

SQL funkcije

Funkcija	Opis	Primer
ABS(n)	Absolutna vrednost broja n	select abs(-55) from dual;
CEIL(n)	Prvi celi broj veći od n	select ceil(17.8) from dual;
COS(n)	Trigonometrijska funkcija kosinus	select cos(3.14159/2) from dual;
EXP(n)	Eksponencijalna funkcija	select exp(-2) from dual;
FLOOR(n)	Prvi celi broj manji ili jednak n	select floor(17.8) from dual;
LN(n)	Prirodni logaritam od n	select ln(100) from dual;
LOG(m,n)	Logaritam sa osnovom m od broja n	select log(10,100) from dual;
MOD(m,n)	Ostatak deljenja broja m sa n	select mod(10,3) from dual;
POWER(m,n)	m na n-ti stepen	select power(4,3) from dual;
ROUND(m,n)	Zaokružuje m na n decimalnih mesta	select round(12.45678,2) from dual;
SIGN(n)	Znak broja n	select sign(-45) from dual;
SIN(n)	Trigonometrijska funkcija sinus	select sin(3.14159/2) from dual;
SQRT(n)	Kvadratni koren broja n	select sqrt(25) from dual;
TAN(n)	Trigonometrijska funkcija tangens	select tan(3.14159/4) from dual;
TRUNC(m,n)	Vraća vrednost broja m "odsečenu" na n decimala	select trunc(134,-2) from dual;

Primer 1.

```
SQL> select proizvod_id, round(min_cena,0)
      from cena
      where kataloska_cena>5
```

Izlaz:

```
PROIZVOD_ID ROUND(MIN_CENA,0)
-----
100890      46
100890      41
104352      38
104352      41
```

Primer 2.

```
SQL> select proizvod_id,
      5*exp(1+cos(min_cena/3)) Funkcija
      from cena
      where kataloska_cena>50;
```

Izlaz:

```
PROIZVOD_ID  FUNKCIJA
-----
      100890   5.146974
      100890  24.63969
      104352  36.924397
      104352  22.672254
```

Grupne funkcije

Funkcija	Opis	Primer
AVG (n)	Izračunava srednju vrednost od n	<code>select avg(min_cena) from cena</code>
COUNT(* / n)	Izračunava broj redova u rezultatu upita	<code>select count(*) from kupac;</code>
MAX(izraz)	Maksimalna vrednost brojeva definisanih izrazom	<code>select max(min_cena) from cena;</code>
MIN(izraz)	Minimalna vrednost brojeva definisanih izrazom	<code>select min(min_cena) from cena;</code>
STDDEV(x)	Standardna devijacija od x	<code>select stddev(min_cena) from cena;</code>
SUM(n)	Zbir	<code>select sum(kolicina) from artikal;</code>
VARIANCE(x)	Varijansa od x	<code>select variance(min_cena) from price;</code>

Primer 1.

```
SQL> select count(*) BROJ,
        sum(total) SUMA,
        avg(total) AVG,
        max(total) MAX,
        min(total) MIN
        from artikal;
```

Izlaz:

```
      BROJ      SUMA      AVG      MAX      MIN
-----
      271 237236.75 875.41236      29000      2.4
```

SORTIRANJE

Često se javlja potreba da se rezultati upita uredi po nekom kriterijumu. Upitni blok SELECT FROM daje kao rezultat listu redova koji su uređeni onim redom kojim su i uneti. Klauzula ORDER BY omogućuje uređenje podataka po kolonama koje su navedene.

```
SQL> select kupac_id, ime, kredit_limit
        from kupac
        where kredit_limit > 8000
        order by kredit_limit;
```

Izlaz:

KUPAC_ID	IME	KREDIT_LIMIT
101	TKB SPORT SHOP	10000
104	EVERY MOUNTAIN	10000
107	WOMENS SPORTS	10000
201	STADIUM SPORTS	10000
221	WHEELS AND DEALS	10000
203	REBOUND SPORTS	10000
204	THE POWER FORWARD	12000

7 rows selected.

Sortiranje u ovom primeru izvršeno je po opadajućem redosledu. Definisane redosleda sortiranja vrši se dodavanjem ključnih reči ASC (po rastućem redosledu - retko se koristi jer je podrazumevano sortiranje) i DESC (po opadajućem redosledu)

```
SQL> select kupac_id, ime, kredit_limit
      from kupac
      where kredit_limit>8000
      order by kredit_limit desc;
```

KUPAC_ID	IME	KREDIT_LIMIT
204	THE POWER FORWARD	12000
101	TKB SPORT SHOP	10000
104	EVERY MOUNTAIN	10000
107	WOMENS SPORTS	10000
221	WHEELS AND DEALS	10000
201	STADIUM SPORTS	10000
203	REBOUND SPORTS	10000

Kolone po kojima se sortira na moraju da budu uvršćene u SELECT klauzuli. Takođe, možete zadati redosled sortiranja za svaku kolonu nezavisno.

```
SQL> select kupac_id, ime, kredit_limit
      from kupac
      where kredit_limit>8000
      order by kredit_limit desc, kupac_id;
```

KUPAC_ID	IME	KREDIT_LIMIT
204	THE POWER FORWARD	12000
101	TKB SPORT SHOP	10000
104	EVERY MOUNTAIN	10000
107	WOMENS SPORTS	10000
201	STADIUM SPORTS	10000
203	REBOUND SPORTS	10000
221	WHEELS AND DEALS	10000

GROUP BY

Posmatrajmo sledeći primer:

```
SQL> select sum(iznos) from prodaja;
```

```
SUM( IZNOS )
-----
 237236.75
```

Kao rezultat, dobili smo ukupnu sumu. U pojedinim slučajevima potrebno je izvršiti grupne funkcije po pojedinim stavkama tj. interesuju nas parcijalne sume po pojedinim stavkama. SQL to omogućuje klauzulom GROUP BY iza koje stoji ime kolone po kojoj se vrši grupna funkcija:

```
SQL> select kupac, sum(iznos) total from prodaja
      where prodavac_id = 7844
      group by kupac
      order by total desc;
```

Izlaz:

KUPAC	TOTAL
K + T SPORTS	46370
NORTH WOODS HEALTH AND FITNESS SUPPLY CENTER	6400
JOCKSPORTS	5285.9

Primer 1.

```
SQL> select prodavac_id, sum(iznos) total,
      count(*) broj
      from prodaja
      group by prodavac_id
      order by total desc;
```

Izlaz:

PRODAVAC_ID	TOTAL	BROJ
7844	58055.9	20
7560	36191.2	27
7557	33655.65	36
7789	32274.2	12
7654	27348	4
7820	20796.8	26
7555	11060	11
7521	9984.2	16
7499	7870.8	11

HAVING

Posmatrajmo sledeći primer:

```
SQL> select prodavac_id, iznos from prodaja;
```

PRODAVAC_ID	IZNOS
7499	846.8
7499	24
7844	560
7844	4800
7844	440
7844	600

163 rows selected.

Izračunajmo sumu iznosa (IZNOS) po svakoj šifri prodavca (PRODAVAC_ID):

```
SQL> select prodavac_id, sum(iznos) total from prodaja
      group by prodavac_id
      order by total desc;
```

PRODAVAC_ID	TOTAL
7844	58055.9
7560	36191.2
7557	33655.65
7789	32274.2
7654	27348
7820	20796.8
7555	11060
7521	9984.2
7499	7870.8

9 rows selected.

Ako bi smo sada želeli da dobijemo sifre prodavaca koji su ostvarili prodaju veću od 30000 moramo da koristimo klauzulu HAVING.

```
SQL> select prodavac_id, sum(iznos) total from prodaja
      group by prodavac_id
      having sum(iznos) > 30000
      order by total desc;
```

Izlaz:

PRODAVAC_ID	TOTAL
7844	58055.9
7560	36191.2
7557	33655.65
7789	32274.2