_NAME	NAME	EVENT_DATE
Jennifer	cho	Vigil wedding	28-APR-04

Non-Equijoin

Ako se kod uslova spajanja koriste operatori nejednakosti kao što su <, > ili BETWEEN, onda se uslov spajanja navodi u klauzuli ON.

Primer:

Prikazati sve zaposlene (iz tabele EMPLOYEES) iz odeljenja 20 koji imaju veću platu od nekog ili nekih radnika iz odeljenja 80.

```
SELECT vp.last_name, vp.department_id, vp.salary, mp.last_name,  
       mp.department_id, mp.salary  
FROM employees vp JOIN employees mp  
ON vp.salary > mp.salary  
WHERE vp.department_id=20 AND mp.department_id=80;
```

Non-Equijoin

Tabela EMPLOYEES spojena je sa samom sobom pri čemu su korišćeni alijasi:

vp (veća plata) → zaposleni sa većom platom,

mp (manja plata) → zaposleni sa manjom platom.

Dobijeni rezultati pokazuju da Hartstein iz odeljenja 20 ima veću platu nego Zlotkey, Abel i Taylor iz odeljenja 80.

LAST_NAME	DEPARTMENT_ID	SALARY	LAST_NAME	DEPARTMENT_ID	SALARY
Hartstein	20	13000	Zlotkey	80	10500
Hartstein	20	13000	Abel	80	11000
Hartstein	20	13000	Taylor	80	8600

Non-Equijoin

Napraviti poređenje dobijenih rezultata sa spiskom zaposlenih iz odeljenja 20 i 80.

```
SELECT last_name, department_id, salary  
FROM employees  
WHERE department_id IN (20, 80)  
ORDER BY department_id;
```

LAST_NAME	DEPARTMENT_ID	SALARY
Hartstein	20	13000
Fay	20	6000
Zlotkey	80	10500
Abel	80	11000
Taylor	80	8600

Spoljašnje spajanje tabela (Outer Join)

Unutrašnje spajanje dve tabele (Inner join) vraća podatke koji zadovoljavaju uslov spajanja. Ponekad je potrebno prikazati redove jedne ili obe tabele koji se, prema datom uslovu spajanja, ne mogu spojiti ni sa jednim redom druge tabele.

Spoljašnje spajanje tabela (Outer join), dozvoljava ovu funkcionalnost u tri moguća oblika:

➤ **LEFT OUTER JOIN (levo spoljašnje spajanje)**

leva_tabela LEFT OUTER JOIN desna_tabela

vraća sve redove leve tabele spojene sa odgovarajućim redovima desne tabele za koje je uslov spajanja zadovoljen i redove iz leve tabele koji nemaju par u desnoj (redovi koji nemaju svoj par spajaju se sa null vrednostima).

Spoljašnje spajanje tabela (Outer Join)

➤ **RIGHT OUTER JOIN** (desno spoljašnje spajanje)

leva_tabela RIGHT OUTER JOIN desna_tabela

vraća sve redove desne tabele spojene sa odgovarajućim redovima leve tabele za koje je uslov spajanja zadovoljen i redove iz desne tabele koji nemaju par u levoj (redovi koji nemaju svoj par spajaju se sa null vrednostima).

➤ **FULL OUTER JOIN** (potpuno spoljašnje spajanje)

leva_tabela FULL OUTER JOIN desna_tabela

vraća sve redove leve tabele spojene sa odgovarajućim redovima desne tabele za koje je uslov spajanja zadovoljen, redove iz leve tabele koji nemaju par u desnoj, kao i redove iz desne tabele koji nemaju par u levoj (redovi koji nemaju svoj par spajaju se sa null vrednostima).

Spoljašnje spajanje tabela (Outer Join)

Primer:

```
SELECT last_name AS radnik, department_name AS odeljenje  
FROM employees LEFT OUTER JOIN departments  
ON employees.department_id=departments.department_id;
```

RADNIK	ODELJENJE
Whalen	Administration
Fay	Marketing
Hartstein	Marketing
Vargas	Shipping
Matos	Shipping
...	...
Higgins	Accounting
Grant	-

Dobijeni rezultati pokazuju da zaposleni Grant nije raspoređen ni u jedno odeljenje.

To ne bi bilo tako vidljivo da je korišćeno unutrašnje spajanje tabela.

Spoljašnje spajanje tabela (Outer Join)

Primer sa funkcijom NVL:

```
SELECT NVL(last_name, 'Nema zaposlenih') "Radnik",  
       NVL(department_name, 'Nema odeljenje') "Odeljenje"  
FROM employees FULL OUTER JOIN departments  
ON employees.department_id=departments.department_id;
```

Radnik	Odeljenje
Whalen	Administration
...	...
Higgins	Accounting
Grant	<u>Nema odeljenje</u>
Nema zaposlenih	Contracting