

Laborator 3

Functii. Popularea tabelelor cu date si modificarea acestora in MySQL

MySQL oferă trei categorii de funcții: funcții care returnează un set de înregistrări, funcții agregat și funcții scalare. Funcțiile agregat operează asupra unei colecții de valori, de regulă provenite din una sau mai multe coloane ale unor tabele și returnează o singură valoare agregat. Funcțiile de agregare pot fi folosite exclusiv în cadrul frazelor SELECT și vor fi prezentate în contextul acestora.

Funcțiile scalare operează asupra unei singure valori și returnează tot o singură valoare. O funcție scalară poate fi folosită oriunde este legală plasarea unei expresii. În cele ce urmează sunt prezentate o parte din funcțiile scalare furnizate de către MySQL, restul putând fi regăsite în manualul de utilizare a MySQL.

De menționat este faptul că pentru rularea acestor funcții se utilizează funcția SELECT.

Tabel 1. Funcții pentru manipularea valorilor de tip șir de caractere

Nume	Descriere
ASCII(<i>str</i>)	Returnează codul ASCII al celui mai din stânga caracter al argumentului
BIN(<i>N</i>)	Returnează reprezentarea binară a unui număr
BIT_LENGTH(<i>str</i>)	Returnează lungimea unui argument în biți
CHAR_LENGTH(<i>str</i>), CHARACTER_LENGTH(<i>str</i>)	Returnează numărul de caractere dintr-un argument
CONCAT_WS(<i>separator, str1, str2, ...</i>)	Returnează un șir de caractere concatenate, fiind specificat și un separator
CONCAT(<i>str1, str2, ...</i>)	Returnează un șir de caractere concatenat
ELT(<i>N, str1, str2, str3, ...</i>)	Returnează al N-lea element dintr-o listă de șiruri de caractere; returnează NULL dacă N este mai mic ca 1 sau mai mare decât numărul de argumente
FIELD(<i>str, str1, str2, str3, ...</i>)	Returnează indexul (poziția) primului argument din lista de argumente ce urmează după acesta
FIND_IN_SET(<i>str, strlist</i>)	Returnează poziția primului argument în cel de al doilea argument
INSERT(<i>str, pos, len, newstr</i>)	Inserează un substring într-o poziție specificată pe un număr specificat de caractere
INSTR(<i>str, substr</i>)	Returnează poziția primeni apariții a unui substring
LCASE(<i>str</i>), LOWER(<i>str</i>)	Șirul de caractere rezultat în urma transformării în litere mici a șirului de caractere str.
LEFT(<i>str, len</i>)	Subșirul începând cu poziția 0 de dimensiune len.
LENGTH(<i>str</i>)	Lungimea în caractere a șirului de caractere str, exclusiv spațiile de la sfârșitul șirului.

LTRIM(str)	Șterge spațiul gol din fața unui șir de caractere
OCT(), HEX()	Returnează valorile argumentului în alte baze de numerație
REPEAT()	Repetă un șir de caractere de un număr specificat de ori
REPLACE(str,from_str,to_str)	Înlocuiește aparițiile unui șir de caractere specificat
REVERSE(str)	Șirul de caractere rezultat în urma inversării ordinii literelor șirului de caractere str.
RIGHT(str,len)	Subșirul începând cu poziția N de dimensiunea lungimii specificate
RTRIM()	Șterge spațiul gol din finalul unui șir de caractere
SPACE(N)	Returnează un string cu N spații
STRCMP()	Compară două șiruri de caractere
SUBSTR()	sinonim cu substring
SUBSTRING(str,pos,len), MID(str,pos,len), SUBSTRING(str,pos) SUBSTRING(str FROM pos), SUBSTRING(str FROM pos FOR len)	Subșirul din str care începe din poziția pos și are lungime caractere len.
TRIM()	Șterge spațiile de la începutul și sfârșitul unui șir de caractere
UCASE(str), UPPER(str)	Șirul de caractere rezultat în urma transformării în litere mari a șirului de caractere str.

Tabel 2. Funcții matematice

Nume	Descriere
ABS(X)	Returnează valoarea absolută
ACOS(x)	Returnează valoarea în radiani a unghiului al cărui cosinus este egal cu argumentul
ASIN(x)	Returnează valoarea în radiani a unghiului al cărui sinus este egal cu argumentul
ATAN(Y,X), ATAN2(Y,X)	Returnează valoarea arctangentei din Y / X
CEIL(X), CEILING(X)	Returnează cea mai mică valoare de tip integer mai mare decât argumentul
CONV(N,from_base,to_base)	Convertește numerele de la o bază de date la alta
COS(X)	Returnează valoarea cosinusului, când X este dat în radiani
COT(X)	Returnează valoarea cotangentei, când X este dat în radiani
DEGREES(X)	Convertește radiani în grade
EXP(X)	Ridică e la puterea X
FLOOR()	Returnează cea mai mare valoare de tip integer mai mică decât argumentul
LN(X)	Returnează logaritmul natural din argument
LOG10(X)	Returnează logaritmul în baza 10 din argument
LOG2(X)	Returnează logaritmul în baza 2 din argument
LOG(B,X)	Returnează logaritm în baza B din X
MOD(N,M)	Returnează restul împărțirii lui N la M
PI()	Valoarea constantei PI
POW(X,Y), POWER(X,Y)	X la puterea Y
RADIANS()	Returnează argumentul în radiani
ROUND(x), ROUND(X,D)	Valoarea argumentului rotunjită la lungimea sau precizia specificată.
SIGN(X)	Returnează semnul argumentului
SIN(X)	Returnează sin din argument
SQRT(X)	Returnează valoarea radicalului din argumentul pozitiv
TAN(X)	Returnează tangent din argument

TRUNCATE(X,D)	Redă numărul cu un număr D de zecimale
----------------------	--

Tabel 3. Funcții pentru manipularea valorilor de tip data calendaristice

Nume	Descriere
ADDDATE(date,INTERVAL expr unit) ADDDATE(expr,days)	Adaugă la dată o valoare egală cu valoarea expresie. Unit precizează ce semnificație are valoarea expresiei (secunde, minute, ani etc.)
ADDTIME(expr1,expr2)	Adaugă expr2 la expr1 și returnează rezultatul. expr1 este o data de tip time sau datetime și expr2 este o expresie de tip time
CURDATE() CURRENT_DATE CURRENT_DATE()	Returnează data curentă
CURRENT_TIME() CURRENT_TIME CURTIME()	Returnează timpul sub forma HH:MM:SS sau HHMMSS, în funcție de funcția utilizată
CURRENT_TIMESTAMP() CURRENT_TIMESTAMP NOW()	Returnează timpul
DATE_ADD(date,INTERVAL expr unit)	Adună valori de timp la o valoare de timp data
DATE_SUB(date,INTERVAL expr unit)	Sustrage un interval de timp dintr-o dată
DATE(expr)	Extrage data din expr care poate fi de tip datetime sau date
DATEDIFF(expr1,expr2)	Returnează expr1-expr2 sub forma unui număr de zile dintre o dată și cealaltă.
DAY(date), DAYOFMONTH(date)	Returnează a câta zi din lunii este data primită ca și argument
DAYNAME(date)	Returnează numele zilei precizate prin date
DAYOFWEEK(date)	Returnează a câta zi din săptămânii este data primită ca și argument. (1 = Duminică, 2 = Luni, etc.)
DAYOFYEAR(date)	Returnează a câta zi din an este data primită ca și argument. (1 = Duminică, 2 = Luni, etc.)
HOURL(time)	Extrage ora dintr-o variabilă de tip timp
LAST_DAY(date)	Returnează ultima zi din lună pentru argument
MAKEDATE(year,dayofyear)	Creează o data din an și ziua din an conținute ca argumente
MAKETIME(hour,minute,second)	Returnează o variabilă de tip timp creată cu ajutorul argumentelor
MINUTE(time)	Returnează minutele din variabila time într-o gamă de valori de la 0 la 59
MONTH(date)	Returnează un întreg reprezentând valoarea lunii din an a datei furnizate ca argument
MONTHNAME(date)	Returnează un șir de caractere reprezentând valoarea lunii din an a datei furnizate ca argument
SEC_TO_TIME(seconds)	Convertește secunde în format 'HH:MM:SS'
SECOND(seconds)	Returnează secunda (0-59) din timp
TIME(expr)	Extrage timpul dintr-o variabilă de tip datetime
TIMEDIFF(expr1,expr2)	Returnează expr1-expr2 sub forma unei variabile de tip timp
WEEK()	Returnează un întreg reprezentând săptămâna anului pentru data furnizată ca argument.

WEEKDAY (<i>date</i>)	Returnează indexul zilei din săptămână (0 = Luni, 1 = Marti, ... 6 = Duminică).
WEEKOFYEAR (<i>date</i>)	Returnează săptămâna din an, o valoare între (0-53)
YEAR ()	Returnează un întreg reprezentând anul din data furnizată ca argument

Operații de inserare și actualizare

- **Instrucțiunea INSERT**

Instrucțiunea **INSERT** este utilizată pentru a insera rânduri într-un tabel dintr-o bază de date. Această instrucțiune se poate folosi pentru:

- a insera un singur rând complet
- a insera un singur rând parțial
- a insera rezultatele unei interogări

Sintaxa instrucțiunii este destul de simplă, iar mai jos sunt prezentate cele 3 moduri de a construi o instrucțiune insert.

```
INSERT [INTO]<nume_tabelă> [(<nume_coloană>,...)]
{VALUES | VALUE} ({<expresie> | DEFAULT},...),(...),...
```

```
INSERT [INTO]<nume_tabelă>
SET<nume_coloană>={<expresie> | DEFAULT}, ...
```

```
INSERT [INTO]<nume_tabelă> [(<nume_coloană>,...)]
SELECT ...
```

unde:

- <nume_tabelă> reprezintă numele tabelului în care se face inserarea.
- <nume_coloană> o listă de una sau mai multe coloane separate prin virgulă; o valoare trebuie să se regăsească în lista de după VALUES pentru fiecare dintre coloanele amintite în lista de coloane

În cazul în care nu se specifică o listă cu coloanele, trebuie specificate valori pentru fiecare dintre coloanele din tabel în lista de după cuvântul cheie VALUES în ordinea în care apar aceste coloane în definiția tabelului.

Dacă într-o coloană nu se regăsește nici o valoare, se va folosi valoarea NULL.

Exemplul 1:

Pentru exemplele de mai jos, se presupune existența unei tabele denumite produse cu următoarele câmpuri: id, nume_produș, preț. Se vor insera valori în această tabelă astfel:

Caz 1: INSERT INTO produse ('id', 'nume_produș', 'preț')
VALUES (1, 'suc', 3.5)

Caz 2: INSERT INTO produse
SET 'id'=1, nume_produș='suc', preț=3.5

Dacă lista de coloane este omisă, atunci valorile ce sunt inserate trebuie să fie în aceeași ordine ca și ordinea coloanelor definită în tabelă.

Exemplu corect:

```
INSERT INTO produse  
VALUES (1, 'suc', 3.5)
```

Exemplu incorect:

```
INSERT INTO produse  
VALUES (1, 3.5, 'suc')
```

Valorile de pe coloane pot fi furnizate în mai multe moduri:

Caz 1. Pentru coloanele care nu apar în lista de nume de coloane, se va seta valoarea implicită definită în definiția tablei, iar pentru restul se vor lua informațiile specificate în lista de valori:

```
INSERT INTO produse ('id', 'nume_produș')  
VALUES (1, 'unt')
```

Caz 2. Pentru a seta în mod explicit valoarea implicită pentru un câmp, se folosește cuvântul cheie DEFAULT

```
INSERT INTO produse  
VALUES (1, 'unt', DEFAULT)
```

Caz 3. Se pot seta valori ce sunt rezultatul unei expresii:

```
INSERT INTO produse  
VALUES (1, 'unt', 2*12)
```

Caz 4. Pentru instrucțiunea INSERT... VALUES, este posibilă inserarea mai multor înregistrări printr-o singură comandă.

```
INSERT INTO produse ('id', 'nume_produș', 'preț')  
VALUES (1, 'paine', 3.5), (2, 'unt', 4.5), (3, 'salam', 4.4);
```

Se pot omite coloane din funcția INSERT dacă definiția tabelului o permite. Pentru aceasta trebuie îndeplinită una dintre următoarele condiții:

Caz 5. Coloana este definită ca acceptând valori NULL

Caz 6. În definiția tabelului este definită o valoare prestabilită

O altă formă a instrucțiunii INSERT care se poate utiliza pentru a insera într-un tabel rezultatul instrucțiunii SELECT este instrucțiunea INSERT SELECT. Această funcție este foarte utilă în momentul în care trebuie să copiem înregistrări dintr-un tabel într-un nou tabel.

Exemplul 2:

Dacă dorim să importăm o listă de clienți din tabelul Clienti Noi în Clienti se va folosi:

```
INSERT INTO Clienti(id, contact,nume, adresa)
SELECT id,contact, nume, adresa
FROM ClientiNoi;
```

- **Instrucțiunea UPDATE**

Pentru a actualiza (modifica) datele dintr-un tabel se utilizează instrucțiunea UPDATE.

Această instrucțiune se poate folosi în două moduri:

- pentru a actualiza anumite rânduri ale unui tabel
- pentru actualizarea tuturor rândurilor dintr-un tabel

Clauza SET indică coloanele care vor fi modificate și valorile cu care se fac modificările. Fiecare valoare poate fi dată ca o expresie sau ca și cuvântul cheie DEFAULT.

Clauza WHERE specifică condiția cu care se identifică înregistrările care vor fi actualizate, iar ORDER BY va determina ordinea înregistrărilor actualizate.

```
UPDATE <nume_tabelă>
SET col_name1={expr1|DEFAULT} [,
    col_name2={expr2|DEFAULT}] ...
[WHERE where_condition]
[ORDER BY ...]
[LIMIT row_count]
```

Exemplul 3:

Avem o tabelă cu prețuri și dorim să scădem cu 10% prețurile pentru oferte cu preț peste 100

```
UPDATE oferte
SET pret=0,9*pret
WHERE pret>100;
```



Aplicații

Pentru a rula exemplele ce urmează este nevoie să porniți serverul MySQL și utilitarul MySQL Workbench unde se vor rula comenzile de mai jos. De asemenea se va lucra în baza de date creată anterior, denumită gestiune_laboratoare

Obs: Întotdeauna se definesc primele datele din tabelul care conține cheile primare la care se face referire din tabellele care conțin chei străine

Aplicatia 1 Inserați în tabela Aparatura următoarele înregistrări:

ID_aparatura	Denumire	Expirare_garanție	verificări	spec_tehn
1	Calculatoare	10/10/2008	1	NULL
2	Analog digital scope	2/22/2012	1	HM 1507-3
3	Programable power supply	2/22/2012	1	HM 8142
4	Digital multimeter	2/22/2012	1	HM 8027
5	Programable 15 MHz function generator	2/22/2012	1	HM 8131-2
6	Programable counter	2/22/2012	1	HM 8122
7	Spectrum analyzer	2/22/2012	1	HM 5102
8	Programable Synthesizer	2/22/2012	1	HM 8134-2
9	Digital multimeter	2/22/2012	0	HM 8011-3
10	Function generator	2/22/2012	1	HM 8030-5
11	Rezistente variabile	2/22/2012	1	NULL
12	cablu conector	NULL	NULL	NULL
13	multimetru	2/22/2012	0	NULL
14	electroseparator	2/22/2013	1	NULL
15	autotransformator	2/22/2013	1	ATR-8;0-250V
16	Voltmetru de curent alternativ	2/22/2012	0	0-250
17	Voltmetru	2/22/2012	0	DU 20 pe scara de 100V
18	Sonda pentru cautarea punctelor echipotentiale	2/22/2013	0	NULL
19	Ampermetru	2/22/2013	1	NULL
20	Intrerupator bipolar capsulat	2/22/2013	0	25 A, 380 V
21	Sonda	2/22/2013	0	NULL
22	Sursa de curent continuu	2/22/2013	0	NULL
23	Ampermetru de curent continuu magnetoelectrice	2/22/2013	0	NULL
24	Voltmetru de curent continuu magnetoelectric	2/22/2013	1	30 V
25	Rezistoare montate pe placa de Plexiglas	NULL	NULL	NULL

26	Reostat cu cursor	2/22/2013	NULL	NULL
27	Comutator bipolar cu parghie	2/22/2013	0	NULL
28	bobina etalon	2/22/2013	0	NULL
29	Cordon magnetic	2/22/2013	0	NULL
30	Galvanometru ballistic	2/22/2013	0	NULL
31	Ampermetru de curent alternative	2/22/2013	0	NULL
32	Bobina de inductivitate variabila	2/22/2013	0	0.005..0.15H;5A
33	Condensator	2/22/2013	0	100uF;400V
34	Wattmetru	2/22/2013	0	60V;5A

Rezolvare

use gestiune_laboratoare;

insert into aparatura(denumire, Expirare_garantie, verificare, spec_tehn)
values ('calculatoare','2008-10-10',true,null);

insert into aparatura(denumire, Expirare_garantie, verificare, spec_tehn)
values('Analog digital scope','2012-02-22',true,'HM 1507-3');

insert into aparatura(denumire, Expirare_garantie, verificare, spec_tehn)
values('Programable power supply','2012-02-22',true,'HM 8142'),
('Digital multimeter','2012-02-22',true,'HM 8027'),
('Programable 15 MHz function generator','2012-02-22',true,'HM 8131-2');

insert into aparatura(denumire, Expirare_garantie, verificare, spec_tehn)
values('Programable counter','2012-02-22',true,'HM 8122'),
('Spectrum analyzer','2012-02-22',true,'HM 5102'),
('Programable Synthesizer','2012-02-22',true,'HM 8134-2'),
('Digital multimeter','2012-02-22',false,'HM 8011-3'),
('Function generator','2012-02-22',true,'HM 8030-5');

insert into aparatura(denumire, Expirare_garantie, verificare, spec_tehn)
values('Rezistente variabile','2012-02-22',true,null),
('cablu conector',null,null,null),
('multimetru','2012-02-22',false,null),
('electroseparator','2013-02-22',true,null),
('autotransformator','2013-02-22',true,'ATR-8;0-250V');

insert into aparatura(denumire, Expirare_garantie, verificare, spec_tehn)
values('Voltmetru de curent alternativ','2012-02-22',false,'0-250'),
('voltmetru','2012-02-22',false,'DU 20 pe scara de 100V'),
('Sonda pentru cautarea punctelor echipotentiale','2013-02-22',false,null),
('ampermetru','2013-02-22',true,null),
('Intrerupator bipolar capsulat','2013-02-22',false,'25 A, 380 V'),
('Sonda','2013-02-22',false,null),
('Sursa de curent continuu','2013-02-22',false,null),
('Ampermetru de curent continuu magnetoelectrice','2013-02-22',false,null),
('Voltmetru de curent continuu magnetoelectric','2013-02-22',true,'30 V'),
('Rezistoare montate pe placa de plexiglas',null,null,null),
('Reostat cu cursor','2013-02-22',null,null),
('Comutator bipolar cu parghie','2013-02-22',false,null),
('bobina etalon','2013-02-22',false,null),
('Cordon magnetic','2013-02-22',false,null),
('Galvanometru ballistic','2013-02-22',false,null),
('Ampermetru de curent alternativ','2013-02-22',false,null),
('Bobina de inductivitate variabila','2013-02-22',false,'0.005...0.15H;5A');


```
('Condensator','2013-02-22',false,'100uF;400V'),  
('Wattmetru','2013-02-22',false,'60V;5A');
```

Aplicatia 2 Modificați valoarea câmpului Expirare_garanție din tabela aparatura utilizând funcția update astfel încât multimetrul să aibă data de verificare '2013-02-22'

Rezolvare

```
update aparatura  
set Expirare_garantie='2013-02-22'  
where denumire='multimetru';
```

Obs: înainte se dezactivează Safe Updates din Home->Edit->Preferences->SQL Queries

Aplicatia 3 Inserați în tabela Functii următoarele înregistrări

ID	denumire	Descriere
1	asistent	descriere functie asistent
2	sef lucrari	descriere functie sef lucrari
3	conferentiar	descriere functie conf
4	tehnician	descriere functie tehnician

Rezolvare

```
insert into functii  
values(1,'asistent','descriere functie asistent');
```

```
insert into functii  
values (2,'sef lucrari','descriere functie sef lucrari'),  
(3,'conferentiar','descriere functie conf'),  
(4,'tehnician','descriere functie tehnician');
```



Exerciții propuse

1. Inserați în tabela **Responsabili** următoarele înregistrări

ID	nume	prenume	Adresa	ID_functie	varsta
1	Georgescu	Alin	Str. Ariesului nr. 2	1	31
2	Tamas	George	Str. Dacia nr.3	3	40
3	Gog	Grigore	Str. 1 Mai	2	45
4	Vasile	Ion	Str. 2 Decembrie	4	40
5	Georgescu	Ioan	Str.Cosbuc nr.5	1	30

2. Utilizând funcția UPDATE pentru a modifica adresa lui Gog Grigore astfel încât aceasta să fie Str. Observatorului nr.5

3. Inserați în tabela **Laboratoare** următoarele înregistrări

ID	denumire	Adresa	contact	ID_responsabil	suprafata	capacitate
----	----------	--------	---------	----------------	-----------	------------

1	Laborator de CAD in electromagnetism si circuite electrice	Str. Baritiu nr. 16B	332458	1	60	20
2	Laborator de compatibilitate electromagnetica	Str. Dorobantilor 71-73	251445	4	102	40
3	Laborator de campuri electrice intense	Str Baritiu, 356	264725	3	60	15
4	Laborator de metode numerice	Str. Baritiu nr.26	154822	1	42	14
5	Laborator de bazele electrotehnicii	Str.Baritiu nr.26	256545	1	80	30
6	laborator inginerie medicala	Str.Baritiu nr.26	187412	2	56	15

4. Inserați în tabela **lab_ap** următoarele înregistrări

ID_lab	ID_ap	nr_ap
1	1	16
2	1	4
2	2	1
2	3	1
2	4	1
2	5	1
2	6	1
2	7	1
2	9	1
3	1	4
3	14	1
4	1	12
5	2	3
5	3	3
5	4	3
5	7	3
5	12	3
5	13	30
6	1	12

4. Inserați în tabela **lucrari_de_lab** următoarele înregistrări

ID	Denumire	ID_lab
1	Determinarea spectrului si a suprafetelor echipotentiale ale unui camp electric cu un model electrocinetic	5
2	Model electric pentru ecuatia lui Laplace in diferente finite aplicata la determinarea suprafetelor echipotentiale si spectrului unui camp electrostatic	5
3	Studiul unui circuit de curent continuu	5
4	Verificarea legii circuitului magnetic in regim stationar cu ajutorul cordonului magnetic	5
5	Studiul unui circuit magnetic	5

6	Ridicarea unui ciclu de histerezis si masurarea pierderilor in fier cu ajutorul osciloscopulu	5
7	Studiul circuitului R,L,C serie si a rezonantei de tensiuni	5
8	Studiul circuitului R,L,C paralel si a rezonantei de curenti	5
9	Studiul unor circuite electrice trifazate	5
10	Studiul unui cuadripol pasiv	5



Aplicații

Aplicatia 4 Inserați în tabela **lucrarilab_aparate** următoarele înregistrări

ID_lucrare	ID_aparat	nr
1	15	1
1	16	1
1	17	1
1	18	1
1	19	1
2	15	1
2	17	1
2	20	2
2	21	1
3	20	1
3	22	2
3	23	3
3	24	1
3	25	1
3	26	1
3	27	2
4	19	1
4	26	1
4	28	1
4	29	1
4	30	1
5	15	1
5	16	1
5	31	1
7	15	1
7	16	4

7	20	1
7	26	1
7	31	1
7	32	1
7	33	1
7	34	1
8	15	1
8	16	1
8	20	1
8	26	1
8	31	4
8	32	1
8	33	1
8	34	1

Rezolvare

```
insert into lucrari_lab_aparate(ID_lucrare,ID_aparat,nr)
values(1,15,1),(1,16,1),(1,17,1),(1,18,1),(1,19,1),(2,15,1),(2,17,1),(2,20,2),
(2,21,1),(3,20,1),(3,22,2),(3,23,3),(3,24,1),(3,25,1),(3,26,1),(3,27,2),(4,19,1),
(4,26,1),(4,28,1),(4,29,1),(4,30,1),(5,15,1),(5,16,1),(5,31,1),(7,15,1),(7,16,4),
(7,20,1),(7,26,1),(7,31,1),(7,32,1),(7,33,1),(7,34,1),(8,15,1),(8,16,1),(8,20,1),(8,26,1),
(8,31,4),(8,32,1),(8,33,1),(8,34,1);
```



Exerciții propuse

1. Utilizați funcțiile următoare utilizate pentru șiruri de caractere și observați efectul acestora:

```
SELECT ASCII('2');
SELECT BIN(12);
SELECT BIT_LENGTH('text');
select char_length('acesta este un laborator');
SELECT CONCAT('My', 'S', 'QL');
SELECT CONCAT('My', NULL, 'QL');
SELECT CONCAT(14.3);
SELECT CONCAT_WS(',', 'nume', 'prenume 1', 'prenume 2');
SELECT CONCAT_WS(',', 'nume', NULL, 'prenume 2');
SELECT ELT(1, 'ej', 'Heja', 'hej', 'foo');
SELECT ELT(4, 'ej', 'Heja', 'hej', 'foo');
SELECT FIELD('ej', 'Hej', 'ej', 'Heja', 'hej', 'foo');
SELECT FIELD('fo', 'Hej', 'ej', 'Heja', 'hej', 'foo');
SELECT FIND_IN_SET('b', 'a,b,c,d');
```

```
SELECT INSERT('inginerie', 3, 4, 'care');
SELECT INSERT('inginerie', -1, 4, 'care');
SELECT INSERT('inginerie', 3, 100, 'care');
SELECT INSTR('foobarbar', 'bar');
SELECT INSTR('xbar', 'foobar');
SELECT LOWER('INGINERIE')
SELECT LEFT('inginerie', 5);
SELECT LENGTH('INGINERIE')
SELECT REPEAT('MySQL', 3);
SELECT REPLACE('www.mysql.com', 'w', 'Ww');
SELECT REVERSE('INGINERIE')
SELECT RTRIM('barbar ');
SELECT SUBSTRING('facultate',5);
SELECT SUBSTRING('foobarbar' FROM 4);
SELECT UPPER('Inginerie');
```

2. Utilizați funcțiile matematice de mai jos și observați efectul acestora

```
SELECT ABS(2);
SELECT ABS(-32);
SELECT ACOS(0);
SELECT ATAN(-2,2);
SELECT CEILING(1.23);
SELECT CEILING(-1.23);
SELECT CONV('6E',18,8);
SELECT COS(PI());
SELECT DEGREES(PI() / 2);
SELECT EXP(2);
SELECT EXP(-2);
SELECT FLOOR(1.23);
SELECT FLOOR(-1.23);
SELECT LOG(2,65536);
SELECT MOD(34.5,3);
SELECT POWER(3,2)
SELECT RADIANS(90);
SELECT ROUND(-1.23);
SELECT ROUND(-1.58);
SELECT ROUND(1.298, 0);
SELECT ROUND(23.298, -1);
SELECT SQRT(4);
SELECT SQRT(-16);
SELECT TRUNCATE(1.223,1);
SELECT TRUNCATE(1.999,1);
```

```
SELECT TRUNCATE(122,-2);
```

3. Să se indice rezultatul returnat de funcțiile următoare:

```
SELECT DATE_ADD('20012-01-02', INTERVAL 31 DAY);
SELECT ADDDATE('20013-01-02', INTERVAL 31 DAY);
SELECT ADDDATE('20013-01-02', 31);
SELECT ADDTIME('20013-12-31 23:59:59.999999', '1 1:1:1.000002');
SELECT ADDTIME('01:00:00.999999', '02:00:00.999998');
SELECT CURTIME();
SELECT DATE('2005-12-31 01:02:03');
SELECT DATEDIFF('2012-12-31 23:59:59', '2007-12-30');
SELECT DATEDIFF('2010-11-30 23:59:59', '2010-12-31');
SELECT DATE_ADD('2010-12-31 23:59:59', INTERVAL 1 DAY);
SELECT DATE_ADD('2100-12-31 23:59:59', INTERVAL '1:1' MINUTE_SECOND);
SELECT DATE_SUB('2005-01-01 00:00:00', INTERVAL '1 1:1:1' DAY_SECOND);
SELECT DATE_ADD('1900-01-01 00:00:00', INTERVAL '-1 10' DAY_HOUR);
SELECT DATE_SUB('1998-01-02', INTERVAL 31 DAY);
SELECT DATE_ADD('1992-12-31 23:59:59.000002', INTERVAL '1.999999' SECOND_MICROSECOND);
SELECT DAYNAME('2012-02-03');
SELECT DAYOFMONTH('2013-02-03');
SELECT DAYOFWEEK('2013-02-03');
SELECT DAYOFYEAR('2013-02-03');
SELECT HOUR('10:05:03');
SELECT LAST_DAY('2003-02-05');
SELECT LAST_DAY('2003-03-32');
SELECT MAKEDATE(2011,31), MAKEDATE(2011,32);
SELECT MAKEDATE(2011,0);
SELECT MINUTE('2008-02-03 10:05:03');
SELECT MONTH('2008-02-03');
SELECT MONTHNAME('2008-02-03');
SELECT SEC_TO_TIME(2378);
SELECT SECOND('10:05:03');
SELECT TIME('2003-12-31 01:02:03');
SELECT TIME('2003-12-31 01:02:03.000123');
SELECT TIMEDIFF('2000:01:01 00:00:00', '2000:01:01 00:00:00.000001');
SELECT TIMEDIFF('2008-12-31 23:59:59.000001', '2008-12-30 01:01:01.000002');
SELECT WEEK('2008:12:12');
SELECT WEEKDAY('2008-02-03 22:23:00');
SELECT WEEKDAY('2007-11-06');
SELECT WEEKOFYEAR('2008-02-20');
SELECT YEAR('1987-01-01');
```

4. Rulați comenzile necesare pentru a afla următoarele informații:

- Rezultatul ridicării la puterea a 5 a numărului 20;
- Sinus ($\Pi/2$)
- Data curentă
- Pentru data '2007-03-17' aflați :
 - numele zilei săptămânii
 - ziua din an
 - numărul săptămânii din anului.