

UGRAĐENE I KORISNIČKE SQL FUNKCIJE

Predmet: Administriranje baze podataka

Predavač: dr Dušan Stefanović

Trigeri, Funkcije i Procedure

- Sličnosti -

- To su objekti baze podataka koji sadrže kod koji se izvršava kao zasebna celina
- Slični su metodama, subrutinama ili modulima u drugim programskim jezicima
- Obično se pišu u T-SQL ali mogu da se napišu u bilo kom NET jeziku(VBA ili C#)

- Razlike -

	Trigeri	Funkcije	Store Procedure
Promena podataka	da	ne	da
Vraćanje vrednosti	nikad	uvek	ponekad
Način pozivanja	dogadjaj	u iskazu	exec

Trigeri, Funkcije i Procedure

- Razlike -

	Trigeri	Funkcije	Store Procedure
Promena podataka	da	ne	da
Vraćanje vrednosti	nikad	uvek	ponekad
Način pozivanja	dogadjaj	u iskazu	exec

- Trigerima i Procedurama je dozvoljeno da **menjaju podatke u bazi** što obično i rade dok **funkcijama to nije dozvoljeno**
- **Suština funkcija je da vraćaju vrednost koja može biti skalarna(jedna vrednost) ili tabelu**
- Trigeri nikad ne vraćaju vrednost jer oni manipulišu sa podacima onog trenutka kada se desi dogadjaj.
- Procedure obično imaju zadatak da promene vrednost u tabeli ali mogu i da vrate vrednost obično u formi 1 ili 0 u zavisnosti da li je zadatak uređen ili nije

Trigeri, Funkcije i Procedure

- Razlike -

	Trigeri	Funkcije	Store Procedure
Promena podataka	da	ne	da
Vraćanje vrednosti	nikad	uvek	ponekad
Način pozivanja	dogadjaj	u iskazu	exec

- Najveća razlika je u načinu pozivanja.
 - Procedure se pozivaju komandom exec
 - **Funkcija je deo Select, Delete ili Update iskaza**
 - Ukoliko vraća skalarnu vrednost poziva se u Select, Where ili Order segmentu
 - Ukoliko vraća tabelu poziva se u From segmentu SQL iskaza
 - Ne postoji komanda kojom se triger pokreće, već se on pokreće kao reakcija na neki događaj(Update, Insert ili Delete)

STRING FUNKCIJE

CHAR_LENGTH() = CHARACTER_LENGTH()

Vraća dužinu tj. **broj karaktera** u stringu

```
SELECT CHAR_LENGTH(CustomerName) AS LengthOfName  
FROM Customers;
```

LENGTH(string)

Vraća dužinu tj. **broj bajtova** u stringu

```
SELECT LENGTH(CustomerName) AS LengthOfName  
FROM Customers;
```

STRING FUNKCIJE

CONCAT(expression1, expression2, expression3,...)

CONCAT_WS(separator, expression1, expression2, expression3,...)

Spaja više iskaza

Ukoliko je bilo koji od iskaza null vratiće null

```
SELECT CustomerName, Contactname, CONCAT(CustomerName," ",Contactname)
FROM Customers;
```

```
SELECT CustomerName, Contactname, CONCAT_WS("-",CustomerName," ",Contactname)
FROM Customers;
```

STRING FUNKCIJE

POSITION(substring IN string)=LOCATE(substring, string, start)

Vraća poziciju(indeks) karaktera koji se traži

Pravi razliku između velika i mala slova

Ukoliko traženi karakter nije pronađen funkcija vraća 0

```
SELECT POSITION("a" IN CustomerName)  
FROM Customers;
```

```
SELECT LOCATE(" ", „it svet je lep“, 4) AS MatchPosition;
```

STRING FUNKCIJE

LEFT(string, number_of_chars)

Prikazuje broj karaktera iz stringa počevši sa leve strane

```
SELECT LEFT(CustomerName, 5) AS ExtractString  
FROM Customers;
```

RIGHT(string, number_of_chars)

Prikazuje broj karaktera iz stringa počevši sa desne strane

```
SELECT RIGHT(CustomerName, 5) AS ExtractString  
FROM Customers;
```


STRING FUNKCIJE

Zadatak

Iz kolone NazivKlijenta koja sadrži ime i prezime klijenta izdvojiti posebno ime a posebno prezime

NazivKlijenta	Ime	Prezime
Alfreds Futterkiste	Alfreds	Futterkiste
Ana Trujillo Emparedados y helados	Ana	Trujillo Emparedados y helados
Antonio Moreno Taquería	Antonio	Moreno Taquería
Around the Horn	Around	the Horn
Berglunds snabbköp	Berglunds	snabbköp
Blauer See Delikatessen	Blauer	See Delikatessen

```
SELECT NazivKlijenta,  
       Left(NazivKlijenta, Locate(" ", NazivKlijenta)-1) as Ime,  
       Right(NazivKlijenta, Character_Length(NazivKlijenta) - Locate(" ", NazivKlijenta)) as Prezime  
FROM Klijent;
```

STRING FUNKCIJE

LOWER(text)=LCASE(text)

Konvertuje string u mala slova

```
SELECT LCASE(CustomerName) AS LowercaseCustomerName  
FROM Customers;
```

UPPER(text)=UCASE(text)

Konvertuje string u velika slova

STRING FUNKCIJE

TRIM(string)

Uklanja vodeće i krajnje blanko karaktere iz stringa

Ne uklanja blanko karaktere iz sredine stringa

```
SELECT
    Character_Length(TRIM("  SQL  Obuka  ")) AS TrimmedString,
    Character_Length("  SQL  Obuka  ") AS OriginalString;
```

TrimmedString	OriginalString
13	20

LTRIM(string)

Uklanja vodeće blanko karaktere iz stringa

RTRIM(string)

Uklanja krajnje blanko karaktere iz stringa

STRING FUNKCIJE

LPAD(string, length, lpad_string)

Dodaje originalnom stringu sa leve strane karaktere(lpad_string) tako da zbirna dužina stringa odgovara dužini (length)

```
SELECT LPAD("SQL Obuka", 20, " VTS_ ") as LeftPadding;
```

LeftPadding
VTS_ VTS_ SQL Obuka

RPAD(string, length, rpad_string)

Dodaje originalnom stringu sa desne strane karaktere(rpad_string) tako da zbirna dužina stringa odgovara dužini (length)

STRING FUNKCIJE

MID(string, start, length)

Prikazuje deo stringa iz originalnog stringa koji počinje od pozicije start i ima dužinu length

```
SELECT MID(CustomerName, 5, 7) AS MidCustomerName  
FROM Customers;
```

Ulazni parametri funkcije	Opis
<code>string</code>	Obavezan. Originalni string
<code>start</code>	Obavezan. Početna pozicija. Može biti pozitivan ili negativan broj. Ukoliko je pozitivan broj funkcija vadi podatak od početka stringa. Ukoliko je negativan broj funkcija vadi podatak od kraja stringa
<code>length</code>	Obavezan. Broj karaktera koji će substring da sadrži

STRING FUNKCIJE

Zadatak

Iz kolone NazivKlijenta prikazati tekst između drugog i trećeg blanko razmaka

NazivKlijenta	Ime
Niš ElektroLux Kolonija 9	ElektoLux
Beograd Comtrade Prvomajska 10	Comtrade
Nis Vts Aleksandra Medvedeva 20	Vts
Leskovac Logistika Kozaračka 13	Logistika

Određujemo poziciju drugog blanko karaktera odakle počinjemo da prikazujemo substring

```
SELECT MID(NazivKlijenta,  
  Locate(" ", NazivKlijenta, locate(" ", NazivKlijenta)+1),  
  Locate(" ", NazivKlijenta, 1+locate(" ", NazivKlijenta, 1+locate(" ", NazivKlijenta))) -  
  Locate(" ", NazivKlijenta, locate(" ", NazivKlijenta)+1)) as Ime
```

Određujemo dužinu substringa tako što oduzimamo dužine između drugog i trećeg blanko karaktera

STRING FUNKCIJE

SUBSTR(string, start, length)=SUBSTRING (string, start, length)

Prikazuje deo stringa iz originalnog stringa koji počinje od pozicije start i ima dužinu length

```
SELECT SUBSTR(CustomerName, 5, 7) AS MidCustomerName  
FROM Customers;
```

Ulazni parametri funkcije	Opis
<code>string</code>	Obavezan. Originalni string
<code>start</code>	Obavezan. Početna pozicija. Može biti pozitivan ili negativan broj. Ukoliko je pozitivan broj funkcija vadi podatak od početka stringa. Ukoliko je negativan broj funkcija vadi podatak od kraja stringa
<code>length</code>	Opcion. Ukoliko je izostavljen ceo substring biće vraćen od startne pozicije

STRING FUNKCIJE

SUBSTRING_INDEX(string, delimiter, number)

Prikazuje deo stringa iz originalnog stringa koji se nalazi pre pojave unetog delimitera kao i nakon kog ponavljanja.

Ulazni parametri funkcije	Opis
<i>string</i>	Obavezan. Originalni string
<i>delimiter</i>	Obavezan. Delimiter koji se traži u originalnom stringu
<i>broj</i>	Obavezan. Od kog po redu delimitera da se prikaže novi string. Ukoliko je pozitivna vrednost funkcija prikazuje string sa leve strane do delimitera. Ukoliko je negativan broj, funkcija prikazuje string sa desne strane do delimitera.

```
SELECT SUBSTRING_INDEX("www.vtsnis.edu.rs", ".", 2);
```


STRING FUNKCIJE

Zadatak

Iz kolone NazivKlijenta prikazati tekst posle poslednjeg blanko karaktera

Napomena: Problem je što funkcija Locate pretragu uvek počinje sa leva na desno

```
SELECT SUBSTRING_INDEX(NazivKlijenta, " ", -1);
```

STRING FUNKCIJE

Zadatak

Iz kolone NazivKlijenta prikazati tekst posle poslednjeg blanko karaktera

Napomena: Koristiti funkcije **CHAR_LENGTH** = Len() -> Excel, **Locate** = Find->Excel, **Replace** =!
Substitute -> Excel, **Right** = Right -> Excel

`LEN(D6)-LEN(SUBSTITUTE(D6;" ";""))` → broj koliko u stringu ima razmaka

`SUBSTITUTE(D6;" ";"$";LEN(D6)-LEN(SUBSTITUTE(D6;" ";"")))` -> umesto poslednjeg space ubacujem delimiter u ovom slucaju \$

`FIND("$";SUBSTITUTE(D6;" ";"$";LEN(D6)-LEN(SUBSTITUTE(D6;" ";""))))` -> brojimo koliko ima karaktera do delimitera

`LEN(D6)-FIND("$";SUBSTITUTE(D6;" ";"$";LEN(D6)-LEN(SUBSTITUTE(D6;" ";""))))` -> razlika mi pokazuje broj karaktera od delimitera do kraja

`RIGHT(D6;LEN(D6)-FIND("$";SUBSTITUTE(D6;" ";"$";LEN(D6)-LEN(SUBSTITUTE(D6;" ";""))))` -> Prikazujemo karaktere posle poslednjeg razmaka

STRING FUNKCIJE

INSERT(string, position, number, string2)

Ubacuje string u originalni string na definisanoj poziciji i za određeni broj karaktera koje želimo da promenimo

Ulazni parametri funkcije	Opis
<i>string</i>	Obavezan. Originalni string
<i>position</i>	Obavezan. Pozicija na koju da se ubacuje <i>string2</i>
<i>number</i>	Obavezan. Broj karaktera koji se menja
<i>string2</i>	Obavezan. String koji se ubacuje

```
select insert('Visoka Tehnicka Skola',7,1,'-');
```

- Ukoliko je pozicija izvan opsega stringa, funkcija vraća originalan string
- Ukoliko je broj karaktera veći od dužine ostatka stringa, funkcija menja string od date pozicije do kraja stringa

STRING FUNKCIJE

INSTR(string1, string2)

Vraća **poziciju** prve pojave stringa u drugom stringu

Pretraga pravi razliku između velikih i malih slova

Ulazni parametri funkcije	Opis
<i>string1</i>	Obavezan. Originalni string koji se pretražuje
<i>string2</i>	Obavezan. String koji se traži u originalnom stringu

```
SELECT INSTR("www.vtsnis.edu.rs", "rs");
```

STRING FUNKCIJE

REPLACE(string, from_string, new_string)

Zamenjuje sva ponavljanja substringa sa novim stringom

Funkcija pravi razliku između malih i velikih slova

Ulazni parametri funkcije	Opis
string	Obavezan. Originalni string
from_string	Obavezan. Tekst koji se menja
new_string	Obavezan. Novi tekst koji se ubacuje

```
SELECT REPLACE("XYZ FGH XYZ", "X", "m");
```

STRING FUNKCIJE

STRCMP(string1, string2)

Upoređuje dva stringa

If string1 = string2, this function returns 0

If string1 < string2, this function returns -1

If string1 > string2, this function returns 1

STRING FUNKCIJE

FORMAT(number, decimal_places)

Formatira broju u obliku "#,###,###.##", zaokružuje na definisan broj decimala

Vraća rezultat u string formatu

Ulazni parametri funkcije	Opis
<i>number</i>	Obavezan. Broj koji se formatira
<i>decimal places</i>	Obavezan. Broj decimala koji se prikazuje. Ukoliko je 0 funkcija vraća string bez decimalnog prikaza

```
SELECT FORMAT(250500.5634, 0);
```

NUMERIČKE FUNKCIJE

CEIL(number)=CEILING(number)

Vraća prvu veću celobrojnu vrednost

```
SELECT CEIL(25,45);
```

AVG(expression)

Vraća srednju vrednost izraza

Null polja se ingorišu

COUNT(expression)

Vraća broj redova u iskazu

Null polja se ingorišu

NUMERIČKE FUNKCIJE

x DIV y

Vraća celobrojnu vrednost nakon deljenja

```
SELECT 8 DIV 3
```

FLOOR(number)

Vraća prvu manju celobrojnu vrednost

MOD(x, y)

Vraća ostatak

```
SELECT MOD(8,3);
```

NUMERIČKE FUNKCIJE

RAND(seed)

Vraća slučajnu vrednost između 0 (uključena) i 1 (isključena)

Seed je opcioni parametar i uvek vraća istu vrednost slučajnog broja

```
SELECT FLOOR(RAND()*(10-5+1)+5);
```

ROUND(number, decimals)

Vraća zaokružen broj na definisan broj cifara

Ulazni parametri funkcije	Opis
<i>number</i>	Obavezan. Broj koji se zaokružuje
<i>decimals</i>	Opcioni. Na koliko decimalnih mesta se zaokružuje. Ukoliko je izostavljen, vraća integer (bez decimala)

```
SELECT ImeProizvoda, Cena, ROUND(Cena, 1) AS ZaokruzenaCena  
FROM Proizvodi;
```

NUMERIČKE FUNKCIJE

SIGN(number)

Vraća znak za broj

Ako je broj > 0 , vraća 1

Ako je broj $= 0$, vraća 0

Ako je broj < 0 , vraća -1

TRUNCATE(number, decimals)

Vraća skraćenu (odsečenu) vrednost na uneti broj decimala

```
SELECT TRUNCATE(345.156, 0);
```

DATUMSKE FUNKCIJE

ADDDATE(date, INTERVAL value addunit) = ADDDATE(date, days)

Dodaje ili oduzima vreme ili datum na postojeći datum i **vraća novi datum**

Ulazni parametri funkcije	Opis
<i>date</i>	Obavezan. Datum koji se menja
<i>days</i>	Obavezan. Broj dana koji se dodaje datumu
<i>value</i>	Obavezan. Vrednost time/date interval koji se dodaje. Pozitivne i negativne vrednosti su dozvoljene
<i>addunit</i>	<ul style="list-style-type: none">• Obavezan. Tip intervala koji se dodaje.• <i>MICROSECOND SECOND MINUTE HOUR</i>• <i>DAY WEEK MONTH QUARTER YEAR</i>• <i>SECOND_MICROSECOND MINUTE_MICROSECOND MINUTE_SECOND</i>• <i>HOUR_MICROSECOND HOUR_SECOND HOUR_MINUTE</i>• <i>DAY_MICROSECOND DAY_SECOND DAY_MINUTE DAY_HOUR YEAR_MONTH</i>

```
SELECT ADDDATE("2017-06-15 09:34:21", INTERVAL 15 MINUTE);
```

```
SELECT ADDDATE("2017-06-15", INTERVAL -3 DAY);
```

DATUMSKE FUNKCIJE

CURDATE() = CURRENT_DATE()

Vraća trenutni datum

Datum se vraća u formatu YYYY-MM-DD (string) ili YYYYMMDD (broj)

Zadatak

Na trenutni datum dodati 2 meseca

```
SELECT ADDDATE(CURDATE(), INTERVAL 2 MONTH);
```

CURRENT_TIME()

Vraća trenutno vreme

Vreme se vraća u formatu HH-MM-SS (string) ili HHMMSS.uuuuuu (broj)

DATUMSKE FUNKCIJE

DATE(expression)

Vadi samo datum iz date/time vrednosti

Datum se vraća u formatu YYYY-MM-DD (string) ili YYYYMMDD (broj)

Vraća null ukoliko iskaz nije date ili date/time

DATEDIFF(date1, date2)

Vraća broj dana između dva datuma (date1 – date2)

```
SELECT DATEDIFF("2017-01-01", "2016-12-24");
```

FROM_DAYS(number of days)

Vraća koliko ima godina, meseca i dana u unetom broju dana

```
SELECT FROM_DAYS(DATEDIFF("2017-01-01", "2016-12-24"));
```

DATUMSKE FUNKCIJE

DATE_FORMAT(date, format)

Formatira prikaz datuma

Format	Opis
%a	Skraćuje prikaz dana u nedelji (Sun to Sat)
%b	Skaćuje prikaz meseca u godini (Jan to Dec)
%c	Numerički prikaz meseci(0 to 12)
%D	Numerički prikaz dana uz dodati sufiks (1st, 2nd, 3rd, ...)
%d	Numerički prikaz dana (01 to 31)
%e	Numerički prikaz dana (0 to 31)
%f	Microseconds (000000 to 999999)
%H	Sati (00 to 23)
%h	Sati (00 to 12)
%I	Sati (00 to 12)
%i	Minuti (00 to 59)
%j	Dan u godini (001 to 366)
%k	Sati (0 to 23)
%l	Sati (1 to 12)
%M	Puno ime meseca (January to December)
%m	Numerički prikaz meseca (00 to 12)

Format	Opis
%p	AM or PM
%r	Time in 12 hour AM or PM format (hh:mm:ss AM/PM)
%S	Seconds (00 to 59)
%s	Seconds (00 to 59)
%T	Vreme 24h format (hh:mm:ss)
%U	Week where Sunday is the first day of the week (00 to 53)
%u	Week where Monday is the first day of the week (00 to 53)
%V	Week where Sunday is the first day of the week (01 to 53). Used with %X
%v	Week where Monday is the first day of the week (01 to 53). Used with %X
%W	Weekday name in full (Sunday to Saturday)
%w	Day of the week where Sunday=0 and Saturday=6
%X	Year for the week where Sunday is the first day of the week. Used with %V
%x	Year for the week where Monday is the first day of the week. Used with %V
%Y	Godina , 4-digit value
%y	Godina, 2-digit value

```
SELECT DATE_FORMAT("2017-06-15", "%d %M %Y");
```

DATUMSKE FUNKCIJE

DATE_SUB(date, INTERVAL value interval)

Oduzima od trenutnog datuma zadat interval

Ulazni parametri funkcije	Opis
<i>date</i>	Obavezan. Datum koji se menja
<i>value</i>	Obavezan. Vrednost time/date interval koji se dodaje. Pozitivne i negativne vrednosti su dozvoljene. Ukoliko je pozitivna vrednost radi se oduzimanje a kod negativne dodavanje
<i>interval</i>	<ul style="list-style-type: none">•Obavezan. Tip intervala koji se dodaje.• <i>MICROSECOND SECOND MINUTE HOUR</i>• <i>DAY WEEK MONTH QUARTER YEAR</i>• <i>SECOND_MICROSECOND MINUTE_MICROSECOND MINUTE_SECOND</i>• <i>HOUR_MICROSECOND HOUR_SECOND HOUR_MINUTE</i>• <i>DAY_MICROSECOND DAY_SECOND DAY_MINUTE DAY_HOUR YEAR_MONTH</i>

```
SELECT DATE_SUB("2017-06-15 09:34:21", INTERVAL 3 HOUR);
```


DATUMSKE FUNKCIJE

DAY(date) = DAYOFMONTH(date)

Prikazuje dan iz datuma

```
SELECT DAY(CURDATE());
```

MONTH(date)

Prikazuje mesec iz datuma

```
SELECT MONTH(CURDATE());
```

YEAR(date)

Prikazuje mesec iz datuma

```
SELECT YEAR(CURDATE());
```

DATUMSKE FUNKCIJE

DAYNAME(date)

Prikazuje naziv dana iz datuma

```
SELECT DAYNAME(CURDATE());
```

MONTHNAME(date)

Prikazuje naziv meseca iz datuma

```
SELECT MONTHNAME(CURDATE());
```

DATUMSKE FUNKCIJE

WEEKDAY(date)

Vraća INDEKS za svaki dan u nedelji

0=Ponedeljak, 1=Utorak, 2=Sreda, 3=Četvrtak, 4=Petak, 5=Subota, 6=Nedelja

Zadatak

Prikazati za svaki datum, datum koji je prvi ponedeljak u tom datumu

```
SELECT DATE_FORMAT(BirthDate- WEEKDAY(BirthDate), "%a %m %Y")  
from Employees
```

DATUMSKE FUNKCIJE

DAYOFYEAR(date)

Vraća dan (1 – 366) u godini za uneti datum

```
SELECT DAYOFYEAR(CURDATE());
```

DATUMSKE FUNKCIJE

EXTRACT(part FROM date)

Prikazuje željeni deo iz unetog datuma

Ulazni Parametri Funkcije	Opis
<i>part</i>	<ul style="list-style-type: none">•SECOND•MINUTE•HOUR•DAY•WEEK•MONTH•QUARTER•YEAR

Ulazni Parametri Funkcije	Opis
<i>part</i>	<ul style="list-style-type: none">•SECOND_MICROSECOND•MINUTE_MICROSECOND•MINUTE_SECOND•HOUR_MICROSECOND•HOUR_SECOND•HOUR_MINUTE•DAY_MICROSECOND•DAY_SECOND•DAY_MINUTE•DAY_HOUR•YEAR_MONTH
<i>date</i>	Obavezan. Uneti datum

```
SELECT EXTRACT(WEEK FROM "2017-06-15");
```

DATUMSKE FUNKCIJE

PERIOD_ADD(period, number)

Dodaje željeni broj meseci za uneti period

Vraća rezultat u formatu YYYYMM

Ulazni Parametri Funkcije	Opis
<i>period</i>	Obavezan. Format za period YMMM ili YYYYMM
<i>number</i>	Obavezan. Broj meseci koji se dodaje na period. Dozvoljene su pozitivne i negativne vrednosti

```
SELECT PERIOD_ADD(CURDATE(), -2);
```

DATUMSKE FUNKCIJE

PERIOD_DIFF(period1, period2)

Vraća razliku između dva perioda

Rezultat je u **mesecima** i oba perioda moraju da koriste isti format

Ulazni Parametri Funkcije	Opis
<i>period1</i>	Obavezan. Format za prvi period YYMM ili YYYYMM
<i>period2</i>	Obavezan. Format za drugi period YYMM ili YYYYMM

```
SELECT PERIOD_DIFF(Extract(YEAR_MONTH FROM curdate()),201803);
```

DATUMSKE FUNKCIJE

TIMESTAMPDIFF(unit,datetime_expr1,datetime_expr2)

Vraća razliku između dva perioda expr2-expr1

Rezultat je vrednost nakon oduzimanja dva datuma

Jedan iskaz može da bude date a drugi datetime

Unit može da bude:

FRAC_SECOND (microseconds), SECOND, MINUTE, HOUR, DAY, WEEK, MONTH, QUARTER ili YEAR

```
SELECT TIMESTAMPDIFF(MONTH, '2019-05-18', '2019-07-29');
```


DATUMSKE FUNKCIJE

STR_TO_DATE(string, format)

Vraća datum na osnovu string vrednosti i formata

Ulazni Parametri Funkcije	Opis
<i>string</i>	Required. The string to be formatted to a date
<i>format</i>	Required. The format to use. Can be one or a combination of the following values:
Format	Description
%a	Abbreviated weekday name (Sun to Sat)
%b	Abbreviated month name (Jan to Dec)
%c	Numeric month name (0 to 12)
%D	Day of the month as a numeric value, followed by suffix (1st, 2nd, 3rd, ...)
%d	Day of the month as a numeric value (01 to 31)
%e	Day of the month as a numeric value (0 to 31)
%f	Microseconds (000000 to 999999)
%H	Hour (00 to 23)
%h	Hour (00 to 12)
%l	Hour (00 to 12)
%i	Minutes (00 to 59)
%j	Day of the year (001 to 366)
%k	Hour (0 to 23)
%l	Hour (1 to 12)

Ulazni Parametri Funkcije	Opis
%M	Month name in full (January to December)
%m	Month name as a numeric value (00 to 12)
%p	AM or PM
%r	Time in 12 hour AM or PM format (hh:mm:ss AM/PM)
%S	Seconds (00 to 59)
%s	Seconds (00 to 59)
%T	Time in 24 hour format (hh:mm:ss)
%U	Week where Sunday is the first day of the week (00 to 53)
%u	Week where Monday is the first day of the week (00 to 53)
%V	Week where Sunday is the first day of the week (01 to 53). Used with %X
%v	Week where Monday is the first day of the week (01 to 53). Used with %X
%W	Weekday name in full (Sunday to Saturday)
%w	Day of the week where Sunday=0 and Saturday=6
%X	Year for the week where Sunday is the first day of the week. Used with %V
%x	Year for the week where Monday is the first day of the week. Used with %V
%Y	Year as a numeric, 4-digit value
%y	Year as a numeric, 2-digit value

```
SELECT STR_TO_DATE("Monday, August 14, 2017", "%W %M %e %Y");
```

USLOVNE FUNKCIJE

CASE STRUKTURA

Simple Case

```
CASE vrednost  
  WHEN vrednost1 THEN rezultat1  
  WHEN vrednost2 THEN rezultat2  
  ...  
  [ELSE else_result]  
END
```

Searched Case

```
CASE  
  WHEN uslov1 THEN rezlutat1  
  WHEN uslov2 THEN rezlutat2  
  WHEN uslovN THEN rezlutatN  
  ELSE rezlutat  
END;
```

USLOVNE FUNKCIJE

SIMPLE CASE STRUKTURA

```
SELECT OrderID, Status,  
CASE Status  
    WHEN 1 THEN "Odobren"  
    WHEN 2 THEN "U obradi"  
    ELSE "Nije odobren"  
END  
FROM OrderDetails;
```

SEARCHED CASE STRUKTURA

```
SELECT OrderID, Quantity,  
CASE  
    WHEN Quantity >=0 AND Quantity < 10 THEN "Kolicina izmedju 0 i 9"  
    WHEN Quantity >= 10 AND Quantity <= 20 THEN "Kolicina izmedju 10 i 20"  
    ELSE "Kolicina veca od 20"  
END AS Kolicina  
FROM OrderDetails;
```

Šema tabele Books

```
CREATE TABLE books
```

```
(
```

```
    book_id INT NOT NULL AUTO_INCREMENT,
```

```
    title VARCHAR(100),
```

```
    author_fname VARCHAR(100),
```

```
    author_lname VARCHAR(100),
```

```
    released_year INT,
```

```
    stock_quantity INT,
```

```
    pages INT,
```

```
    PRIMARY KEY(book_id)
```

```
);
```

Case Iskaz

1. Prikazati naslove knjiga i opis za svaku knjigu i to tako da za izdanja posle 2000 stoji *Modern* dok za izdanja pre 2000 stoji *20th Century*

```
SELECT title, released_year,  
       CASE  
         WHEN released_year >= 2000 THEN 'Modern'  
         ELSE '20th Century Lit'  
       END AS GENRE  
FROM books;
```

2. Prikazati opisno u formi zvezdica(*) količinu knjiga na stanju i to tako da ukoliko je količina između 0 i 50 se upiše 1 zvezdica, ukoliko je količina između 51 i 100 se upišu 2 zvezdice a za sve ostale vrednosti 3 zvezdice

```
SELECT title,  
       CASE  
         WHEN stock_quantity BETWEEN 0 AND 50 THEN '*'  
         WHEN stock_quantity BETWEEN 51 AND 100 THEN '**'  
         ELSE '***'  
       END AS STOCK  
FROM books;
```

Case Iskaz

3. Prikazati naslove, ime autora i opis i to tako da za naslove koje sadrže *stories* stoji *Short Stories* dok za *Just Kids* ili nalove koji počinju sa *A Heartbreaking* stoji *Memoir* dok za sve ostale *Novel*

```
SELECT title, author_lname,  
CASE  
    WHEN title LIKE '%stories%' THEN 'Short Stories'  
    WHEN title = 'Just Kids' OR title = 'A Heartbreaking %' THEN  
'Memoir'  
    ELSE 'Novel'  
END AS TYPE  
FROM books;
```

4. Prikazati ime i prezime autora kao i broj napisanih knjiga i to tako da pored svakog broja stoji *book* ili *books* u zavisnosti da li je autor napisao jednu ili više knjiga

```
SELECT author_fname, author_lname,  
CASE  
    WHEN COUNT(*) = 1 THEN '1 book'  
    ELSE CONCAT(COUNT(*), ' books')  
END AS COUNT  
FROM books  
GROUP BY author_lname, author_fname;
```

USLOVNE FUNKCIJE

CASE STRUKTURA U AGREGACIONIM FUNKCIJAMA

```
SELECT
  SUM(CASE WHEN status = 'Shipped' THEN 1 ELSE 0
  END) AS 'Shipped',
  SUM(CASE WHEN status = 'On Hold' THEN 1 ELSE 0
  END) AS 'On Hold',
  SUM(CASE WHEN status = 'In Process' THEN 1 ELSE 0
  END) AS 'In Process',
  SUM(CASE WHEN status = 'Resolved' THEN 1 ELSE 0
  END) AS 'Resolved',
  SUM(CASE WHEN status = 'Cancelled' THEN 1 ELSE 0
  END) AS 'Cancelled',
  COUNT(*) AS Total
FROM
  orders;
```

```
SELECT shop_id,
  SUM(CASE WHEN currency= "GBP" THEN price ELSE 0 END) AS POUND,
  SUM(CASE WHEN currency= "USD" THEN price ELSE 0 END) AS USDOLLAR
GROUP BY shop_id
```

USLOVNE FUNKCIJE ZA KONTROLU TOKA

IF STRUKTURA

IF(condition, value_if_true, value_if_false)

```
SELECT OrderID, Quantity, IF(Quantity>10, "MORE", "LESS")  
FROM OrderDetails;
```


FUNKCIJE ZA KONTROLU TOKA

IF STRUKTURA

book_id	book_name	isbn_no	cate_id	aut_id	pub_id	dt_of_pub	pub_lang	no_page	book_price
BK001	Introduction to Electrodynamics	0000979001	CA001	AUT001	P003	2001-05-08	English	201	85.00
BK002	Understanding of Steel Construction	0000979002	CA002	AUT002	P001	2003-07-15	English	300	105.50
BK003	Guide to Networking	0000979003	CA003	AUT003	P002	2002-09-10	Hindi	510	200.00
BK004	Transfer of Heat and Mass	0000979004	CA002	AUT004	P004	2004-02-16	English	600	250.00
BK005	Conceptual Physics	0000979005	CA001	AUT005	P006	2003-07-16	NULL	345	145.00
BK006	Fundamentals of Heat	0000979006	CA001	AUT006	P005	2003-08-10	German	247	112.00
BK007	Advanced 3d Graphics	0000979007	CA003	AUT007	P002	2004-02-16	Hindi	165	56.00
BK008	Human Anatomy	0000979008	CA005	AUT008	P006	2001-05-17	German	88	50.50
BK009	Mental Health Nursing	0000979009	CA005	AUT009	P007	2004-02-10	English	350	145.00
BK010	Fundamentals of Thermodynamics	0000979010	CA002	AUT010	P007	2002-10-14	English	400	225.00

Uslov

Ako je uslov
ispunjen

Ako uslov nije
ispunjen ispunjen

```
SELECT book_name, isbn_no,  
IF(  
  (SELECT COUNT(*) FROM book_mast WHERE pub_lang='English') >  
  (SELECT COUNT(*) FROM book_mast WHERE pub_lang<>'English'),  
  (CONCAT("Pages: ", no_page)),  
  (CONCAT("Price: ", book_price))) AS "Page / Price"  
FROM book_mast;
```

FUNKCIJE ZA KONTROLU TOKA

IF STRUKTURA

book_id	book_name	isbn_no	cate_id	aut_id	pub_id	dt_of_pub	pub_lang	no_page	book_price
BK001	Introduction to Electrodynamics	0000979001	CA001	AUT001	P003	2001-05-08	English	201	85.00
BK002	Understanding of Steel Construction	0000979002	CA002	AUT002	P001	2003-07-15	English	300	105.50
BK003	Guide to Networking	0000979003	CA003	AUT003	P002	2002-09-10	Hindi	510	200.00
BK004	Transfer of Heat and Mass	0000979004	CA002	AUT004	P004	2004-02-16	English	600	250.00
BK005	Conceptual Physics	0000979005	CA001	AUT005	P006	2003-07-16	NULL	345	145.00
BK006	Fundamentals of Heat	0000979006	CA001	AUT006	P005	2003-08-10	German	247	112.00
BK007	Advanced 3d Graphics	0000979007	CA003	AUT007	P002	2004-02-16	Hindi	165	56.00
BK008	Human Anatomy	0000979008	CA005	AUT008	P006	2001-05-17	German	88	50.50
BK009	Mental Health Nursing	0000979009	CA005	AUT009	P007	2004-02-10	English	350	145.00
BK010	Fundamentals of Thermodynamics	0000979010	CA002	AUT010	P007	2002-10-14	English	400	225.00

```
SELECT book_id, book_name,  
       IF(pub_lang IS NULL, 'N/A', pub_lang) AS "Pub. Language"  
FROM book_mast;
```

FUNKCIJE ZA KONTROLU TOKA

IF STRUKTURA

book_id	book_name	isbn_no	cate_id	aut_id	pub_id	dt_of_pub	pub_lang	no_page	book_price
BK001	Introduction to Electrodynamics	0000979001	CA001	AUT001	P003	2001-05-08	English	201	85.00
BK002	Understanding of Steel Construction	0000979002	CA002	AUT002	P001	2003-07-15	English	300	105.50
BK003	Guide to Networking	0000979003	CA003	AUT003	P002	2002-09-10	Hindi	510	200.00
BK004	Transfer of Heat and Mass	0000979004	CA002	AUT004	P004	2004-02-16	English	600	250.00
BK005	Conceptual Physics	0000979005	CA001	AUT005	P006	2003-07-16	NULL	345	145.00
BK006	Fundamentals of Heat	0000979006	CA001	AUT006	P005	2003-08-10	German	247	112.00
BK007	Advanced 3d Graphics	0000979007	CA003	AUT007	P002	2004-02-16	Hindi	165	56.00
BK008	Human Anatomy	0000979008	CA005	AUT008	P006	2001-05-17	German	88	50.50
BK009	Mental Health Nursing	0000979009	CA005	AUT009	P007	2004-02-10	English	350	145.00
BK010	Fundamentals of Thermodynamics	0000979010	CA002	AUT010	P007	2002-10-14	English	400	225.00

```
SELECT SUM(IF(pub_lang = 'English', 1, 0)) AS English,  
        SUM(IF(pub_lang <> 'English', 1, 0)) AS "Non English"  
FROM purchase;
```

FUNKCIJE ZA KONTROLU TOKA

IFNULL

Vraća željenu vrednost ukoliko je iskaz Null u suprotnom vraća vrednost iskaza

```
IFNULL(expression, alt_value)
```

Upisuje vrednosti iz kolone part_id, ukoliko je vrednost u koloni part_id null, tada će upisati vrednost iz kolone catalog_id

```
SELECT IFNULL(part_id, catalog_id) as identifier from parts;
```

=

```
SELECT IF(part_id is not null, part_id, catalog_id) as identifier from parts
```

FUNKCIJA ZA KONVERZIJU TIPA PODATAKA

CAST

Ukoliko u jednoj koloni želimo da prikazemo tekstualni i numerički podataka potrebno je da numerički podatak konvertujem u string

```
select concat("Number of books: ", cast(count(*) as char)) from books;
```

Zadatak

Matematičar Kurt Godel je rođen Aprila 28, 1906 a preminuo je u Januaru 14, 1978

- Kog dana u nedelji je rođen i kog dana u nedelji je preminuo
- Koliko godina je bio star (broj godina, broj meseca i broj dana)
- Da je živeo još 99 godina, 6 meseca i 3 dana kog datuma bi preminuo

A)

```
set @born = "April 28, 1906";  
set @died = "January 14, 1978";
```

```
select dayname(str_to_date(@born, "%M %d, %Y"));  
select dayname(str_to_date(@died, "%M %d, %Y"));
```

B)

```
select from_days(datediff(str_to_date(@died, "%M %d, %Y"), str_to_date(@born, "%M %d, %Y")));
```

C)

```
select @newdied := str_to_date(@born, "%M %d, %Y") + interval 99 year + interval 6 month + interval 3 day;
```

Korisničke definisane funkcije

- Sačuvana funkcija (*Stored function*) je poseban tip sačuvanog programa koji **vraća samo jednu vrednost**.
- Kreirane funkcije se obično koriste da enkapsuliraju (zatvore) opšte formule ili pravila u poslovanju koja se često pozivaju kroz SQL iskaze, najčešće u select iskazu
- U funkcijama ne mogu da se koriste kursori, transakcije, ažuriranje tabela i pisanje agregatnih funkcija kao što je npr. sum funkcija
- Funkcija sadrži samo ulazne (in) parametre, ne sadrži out parametre što je slučaj sa procedurama

```
DELIMITER $$  
CREATE FUNCTION function_name( param1, param2,... ) RETURNS datatype [NOT] DETERMINISTIC  
BEGIN  
-- statements  
END $$  
DELIMITER ;
```

Struktura funkcije

- Prvi korak je definsanje naziva za funkciju
- Drugi korak je lista ulaznih parametara
- Treći korak je tip podataka za vrednost koju funkcija vraća
- Četvri korak je izbor determinističke funkcije koja uvek vraća isti rezultat za iste ulazne parametre dok nedeterministička (default) funkcija vraća različit rezultat za iste ulazne parametre
- Peti korak je logika funkcije koja mora da sadrži RETURN iskaz unutar BEGIN i END bloka
 - Return iskaz vraća vrednost i izlazi se iz funkcije

```
DELIMITER $$  
  
CREATE FUNCTION function_name( param1, param2,... ) RETURNS datatype [NOT] DETERMINISTIC  
  
BEGIN  
  
-- statements  
  
END $$  
  
DELIMITER ;
```


Struktura funkcije



```
DELIMITER $$
```

SQL

```
CREATE FUNCTION function_name(parameter data_type) RETURNS data_type
```

```
DECLARE variable_name data_type
```



```
BEGIN
```

```
    SELECT ...
```

```
    RETURN variable_name
```

```
END$$
```

```
DELIMITER ;
```



here you have no OUT parameters to define between the parentheses after the object's name

all parameters are IN, and since this is well known, you need not explicitly indicate it with the word, 'IN'



```
DELIMITER $$
```

SQL

```
CREATE FUNCTION function_name(parameter data_type) RETURNS data_type
```

```
DECLARE variable_name data_type
```



```
BEGIN
```

```
    SELECT ...
```

```
    RETURN variable_name
```

```
END$$
```

```
DELIMITER ;
```



although there are no OUT parameters, there is a 'return value'

it is obtained after running the query contained in the body of the function

Primer korisničke funkcije

- Kreirati funkciju za bazu **ig_clone** koja će na osnovu username korisnika da vrati broj slika koje je taj korisnik postavio

```
select username, broj_slika(username)
from users;
```

```
delimiter $$
create function broj_slika(v_username varchar(40))
returns integer
deterministic
begin
declare v_broj_slika integer;
select count(*) into v_broj_slika
from users inner join photos on users.id=photos.user_id
where username=v_username;
return v_broj_slika;
end$$
delimiter ;
```

Primer korisničke funkcije

- Kreirati funkciju koja će na osnovu unete vrednosti za cenu da vrati jednu od tri vrednosti
 - Jeftino ukoliko je cena manja od 3\$
 - Povoljno ukoliko je cena između 3\$ i 8\$
 - Skupo ukoliko je cena veća od 8\$

```
SHOW FUNCTION STATUS
WHERE db = 'online_shop';
```

```
select name, price, opis_cene(price)
from products;
```

```
delimiter $$
create function opis_cene(price decimal(10,2))
returns varchar(20)
deterministic
begin
declare opis varchar(20);
if price < 3
then set opis = 'Jeftino';
Else if price < 7
then set opis = 'Povoljno';
Else set opis='Skupo';
end if;
return (opis);
end$$
delimiter ;
```