

Spajanje tabela

Upiti nad dve tabele

Sve do sada, vaše iskustvo u korišćenju SQL-a je bilo ograničeno na pravljenje upita i dobijanje informacija iz jedne tabele u jednom trenutku vremena.

To ne bi bio problem kada bi svi podaci bili smešteni u samo jednoj tabeli. Ali, u toku modelovanja podataka smo razdvajali podatke u više tabela radi normalizacije i minimiziranja redundanse. Razdvajanje podataka u individualne tabele i mogućnost međusobnog povezivanja tabela čine suštinu relacionih baza podataka. SQL koristi spajanje tabela (JOIN) radi upita nad dve ili više različitih tabela.

Tako se informacije dobijene iz različitih tabela kombinuju u jedan izveštaj kakav je potreban korisniku.

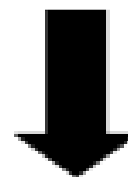
Spajanje tabela

EMPLOYEES

EMPLOYEE_ID	LAST_NAME	DEPT_ID
100	King	90
101	Kochhar	90
...		
202	Fay	20
205	Higgins	110
206	Gietz	110

DEPARTMENTS

DEPARTMENT_ID	DEPT_NAME	LOCATION_ID
10	Administration	1700
20	Marketing	1800
50	Shipping	1500
60	IT	1400
80	Sales	2500
90	Executive	1700
110	Accounting	1700
190	Contracting	1700



EMPLOYEE_ID	DEPT_ID	DEPT_NAME
200	10	Administration
201	20	Marketing
202	20	Marketing
...		
102	90	Executive
205	110	Accounting
206	110	Accounting

Dekartov (kartezijski) proizvod tabela je bezuslovno spajanje dve tabele, kojim se dobijaju sve kombinacije svih redova jedne tabele sa redovima druge tabele. Zato ovakvo spajanje retko kad daje neku korisnu informaciju.

```
SELECT * FROM d_songs, d_types;
```

ID	TITLE	DURATION	ARTIST	TYPE_CODE	CODE	DESCRIPTION
45	Its Finally Over	5 min	The Hobbits	12	1	Jazz
47	Hurrah for Today	3 min	The Jubilant Trio	77	1	Jazz
48	Meet Me At the Altar	6 min	Bobby West	1	1	Jazz
	...					
45	Its Finally Over	5 min	The Hobbits	12	12	Pop
	...					

Da bi se kombinovanjem podataka iz dve tabele dobile korisne informacije, potrebno je zadati uslov spajanja kojim se ograničava kombinovanje redova jedne tabele sa redovima druge tabele.

```
SELECT kolone_koje_nas_interesuju  
FROM table1, table2  
WHERE uslov;
```

Uslov spajanja, ili kombinacija uslova, zavisi od toga kakvu informaciju tražimo. Često, ali ne uvek, taj uslov predstavlja jednakost kolone spoljnog ključa iz jedne tabele sa kolonom primarnog ključa druge tabele koju spoljni ključ referencira.

Na primer, ako nam treba spisak pesama iz naše kolekcije sa informacijom o žanru svake pesme, uslov spajanja će biti jednakost kolone `d_songs.type_code` (strani ključ) sa kolonom `d_types.code` (referencirani primarni ključ)

```
SELECT title AS "Pesma", description AS "Žanr"  
FROM d_songs, d_types  
WHERE d_songs.type_code = d_types.code;
```

Pesma	Žanr
Meet Me At the Altar	Jazz
Its Finally Over	Pop
Im Going to Miss My Teacher	Pop
Hurrah for Today	New Age
Lets Celebrate	New Age
All These Years	Country

DEKARTOV PROIZVOD SPAJANJA

Ako dve tabele u join upitu nemaju join uslov preciziran u WHERE klauzuli ili je join uslov neispravan, Oracle Server vraća Dekartov proizvod navedenih tabela. **To je kombinacija svakog reda iz jedne tabele sa svakim redom iz druge tabele. Dekartov proizvod uvek daje mnogo redova i retko je koristan.**

Na primer, Dekartov proizvod dve tabele, od kojih svaka ima 100 redova, iznosi 10 000 redova! To možda nije ono što ste hteli da dobijete kao rezultat. Da biste izbegli Dekartov proizvod, uvek koristite ispravan join uslov u WHERE klauzuli.

Istu informaciju možemo dobiti pomoću operacije JOIN u klauzuli FROM i klauzulom ON umest WHERE.

```
SELECT title AS "Pesma", description AS "Žanr"  
FROM d_songs JOIN d_types  
ON d_songs.type_code = d_types.code;
```

Pesma	Žanr
Meet Me At the Altar	Jazz
Its Finally Over	Pop
Im Going to Miss My Teacher	Pop
Hurrah for Today	New Age
Lets Celebrate	New Age
All These Years	Country

Ako su nazivi kolona različiti, ne moraju im se dodavati nazivi tabela kojima pripadaju. Isti rezultat kao u prethodnom primeru dobijamo upitom:

```
SELECT title AS "Pesma", description AS "Žanr"  
FROM d_songs JOIN d_types  
ON type_code = code;
```

Pesma	Žanr
Meet Me At the Altar	Jazz
Its Finally Over	Pop
Im Going to Miss My Teacher	Pop
Hurrah for Today	New Age
Lets Celebrate	New Age
All These Years	Country

Međutim, kada se pojavi isto ime kolone u obe tabele, onda se ispred imena kolone mora dodati ime tabele kojoj pripada. To se zove "**qualifying columns**" (kvalifikovanje kolona).

Kombinacijom imena tabele i imena kolone se eliminišu dvosmisleni nazivi kada dve tabele sadrže kolone sa istim imenom. Tako, u narednom primeru, uslov spajanja glasi:

```
employees.department_id = departments.department_id
```

Primenom operacije JOIN na tabele EMPLOYEES i DEPARTMENTS može se dobiti spisak zaposlenih sa nazivima organizacionih jedinica u kojima rade.

```
SELECT
    last_name AS "Radnik",
    department_name AS "Organizaciona jedinica"
FROM employees JOIN departments
ON employees.department_id=departments.department_id;
```

Radnik	Organizaciona jedinica
Whalen, Jennifer	Administration
Hartstein, Michael	Marketing
Fay, Pat	Marketing
Davies, Curtis	Shipping
...	

Kvalifikovani nazivi kolona mogu da budu predugi, a mogu se skratiti uvođenjem alijasa tabela.

```
SELECT
    last_name AS "Radnik",
    d.department_name AS "Organizaciona jedinica"
FROM employees e JOIN departments d
ON e.department_id = d.department_id
WHERE d.department_id < 21;
```

Ako su uvedeni alijasi tabela, oni se moraju koristiti u svim kvalifikovanim nazivima kolona u klauzulama SELECT, ON, WHERE i dr.

Radnik	Organizaciona jedinica
Whalen	Administration
Hartstein	Marketing
Fay	Marketing

A šta ako nam treba spisak radnika sa imenima njihovih šefova? I jedni i drugi su u istoj tabeli EMPLOYEES. To je situacija koju smo na ERD rešili rekurzivnom vezom.

Dakle potrebno je spojiti tabelu EMPLOYEES samu sa sobom (**Self Join**) uz korišćenje aliasa tabele, jednog za EMPLOYEES kao tabelu radnika (rad), a drugog za istu tabelu kao tabelu rukovodilaca (ruk).

```
SELECT rad.last_name AS "Radnik",  
       ruk.last_name AS "Rukovodilac"  
FROM employees rad, employees ruk  
WHERE rad.manager_id = ruk.employee_id;
```

ili

```
SELECT rad.last_name AS "Radnik",  
       ruk.last_name AS "Rukovodilac"  
FROM employees rad JOIN employees ruk  
ON rad.manager_id = ruk.employee_id;
```

Radnik	Rukovodilac
Zlotkey	King
Mourgos	King
Kochhar,	King
Hartstein	King
De Haan	King
Whalen	Kochhar
Higgins	Kochhar
Hunold	De Haan
...	

Provera

```
SELECT employee_id AS "Šifra radnika",  
       last_name AS "Prezime",  
       first_name AS "Ime",  
       manager_id AS "Šifra rukovodioca"  
FROM employees;
```

Šifra radnika	Prezime	Ime	Šifra rukovodioca
100	King	Steven	-
101	Kochhar	Neena	100
102	De Haan	Lex	100
200	Whalen	Jennifer	101
205	Higgins	Shelley	101
...			

Spajanje tabela. SELFJOIN

Šifra radnika	Prezime	Ime	Šifra rukovodioca
100	King	Steven	-
101	Kochhar	Neena	100
102	De Haan	Lex	100
200	Whalen	Jennifer	101
205	Higgins	Shelley	101
...			

Radnik	Rukovodilac
Kochhar, Neena	King, Steven
Higgins, Shelley	Kochhar, Neena
...	