

levads datu bāzēs

Roberts Polis
Vjačeslavs Pēteris



LATVIJAS UNIVERSITĀTE
**BIZNESA, VADĪBAS
UN EKONOMIKAS
FAKULTĀTE**



VADĪBAS UN
UZNĒMĒJDARBĪBAS
MĀCĪBU CENTRS

ESF projekts Nr. 8.4.1.0/16/I/001
"Nodarbināto personu profesionālās kompetences pilnveide"

NACIONĀLAIS
ATTĪSTĪBAS
PLĀNS 2020



EIROPAS SAVIENĪBA
Eiropas Sociālais
fonds

Funkcijas.

8. lekcija



Šodienas lekcijā

8.1. Funkciju tipi.

8.2. Funkciju lietošana.

8. patstāvīgais mājas darbs

Dota jautājumu tīmekļa forma. Aizpildīt šo formu un iesūtīt. Jautājumi par funkciju lietošanu.

Izpildīt uzdevumu ar SQL darba vidi (Workbench).

Iebūvēto (String), un (Temporal) funkciju lietošana SQL pieprasījumos.

Iebūvēto skaitlisko un vadības kontroles funkciju lietošana SQL pieprasījumos.

Agregātfunkciju lietošana.

Funkcijas MySQL izteiksmēs

- Funkcijas veic aprēķinus ar datiem.
- Funkcijas atgriezto vērtību var izmantot SQL izteiksmēs.
- Sintakse:
`function_name([<arg1> [, <arg2>, ..., <argn>]])`
- Jālieto iekavas arī tad ja nav argumentu..
- Atdalīt vērtības ar komatiem.
- Piemēri:
`SELECT NOW():` Atgriež tekošo datumu un laiku.
`SELECT VERSION():` Atgriež MySQL Servera versiju

Funkciju izmantošana

- Funkcijas var izmantot visur kur drīkst lietot vērtību izteiksmes.
- Lielākoties funkcijām nepieciešami argumenti.
- Kolonas (ievērojot datu tipus) var izmantot kā funkciju argumentus.
- Vienas funkcijas rezultātu var izmantot kā argumentu citai funkcijai.
- **Izteiksme ar NULL vērtību vienmēr rada NULL rezultātu.**
 - Ar retiem izņēmumiem
- Matemātiskās funkcijas atgriež 0 kļūdas gadījumā.
- Piemēram, dalīšana ar 0.

Funkciju tipi

- **String:** Darbības ar teksta rindām
- **Temporal:** Darbības ar datumu un laiku
- **Numeric:** Matemātiskās darbības
- **Vadības kontrole:** Vērtības izvēle balstoties uz izteiksmes vērtības.
- **Aggregate:** atgriež vienu vērtību ierakstu grupai

String[teksta] Funkcijas

Veic tādas darbības kā:

- Aprēķina teksta garumu
- Ekstraktē teksta daļas
- Meklē vai aizvieto apakšrindas
- Veic augš/apakš. reģistra (upperCase/lowerCase) pārveidojumus

String funkcijas pēc atgriežamā tipa iedala 2 kategorijās:

- Numeric: Atgriež skaitli
- String: Atgriež strings

Teksta funkcijas

```
mysql> select length(_utf8 '€'), char_length(_utf8 '€');
```

length(_utf8 '€')	char_length(_utf8 '€')
3	1

```
1 row in set, 2 warnings (0.01 sec)
```

```
mysql> SELECT INSTR('MySQL', 'SQL');
```

INSTR('MySQL', 'SQL')
3

```
1 row in set (0.00 sec)
```

```
mysql> SELECT STRCMP('abc','def'),
```

```
-> STRCMP('def','def'),
```

```
-> STRCMP('def','abc');
```

STRCMP('abc','def')	STRCMP('def','def')	STRCMP('def','abc')
-1	0	1

```
1 row in set (0.00 sec)
```

- LENGTH un CHAR_LENGTH atgriež teksta rindas garumu.
- INSTR atgriež pirmās rindas pozīciju iekš otrās
- STRCMP salīdzina 2 rindas un atgriež:
 - >0 pirmā rinda lielāka
 - =0 abas rindas vienādas
 - <0 pirmā rinda mazāka

Funkcijas: CONCAT, REVERSE, LEFT, RIGHT

```
mysql> SELECT CONCAT('A', '-', 'Z');
```

```
+-----+
| CONCAT('A', '-', 'Z') |
+-----+
|          A-Z          |
+-----+
1 row in set (0.01 sec)
```

```
mysql> SELECT REVERSE('MySQL');
```

```
+-----+
| REVERSE('MySQL') |
+-----+
|          LQSyM    |
+-----+
1 row in set (0.00 sec)
```

```
mysql> SELECT LEFT('MySQL', 3);
```

```
+-----+
| LEFT('MySQL', 3) |
+-----+
|          MyS     |
+-----+
1 row in set (0.00 sec)
```

```
mysql> SELECT RIGHT('MySQL', 3);
```

```
+-----+
| RIGHT('MySQL', 3) |
+-----+
|          SQL      |
+-----+
1 row in set (0.01 sec)
```

- CONCAT savieno vairākas rindas (CONCATENATE)
- REVERSE atgriež simbolus pretēejā virzienā
- LEFT - atgriež n simbolus no kreisās puses
- RIGHT - atgriež n simbolus no labās puses.

Funkcijas: upper, lower, lpad, rpad

```
mysql> SELECT LOWER('MySQL');
+-----+
| LOWER('MySQL') |
+-----+
|      mysql      |
+-----+
1 row in set (0.00 sec)

mysql> SELECT UPPER('MySQL');
+-----+
| UPPER('MySQL') |
+-----+
|     MYSQL      |
+-----+
1 row in set (0.00 sec)

mysql> SELECT LPAD('MySQL', 8, '.');
+-----+
| LPAD('MySQL', 8, '.') |
+-----+
| ...MySQL             |
+-----+
1 row in set (0.01 sec)

mysql> SELECT RPAD('MySQL', 8, '.');
+-----+
| RPAD('MySQL', 8, '.') |
+-----+
| MySQL...             |
+-----+
1 row in set (0.00 sec)
```

- LOWER - atgriež teksta rindu apakšējā reģistrā
- UPPER - atgriež teksta rindu augšējā reģistrā
- LPAD papildina rindu no kreisās puses ar norādīto simbolu skaitu līdz norādītajam garumam.
- RPAD papildina rindu no labās puses ar norādīto simbolu skaitu līdz norādītajam garumam.

Funkcija: trim

```
mysql> SELECT TRIM('  MySQL  ') AS str;
+-----+
| str    |
+-----+
| MySQL  |
+-----+
1 row in set (0.00 sec)

mysql> SELECT TRIM(LEADING 'Q' FROM 'QQQMySQLQQQ');
+-----+
| TRIM(LEADING 'Q' FROM 'QQQMySQLQQQ') |
+-----+
| MySQLQQQ                             |
+-----+
1 row in set (0.00 sec)

mysql> SELECT TRIM('Q' FROM 'QQQMySQLQQQ');
+-----+
| TRIM('Q' FROM 'QQQMySQLQQQ') |
+-----+
| MySQL                         |
+-----+
1 row in set (0.00 sec)
```

- TRIM funkcija izmet norādīto priekšējo, aizmugurējo simbolu virkni.

Funkcija: substring, substring_index

```
mysql> SELECT SUBSTRING('MySQL', 3);
+-----+
| SUBSTRING('MySQL', 3) |
+-----+
| SQL                  |
+-----+
1 row in set (0.00 sec)

mysql> SELECT SUBSTRING('MySQL', 2, 2);
+-----+
| SUBSTRING('MySQL', 2, 2) |
+-----+
| yS                      |
+-----+
1 row in set (0.00 sec)

mysql> SELECT SUBSTRING_INDEX('training@mysql.com', '@', 1);
+-----+
| SUBSTRING_INDEX('training@mysql.com', '@', 1) |
+-----+
| training                                     |
+-----+
1 row in set (0.00 sec)

mysql> SELECT SUBSTRING_INDEX('www.mysql.com', '.', -2);
+-----+
| SUBSTRING_INDEX('www.mysql.com', '.', -2) |
+-----+
| mysql.com                                |
+-----+
1 row in set (0.01 sec)
```

- SUBSTRING funkcija Atgriež apakšrindu.
- No dotā simbola nummura ar simbolu skaitu

Laika funkcijas

- Izpilda kalkulācijas ar datumu/laiku:
 - Datuma/Laika daļu izvilkšana
 - Vērtību pārformatēšana
 - Konvertēt vērtību uz sekundēm vai gadiem
- Laika datus var ģenerēt dažādi:
 - Kopējot esošos datus.
 - Izmantojot iebūvētās funkcijas.
 - Izveidot teksta versiju.

Funkcija: substring, substring_index

```
mysql> SELECT CURDATE(), CURTIME(), DAYNAME(NOW());
+-----+-----+-----+
| CURDATE() | CURTIME() | DAYNAME(NOW()) |
+-----+-----+-----+
| 2021-08-28 | 23:54:16 | Saturday      |
+-----+-----+-----+
1 row in set (0.01 sec)
```

```
mysql> SELECT NOW(), NOW() + INTERVAL 5 DAY;
+-----+-----+
| NOW()          | NOW() + INTERVAL 5 DAY |
+-----+-----+
| 2021-08-28 23:54:28 | 2021-09-02 23:54:28   |
+-----+-----+
1 row in set (0.01 sec)
```

```
mysql> SELECT NOW(), DATE_FORMAT(NOW(), '%W the %D of %M');
+-----+-----+
| NOW()          | DATE_FORMAT(NOW(), '%W the %D of %M') |
+-----+-----+
| 2021-08-28 23:57:59 | Saturday the 28th of August          |
+-----+-----+
1 row in set (0.00 sec)
```

- Tekošais datums, laiks
- Pieskaitīt dienu skaitu

Tips	Noklusētais formāts
DATE	YYYY-MM-DD
TIME	HH:MM:SS
DATETIME	YYYY-MM-DD HH:MM:SS
TIMESTAMP	YYYY-MM-DD HH:MM:SS
YEAR	YYYY

Laika funkcijas

- NOW()
- CURDATE()
- CURTIME()
- YEAR(<date_expression>)
- MONTH(<date_expression>)

DAYOFMONTH(<date_expression>)

DAY(<date_expression>)

DAYNAME(<date_expression>)

HOUR(<date_expression>)

MINUTE(<date_expression>)

SECOND(<date_expression>)

```
mysql> select YEAR(NOW()),MONTH(NOW()),DAYOFMONTH(NOW()),DAY(NOW()),DAYNAME(NOW()),HOUR(NOW()),MINUTE(NOW()),SECOND(NOW());
+-----+-----+-----+-----+-----+-----+-----+-----+
| YEAR(NOW()) | MONTH(NOW()) | DAYOFMONTH(NOW()) | DAY(NOW()) | DAYNAME(NOW()) | HOUR(NOW()) | MINUTE(NOW()) | SECOND(NOW()) |
+-----+-----+-----+-----+-----+-----+-----+-----+
|          2021 |           8 |           29 |          29 |          Sunday |           0 |           5 |           44 |
+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Skaitliskās funkcijas

Veikt skaitliskās darbības kā:

- Noapaļošana
- Precizitātes samazināšana
- Trigonometriskie aprēķini
- Varbūtības skaitļu ģenerēšana

ABS(<number>) - absolūtā skaitļa
vērtība

SIGN(<number>) skaitļa zīme +1, 0, -1

TRUNCATE(<number>, <decimals>)

- nomest sskaitli uz norādīto
decimālo zīmju skaitu

FLOOR(<number>) - tuvākais
veselais skaitlis uz augšu

CEILING(<number>) - tuvākais
veselais skaitlis uz augšu

ROUND(<number>) - noapaļot
skaitli uz norādīto decimālo zīmju

```
mysql> select YEAR(NOW()),MONTH(NOW()),DAYOFMONTH(NOW()),DAY(NOW()),DAYNAME(NOW()),HOUR(NOW()),MINUTE(NOW()),SECOND(NOW());
+-----+-----+-----+-----+-----+-----+-----+-----+
| YEAR(NOW()) | MONTH(NOW()) | DAYOFMONTH(NOW()) | DAY(NOW()) | DAYNAME(NOW()) | HOUR(NOW()) | MINUTE(NOW()) | SECOND(NOW()) |
+-----+-----+-----+-----+-----+-----+-----+-----+
|          2021 |           8 |           29 |       29 |       Sunday |           0 |           5 |           44 |
+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

skaitu

Skaitliskās funkcijas

```
mysql> SELECT ABS(-42), ABS(42);
```

```
+-----+-----+
| ABS(-42) | ABS(42) |
+-----+-----+
|      42 |      42 |
+-----+-----+
1 row in set (0.00 sec)
```

```
mysql> SELECT SIGN(-42), SIGN(-1), SIGN(0), SIGN(1), SIGN(42);
```

```
+-----+-----+-----+-----+-----+
| SIGN(-42) | SIGN(-1) | SIGN(0) | SIGN(1) | SIGN(42) |
+-----+-----+-----+-----+-----+
|      -1 |      -1 |       0 |       1 |       1 |
+-----+-----+-----+-----+-----+
1 row in set (0.01 sec)
```

```
mysql> SELECT ROUND(135.375, 2), TRUNCATE(135.375, 2);
```

```
+-----+-----+
| ROUND(135.375, 2) | TRUNCATE(135.375, 2) |
+-----+-----+
|      135.38 |      135.37 |
+-----+-----+
```

ABS(<number>) - absolūtā skaitļa
vērtība

SIGN(<number>) skaitļa zīme +1, 0, -1

TRUNCATE(<number>, <decimals>)

- nomest sskaitli uz norādīto

decimālo zīmju skaitu

FLOOR(<number>) - tuvākais

veselais skaitlis uz augšu

CEILING(<number>) - tuvākais

veselais skaitlis uz augšu

ROUND(<number>) - noapaļot

skaitli uz norādīto decimālo zīmju

skaitu

Kontroles vadības funkcijas

```
mysql> SELECT IF(1 > 0, 'YES', 'NO');
```

```
+-----+  
| IF(1 > 0, 'YES', 'NO') |  
+-----+  
| YES          |  
+-----+  
1 row in set (0.00 sec)
```

```
mysql> SELECT CASE 3 WHEN 1 THEN 'one'  
-> WHEN 2 THEN 'two'  
-> ELSE 'more' END;
```

```
+-----+  
| CASE 3 WHEN 1 THEN 'one' WHEN 2 THEN 'two' ELSE 'more' END. |  
+-----+  
| more                                                         |  
+-----+  
1 row in set (0.00 sec)
```

IF(izteiksme, patiesa_vērt,
nepatiesa_vērt)

funkcija parāda otro argumentu ja
izteiksme patiesa vai trešo pretējā
gadījumā

CASE value

WHEN <compare_value> THEN
<result>

[WHEN <compare_value> THEN
<result> ...]

[ELSE <result>]

END

Case funkcija

```
mysql> SELECT emp_no,  
-> CASE gender  
-> WHEN 'M' THEN 'Vīrietis'  
-> WHEN 'F' THEN 'Sieviete'  
-> ELSE 'nav zinams'  
-> END as dzimums  
-> FROM employees LIMIT 10;
```

```
+-----+-----+  
| emp_no | dzimums |  
+-----+-----+  
| 10002 | Sieviete |  
| 10003 | Vīrietis |  
| 10004 | Vīrietis |  
| 10005 | Vīrietis |  
| 10006 | Sieviete |  
| 10007 | Sieviete |  
| 10008 | Vīrietis |  
| 10009 | Sieviete |  
| 10010 | Sieviete. |  
| 10011 | Sieviete |  
+-----+-----+  
10 rows in set (0.00 sec)
```

Divu veidu izteiksmes spārbaudes funkcijas

CASE value

WHEN <compare_value> THEN <result>

[WHEN <compare_value> THEN <result> ...]

[ELSE <result>]

END

Case funkcijas piemērs

```
mysql>
mysql> SELECT emp_no,
-> CASE
->   WHEN salary > 70000 THEN 'virs 70 TK'
->   WHEN salary > 60000 THEN 'virs 60 TK'
->   ELSE 'zem 60 TK'
-> END as category
-> FROM salaries LIMIT 10;
```

```
+-----+-----+
| emp_no | category |
+-----+-----+
| 10002 | virs 60 TK |
| 10002 | virs 60 TK |
| 10002 | virs 60 TK |
| 10002 | virs 60 TK |
| 10002 | virs 70 TK |
| 10002 | virs 70 TK |
| 10003 | zem 60 TK |
| 10003 | zem 60 TK |
| 10003 | zem 60 TK |
| 10003 | zem 60 TK |
+-----+-----+
10 rows in set (0.00 sec)
```

Divu veidu izteiksmes spārbaudes funkcijas

CASE

WHEN <condition> THEN <result>

[WHEN <condition> THEN <result> ...]

[ELSE <result>]

END

Agregātfunkcijas

```
mysql> select MIN(salary),MAX(salary),AVG(salary),COUNT(salary) FROM
salaries;
```

```
+-----+-----+-----+-----+
| MIN(salary) | MAX(salary) | AVG(salary) | COUNT(salary) |
+-----+-----+-----+-----+
|          38623 |          155377 |    63810.5123 |          2844025 |
+-----+-----+-----+-----+
1 row in set (0.65 sec)
```

Agregātfunkcijass domātas izmantot kopā ar GROUP BY klausi
GROUP by sadala tabulas ierakstus grupās
un agregātfunkcija rēķina vērtības no grupas;
Ja izmanto bez tās tad visu tabulu uztver par
1 grupu

MIN(<column_name>) - kolonas

minimālā vērtība

MAX(<column_name>) - kolonass

maksimālā vērtība

SUM(<column_name>) - kolonaass

summa

AVG(<column_name>) - vidējā

kolonas vērtība

COUNT(<column_name>) - ierakstu

skaits ar NON NUL: L bvērtībām

GROUP_CONCAT(<column_name>)

- gripaass koncat vērtībaa

Aggregātfunkcijas ar GROUP BY



Ievads datu bāzēs

2021

```
mmysql> select count(*),gender from employees group by gender;
```

```
+-----+-----+
| count(*) | gender |
+-----+-----+
|    120051 | F     |
|    179972 | M     |
+-----+-----+
2 rows in set (0.14 sec)
```

```
mysql> SELECT release_year,rating, AVG(rental_duration) FROM
sakila.film group by release_year,rating;
```

```
+-----+-----+-----+
| release_year | rating | AVG(rental_duration) |
+-----+-----+-----+
|          2006 | PG     |          5.0825 |
|          2006 | G      |          4.8371 |
|          2006 | NC-17  |          5.1429 |
|          2006 | PG-13  |          5.0538 |
|          2006 | R      |          4.7744 |
+-----+-----+-----+
5 rows in set (0.01 sec)
```

```
SELECT release_year,GROUP_CONCAT(rating) ratings, AVG(rental_duration)
FROM sakila.film group by release_year;
```

```
|          2006 | PG,G,NC-17,G,G,PG,PG-13,R,PG-13,NC-17,G,PG,PG,NC-17,NC-
17,NC-173,PG-13,NC-17,R,G,PG,NC-17,G,G,NC-17,R,G,PG,PG-13,NC-17,NC-17 |
4.9850 |
```

MIN(<column_name>) - kolonas
minimālā vērtība

MAX(<column_name>) - kolonass
maksimālā vērtība

SUM(<column_name>) - kolonaass
summa

AVG(<column_name>) - vidējā
kolonas vērtība

COUNT(*) - ierakstu skaits

COUNT(<column_name>) - ierakstu
skaits ar NON NUL:L bvērtībām

GROUP_CONCAT(<column_name>)
- grupas ierakstu salīmēta kolonas
vērtību

GROUP BY un HAVING

```
mysql> SELECT release_year,rating, AVG(rental_duration) FROM  
sakila.film group by release_year,rating;
```

release_year	rating	AVG(rental_duration)
2006	PG	5.0825
2006	G	4.8371
2006	NC-17	5.1429
2006	PG-13	5.0538
2006	R	4.7744

5 rows in set (0.00 sec)

```
mysql> SELECT release_year,rating, AVG(rental_duration) FROM  
sakila.film group by release_year,rating HAVING AVG(rental_duration)>5;
```

release_year	rating	AVG(rental_duration)
2006	PG	5.0825
2006	NC-17	5.1429
2006	PG-13	5.0538

3 rows in set (0.01 sec)

- HAVING ir grupu filtrs
- varam norādīt izteiksmes
bāzētas uz grupas kategorijām
(to kas attiecas uz grupu)
 - Grupas kolonas
 - Agregātfunkcijas

WITH ROLLUP

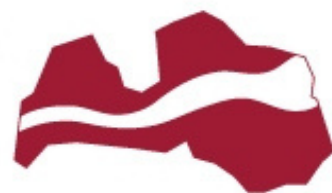
```
mysql> SELECT release_year,rating, AVG(rental_duration) FROM  
sakila.film group by release_year,rating WITH ROLLUP;;
```

```
+-----+-----+-----+  
| release_year | rating | AVG(rental_duration) |  
+-----+-----+-----+  
|          2006 | G      |          4.8371 |  
|          2006 | PG     |          5.0825 |  
|          2006 | PG-13  |          5.0538 |  
|          2006 | R      |          4.7744 |  
|          2006 | NC-17  |          5.1429 |  
|          2006 | NULL   |          4.9850 |  
|          NULL | NULL   |          4.9850 |  
+-----+-----+-----+  
7 rows in set (0.01 sec)
```

- WITH ROLLUP iekļauj augstāka līmeņa agregācijass (grupas apkopojumus ar agregātfunkcijām)
- Kolonass vērtībaa ir NULL rindai kas atbilst ROLLUP.

ESF projekts Nr. 8.4.1.0/16/I/001
"Nodarbināto personu profesionālās kompetences pilnveide"

NACIONĀLAIS
ATTĪSTĪBAS
PLĀNS 2020



EIROPAS SAVIENĪBA
Eiropas Sociālais
fonds