Prijava na bazu podataka - SQLTools

Hostname: **???**
Username : **???**
Password: **???**
Host string: **etflab**

SQL funkcije

SQL funkcije omogućavaju veću manipulacija sa podacima, kao što je određivanje minimalne i/ili maksimalne, i/ili sume vrijednosti kolone. S druge strane, manipulacija s stringovnim vrijednostima je jednostavnije i konvertovanje svih karaktera nekod stringa u mala slova ili velika slova, jednostavnije manipulisanje sa podstringovnim vrijednostima i slično. Funkcije su jako korisne osobine SQL, a prema mogućnostima koje nude mogu se općenito podijeliti na:

- agregatne funkcije
- funcije za datum i vrijeme
- aritmetičke funkcije
- funkcije za konverziju
- funkcije za rad sa karakterima
- mješovite funkcije.

Ove funkcije znatno povećavaju mogućnosti manipulacije s informacijama koje se dobivaju putem osnovnih SQL funkcija.

Funkcije za datum i vrijeme

Funkcije za datum i vrijeme su vjerovatno jedne od funkcija koje se najviše koriste u upitima za dohvatanje korisnih informacija iz sistema baza podataka. Iz tih razloga datum i vrijeme od strane mnogih baza, pa tako i oracle baze podataka ima popriličan broj pripremljenih funkcija za rad s ovim tipom podataka.

Funkcija SYSDATE
Ova funkcija vraća trenuti datum i vrijeme.

```
SQL> SELECT sysdate
     FROM dual;
```

Funkcija MONTHS_BETWEEN
Funkcija *MONTHS_BETWEEN* vraća broj mjeseci između dva datuma. Tako na primjer, ako je potrebno napisati upit koji će vratiti broj mjeseci zaposlenja svakog zaposelnog iz odjela 10, 30 i 90 upit bi izgledao kako slijedi:

```
SQL> SELECT last_name,
            first_name,
            months_between(hire_date, sysdate) "Broj mjeseci zaposlenja",
     FROM employees
     WHERE department_id IN (10,30,90);
```

Funkcija ADD_MONTHS
Ova funkcija dodaje određeni broj mjeseci za izabrani datum. Na primjer, neka je potrebno dodati za predhodni upit još i kolonu koja datumu zaposlenja dodaje još dodatnih 6 mjeseci.

```
SQL> SELECT last_name,
            first_name,
            months_between(hire_date, sysdate) "Broj mjeseci zaposlenja",
            add_months(hire_date,6) "Datum zaposlenja + 6 mjeseci"
     FROM employees
     WHERE department_id IN (10,30,90);
```

Funkcija LAST_DAY
Funkcija *LAST_DAY* vraća kao rezultat posljednji dan u mjesecu za zadati datum. Tako na primjer, posljednji dan za tekući datum bio bi:

```
SQL> SELECT last_day(sysdate) "Zadnji dan tekućeg mjeseca"
     FROM dual;
```

Funkcija NEXT_DAY
Funkcija *NEXT_DAY* kao rezultat vraća sljedeći dan u sedmici za zadati datum. Tako na primjer, sljedeći petak za tekući datum bio bi:

```
SQL> SELECT next_day(sysdate, 'FRIDAY') "Sljedeći petak"
     FROM dual;
```

Funkcije ROUND i TRUNC
Funkcija *ROUND* vraća kao rezultat zaokružen datum po formatu definisanom kroz drugi parametar po pozivu funkcije. U slučaju da je drugi parametar izostavljen, tada će funkcija kao rezultat vratiti datum sa setovanim satima na 00:00:00 i datumom + jedan dan ako je vrijeme veće od 12:00 sati, u protivnom će vratiti tekući datum.
Funkcija *TRUNC* radi isto što i *ROUND* samo što uvijek vraća tekući datum, bez obzira na trenutno vrijeme.

```
SQL> SELECT round(sysdate) "ROUND bez zadatog formata",
            trunc(sysdate) "TRUNC bez zadatog formata",
            round(sysdate, 'mm') "ROUND sa zadatim formatom",
            trunc(sysdate, 'yyyy') "TRUNC sa zadatim formatom"
     FROM dual;
```

Aritmetičke funkcije

Mnoge implementacije SQL uključuju matematičke funkcije za rad s podacima u bazi podataka. Većina funkcija koja će se ovdje navesti je podržana od strane mnogih baza, osim njih nekolicina koje su specifične samo za Oracle bazu podataka.

Funkcija ABS
Kao rezultat funkcija vraća apsolutnu vrijednost proslijeđenog broja. Na primjer:

```
SQL> SELECT abs(-0.12789)
     FROM dual;
```

Funkcije CEIL i FLOOR
Funkcia CEIL vraća najmanju cjelobrojnu vrijednost koja je veća ili jednaka argumentu, dok funkcija FLOOR radi suprotno, odnosno vraća kao rezltat najveću cjelobrojnu vrijednost koja je jednaka ili manja od argumenta. Na primjer:

```
SQL> SELECT ceil(12.45),
            floor(12.45)
     FROM dual;
```

Funkcije COS, SIN, TAN, COSH, SINH, TANH
Ove funkcije predstavljaju trigonometrijske funkcije. Vrijednosti koje se prosljeđuju kao porametar datim funkcijama izražene su u radijanima.

```
SQL> SELECT cos(12),
            cosh(.75),
            sin(12),
            sinh(.5),
            tan(23),
            tanh(12.45)
     FROM dual;
```

Funkcija EXP
Ova funkcija stepenuje broj *e*.

```
SQL> SELECT exp(-3.134567)
     FROM dual;
```

Funkcije LN i LOG
Funkcija *LN* kao rezultat vraća prirodni logaritam zadatog argumenta, dok *LOG* funkcija vraća isto logaritam po bazi koja se prosljeđuje kao prvi argument funkciji. Na primjer, predpostavimo da je potrebno vratiti *ln* za argument 3.134567 i *log* po bazi 2 za isti broj.

```
SQL> SELECT ln(3.134567),
            log(2,3.134567)
     FROM dual;
```

Funkcija MOD
Funkcija *MOD* kao rezultat vraća ostatak dijeljenja. Na primjer:

```
SQL> SELECT MOD(salary, commission_pct)
     FROM employees;
```

Funkcija POWER
Funkcija *POWER* je funkcija stepenovanja, a kao rezultat funkcija vraća stepenovani prvi argument s drugim.

```
SQL> SELECT POWER(2, 3)
     FROM dual;
```

Funkcija SQRT
Ova funkcija kao rezultat vraća kvadratni korijen argumenta.

```
SQL> SELECT SQRT(16)
     FROM dual;
```

Funkcije za konverziju

Funkcije konverzije, najednostavnije rečeno, konvertuju jedan tip podataka u drugi tip. Generalno gledajući postoje tri takve funkcije koje omogućavaju jednostavan način konvertovanja podatka iz jednog tipa u drugi, dok sam naziv funkcije govori u koji tip podataka se želi konvertovati zadati argument.

Funkcija TO_CHAR
Osnovna namjena ove funkcije je konverzija broja u karaktere. Naravno, data funkcija nije namijenjena samo za konverziju broja u karaktere, nego i za konverziju drugih tipova podataka, kao što je datum u karaktere, gdje se uzma u obzir i drugi argument funkcije za formatiranje podataka datuma u karaktere.

```
SQL> SELECT to_char(-0.12789),
            to_char(sysdate, 'dd.mm.yyyy')
     FROM dual;
```

Kod konverzije broja u karaktere mogu se koristiti neke od sljedećih oznaka (oznake se koriste za formatiranje brojeva u karaktere):

Oznaka	Značenje	Primjer	Rezultat
9	Označava jednu cifru broja	9999999	1024
0	Prikazuje nulu na mjestu gdje nema broj u protivnom prikazuje broj.	0000000	0001024
.	Predstavlja decimalnu tačku u broju koji se prikazuje.	9999.999	1024.000
,	Oznaka koja simbolizira oznaku hiljade	99,999.99	1,024.00
MI	Minus oznaka s desne strane za negativne brojeve	9,999.00MI	1,024.00-
V	Množitelj broja brojem 10 n puta, gdje je n broj označava cifre iza oznake V	9999V999	1024000
B	Prikazuje 0 vrijednost kao blanko znak	B9999999	1024

Kada je u pitanju datum, tada se za konverziju datuma u karaktere mogu koristiti sljedeće oznake konverzije:

Oznaka	Značenje
SCC ili CC	Stoljeće, S prefiks za ozaku datuma prije nove ere (datum sa oznakom -)
YYYY ili SYYYY	Oznaka za godinu sa četiri cifre, na primjer 2007
YYY, YY ili Y	Oznaka za tri, dvije ili jednu cifru godine
IYYY, IYY, IY ili I	Oznaka za četiri, tri, dvije ili jednu cifru godine zasnovane na ISO standardu
BC ili AD	BD ili AD indikator
B.C. ili A.D.	BD ili AD indikator
Q	Oznaka za kvartal u godini
MM	Dvije cifre za oznaku mjececa
MONTH	Naziv mjeseca sa nadopunjenim blanko znakovima dužine 9 karaktera
MON	Skračeni naziv mjeseca dužine tri karaktera
DD	Dvije cifre za oznaku dana
DDD	Broj dana od 01.01.---- godine

DAY	Naziv dana u sedmici sa nadopunjenim blanko znakovima dužine 9 karaktera
DY	Skračeni naziv za dan u sedmici dužine tri karaktera

Kod Oracle baze podataka je specifično to što se i vrijeme nalazi u sastavu datuma, tako da se vrijeme može dobiti iz datuma na osnovu sljedećih oznaka konverzije:

Oznaka	**Značenje**
AM ili PM	Oznaka za sate prije podne ili poslije podne.
A.M. ili P.M.	Oznaka za sate prije podne ili poslije podne.
HH, HH12, HH24	Dvije cifre za oznaku sata, oznaka za sate od 1 do 12, oznaka za sate od 0 do 23
MI	Dvije cifre za oznaku minuta
SS	Dvije cifre za oznaku sekunda

```
SQL>  SELECT to_char(sysdate, 'day - month syyyy, hh24:mi:ss')
      FROM dual;
```

Funkcija TO_NUMBER
Osnovna namjena ove funkcije je konverzija karaktera u broj. Prvi argument sadrži karaktere koji se konvertuju, a drugi argument format po kojem se vrši konverzija

```
SQL>  SELECT to_number('1,024.890', '9,999.999')
      FROM dual;
```

Funkcija TO_DATE
Datum je najvjerovatnije, najčešće korišten tip podataka na osnovu kojeg se vrše određena sumiranja, poređenja, analize i sl. Iz tih razloga, kao što se može vidjeti u tabeli konverzija iz datuma u karaktere postoji jako bogata kolekacija oznaka koje se mogu koristiti za konverziju karaktera u datume. Logika koja se koristi za ovaj tip poređenja je skori po isti kao i za konverziju datuma u karatere.

```
SQL>  SELECT to_date('01.01.2007' , 'dd.mm.yyyy'),
              to_date('01-01-07 23/12:12', 'dd-mm-yy hh24/mi:ss')
      FROM dual;
```

Funkcije za rad sa karakterima

Mnogobrojne implementacije SQL uključuju značajan broj karakterih funkcija koje omogućavaju veću fleksibilnost za rad sa karakterima. U tekstu koji slijedi date su najčešće korišene funkcije u svakodnevnoj upotrebi SQL-a.

Funkcija CHR
Funkcija *CHR* vraća kao rezultat karakter koji odgovara ASCII broju, ili setovanom kodu karaktera baze podataka koji se prosljeđuje kao argument funkciji.

```
SQL>  SELECT chr(65)
      FROM dual;
```

Funkcija CONCAT
Ova funkcija se koristi za spajanje dva karaktera. Ekvivalent "||" označava spajanje dva ili više karaktera (vježbe I).

```
SQL> SELECT concat(first_name, last_name) "Naziv zaposlenog"
     FROM employees
     WHERE salary > 2500;
```

Funkcije LOWER, UPPER i INITCAP

Funkcije *LOWER*, *UPPER*, *INITCAP* su namjenjene za alfa karakterene konverzije podataka. Naime, funkcija *LOWER* pretvara sve alfa karakterne oznake u mala slova, *UPPER* u velika slova, dok *INITCAP* predvara svaki prvi alfa karakter stringa u veliko slova, a ostala u mala.

```
SQL> SELECT lower(first_name || ' ' || last_name) "LOWER funkcija",
            upper(first_name || ' ' || last_name) "UPPER funkcija",
            initcap(first_name || ' ' || last_name) "INITCAP funkcija"
     FROM employees
     WHERE department_id IN (10, 30, 80, 90);
```

Funkcije LPAD i RPAD

Funkcije LPAD i RPAD su kao što im i samo ime kaže koriste za nadopunjavanje karakternih vrijednosti do određene dužine karakternim oznakama definisanim kroz argument funkcije. Obje funkcije imaju minimalno dva, a maksimalno tri argumenta za poziv. Prvi argument označava string nad kojim se vrši operacija nadoupnjavanja stringa. Drugi argument označava dužinu karaktera koji će se vratiti po pozivu funkcije. Treći argument je karakter koji se koristi da zamijeni prazna mjesta s datim karakterom. U slučaju da se izostavi treći argument funkcije predefinisana vrijednost oznake za nadopunjavanje karaktera je blanko znak.

```
SQL> SELECT lpad(first_name || ' ' || last_name, 100, '*') "LPAD funkcija",
            rpad(first_name || ' ' || last_name, 100, '*') "RPAD funkcija"
     FROM employees;
```

Funkcije LTRIM i RTRIM

Funkcije LTRIM i RTRIM za razliku od predhodne dvije funkcije radi obratnu operaciju, tj. iz stringa izuzima s lijeve ili desne strane određene karaktere definisane putem drugog argumenta funkcije. Funkcije posjeduju dva argumenta za poziv pri ćemu je prvi argument string koji se prosljeđuje, a drugi argument je karakter koji se koristi kao oznaka za izuzimanje iz stringa s lijeve ili desne strane. U slučaju da se drugi argument izostavi defaultna vrijednost za ovaj argument će biti blanko znak.

```
SQL> SELECT ltrim('     O B P     ') "LTRIM funkcija",
            rtrim('***** O B P *****', '*') "RTRIM funkcija"
     FROM dual;
```

Funkcija REPLACE

Funkcije REPLACE koristi se za zamjenu određenog karaktera ili niza karaktera u stringu. Funkcija posjeduje tri argumenta od kojih prvi označava string koji se prosljeđuje, drugi argument karakter ili niza karaktera koji se mijenjaju u stringu i treći argument označava karakter ili niz karaktera kojim se mijenjaju karakteri drugog argumenta. U slučaju da se treći argument izostavi tada se karakteri za zamjenu mijenjaju sa NULL vrijednošću, odnosno zamjenski karakteri se izbacuju iz stringa.

```
SQL> SELECT first_name || ' ' || last_name,
            replace(first_name || ' ' || last_name, 'al' , '*****')
     FROM employees;
```

Funkcija SUBSTR

Ova funkcija, kao što joj i samo ime kaže, koristi se za izdvajanje podstringa iz stringa koji se prosljeđuje kao prvi argument funkcije. Funkcija posjeduje tri argumenta od kojih

drugi označava poziciju u stringu od koje se uzima prvi karakter i treći argument označava broj karaktera koji se izdvajuju za podstring od prvog uzetog karaktera.

```
SQL> SELECT first_name || ' ' || last_name,
            substr(first_name || ' ' || last_name, 3, 4)
     FROM employees;
```

Funkcija INSTR
Funkcija INSTR se koristi za pronalaženje pozicije u stringu koja posjeduje karakter koji se pretražuje. Funkcija posjeduje četiri argumenta od kojih prvi označava string koji se prosljeđuje. Drugi argument je karakter koji se pretražuje. Treći argument određuje poziciju od koje se vrši pretraživanje. Četvrti argument određivanje koji će se karakter uzeti u obzir prilikom prikaza pozicije karaktera u stringu. Tako na primjer, neka je potrebno pronaći poziciju karaktera 'a' u nazivima zaposlenih počevši od 2 pozicije u stringu i potrebno je uzeti u obzir pozicije drugog karaktera 'a' prilikom pretraživanja.

```
SQL> SELECT instr(first_name || ' ' || last_name, 'a', 2, 2)
     FROM employees;
```

Funkcija LENGTH
Ova funkcija vraća kao rezultat dužinu stringa koji je njen argument po pozivu funkcije. Predpostavimo da trebamo napisati upit koji će vratiti naziv zaposlenog i dužinu imena zaposlenog za sve zaposlene iz odjela 90.

```
SQL> SELECT first_name,
            length(first_name)
     FROM employees
     WHERE department_id = 90;
```

Mješovite funkcije

U zavisnosti od različitih baza podataka, odnosno SQL implementacija ove funkcije mogu biti specifične samo za datu implementaciju SQL, dok u drugim implementacijama nisu uopšte zastupljene ili se nazivaju drugačijim imenom. Iz tih razloga ovdje će biti predstavljene samo dvije funkcije od kojih je jedna zajednička za sve SQL implementacije koje se pridržavaju ISO standarda, i druga koja je specifična samo za Oracle bazu podataka.

Funkcija USER
Funkcija *USER* vraća naziv korisnika (username) koji je trenutno konektovan na bazu podataka. Ova funkcija se koristi gotovo kod svih baza podataka koji se pridržavaju ISO standarda.

```
SQL> SELECT user
     FROM dual;
```

Funkcija DECODE
Ova funkcija je specifična samo za Oracle bazu podataka, dok je i kod Oracle baze podataka podržana ekvivalentna funkcija njoj, kao i u drugim bazama podataka, tj. funkcija *CASE*. *DECODE* funkcija se koristi za dekodiranje izraza na sličan način kao što je IF – THEN – ELSE logika, tj. logici koja se koristi u različitim programski jezicima. Na primjer, neka je potrebno napisati upit koji će vratiti naziv zaposlenog, šifru odjela, platu i platu:

- uvećanu za 10% za zaposlene iz odjela 10,

- uvećanu za 20% za zaposlene iz odjela 30,
- uvećanu za 15% za zaposlene iz odjela 90 i
- za sve ostale zaposlene uvećanu za 5%.

```
SQL> SELECT first_name || ' ' || last_name zaposlenik,
            department_id odjel,
            salary stara_plata,
            decode(department_id,
                    10, salary * 1.1,
                    30, salary * 1.2,
                    90, salary * 1.15,
                    salary * 1.05) nova_plata
     FROM employees;
```

Zadaci

1. Napisati upit koji će prikazati trenutni datum i korisnika logiranog na bazu podataka. Labele za kolone su date i user respektivno.
2. Napisati upit koji će prikazati šifru, ime, prezime, platu i platu uvećanu za 25% kao cijeli broj. Labela za novu platu je «plata uvećana za 25%».
3. Modificirati upit 2 tako da se doda nova kolona koja će iz nove plata izdvojiti posljednje 2 cifre plate i prikazati kao novu kolonu koja će se zvati «ostatak plate».
4. Napisati upit koji će prikazati naziv zaposlenog, datum zaposlenja i datum prvog ponedjeljka nakon 6 mjeseci rada zaposlenog. Datume predstaviti u formatu naziv dana – naziv mjeseca, godina.
5. Za sve zaposlene iz tabele zaposlenih prikazati naziv zaposlenog, naziv odjela i kontinent, kao i broj mjeseci zaposlenja zaposlenika. Broj mjeseci zaokružiti na cjelobrojnu vrijednost.
6. Napisati upit koji će prikazati za sve zaposlene iz odjela 10, 30 i 50 sljedeće: «naziv zaposlenog» prima platu «iznos plate» mjesečno ali on bi želio platu «plata uvećana za procenat dodataka na platu i pomnožena sa 4,5 puta». Labela za kolonu je «plata o kojoj možeš samo sanjati».
7. Napisati upit koji će vratiti jednu kolonu "Ime + Plata" od naziva zaposlenog i njegove plate za sve zaposlene. Formatirati "Ime + plata" tako da je vraćena kolona dužine 50 karaktera i s lijeve strane nadopunjena s «$» karakterom.
8. Napisati upit koji će prikazati naziv zaposlenog i dužinu naziva zaposlenog za sve zaposlene čija imena počinju sa slovima A, J, M i S. Naziv zaposlenog treba prikazati tako da je prvi karakter naziva predstavljen malim slovom, a ostali karakteri velikom slovima.
9. Napisati upit koji će prikazati naziv, datum zaposlenja i dan u sedmici kada je zaposleni počeo da radi. Rezultati sortirati po danima u sedmici počevši od ponedjeljka.
10. Napisati upit koji će prikazati naziv zaposlenog, grad u kojem zaposlenik radi, kao i iznos dodatka na platu. Za one zaposlene koji ne dobivaju dodatak na platu ispisati «zaposlenik ne prima dodatak na platu».
11. Napisati upit koji će prikazati naziv zaposlenog, platu i indikator plate izražene za znakom «*». Svaka zvjezdica označava jednu hiljadu od plate. Na primjer ako uposleni prima 2600 KM platu, tada treba za indikator plate ispisati ***, a ako prima 2400 onda **.
12. Napisati upit koji će prikazati sve zaposlene s stepenom posla. Stepen posla potrebnoa je uraditi prema sljedećoj specifikaciji:

Posao	Stepen
President	A
Manager	B
Analyst	C
Sales manager	D
Programmer	E
Ostali	X