

Databases. Tables, columns, foreign keys. MySql Workbench. SQL queries



IT Learning &
Outsourcing Center

www.pragmatic.bg

Lector: Stefan Vadev
Skype: stefanvadev77
E-mail: stefan.vadev@gmail.com
Facebook:
<https://www.facebook.com/stefan.vadev.7>
Copyright © Pragmatic LLC

2013–2018



Summary

- What is a database
- MySQL Server, MySQL Workbench
- MySQL data types
- Creating tables
- Executing SQL queries



Intro to databases

- We use databases because we need to store business related information somewhere
- Database - a structured set of data held in a computer, especially one that is accessible in various ways.





Database types

- We use databases because we need to store business related information somewhere
- Database - a structured set of data held in a computer, especially one that is accessible in various ways.





Tables and columns

- A **table** is a collection of related data held in a structured format within a database. It consists of columns, and rows.
- In a relational database, a **column** is a set of data values of a particular simple type, one value for each row of the database. A column may contain text values, numbers, or even pointers to files in the operating system. Some relational database systems allow columns to contain more complex data types; whole documents, images or even video clips are examples.





This is a table

Columns stores a specific data type



Row →
Or record

| Emp No | Name | Age | Department | Salary |
|--------|----------|-----|------------|--------|
| 001 | Alex S | 26 | Store | 5000 |
| 002 | Golith K | 32 | Marketing | 5600 |
| 003 | Rabin R | 31 | Marketing | 5600 |
| 004 | Jons | 26 | Security | 5100 |



Another table

| ID | NAME | CLASS | MARK | SEX |
|----|-------------|-------|------|--------|
| 1 | John Deo | Four | 75 | female |
| 2 | Max Ruin | Three | 85 | male |
| 3 | Arnold | Three | 55 | male |
| 4 | Krish Star | Four | 60 | female |
| 5 | John Mike | Four | 60 | female |
| 6 | Alex John | Four | 55 | male |
| 7 | My John Rob | Fifth | 78 | male |
| 8 | Asruid | Five | 85 | male |
| 9 | Tes Qry | Six | 78 | male |
| 10 | Big John | Four | 55 | female |



SQL data types

Properly defining the fields in a **table** is important to the overall optimization of your **database**. You should use only the type and size of field you really need to use. For example, do not define a field 10 characters wide, if you know you are only going to use 2 characters. These type of fields (or columns) are also referred to as data types, after the **type of data** you will be storing in those fields.

MySQL uses many different data types broken into three categories

- **Numeric**
- **Date and Time**
- **String Types.**
- **Let us now discuss them in detail.**



Numeric data types

Numeric Data Types

MySQL uses all the standard ANSI SQL numeric data types, so if you're coming to MySQL from a different database system, these definitions will look familiar to you. The following list shows the common numeric data types and their descriptions:

INT – A normal-sized integer that can be signed or unsigned. If signed, the allowable range is from -2147483648 to 2147483647. If unsigned, the allowable range is from 0 to 4294967295. You can specify a width of up to 11 digits.

TINYINT – A very small integer that can be signed or unsigned. If signed, the allowable range is from -128 to 127. If unsigned, the allowable range is from 0 to 255. You can specify a width of up to 4 digits.



Numeric data types

SMALLINT – A small integer that can be signed or unsigned. If signed, the allowable range is from -32768 to 32767. If unsigned, the allowable range is from 0 to 65535. You can specify a width of up to 5 digits.

MEDIUMINT – A medium-sized integer that can be signed or unsigned. If signed, the allowable range is from -8388608 to 8388607. If unsigned, the allowable range is from 0 to 16777215. You can specify a width of up to 9 digits.



Numeric data types

BIGINT – A large integer that can be signed or unsigned. If signed, the allowable range is from -9223372036854775808 to 9223372036854775807. If unsigned, the allowable range is from 0 to 18446744073709551615. You can specify a width of up to 20 digits.

FLOAT(M,D) – A floating-point number that cannot be unsigned. You can define the display length (M) and the number of decimals (D). This is not required and will default to 10,2, where 2 is the number of decimals and 10 is the total number of digits (including decimals). Decimal precision can go to 24 places for a FLOAT.



Numeric data types

DOUBLE(M,D) – A double precision floating-point number that cannot be unsigned. You can define the display length (M) and the number of decimals (D). This is not required and will default to 16,4, where 4 is the number of decimals. Decimal precision can go to 53 places for a DOUBLE. REAL is a synonym for DOUBLE.

DECIMAL(M,D) – An unpacked floating-point number that cannot be unsigned. In the unpacked decimals, each decimal corresponds to one byte. Defining the display length (M) and the number of decimals (D) is required. NUMERIC is a synonym for DECIMAL.



Date and time types

Date and Time Types

The MySQL date and time datatypes are as follows:

DATE – A date in YYYY-MM-DD format, between 1000-01-01 and 9999-12-31. For example, December 30th, 1973 would be stored as 1973-12-30.

DATETIME – A date and time combination in YYYY-MM-DD HH:MM:SS format, between 1000-01-01 00:00:00 and 9999-12-31 23:59:59. For example, 3:30 in the afternoon on December 30th, 1973 would be stored as 1973-12-30 15:30:00.



Date and time types

TIMESTAMP – A timestamp between midnight, January 1st, 1970 and sometime in 2037. This looks like the previous DATETIME format, only without the hyphens between numbers; 3:30 in the afternoon on December 30th, 1973 would be stored as 19731230153000 (YYYYMMDDHHMMSS).

TIME – Stores the time in a HH:MM:SS format.

YEAR(M) – Stores a year in a 2-digit or a 4-digit format. If the length is specified as 2 (for example YEAR(2)), YEAR can be between 1970 to 2069 (70 to 69). If the length is specified as 4, then YEAR can be 1901 to 2155. The default length is 4.



String types

Although the numeric and date types are fun, most data you'll store will be in a string format. This list describes the common string datatypes in MySQL.

CHAR(M) – A fixed-length string between 1 and 255 characters in length (for example CHAR(5)), right-padded with spaces to the specified length when stored. Defining a length is not required, but the default is 1.

VARCHAR(M) – A variable-length string between 1 and 255 characters in length. For example, VARCHAR(25). You must define a length when creating a VARCHAR field.



String types

BLOB or TEXT – A field with a maximum length of 65535 characters. BLOBs are "Binary Large Objects" and are used to store large amounts of binary data, such as images or other types of files. Fields defined as TEXT also hold large amounts of data. The difference between the two is that the sorts and comparisons on the stored data are **case sensitive** on BLOBs and are **not case sensitive** in TEXT fields. You do not specify a length with BLOB or TEXT.

TINYBLOB or TINYTEXT – A BLOB or TEXT column with a maximum length of 255 characters. You do not specify a length with TINYBLOB or TINYTEXT.

MEDIUMBLOB or MEDIUMTEXT – A BLOB or TEXT column with a maximum length of 16777215 characters. You do not specify a length with MEDIUMBLOB or MEDIUMTEXT.



String types

LONGBLOB or LONGTEXT – A BLOB or TEXT column with a maximum length of 4294967295 characters. You do not specify a length with LONGBLOB or LONGTEXT.

ENUM – An enumeration, which is a fancy term for list. When defining an ENUM, you are creating a list of items from which the value must be selected (or it can be NULL). For example, if you wanted your field to contain "A" or "B" or "C", you would define your ENUM as ENUM ('A', 'B', 'C') and only those values (or NULL) could ever populate that field.



Primary key

A **primary key** is a special relational database table column (or combination of columns) designated to uniquely identify all table records. A **primary key's** main features are: It must contain a unique value for each row of data. It cannot contain null values.





Foreign Key

A **FOREIGN KEY** is a key used to link two tables together.

A **FOREIGN KEY** is a field (or collection of fields) in one table that refers to the **PRIMARY KEY** in another table.

The table containing the foreign key is called the child table, and the table containing the candidate key is called the referenced or parent table.

Let's look at the following two tables:



Foreign Key

| PersonID | LastName | FirstName | Age |
|----------|-----------|-----------|-----|
| 1 | Hansen | Ola | 30 |
| 2 | Svendson | Tove | 23 |
| 3 | Pettersen | Kari | 20 |

"Orders" table:

| OrderID | OrderNumber | PersonID |
|---------|-------------|----------|
| 1 | 77895 | 3 |
| 2 | 44678 | 3 |
| 3 | 22456 | 2 |
| 4 | 24562 | 1 |



SQL Workshop

Now all we need is here:

<https://www.w3schools.com/sql/default.asp>





Summary

- What is a database
- MySQL Server, MySQL Workbench
- MySQL data types
- Creating tables
- Executing SQL queries