

4.4 Creating and dropping tables

SQL data types

When creating a table, every table column must be assigned a data type. Common data types include: integer, decimal, date and time, and character. Most databases allow integer and decimal numbers to be signed or unsigned.

- A **signed** number is a number that may be negative.
- An **unsigned** number is a number that cannot be negative.

Data types vary in size requirements. Ex: A character data type generally uses 1 byte per character to store a character string, but an integer data type uses a fixed number of bytes to store a number. Unsigned data types can store larger numbers than the signed version of the same data type.

A database designer should choose the data type with the smallest storage requirements to minimize the overall table size. Ex: Any integer data type can store integers that range from -100 to 100, but TINYINT has the smallest storage requirements (1 byte) for storing numbers in the range -100 to 100.

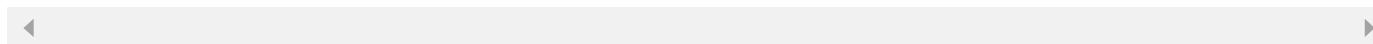
The table below summarizes the commonly used data types available in MySQL. The table's Storage column lists the number of bytes required to store values for the data type.

Table 4.4.1: Common MySQL data types.

Category	Example	Data type	Storage	Notes
Integer	34 and -739448	TINYINT	1 byte	Signed range: -128 to 127 Unsigned range: 0 to 255
		SMALLINT	2 bytes	Signed range: -32,768 to 32,767 Unsigned range: 0 to 65,535

Category	Example	Data type	Storage	Notes
		MEDIUMINT	3 bytes	Signed range: -8,388,608 to 8,388,607 Unsigned range: 0 to 16,777,215
		INTEGER or INT	4 bytes	Signed range: -2,147,483,648 to 2,147,483,647 Unsigned range: 0 to 4,294,967,295
		BIGINT	8 bytes	Signed range: -2^{63} to $2^{63} - 1$ Unsigned range: 0 to $2^{64} - 1$
Decimal	123.4 and -54.29685	DECIMAL(M,D)	Varies depending on M and D	Exact decimal number where M = number of significant digits, D = number of digits after decimal point
		FLOAT	4 bytes	Approximate decimal numbers with range: -3.4E+38 to 3.4E+38
		DOUBLE	8 bytes	Approximate decimal numbers with range: -1.8E+308 to 1.8E+308
Date and time	'1776-07-04 13:45:22'	DATE	3 bytes	Format: YYYY-MM-DD. Range: '1000-01-01' to '9999-12-31'
		TIME	3 bytes	Format: hh:mm:ss
		DATETIME	5 bytes	Format: YYYY-MM-DD hh:mm:ss. Range: '1000-01-01 00:00:00' to '9999-12-31 23:59:59'.

Category	Example	Data type	Storage	Notes
Character	'string'	CHAR(N)	N bytes	Fixed-length string of length N ; $0 \leq N \leq 255$
		VARCHAR(N)	Length of characters + 1 bytes	Variable-length string with maximum N characters; $0 \leq N \leq 65,535$

[Feedback?](#)**PARTICIPATION ACTIVITY****4.4.1: MySQL data types.**

Choose the best data type for each scenario with the minimum storage requirements.

- 1) Storing a city's population, which ranges from a dozen to 24 million people.

- unsigned MEDIUMINT
- unsigned INTEGER
- unsigned BIGINT

Correct

An INTEGER requires 4 bytes and is sufficient to represent populations that are up to 4.3 billion people. The next smallest data type, MEDIUMINT, can only represent populations up to 16.8 million.



- 2) Storing the annual gain or loss in a city's population, which ranges from -1 million to 1 million.

- signed SMALLINT
- unsigned MEDIUMINT
- signed MEDIUMINT

Correct

A signed MEDIUMINT requires 3 bytes and can store -8,388,608 to 8,388,607.



- 3) Storing the price of an item that ranges from a few dollars to a few hundred dollars.

- FLOAT
- DECIMAL(2,5)
- DECIMAL(5,2)

Correct

$M = 5$ and $N = 2$, so the number of significant digits is 5, and the number of places after the decimal point is 2. A price like \$599.27 has 5 significant digits and 2 numbers after the decimal point.

- 4) Storing the date and time an item is purchased.

- DATE
- TIME
- DATETIME

Correct

DATETIME stores both a date and a time. Ex: If an item is purchased on July 4, 2020 at 2:30 pm, the value stored in DATETIME is '2020-07-04 14:30:00'.

- 5) Storing a student's assigned letter grade, like A or D.

- TINYINT
- CHAR(1)
- VARCHAR(1)

Correct

A grade is a single character, and $N = 1$ uses 1 byte to store each grade.

- 6) Storing a student's email address.

- VARCHAR(100)
- VARCHAR(10)
- CHAR(100)

Correct

Each student's email address varies in length, so a VARCHAR is ideal. Using $N = 100$ means no email address can exceed 100 characters, which is a reasonable assumption.

- 7) Storing a yes or no value.

- TINYINT
- CHAR(3)
- VARCHAR(3)

Correct

Typically, a yes/no or true/false value is stored as a 1-byte TINYINT, where 1 means yes/true, and 0 means no/false. A common term that refers to yes/no is Boolean.

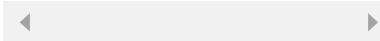
[Feedback?](#)

CREATE TABLE and DROP TABLE statements

The **CREATE TABLE** statement creates a new table by specifying the table name, column names, and column data types.

Figure 4.4.1: CREATE TABLE syntax.

```
CREATE TABLE TableName (
    Column1 DATA_TYPE,
    Column2 DATA_TYPE,
    ...
    ColumnN DATA_TYPE
);
```

[Feedback?](#)

The **DROP TABLE** statement deletes a table, along with all the table's rows, from a database. Ex: `DROP TABLE Employee;` deletes the Employee table.

PARTICIPATION
ACTIVITY

4.4.2: Creating an Employee table.



1 2 ⏪ ✓ 2x speed

table name
↓
CREATE TABLE Employee (
 ID SMALLINT UNSIGNED,
 Name VARCHAR(60),
 BirthDate DATE,
 } data types
 { column names

ID	Name	BirthDate	Salary

```
        Salary      DECIMAL(7,2)
    );
```

The column names and data types are separated by commas.

Captions ^

1. The CREATE TABLE statement names the new table "Employee".
2. The column names and data types are separated by commas.

[Feedback?](#)

PARTICIPATION
ACTIVITY

4.4.3: Creating and deleting tables.



Refer to the animation above.

- 1) The Employee table is created with 4 columns.

True
 False

Correct



The columns are: ID, Name, BirthDate, and Salary.

- 2) The ID column only holds non-negative integers.

True
 False

Correct



The **SMALLINT UNSIGNED** data type specifies the ID column must be a number between 0 and 65,535. If the company expects to hire more than 65,535 employees, a larger data type like **MEDIUMINT** should be used to create the Employee table.

- 3) The Salary column holds only non-negative numbers.

True
 False

Correct



DECIMAL(7,2) is by default a signed number unless the **UNSIGNED** keyword is specified, so the number may be negative.

- 4) The statement `DROP Employee;` deletes the Employee table.

True
 False

Correct



The keyword TABLE must appear after DROP: `DROP TABLE Employee;`

- 5) The `DROP TABLE` statement does not delete the table unless the table is empty.

True
 False

Correct



The table is deleted regardless of how many rows are in the table.

[Feedback?](#)

PARTICIPATION ACTIVITY

4.4.4: Create a Product table.



Write the SQL to create a Product table with the following columns:

- **ID** - Unsigned integer
- **Name** - Variable-length string with maximum 40 characters
- **ProductType** - Fixed-length string with maximum 3 characters
- **OriginDate** - Year, month, and day
- **Weight** - Decimal number with six significant digits and one digit after the decimal point

Place your `CREATE TABLE` statement before the `INSERT` and `SELECT` statements. Run your solution and verify the result table contains the three inserted rows.

```
1 -- Write your CREATE TABLE statement here:  
2  
3  
4  
5 INSERT INTO Product (ID, Name, ProductType, OriginDate, Weight) VALUES  
6   (100, 'Tricorder', 'COM', '2020-08-11', 2.4),  
7   (200, 'Food replicator', 'FOD', '2020-09-21', 54.2),  
8   (300, 'Cloaking device', 'SPA', '2019-02-04', 177.9);
```

```
9  
10 SELECT *  
11 FROM Product;
```

Run**Reset code****Feedback?**

ALTER TABLE statement

The **ALTER TABLE** statement adds, deletes, or modifies columns on an existing table. The ALTER TABLE statement specifies the table name followed by a clause that indicates what should be altered. The table below summarizes the three ALTER TABLE clauses.

Table 4.4.2: ALTER TABLE ADD, CHANGE, and DROP syntax.

ALTER TABLE clause	Description	Syntax
ADD	Adds a column	ALTER TABLE TableName ADD ColumnName DataType;
CHANGE	Modifies a column	ALTER TABLE TableName CHANGE CurrentColumnName NewColumnName NewDataType;
DROP	Deletes a column	ALTER TABLE TableName DROP ColumnName;

[Feedback?](#)**PARTICIPATION ACTIVITY**

4.4.5: Adding, modifying, and deleting columns.

**1 2 3** 2x speed

```
ALTER TABLE Employee  
ADD Salary DECIMAL(7, 2);
```

column name data type

ID	Name	BirthDate	AnnualSalary
2538	Lisa Ellison	1993-10-02	
5384	Sam Snead	1995-03-15	
6381	Maria Rodriguez	2001-12-21	
7343	Gary Smith	1984-09-22	

```
ALTER TABLE Employee  
CHANGE Salary AnnualSalary INT;
```

current column name new column name new data type

```
ALTER TABLE Employee  
DROP AnnualSalary;
```

column name

The ALTER TABLE statement deletes the AnnualSalary column.

Captions

1. The ALTER TABLE statement adds a Salary column to Employee that holds 7 significant digits and 2 decimal places.
2. The ALTER TABLE statement changes the Salary column's name to AnnualSalary. The data type is changed to INT.
3. The ALTER TABLE statement deletes the AnnualSalary column.

[Feedback?](#)

**PARTICIPATION
ACTIVITY****4.4.6: ALTER TABLE.**

- 1) Add a column called Description to the Department table.

ALTER TABLE Department

VARCHAR(50);

Check**Show answer****Answer**

ADD Description

The ADD keyword is followed by the column name and data type.



- 2) Rename column Description to ShortDesc.

ALTER TABLE Department

VARCHAR(50);

Check**Show answer****Answer**

CHANGE Description ShortDesc

The CHANGE keyword is followed by the current column name, new column name, and data type.



- 3) Change column ShortDesc to accept up to 80 characters.

ALTER TABLE Department

CHANGE

;

Check**Show answer****Answer**

ShortDesc ShortDesc VARCHAR(80)

ShortDesc appears twice, first as the current column name and then as the new column name. VARCHAR(80), the new data type, follows the new column name.



- 4) Delete the column ShortDesc.

ALTER

;

Check**Show answer****Answer**

**TABLE Department
DROP ShortDesc**



The TABLE keyword is followed by the table name being altered. The DROP keyword is followed by the column name to delete.

[Feedback?](#)**CHALLENGE ACTIVITY**

4.4.1: Creating, dropping, and altering tables.



379958.2369558.qx3zqy7

Start

1



2



3

Complete the following statement to create a table named Country.

Choose data types based on the following requirements:

- ISOCode3 stores the country's code, consisting of one to three letters.
- Population stores the country's population, a whole number.

```
/* Your code goes here */ (
    ISOCode3    /* Your code goes here */ ,
    Population /* Your code goes here */
);
```

Enter a statement to delete the above table.

```
/* Your code goes here */ ;
```

1

2

3

Check**Next**[Feedback?](#)

Exploring further:

- [Data Types](#) from MySQL.com
- [CREATE TABLE Syntax](#) from MySQL.com
- [DROP TABLE Syntax](#) from MySQL.com
- [ALTER TABLE Syntax](#) from MySQL.com

How was this
section?



[Provide feedback](#)