MySQL Data Types

In this section

MySQL Data Types 1
Table 1: Most of the available column types for use with MySQL databases 1
To choose your data types: 2
CHAR versus VARCHAR 3
Table 2 Example: Defining the optimal type for each field 3
Activity 1 4

MySQL Data Types

Once you have identified all of the tables and columns that the database will need, you should determine each field's MySQL data type. When creating the database, MySQL requires that you define what sort of information each field will contain.

There are three primary categories, which is true for almost every database software:

- Text
- Numbers
- Dates and times

Within each of these, there are a number of variants—some of which are MySQL-specific—you can use. Choosing your column types correctly not only dictates what information can be stored and how, but also affects the database's overall performance. **Table 1** lists most of the available types for MySQL, how much space they take up, and a brief description.

Table 1: Most of the available column types for use with MySQL databases

MySQL Datatypes		
Type	**Size**	**Description**
CHAR(Length)	Length bytes	A fixed-length field from 1 to 255 characters. Right padded with spaces to specified length. Need not define a length, but default is 1.
VARCHAR(Length)	String length + 1 bytes	A variable-length field from 1 to 255 characters. Must define a length.
TINYTEXT	String length + 1 bytes	A string with a maximum length of 255 characters. You do not specify a length.
TEXT	String length + 2 bytes	A string with a maximum length of 65,535 characters. You do not specify a length.
MEDIUMTEXT	String length + 3 bytes	A string with a maximum length of 16,777,215 characters. You do not specify a length.
LONGTEXT	String length + 4 bytes	A string with a maximum length of 4,294,967,295 characters. You do not specify a length.
ENUM	1 or 2 bytes	An enumeration, which is a fancy term for a list.
TINYINT(Length)	1 byte	Range of -128 to 127 or 0 to 255 unsigned. You can specify a width of up to 4 digits.
SMALLINT(Length)	2 bytes	Range of -32,768 to 32,767 or 0 to 65,535 unsigned. You can specify a width of up to 5 digits.

MEDIUMINT(Length)	3 bytes	Range of -8,388,608 to 8,388,607 or 0 to 16,777,215 unsigned. You can specify a width of up to 9 digits.
INT(Length)	4 bytes	Range of -2,147,483,648 to 2,147,483,647 or 0 to 4,294,967,295 unsigned. You can specify a width of up to 11 digits.
BIGINT(Length)	8 bytes	Range of -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807 or 0 to 18,446,744,073,709,551,615 unsigned. You can specify a width of up to 20 digits.
FLOAT(length, decimals)	4 bytes	A small number with a floating decimal point. You may display the length and number of decimals. If not, then it will default to (10,2). Decimal precision can go to 24 places.
DOUBLE(length, decimals)	8 bytes	A large number with a floating decimal point. You may display the length and number of decimals. If not, then it will default to (16,4). Decimal precision can go to 53 places.
DECIMAL(length, decimals)	Length + 1 or Length + 2 bytes	You must specify the length and number of decimals.
DATE	3 bytes	In the format of YYYY-MM-DD.
DATETIME	8 bytes	In the format of YYYY-MM-DD HH:MM:SS.
TIMESTAMP	4 bytes	In the format of YYYYMMDDHHMMSS, or YYMMDDHHMMSS or YYYYMMDD or YYMMDD
TIME	3 bytes	In the format of HH:MM:SS
YEAR(length)	1 byte	Can store 2 or 4 digits. If not specified it defaults to 4 digits.

The number types can be UNSIGNED—limiting the column to positive numbers or zero—or be defined as ZEROFILL, which means that any extra room will be padded with zeroes (ZEROFILLs are also automatically UNSIGNED).

To choose your data types:

1. Identify whether a column should be a text, number, or date type.

   This is normally an easy and obvious step. You will find that numbers such as post codes and dollar amounts should be text fields if you include their corresponding punctuation (dollar signs, commas, and hyphens), but you'll get better results if you store them as numbers and address the formatting elsewhere.

2. Choose the most appropriate subtype for each column.

   For improved performance, keep in mind two considerations:

   - Fixed-length fields (such as CHAR) are generally faster than variable-length fields (such as VARCHAR), but they also take up more disk space. See the side-bar for more information.
   - The size of any field should be restricted to the smallest possible value, based upon what the largest possible input could be. For example, if the largest a number such as Client ID could be is in the hundreds, set the column as an unsigned three-digit SMALLINT (allowing for up to 999 values).
   - You should keep in mind that if you insert a string five characters long into a CHAR(2) field, the final three characters will be truncated. This is true for any field in which the length is set (CHAR, VARCHAR, INT, etc.).

CHAR versus VARCHAR

There is some debate as to the merits of these two similar types. Both store strings and can be set with a fixed maximum length. One primary difference is that anything stored as a CHAR will always be stored as a string the length of the column (using spaces to pad it). Conversely, VARCHAR strings will be only as long as the stored string.

The two implications of this are

- VARCHAR columns tend to take up less disk space.
- CHAR columns are faster to access and sort than VARCHAR.

Granted, the speed and disk space differences between the two types may be imperceptible in most cases and is therefore not worth dallying over.

There is also a third, minor difference between these two: MySQL trims off extra spaces from CHAR columns when data is retrieved and from VARCHAR when it's inserted.

3. Set the maximum length for text and number columns as well as other attributes such as UNSIGNED (**Table 1**).

   Different developers have different preferences, but the most important factor is to tailor each setting to the information at hand rather than using generic (and inefficient) TEXT and INT types at all times.

Table 2 Example: Defining the optimal type for each field

Accounting Database		
Column Name	**Table**	**Column Type**
Invoice Number	Invoices	SMALLINT(4) UNSIGNED
Client ID	Invoices	SMALLINT(3) UNSIGNED
Invoice Date	Invoices	DATE
Invoice Amount	Invoices	DECIMAL(10,2) UNSIGNED
Invoice Description	Invoices	TINYTEXT
Client ID	Clients	SMALLINT(3) UNSIGNED
Client Name	Clients	VARCHAR(40)
Client Street Address	Clients	VARCHAR(80)
Client City	Clients	VARCHAR(30)
Client State	Clients	CHAR(2)
Client Zip	Clients	MEDIUMINT(5) UNSIGNED
Client Phone	Clients	VARCHAR(14)
Contact Name	Clients	VARCHAR(40)
Contact Email Address	Clients	VARCHAR(60)
Expense ID	Expenses	SMALLINT(4) UNSIGNED
Expense Category ID	Expenses	TINYINT(3) UNSIGNED
Expense Amount	Expenses	DECIMAL(10,2) UNSIGNED
Expense Description	Expenses	TINYTEXT
Expense Date	Expenses	DATE
Expense Category ID	Expense Categories	TINYINT(3) UNSIGNED
Expense Category	Expense Categories	VARCHAR(30)

Activity 1

Determine the data types for each of the fields in the Dental Practice database (Q6 of '1 – introduction to databases' file).