

Upiti nad jednom tabelom

Funkcija SELECT rečenice

- SELECT je ključna reč SQL-a.
- Koristi se za pronalaženje informacija iz baze podataka.
- Ako je iz baze podataka koja sadrži veliki broj informacija potrebno pronaći samo neke od njih, SELECT rečenica može lako da izvuče tražene informacije.

Anatomija SELECT rečenice

U najjednostavnijem obliku, SELECT-rečenica mora da sadrži:

- SELECT-klauzulu, koja specificira koja kolona se prikazuje,
- FROM-klauzulu, koja specificira tabelu kojoj pripada kolona navedena u SELECT-klauzuli.

Anatomija SELECT rečenice

SELECT-rečenicom se izvršavaju operacije:

Projekcija: izbor kolona iz tabele

→ klauzula

Selekcija: izbor redova iz
tabele (restrikcija).

→ klauzula

SELECT

Projection

WHERE

Selection

[illegible]

Prikazivanje svih kolona tabele

Prikazivanje svih kolona podataka u tabeli može se izvršiti korišćenjem ključne reči `SELECT` sa zvezdicom (*).

U prikazanom primeru, * u klauzuli `SELECT` je iskorišćena da prikaže sve kolone iz tabele `D_SONGS`.

```
SELECT * FROM d_songs;
```

ID	TITLE	DURATION	ARTIST	TYPE_CODE
47	Hurrah for Today	2 min	The Jubilant Trio	77
48	Meet Me at the Altar	6 min	Bobby West	1
50	All These Years	10 min	Diana Crooner	88
51	Another One Bites the Dust	7 min	The Shoo Bugs	77
45	Its Finally Over	5 min	The Hobbits	12
46	I'm Going to Miss My Teacher	2 min	Jane Pop	12
49	Lets Celebrate	8 min	The Celebrants	77

Prikazivanje određenih kolona tabele

Za prikaz pojedinih kolona iz tabele potrebno je navesti imena svih kolona koje se prikazuju, razdvojena zapetom, u SELECT klauzuli.

```
SELECT id, title, artist  
FROM d_songs;
```

ID	TITLE	ARTIST
47	Hurrah for Today	The Jubilant Trio
48	Meet Me at the Altar	Bobby West
50	All These Years	Diana Crooner
51	Another One Bites the Dust	The Shoo Bugs
45	Its Finally Over	The Hobbits
46	I'm Going to Miss My Teacher	Jane Pop
49	Lets Celebrate	The Celebrants

Sintaksa osnovne SQL rečenice

SELECT ime_kolone_1, ime_kolone_2, ...

izaberi kolone sa imenom ime_kolone_1,...

FROM ime_tabele

iz tabele sa imenom...

WHERE uslov

gde je ispunjen uslov...

ORDER BY ime_kolone

poređano po vrednostima kolone sa imenom...

Sintaksa osnovne SQL rečenice - primer

SELECT id, salary

izaberi kolone: id,salary

FROM employees

iz tabele sa imenom employees

WHERE salary>10000

gde je salary>10000

ORDER BY salary

poređaj po salary

Ključna reč DISTINCT

SELECT rečenica koja se koristi za projekciju određenih kolona iz neke tabele, prikazuje sve vrednosti iz tih kolona, pri čemu može da se desi da se neke vrednosti ponavljaju.

Na primer, rečenica

```
SELECT year  
FROM d_cds;
```

daje informaciju o godinama iz kojih potiču CD-ovi u CD-teci.

Dva CD-a su iz 2000. godine pa zato postoje dva reda sa tom vrednošću.

YEAR
1997
2000
2002
1999
2000
2001
1998
2004

Kompletan sadržaj tabele D_CDS

```
SELECT * FROM d_cds;
```

CD_NUMBER	TITLE	PRODUCER	YEAR
90	The Celebrants Live in Concert	Old Town Records	1997
91	Party Music for All Occasions	The Music Man	2000
92	Back to the Shire	Middle Earth Records	2002
93	Songs from My Childhood	Old Town Records	1999
94	Carpe Diem	R & B Inc.	2000
95	Here Comes the Bride	The Music Man	2001
96	Graduation Songbook	Tunes Are Us	1998
98	Whirled Peas	Old Town Records	2004

Ključna reč DISTINCT

Ako je potrebno prikazati samo različite vredosti, a ne i koliko puta se pojavljuju, koristiće se ključna reč DISTINCT.

```
SELECT DISTINCT year  
FROM d_cds;
```

Sada je prikazan jedan red manje i u svim redovima su različite vrednosti.

YEAR
1997
2000
2001
1999
2002
2004
1998

Ključna reč DISTINCT

Primeri:

Prikazati iznose plata koje se isplaćuju u departmanu 90?

```
SELECT salary  
FROM employees  
WHERE department_id =90;
```

SALARY
24000
17000
17000

Prikazati različite iznose plata koje primaju zaposleni u departmanu 90?

```
SELECT DISTINCT salary  
FROM employees  
WHERE department_id =90;
```

SALARY
24000
17000

Aritmetički operatori
Rad sa null-vrednostima
Alijasi kolona

Aritmetički operatori

Primer primene aritmetičkih operatora:

```
SELECT last_name, salary, salary + 300  
FROM employees;
```

Dati primer pokazuje upotrebu aritmetičke operacije sabiranja da bi se izračunala plata povećana za 300\$ svim zaposlenima, prikazana kao nova kolona SALARY + 300.

LAST NAME	SALARY	SALARY + 300
King	24000	24300
Kochhar	17000	17300
De Haan	17000	17300
Hunold	9000	9300
Ernst	6000	6300
Hartstein	13000	13300
Fay	6000	6300
Higgins	12000	12300
Gletz	8300	8600

Blanko znak (razmak) pre i posle aritmetičkog operatora ne utiču na ishod prikaza.

Aritmetički operatori

Prioritet aritmetičkih operacija:

1. ()
2. *, /
3. +, -

Primeri primene aritmetičkih operatora:

Operator Precedence

LAST_NAME	SALARY	12 * SALARY + 100
Hartstein	13000	156100
Fay	6000	72100
Higgins	12000	144100

Using Parentheses

LAST_NAME	SALARY	12 * (SALARY+100)
Hartstein	13000	157200
Fay	6000	73200
Higgins	12000	145200

Null vrednosti

- Null je vrednost koja je nedodeljena, nepoznata ili neupotrebljiva.
- Null nije isto što i nula ili blanko znak. U SQL-u nula je broj, blanko je znak, a null znači nepostojanje bilo kakve vrednosti.
- Ponekad, vrednost koju treba smestiti u neku ćeliju nije poznata. Za predstavljanje takvih podataka koristi se NULL. Dakle, NULL znači da nema vrednosti.

Null vrednosti

Ako je vrednost bilo koje kolone u aritmetičkom izrazu null, rezultat je null (nepoznat).

Primer:

Vrednosti u koloni *Nova plata* izračunate su po formuli:

$$\text{Nova plata} = \text{Plata} * (1 + \text{Povećanje})$$

Prezime	Posao	Plata	Povećanje	Nova plata
King	AD-PRES	24000	2%	24480
Kochhar	AD-VP	17000		
Zlotkey	SA-MAN	10500	0%	10500
Abel	SA-REP	8600	5%	9030

Kochhar i Zlotkey nisu imali povećanje pa bi trebalo da ostanu na istoj plati, ali je Kochhar ostao bez plate. Zašto?

Null vrednosti

Ako se pokuša deljenje sa null, rezultat će biti null. Ali, ako se pokuša deljenje sa nulom, javiće se greška!

Primer:

Prosečan budžet izračunava se po formuli:

$$\text{Prosek} = \text{Budžet} / \text{Broj zaposlenih}$$

Odeljenje	Budžet	Broj zaposlenih	Prosek
10	360000	20	18000
60	250000	0	Error!
90	140000		
120	400000	25	16000

Alijasi

- Alijas je način preimenovanja imena kolona na izlazu.
- Bez alijasa, kolone prikazane kao rezultat SQL-rečenice imaju ista imena kao kolone u tabeli ili ime koje pokazuje aritmetičku operaciju, kao $12*(SALARY + 100)$.
- Nekada je poželjno da na izlazu bude prikazano ime kolone koje je lakše razumeti, ime koje je više "friendly". Alijasi kolona dozvoljavaju preimenovanje kolone na izlazu.

Alijasi

Postoji nekoliko pravila kod korišćenja alijasa kolona.

Alijas kolone:

- menja naslov kolone,
- koristan je kod proračuna,
- navodi se odmah posle imena kolone,
- može imati opcionalnu ključnu reč AS između imena kolone i alijasa,
- zahteva znake navoda ako alias sadrži blanko-znak, specijalne znake ili je osetljiv na velika i mala slova.

Alijasi

Primeri upotrebe alijasa kolona:

```
SELECT last_name AS name, commission_pct AS comm  
FROM employees;
```

NAME	COMM
King	
Kochhar	
De Haan	

```
SELECT last_name "NAME", salary*12 AS "Annual Salary"  
FROM employees;
```

NAME	Annual Salary
King	288000
Kochhar	204000
De Haan	204000

Konkatenacija i alijasi kolona

U sledećem primeru je oznaka `||` upotrebljena kako bi se napravio razmak između imena i prezimena.

```
SELECT first_name || ' ' || last_name AS "DJ on Demand  
Clients"  
FROM d_clients;
```

Konkatenacija i alijasi kolona

Uporediti razlike u izlaznom tekstu u upitu koji koristi konkatenaciju i upitu koji je ne koristi.

```
SELECT first_name || ' ' || last_name  
AS "DJ on Demand Clients"  
FROM d_clients;
```

DJs on Demand Clients
Hiram Peters
Serena Jones
Lauren Vigil

```
SELECT first_name, last_name AS  
"DJ on Demand Clients"  
FROM d_clients;
```

FIRST_NAME	DJs on Demand Clients
Hiram	Peters
Serena	Jones
Lauren	Vigil

Restriktivan izbor redova

Operatori poredjenja

Operatori poređenja (komparacije)

Operatori poređenja se koriste u WHERE klauzuli, odnosno u uslovu restrikcije. Neki operatori porede datu kolonu ili izraz sa jednom vrednošću (jednoredni operatori), a drugi porede sa skupom ili intervalom vrednosti (višeredni operatori).

Jednoredni operatori

=

<=

>=

!= , <>

IS

NOT

LIKE

Višeredni operatori

IN

BETWEEN ... AND

ALL

ANY

Restriktivan izbor redova - WHERE klauzula

Kada se zahtevaju podaci iz baze podataka, možda nisu potrebni svi redovi iz tabele pa je neophodno ograničiti redove podataka koji će biti prikazani. Ovo se postiže WHERE klauzulom. Ona sadrži uslov koji mora da se ispuni i navodi se odmah posle FROM klauzule u SQL rečenici.

Alijasi se ne mogu upotrebljavati u WHERE klauzuli!

Sintaksa WHERE klauzule:

WHERE ime_kolone uslov_poređenja ime_kolone | konstanta |
lista_vrednosti

Restriktivan izbor redova - WHERE klauzula

Primeri:

```
SELECT id, last_name, manager_id  
FROM d_partners  
WHERE manager_id = 33;
```

```
SELECT employee_id, last_name, job_id, department_id  
FROM employees  
WHERE department_id = 90 ;
```

Alijas se ne može koristiti u WHERE-klauzuli!

Restriktivna selekcija redova

Nizovi karaktera i datumi u WHERE klauzuli moraju da budu navedeni pod jednostrukim navodnicima ' '.

Brojevi ne bi trebalo da budu navedeni pod jednostrukim navodnicima.

U datom primeru, WHERE klauzula sadrži kao konstantu niz karaktera koji je stavljen pod jednostruke navodnike.

```
SELECT first_name, last_name  
FROM d_clients  
WHERE last_name = 'Jones';
```

Restriktivna selekcija redova

Šta će se desiti ako se WHERE klauzula napiše kao:

```
WHERE last_name = 'jones';
```

Svi pretraživači slovnih podataka su **case-senzitivni**. Pošto su u tabeli D_CLIENTS sva prezimena napisana velikim početnim slovom, ni jedan red neće biti prikazan.

Upotreba ključnih reči UPPER, LOWER i INITCAP omogućava da se izbegnu greške zbog velikih i malih slova.

UPPER – konvertuje sva slova u velika

LOWER – konvertuje sva slova u mala

INITCAP – konvertuje početna slova u velika

Restriktivna selekcija redova

Primer:

TITLES

ID	TITLE	YEAR
a216	Party Music for All Occasions	2000
a300	party music for all occasions	2001
b01	PARTY MUSIC FOR ALL OCCASIONS	2005
b111	Carpe Diem	1999
abc	Carpe diem	2002

Šta će vratiti sledeći upiti:

1. SELECT id, year
FROM titles
WHERE title = 'party music for
all occasions';

2. SELECT id, year
FROM titles
WHERE LOWER(title) = 'party
music for all occasions';

3. SELECT id, year
FROM titles
WHERE INITCAP(title) =
'Carpe Diem';

4. SELECT id, year
FROM titles
WHERE INITCAP(title) =
'Carpe diem';

Restriktivna selekcija redova

Rešenja:

1. title = 'party music for all occasions'

ID	YEAR
a300	2001

2. LOWER(title) = 'party music for all occasions'

ID	YEAR
a216	2000
a300	2001
b01	2005

3. INITCAP(title) = 'Carpe Diem'

ID	YEAR
b111	1999
abc	2002

4. INITCAP(title) = 'Carpe diem'

ID	YEAR
----	------

Operator poređenja BETWEEN...AND

- Operator BETWEEN...AND se koristi za selekciju i prikaz redova na osnovu opsega vrednosti. Kada se operator BETWEEN...AND upotrebi u WHERE klauzuli, na izlazu će biti prikazane vrednosti između odgovarajućih donjih i gornjih granica, **uključujući i vrednosti na tim granicama**.
- Za vrednosti određene BETWEEN...AND operatorom se kaže da su inkluzivne zato što vraćene vrednosti uključuju i vrednost na donjoj granici i vrednost na gornjoj granici.
- Vrednost na donjoj granici mora biti prva navedena u WHERE-klauzuli.

Operator poredenja BETWEEN...AND

Primer:

```
SELECT title, year  
FROM d_cds  
WHERE year BETWEEN '1999' AND '2001';
```

TITLE	YEAR
Party Music for All Occasions	2000
Songs from My Childhood	1999
Carpe Diem	2000
Here Comes the Bride	2001

Operator poređenja BETWEEN...AND

Rezultat dobijen primenom operatora BETWEEN...AND može se dobiti i na drugi način. Na primer, izraz **BETWEEN 2500 AND 3500** ima isto značenje kao izraz:

WHERE salary >= 2500 AND salary <= 3500

Ne postoje neke posebne razlike u performansama kada se koristi BETWEEN...AND u odnosu na druge operatore komparacije. Operator BETWEEN...AND se koristi zbog jednostavnosti u čitanju koda i dobijanju rezultata iz baze podataka.

Operator poređenja IN

- Operator poređenja IN se koristi da bi se proverilo da li neka vrednost pripada određenoj grupi vrednosti.
- Na primer, operator IN može biti upotrebljen za identifikaciju studenata čiji su identifikacioni brojevi 2349, 7354 ili 4333, ili identifikaciju ljudi čiji su internacionalni pozivni brojevi telefona 1735, 82 ili 10.

Operator poređenja IN

Dati primer prikazuje naslove i tipove kodova pesama kod kojih je tip koda 77 ili 12.

```
SELECT title, type_code  
FROM d_songs  
WHERE type_code IN (77, 12);
```

TITLE	TYPE_CODE
Hurrah for Today	77
Another one bites the dust	77
Its Finally Over	12
I'm Going to Miss My Teacher	12
Let's Celebrate	12

Operator poređenja IN

Rezultat dobijen primenom operatora IN može se dobiti i pomoću grupe OR uslova.

Na primer, izraz **WHERE type_code IN (77, 12)** ima isto značenje kao izraz:

WHERE type_code = 77 OR type_code = 12

Kao i kod operatora BETWEEN...AND, tako i operator IN može biti napisan na bilo koji od ova dva načina.

BETWEEN...AND *ili* IN

Po čemu se razlikuju rezultati datih upita:

```
SELECT title, year  
FROM d_cds  
WHERE year BETWEEN 1999 AND 2001;
```

```
SELECT title, year  
FROM d_cds  
WHERE year IN (1999, 2001);
```

Operatori poređenja ALL i ANY

Operatori ALL i ANY se koriste na sledeći način:

kolona/izraz jednoredni_operator ALL (niz vrednosti)

kolona/izraz jednoredni_operator ANY (niz vrednosti)

- Uslov sa **ALL** je ispunjen ako jednoredni operator vraća vrednost *True* za sve vrednosti iz datog niza vrednosti.
- Uslov sa **ANY** je ispunjen ako jednoredni operator vraća vrednost *True* za bar jednu vrednosti iz datog niza vrednosti.

Operatori poredjenja ALL i ANY

Primer:

```
SELECT department_id, last_name, salary  
FROM employees  
WHERE salary > ALL (16000, 18000, 20000)  
ORDER BY department_id;
```

DEPARTMENT_ID	LAST_NAME	SALARY
90	King	24000

Operatori poređenja ALL i ANY

Primer:

```
SELECT department_id, last_name, salary  
FROM employees  
WHERE salary > ANY (16000, 18000, 20000)  
ORDER BY department_id;
```

DEPARTMENT_ID	LAST_NAME	SALARY
90	King	24000
90	De Haan	17000
90	Kochhar	17000

Operatori IS NULL, IS NOT NULL

- Testiranje nedostupnog, nedodeljenog ili nepoznatog može da bude veoma korisno.
- Na primer, možda je potrebno znati sve datume u junu koji trenutno nemaju zakazane koncerte. Ili, možda je poželjno znati koji klijenti nemaju zapisane brojeve telefona u bazi podataka nekog preduzeća.
- Operator IS NULL testira nedostupne, nedodeljene ili nepoznate podatke.
- Nasuprot njemu, operator IS NOT NULL testira podatke koji su prisutni u bazi podataka.

Operatori IS NULL, IS NOT NULL

Primer:

U datom primeru, rezultat upita će biti sva prezimena i id menadžera onih zaposlenih koji nemaju menadžera.

```
SELECT last_name, manager_id  
FROM employees  
WHERE manager_id IS NULL;
```

Operator LIKE

Često se dešava da je potrebno pronaći nešto u bazi podataka, a da nije poznato dovoljno detalja o tome što se traži. Na primer, poznato je da u kompaniji postoji neko čije ime počinje sa El, pri tome nije poznato puno ime, a treba pronaći gde radi ta osoba.

U takvim situacijama koristi se operator LIKE sa sledećim džoker-znacima:

- % – zamenjuje 0, 1 ili više slova u nizu, uključujući i razmak
- _ (donja crta) – zamenjuje jedno slovo ili razmak
- \ – poništava značenje džoker-znaka, tako da se tretira kao obično slovo.

Operator LIKE

Primer:

```
SELECT first_name, last_name, department_id  
FROM employees  
WHERE first_name LIKE 'E%';
```

FIRST_NAME	LAST_NAME	DEPARTMENT_ID
Eleni	Zlotkey	80
Ellen	Abel	80

Operator LIKE

Primer:

Prikazati gde radi osoba koja u prezimenu ima slovo *e* na drugom mestu i još negde iza njega ima slovo *h*. Nije poznato da li je reč o velikim ili malim slovima.

```
SELECT first_name, last_name, department_id  
FROM employees  
WHERE LOWER(last_name) LIKE '_e%h%';
```

FIRST_NAME	LAST_NAME	DEPARTMENT_ID
Lex	De Haan	90