

Jednoredne funkcije

FUNKCIJE

U SQL-u postoji mnogo tipova funkcija koje se koriste da transformišu ulaz, koji je dat u jednom obliku, u izlaz u drugom obliku. Ove funkcije se koriste za manipulaciju vrednostima podataka.

Funkcije su produžetak SQL-a – one omogućavaju programerima da izvode uobičajene zadatke, a da pri tome sami ne moraju da odrade celo programiranje. Funkcije su programi koji izvode operacije (akcije) nad vrednostima ili kolonama. Imaju i ulaz i izlaz. Ulaz u funkciju se naziva argumentom. Funkcije koriste više argumenata da bi izvele proračune i vratile rezultat na izlazu. Ulazne vrednosti bivaju promenjene ili transformisane tako da se na izlazu dobije nešto drugo, novo – to je svrha funkcija.

FUNKCIJE

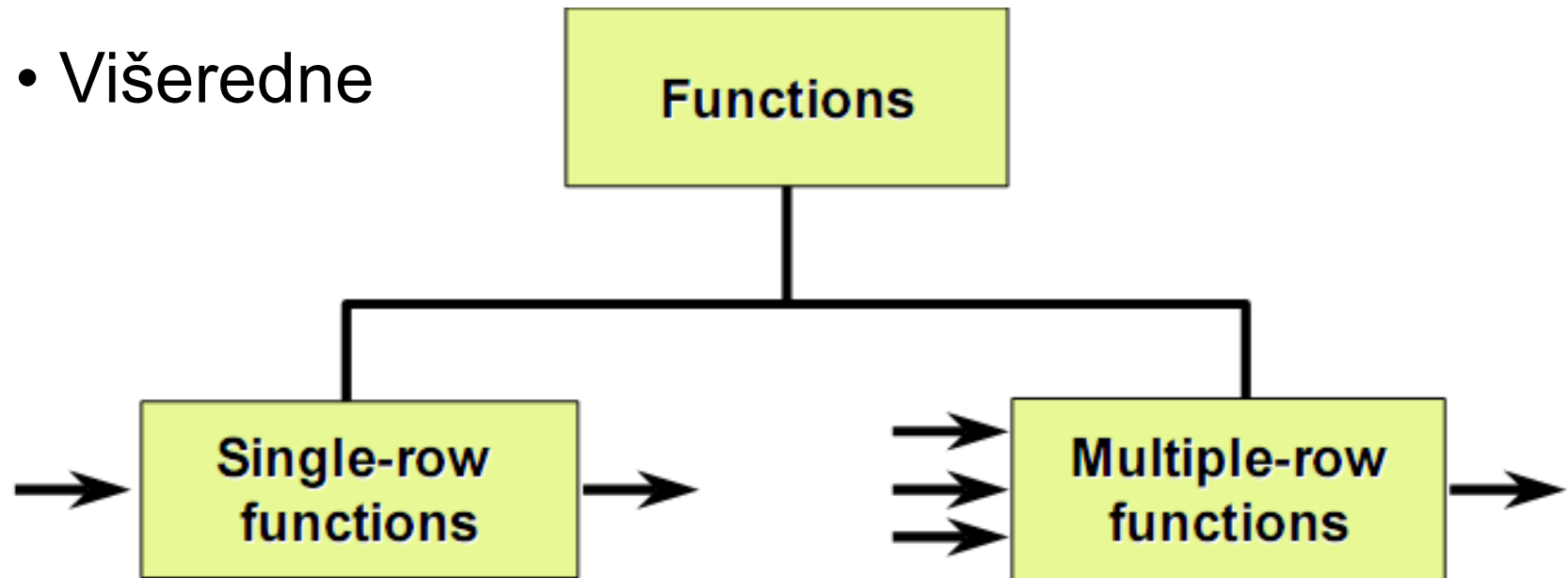
Funkcije predstavljaju moćno sredstvo SQL-a. Mogu biti upotrebljene da:

- Izvode računske operacije kao što su zaokruživanje brojeva na određeni broj decimala
- Modifikuju tekstualne podatke tako što, na primer, menjaju velika slova u mala, dopunjuju ili skraćuju tekst itd.
- Manipulišu izlazom za grupe redova, nalaženjem proseka ili sume za nekoliko redova.
- Menjaju format zapisa datuma.
- ...

FUNKCIJE

Postoje dva različita tipa funkcija:

- Jednoredne
- Višeredne



JEDNOREDNE FUNKCIJE:

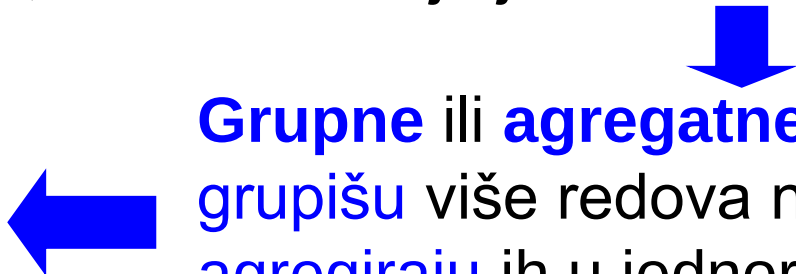
- Manipulišu podacima
- Prihvataju argumente i vraćaju jednu vrednost
- Deluju nad svakim vraćenim redom
- Vraćaju jedan rezultat po svakom redu
- Mogu da modifikuju tip podatka
- Mogu da formatiraju podatke za prikaz
- Mogu da budu ugneždene
- Mogu se upotrebiti u klauzulama
`SELECT`, `WHERE` i `ORDER BY`.

Jednoredne funkcije prihvataju jedan ili više argumenata i vraćaju jedan rezultat po redu. Tako da, ako primenite jednorednu funkciju na 12 redova, dobićete 12 izlaznih rezultata jednoredne funkcije.

VIŠEREDNE FUNKCIJE

Višeredne funkcije operišu nad grupom redova da bi dale jedan rezultat po svakoj grupi. One uzimaju više redova kao ulaz, a onda vraćaju jedan red kao izlaz.

- AVG
- COUNT
- MAX
- MIN
- SUM



Grupne ili **agregatne** funkcije: grupišu više redova na ulazu i agregiraju ih u jednorodni rezultat na izlazu.

```
SELECT MAX(salary), MIN(salary), AVG(salary)
FROM employees;
```

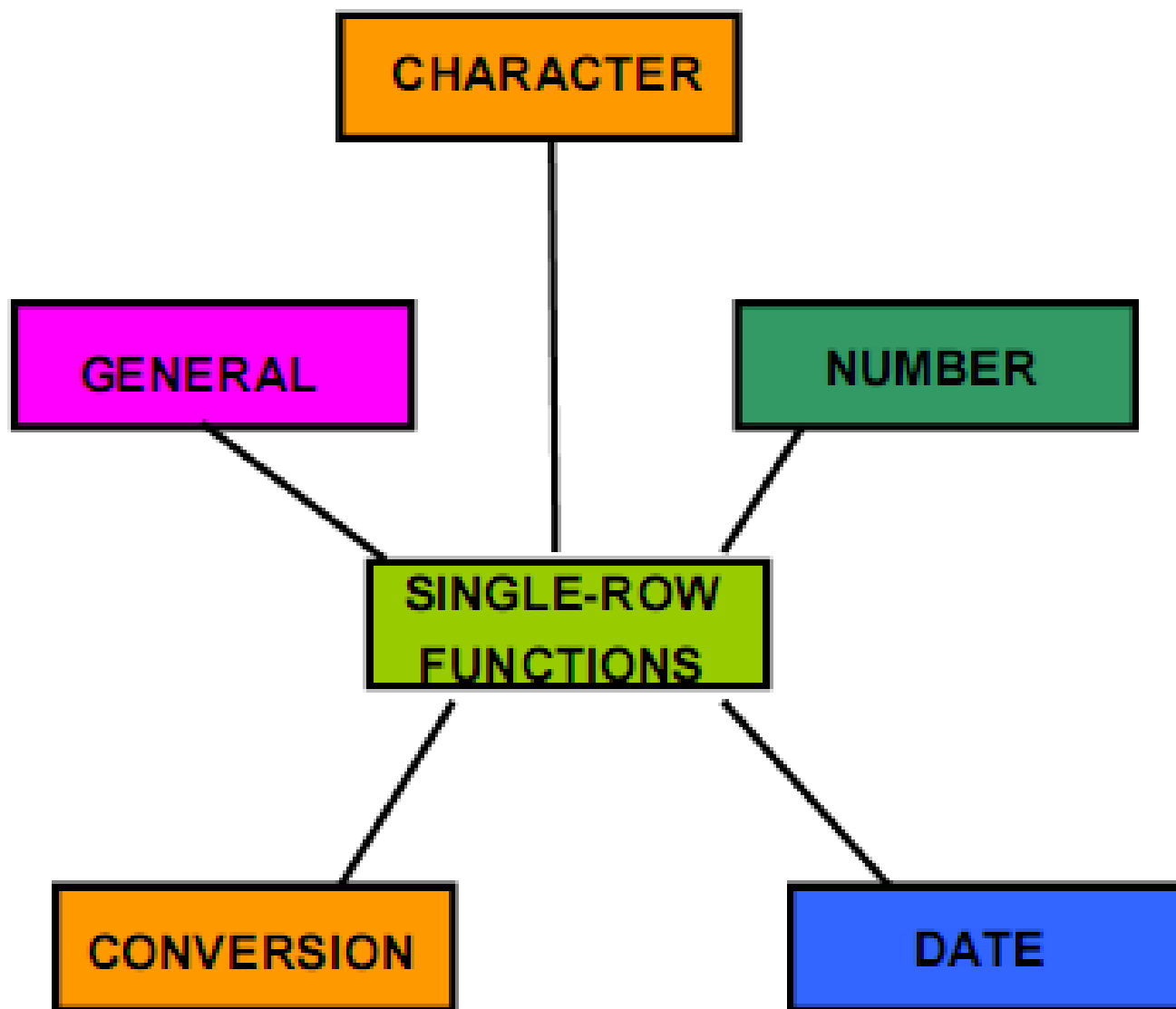
MAX(SALARY)	MIN(SALARY)	AVG(SALARY)
24000	2500	8775

```
SELECT first_name, LENGTH(first_name),  
       UPPER(first_name), LOWER(first_name)  
FROM employees WHERE department_id=80  
ORDER BY first_name;
```

FIRST_NAME	LENGTH (FIRST_NAME)	UPPER (FIRST_NAME)	LOWER (FIRST_NAME)
Eleni	5	ELENI	eleni
Ellen	5	ELLEN	ellen
Jonathon	8	JONATHON	jonathon

```
SELECT MAX(first_name), MIN(first_name),  
       MAX(LENGTH(first_name)), AVG(LENGTH(first_name))  
FROM employees WHERE department_id=80;
```

MAX(FIRST_NAME)	MIN(FIRST_NAME)	MAX(LENGTH (FIRST_NAME))	AVG(LENGTH (FIRST_NAME))
Jonathon	Eleni	8	6

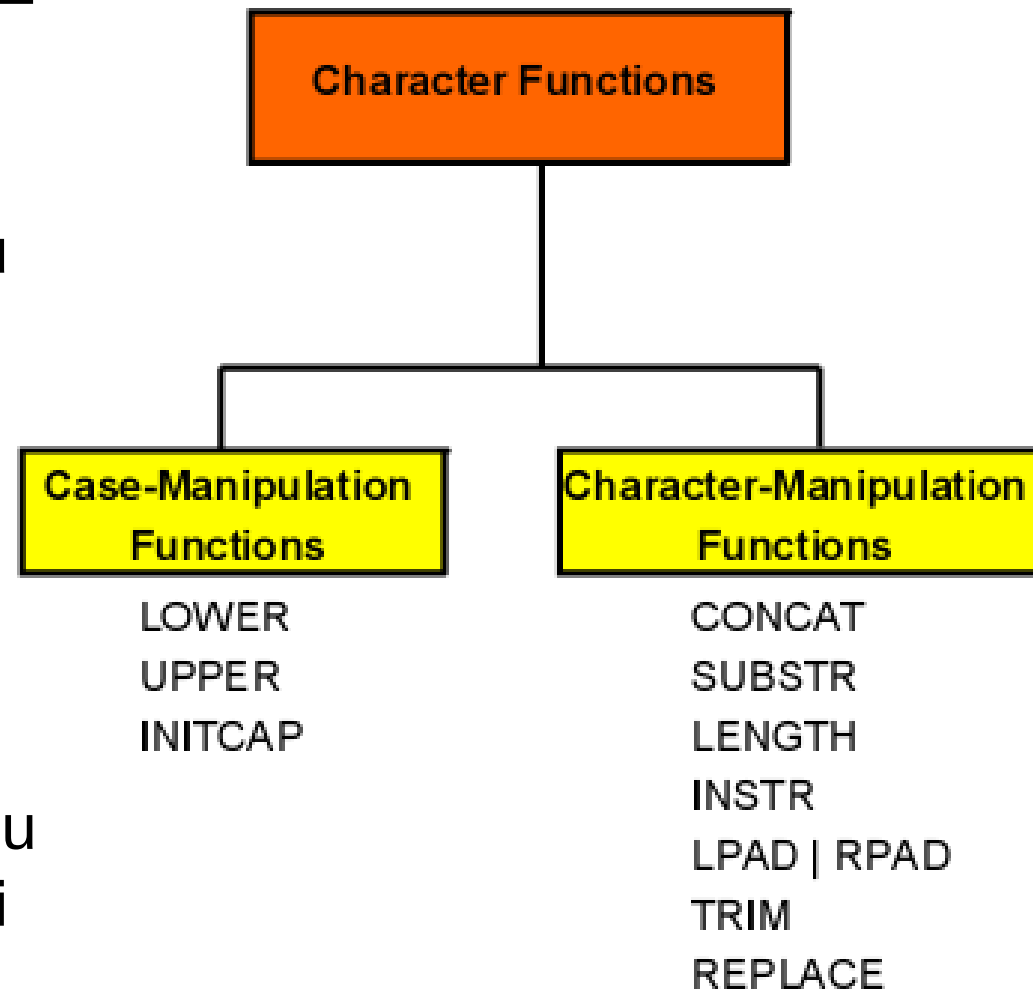


JEDNOREDNE KARAKTER-FUNKCIJE

Case-manipulativne funkcije se koriste za konvertovanje iz malih u velika slova ili obrnuto. Ove konverzije mogu biti upotrebljene za formiranje izlaza i kada vršimo pretragu za određenim stringom.

Dakle, ove funkcije mogu biti upotrebljene u većini SQL rečenica.

CHARACTER FUNCTIONS





Reci mi/Pokaži mi

CASE-MANIPULATION FUNCTIONS

Kao što se može videti u sledećem primeru Case-manipulativne funkcije su često korisne kada tražite određene podatke u Oracle bazi podataka ako ne znate kojim slovima je podataka zapisan. Sa stanovišta Oracla 'V' ili 'v' nisu isti karakteri, tako da neće vratiti zapis ako niste koristili odgovarajuća slova pri pretrazi.

LOWER(kolona | izraz) konvertuje alfa karaktere u mala slova.

```
SELECT title  
FROM d_cds  
WHERE LOWER(title) = 'carpe diem';
```



Reci mi/Pokaži mi

CASE-MANIPULATION FUNCTIONS

UPPER(kolona | izraz) konvertuje alfa karatkere u velika slova.

```
SELECT title  
FROM d_cds  
WHERE UPPER(title) = 'CARPE DIEM';
```

INITCAP(kolona | izraz) konvertuje vrednosti alfa karaktera u veliko prvo slovo svake reči.

```
SELECT title  
FROM d_cds  
WHERE INITCAP(title) = 'Carpe Diem';
```



Reci mi/Pokaži mi

CASE-MANIPULATION FUNCTIONS

Karakter manipulativne funkcije se koriste za izvođenje, menjanje, formatiranje ili na neki način promenu string karaktera. Jedan ili više karaktera ili reči se prenose u funkciju i funkcija tada izvodi svoju funkcijonalnost nad izlazom stringova karaktera i vraća promenjenu, izvođenu, računatu ili izmenjenu vrednost.

CONCAT: Spaja dve vrednosti.

SUBSTR: Izvodi string određene dužine.

Function	Result
CONCAT ('Hello', 'World')	HelloWorld
SUBSTR ('HelloWorld', 1, 5)	Hello



Reci mi/Pokaži mi

CASE-MANIPULATION FUNCTIONS

LENGTH: Prikazuje dužinu stringa kao brojnu vrednost.

INSTR: Pronalazi numeričku poziciju imenovanog karaktera.

LPAD: Popunjava levu stranu karaktera, **rezultujući desno-poravnatom vrednošću**

RPAD: Popunjava desnu stranu karaktera, rezultujući levo poravnatom vrednošću.

TRIM: Uklanja sve specifične karaktere bilo sa početka ili kraja stringa. Sintaksa trim funkcije je: trim ([leading | trailing | both [character(s) to be removed]] string to trim)

Function	Result
LENGTH ('HelloWorld')	10
INSTR ('HelloWorld', 'W')	6
LPAD (salary, 10, '*')	*****24000
RPAD (salary, 10, '*')	24000*****
TRIM ('H', FROM 'HelloWorld')	elloWorld



Reci mi/Pokaži mi

CASE-MANIPULATION FUNCTIONS

REPLACE: Zamenjuje sekvence karaktera u stringu sa drugom grupom karaktera.

Sintaksa za REPLACE funkciju je: (string1, string_to_replace, [replacement_string])

string1 je string koji će imati zamenjene karaktere. string_to_replace je string koji će biti pretražen i uklonjen iz stringa 1. [replacement_string] je novi string koje će biti insertovan u string 1.

```
SELECT REPLACE('JACK and JUE','J','BL') "Changes"  
FROM DUAL
```

Function	Result
REPLACE('JACK and JUE','J','BL')	BLACK and BLUE

NUMERIČKE FUNKCIJE

ROUND: Koristi se za zaokruživanje brojeva na određeni broj decimala. ROUND se takođe može upotrebiti za zaokruživanje brojeva levo od decimalnog zareza. ROUND se može upotrebiti i kod datuma.

ROUND(column|expression, decimal places)

TRUNC: Koristi se da skрати kolonu, izraz ili vredost za određen broj decimalnih mesta. Kada se TRUNC koristi, ako broj decimalnih mesta nije određen, onda je taj broj po difoltu nula. I TRUNC se može koristiti kod datuma.

TRUNC(column|expression, decimal places)

MOD: Koristi se da vrati ostatak deljenja celog broja drugim celim brojem.

MOD(broj1, broj2)

NUMERIČKE FUNKCIJE

ROUND(45.926)	46	TRUNC(45.926)	45
ROUND(45.926, 0)	46	TRUNC(45.926, 0)	45
ROUND(45.926, 1)	45,9	TRUNC(45.926, 1)	45,9
ROUND(45.926, -1)	50	TRUNC(45.926, -1)	40

MOD(45, 2) = 1

MOD(45, 3) = 0

MOD(45, 4) = 1

MOD(45, 13) = 6

MOD se koristi i za ispitivanje parnosti broja.

MOD(n, 2) = 0 → broj n je paran

MOD(n, 2) = 1 → broj n je neparan

Podaci tipa DATE

Oraklove baze podataka čuvaju datume u numeričkom formatu koji uključuje vek, godinu, mesec, dan, sate, minute i sekunde.

Difolt prikaz i izlazni format za bilo koji datum je DD-MON-YY
na primer: **22-MAY-08**.

SYSDATE

SYSDATE je datumska funkcija koja vraća vreme i datum sa tekućeg servera baze podataka, a prikazuje samo datum u formatu DD-MON-YY.

Datumske funkcije

Date Functions

Function	Description
MONTHS_BETWEEN	Number of months between two dates
ADD_MONTHS	Add calendar months to date
NEXT_DAY	Next day of the date specified
LAST_DAY	Last day of the month
ROUND	Round date
TRUNC	Truncate date

Datumske funkcije

Sve datumske funkcije vraćaju vrednost tipa DATE, osim funkcije MONTHS_BETWEEN , koja vraća numerički tip vrednosti.

Pogledajmo datume zaposlenja u tabeli EMPLOYEES za zaposlene u odeljenju 90 i primenimo na njih datumske funkcije iz tabele sa pretodnog slajda.

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

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	HIRE_DATE
100	Steven	King	17-JUN-87
101	Neena	Kochhar	21-SEP-89
102	Lex	De Haan	13-JAN-93

MONTHS_BETWEEN(kasniji_datum, raniji_datum)

vraća broj meseci između datuma datih u zagradi

```
SELECT employee_id, hire_date, SYSDATE,  
MONTHS_BETWEEN(SYSDATE, hire_date) "Trajanje"  
FROM employees WHERE department_id = 90;
```

EMPLOY EE_ID	HIRE_ DATE	SYSDATE	Trajanje
100	17-JUN-87	15-MAY-08	250.95552606033452807646356033452
101	21-SEP-89	15-MAY-08	223.82649380227001194743130227001
102	13-JAN-93	15-MAY-08	184.08455831839904420549581839904

MONTHS_BETWEEN(kasniji_datum, raniji_datum)
vraća broj meseci između datih datuma.

Rezultat je tipa NUMBER pa se mogu primeniti
numeričke funkcije ROUND i TRUNC da bismo
dobili ceo broj meseci.

```
SELECT employee_id, hire_date, SYSDATE,  
       ROUND(MONTHS_BETWEEN(SYSDATE, hire_date))  
           AS "Round(Trajanje)",  
       TRUNC(MONTHS_BETWEEN(SYSDATE, hire_date))  
           AS "Trunc(Trajanje)"  
FROM employees WHERE department_id = 90;
```

EMPLOYEE_ID	HIRE_DATE	SYSDATE	Round (Trajanje)	Trunc (Trajanje)
100	17-JUN-87	15-MAY-08	251	250
101	21-SEP-89	15-MAY-08	224	223
102	13-JAN-93	15-MAY-08	184	184



Reci mi/Pokaži mi

NVL FUNCTION

NVL funkcija konvertuje null vredost u datum, karakter ili broj. Tipovi podataka kolona null vrednosti i nove vrednosti moraju biti iste. NVL funkcija je:

NVL (value that may contain a null, value to replace the null)

Sledeći upit koristi NVL funkciju sa karakter tipom podatka:

```
SELECT NVL(comments, 'no comment')  
FROM D_PLAY_LIST_ITEMS;
```

NVL – zamenjuje null-vrednost nečim drugim

```
SELECT first_name, expertise AS "Stručnost", auth_expense_amt AS "Honorar"  
FROM d_partners;
```

FIRST_NAME	Stručnost	Honorar
Jennifer	Weddings	-
Jason	-	-
Allison	Event Planning	300000

```
SELECT first_name, NVL(expertise, 'nema') AS "Stručnost",  
                NVL(auth_expense_amt, 0) AS "Honorar"  
FROM d_partners;
```

FIRST_NAME	Stručnost	Honorar
Jennifer	Weddings	0
Jason	nema	0
Allison	Event Planning	300000



Reci mi/Pokaži mi

NVL funkcija se može upotrebiti za konvertovanje vrednosti kolona koje sadrže null-ove u broj pre izvođenja proračuna. Kada se aritmetički proračun izvodi sa null, rezultat je null. NVL funkcija može prvo da konvertuje null vrednost u broj pre izvođenja aritmetičkih proračuna da bise izbegao null rezultat.



Reci mi/Pokaži mi

U primeru, kolona `auth_expense_amt` u tabeli `D_PARTNERS` sadrži null vrednosti. `NVL` funkcija je iskorišćena da promeni null u nula pre aritmetičke kalkulacije.

```
SELECT first_name, last_name, NVL(auth_expense_amt, 0) * 1.05 AS  
Expenses  
FROM D_Partners;
```



Reci mi/Pokaži mi

NULLIF FUNCTION

NULLIF funkcija poredi dve funkcije. Ako su jednake, funkcija vraća null. Ako nisu jednake, funkcija vraća prvi izraz.

```
SELECT first_name, LENGTH(first_name) "Expression 1", last_name,  
       LENGTH(last_name) "Expression 2", NULLIF(LENGTH(first_name),  
       LENGTH(last_name)) AS "Compare Them" FROM D_PARTNERS;
```

FIRST_NAME	Expression 1	LAST_NAME	Expression 2	Compare Them
Jennifer	8	Cho	3	8
Jason	5	Tsang	5	(null)
Allison	7	Plumb	5	7

The NULLIF funkcija je: **NULLIF(expression 1, expression 2)**