

Lucrare laborator 4

Tipuri de date

Descrierea și utilizarea tipurilor de date

Dupa cum s-a vazut in lucrarile de laborator anterioare, campurile de date dintr-un tabel sunt definite in principal de anumite tipuri de date. Suportul oferit in acest sens de catre MySQL este foarte variat astfel incat sa acopere toate cerintele programatorilor si proiectantilor de baze de date.

In MySQL tipurile de date se impart in trei categorii :

- Date numerice ;
- Date de tipul data-timp calendaristic ;
- Date de tip string.

Tipurile numerice

Formatele generale ale tipurilor de date numerice sunt:

TINYINT[(M)] [UNSIGNED] [ZEROFILL]

Este un intreg foarte mic.

Domeniul de valori cu semn este de la -128 la 127.

Domeniul de valor fara semn este de la 0 la 255.

BIT sau **BOOL** sunt sinonime pentru TINYINT(1).

SMALLINT[(M)] [UNSIGNED] [ZEROFILL]

Intreg pentru valori mici.

Domeniul de valori cu semn este de la -32768 la 32767.

Domeniul de valor fara semn este de la 0 la 65535.

MEDIUMINT[(M)] [UNSIGNED] [ZEROFILL]

Intreg de dimensiune medie.

Domeniul de valori cu semn este de la -8388608 la 8388607.

Domeniul de valor fara semn este de la 0 la 16777215.

INT[(M)] [UNSIGNED] [ZEROFILL]

Intreg de dimensiune normala.

Domeniul de valori cu semn este de la -2147483648 la 2147483647.

Domeniul de valor fara semn este de la 0 la 4294967295.

INTEGER[(M)] [UNSIGNED] [ZEROFILL]

Este sinonim cu INT.

BIGINT[(M)] [UNSIGNED] [ZEROFILL]

Intreg de dimensiune mare.

Domeniul de valori cu semn este de la -9223372036854775808 la 9223372036854775807.

Domeniul de valor fara semn este de la 0 la 18446744073709551615.

FLOAT (precizie) [UNSIGNED] [ZEROFILL]

Numar cu zecimale variabile.

precizie poate fi <= 24 pentru numar cu simpla precizie si intre 25 si 53 dubla precizie.

Pentru compatibilitati cu alte tipuri de baze de date se poate folosi si sintaxa

FLOAT[(M,D)] [UNSIGNED] [ZEROFILL]

Domeniul de valori admis este de la -3.402823466E+38 la -1.175494351E-38, 0 si de la 1.175494351E-38 la 3.402823466E+38. Daca este specificat si UNSIGNED atunci nu se admit valori negative. M este lungimea numarului afisat si D este numarul de zecimale specificat.

DOUBLE[(M,D)] [UNSIGNED] [ZEROFILL]

Domeniul de valori admis este de la -1.7976931348623157E+308 la -2.2250738585072014E-308, 0 si de la 2.2250738585072014E-308 la 1.7976931348623157E+308. Daca este specificat si UNSIGNED atunci nu se admit valori negative. M este lungimea numarului afisat si D este numarul de zecimale specificat.

REAL[(M,D)] [UNSIGNED] [ZEROFILL]

Este sinonim cu DOUBLE.

DECIMAL[(M[,D])] [UNSIGNED] [ZEROFILL]

Pentru acest tip de date valorile sunt inregistrate ca siruri de caractere.

M reprezinta marimea in caractere a numarului necontorizandu-se '.' si '-' pentru valorile negative. Daca M este omis, valoare implicita a acestuia va fi 10.

D reprezinta numarul de zecimale. Daca D este 0, valorile nu vor avea parte fractionara. Daca D este omis, valoare implicita a acestuia va fi 0.

Domeniul de valori pentru DECIMAL sunt la fel ca la DOUBLE.

DEC[(M[,D])] [UNSIGNED] [ZEROFILL]

si

NUMERIC[(M[,D])] [UNSIGNED] [ZEROFILL]

sunt sinonime cu DECIMAL

MySQL suporta toate tipurile de date numerice din standardul ANSI/ISO SQL92. Aceste tipuri includ tipurile de date numerice exacte (NUMERIC, DECIMAL, INTEGER si SMALLINT) ca si tipurile de date numerice aproximative (FLOAT, REAL si DOUBLE PRECISION).

Dupa cum este permis in standardul mai sus amintit, in MySQL, tipurile de date NUMERIC si DECIMAL sunt implementate identic. Acestea sunt implementate pentru valorile unde precizia este foarte importanta, exemplu fiind datele monetare. Cand o coloana este declarata ca fiind din aceste tipuri trebuie amintit ca se pot specifica precizia si scala datelor ca in exemplul urmator.

```
salariu DECIMAL(5,2);
```

Unde 5 (precizia) este numarul de digiti semnificativi care vor fi inregistrati pentru valori iar 2 (scala) reprezinta numarul de digiti pentru zecimale.

In standardul ANSI/ISO SQL92 sintaxa DECIMAL(p) este echivalenta cu DECIMAL(p,0) iar DECIMAL este echivalent cu DECIMAL(p,0). Daca p nu este specificat, numarul de caractere ce pot fi inregistrate este 11 incluzand si caracterul '.'.

Datorita faptului ca in MySQL valorile pentru DECIMAL si NUMERIC sunt inregistrate sub forma de text si nu sub forma binara, numaru de digiti ce pot fi inregistrati (incluzand si caracterul '.') este dat de suma dintre precizie si scala (5+2=7 digiti in exemplul de mai sus).

Ca o extensie la standardul amintit, MySQL suporta si tipurile TINYINT, MEDIUMINT si BIGINT.

O alta extensie suportata este reprezentata de tipul INTEGER sau prescurtat INT unde i se poate specifica lungimea in caractere (ex INT(3)). Aceasta optiune este utila pentru alinierea la stanga la listarea valorilor acolo unde acestea sunt mai mici decat marimea indicata. Acolo unde INT este utilizat cu extensia ZEROFILL, spatiile vor fi inlocuite cu 0, de exemplu daca pentru INT(4) atribuim valoarea 30, aceasta va fi scrisa 0030.

Toate tipurile de date numerice pot avea o valoare optionala (necostandard) UNSIGNED. Toate valorile cu specificatia UNSIGNED sunt valori intregi sau fara semn.

Tipul de date FLOAT este reprezentantul tipurilor de date numerice aproximative. Acest tip are optional doi parametrii FLOAT(p,q) ce reprezinta ca si la DECIMAL lungimea si respectiv numarul de zecimale dorit. Cand FLOAT este utilizat fara parametrii pentru o coloana, MySQL va utiliza 4 bytes (32 biti) pentru a inregistra valoarea.

In situatia in care se incearca salvarea mai multor zecimale decat cele specificate in tipul FLOAT, MySQL va rotunji valoarea pentru a o trunchia la numarul de digiti specificat.

De precizat este faptul ca tipurile REAL si DOUBLE PRECISION nu accepta precizie specificata.

Pentru toate tipurile de date numerice in situatia in care valoarea inregistrata depaseste limita de domeniu a tipului, se va inregistra valoarea limita a domeniului. De exemplu INT are domeniul de valori intre -2147483648 si 2147483647. Daca valoare ce se doreste a fi inregistrata este -9999999999, atunci se va inregistra valoarea -2147483648 (limita inferioara a domeniului de

valori). Acelasi lucru se va intampla si pentru valori pozitive adica pentru 9999999999 se va inregistra 2147483647.

Tipurile de date calendaristice

Formatele general ale tipurilor de date calendaristice sunt:

DATE

O data calendaristica.

Domeniul de valori este intre '1000-01-01' si '9999-12-31'.

MySQL afiseaza valorile pentru DATE in formatul 'AAAA-LL-ZZ'.

DATETIME

O combinatie intre data si timp.

Domeniul de valori este intre '1000-01-01 00:00:00' si '9999-12-31 23:59:59'.

MySQL afiseaza valorile pentru DATETIME in formatul 'AAAA-LL-ZZ HH:MM:SS'.

TIMESTAMP[(M)]

Un timp formatat.

Domeniul de valori este intre '1970-01-01 00:00:00' pana in '2037-01-01 00:00:00'.

MySQL afiseaza valorile pentru TIMESTAMP in formatul AAAALLZZHHMMSS, unde AAAALLZZHHMMSS este formatat de lungimea M ce poate lua valorile 14 (sau lipsa), 12, 10, 8, 6, 4 sau 2.

TIME

Timpul.

Domeniul de valori este intre '-838:59:59' si '838:59:59'. Motivul pentru care valoarea pentru ore a fost aleasa asa de mare este acela ca se pot afisa si intervale de timp intre anumite evenimente.

YEAR[(2|4)]

Anul este reprezentat in format de 2 sau 4 digiti. Valoare implicita este 4. Domeniul de valori este intre 1901 si 2155 si 0000 pentru formatul de 4 digiti si intre 1970-2069 pentru 2 digiti (70-69).

Fiecare dintre aceste tipuri are un anumit domeniu de valori in care este inclusa si valoarea 0 ce va fi inregistrata in cazul scrierii eronate a unei valori. Valoarea 0 este utila si in situatia in care spre exemplu se doreste sa se inregistreze data de nastere care sa presupunem ca nu este cunoscuta exact si atunci se poate scrie 1980-00-00 (nu este cunoscuta luna si ziua) sau 1989-01-00 (nu este cunoscuta ziua). MySQL afiseaza si inregistreaza valorile de tip data calendaristica in urmatoarea forma 'AAAA-LL-ZZ HH:MM:SS'.

Tipul de date **TIMESTAMP** are ca valoare optionala lungimea, unde aceasta poate fi de maxim 14 caractere (AAAALLZZHHMMSS). Formatul luat de **TIMESTAMP** pentru alte valori de lungime este:

TIMESTAMP(14) AAAALLZZHHMMSS

TIMESTAMP(12) AALLZZHHMMSS

TIMESTAMP(10) AALLZZHHMM

TIMESTAMP(8) AAAALLZZ

TIMESTAMP(6) AALLZZ

TIMESTAMP(4) AALL

TIMESTAMP(2) AA

Sirurile de caractere

CHAR(M) [BINARY]

Este un sir de lungime fixa aliniata la dreapta si completate la stanga cu spatii goale.

Ex: Pentru **CHAR(4)** sirul 'ab' este inregistrat ' ab'.

Valorile vor fi ordonate si afisate facandu-se diferenta intre caractere mari si mici daca optiunea **BINARY** nu a fost adaugata.

Domeniul lui **M** este intre 1 si 255.

CHAR este sinonim cu **CHAR(1)**.

VARCHAR(M) [BINARY]

Sir de caractere cu lungime variabila. **VARCHAR** este prescurtarea pentru **CHARACTER VARYING**.

Spre deosebire de **CHAR** nu sunt adaugate spatii goale pentru completarea dimensiunii specificate in **M**.

Domeniul de valori pentru **M** este acelasi ca la **CHAR** intre 1 si 255.

Valorile vor fi ordonate si afisate facandu-se diferenta intre caractere mari si mici daca optiunea **BINARY** nu a fost adaugata.

TINYBLOB

BLOB reprezinta prescurtarea de la Binary Large Object (obiect binar lungime mare).

Coloane de tip **TINYBLOB** au lungime maxima de 255 caractere.

TINYTEXT

Coloane de tip **TINYTEXT** au lungime maxima de 255 caractere.

BLOB

TEXT

Coloane de tip **BLOB** sau **TEXT** au lungime maxima de 65535 ($2^{16} - 1$) caractere.

MEDIUMBLOB

MEDIUMTEXT

Sunt coloane de tip BLOB sau TEXT cu lungime maxima de 16777215 (2²⁴ - 1) caractere.

LONGBLOB

LONGTEXT

Sunt coloane de tip BLOB sau TEXT cu lungime maxima de 4294967295 (2³² - 1) caractere.

ENUM ('value1','value2',...)

Enumerare.

Defineste un obiect de tip sir de caractere ce poate lua doar una din valorile din lista 'value1', 'value2', ... sau NULL.

Un ENUM poate avea maxim 65535 de valori distincte.

SET('value1','value2',...)

Un set de valori.

Este un obiect de tip sir de caractere care poate avea zero sau mai multe valori alese din lista de valori 'value1', 'value2', ...

Un SET poate avea maxim 64 de membrii.

Tipurile CHAR si VARCHAR sunt identice avand acelasi domeniu de valori ale lungimii (1-255), diferenta este numai modul in care se inregistreaza si afizeaza datele. Spre exemplu daca pentru tipul CHAR(4) se inregistreaza doar 'ab', atunci restul caracterelor necompletate vor fi umplute automat cu spatii goale iar alinierea se va face spre dreapta. Daca pentru tipul VARCHAR(4) se va inregistra 'ab', rezultatul va fi cel inregistrat, adica 'ab'.

Tipul BLOB este un obiect binar de lungime mare care poate suporta o cantitate de date variabila. Cele patru tipuri de BLOB (TINYBLOB, BLOB, MEDIUMBLOB si LONGBLOB) difera doar prin numarul maxim de caractere pe care il pot suporta.

Cele patru tipuri de TEXT (TINYTEXT, TEXT, MEDIUMTEXT si LONGTEXT) corespund la cele patru tipuri de BLOB si au aceeasi lungime de stocare. Singura diferenta intre BLOB si TEXT este aceea ca sortarea si compararea se face tinandu-se cont de caractere mari si mici pentru BLOB si netinandu-se cont pentru TEXT.

Daca se atribuie unei coloane de tip BLOB sau TEXT o valoare mai mare decat domeniul de referinta, valoarea atribuita va fi truncheata pana la valoarea maxima a domeniului de referinta.

Tema de laborator

Sa se creeze un tabel ale carui coloane sa fie create dupa urmatorul tabel:

Nr.	Nume Coloana	Tip
1	tiny_col	tinyint
2	bool_col	bool
3	smallint_col	smallint fara semn
4	medint_col	mediumint
5	int_col	int de 4 caractere
6	bigint_col	bigint
7	float_col	float de 5 caractere cu 2 zecimale
8	double_col	double fara semn
9	dec_col	decimal de 3 caractere cu 2 zecimale
10	date_col	date
11	datetime_col	datetime
12	timestamp_col	timestamp de 8 caractere
13	time_col	time
14	year_col	year de 2 caractere
15	char_col	char de 10 caractere
16	varchar_col	varchar de 10 caractere binary
17	blob_col	blob
18	text_col	mediumtext
19	enum_col	enum cu valorile unu, doi, trei, patru si cinci
20	set_col	set cu valorile unu, doi, trei, patru si cinci