As you already know, SQL is a language used for working with different types of data organized into a table. Usually, data values from the same column in a table have the same meaning and type. For example, a table Car may look like this:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **manufacture\_year** | **color** | **horsepower** | **price** | **electricity** | **...** |
| 2018 | red | 283 | 34.990 | TRUE | ... |
| 2019 | black | 313 | 50.000 | FALSE | ... |
| ... | ... | ... | ... | ... | ... |

We see that values in the manufacture yearcolumn are integer numbers, values in price are decimal, and values in electricity are boolean. SQL databases usually require that each column in a database table has a name and a **data type**. The columndata type restricts the set of values that can be stored in the column and defines a set of possible operations on them.

ANSI standard defines a pretty complex set of data types. Besides, database vendors usually add their non-standard data options. In this topic, we will consider a very basic subset of data types: INTEGER, FLOAT, DECIMAL, VARCHAR, and BOOLEAN.

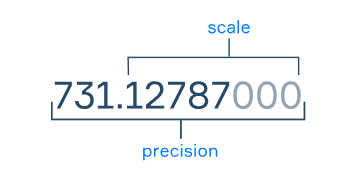
**Numerical data types**

INTEGERis a numeric data type that represents some range of mathematical integers (usually from -2147483648 to +2147483647).INTEGER type is good for counters, numeric identifiers, and any integer business value you can imagine that fits the scale range.

In everyday life, we usually face decimal numbers quite a lot: for example, when measuring body temperature (36.6 degrees Celsius) or counting our precious finances ($103050.79). SQL supports a special data type for such values – DECIMAL(precision, scale).

This type has two parameters: precision and scale.

* **Scale**is the count of digits to the right of the decimal point.
* **Precision**is the total count of digits in the number.



The FLOATdata type is an approximate numeric data type used for floating-point numbers. With the FLOAT data type, we can store very large or very small numbers. Also, it is used for calculations that require fast processing. The FLOAT data type has an optional parameter **n** that specifies the precision and storage size (from 1 to 53).

By the way, sometimes in SQL, you can encounter the REALdata type. And so REAL is FLOAT(24), or FLOAT of certain accuracy.

**Text**

Of course, you may want to process something other than numeric data, and SQL supports a family of data types designed to represent text data. Let's consider one of them, quite universal and basic – VARCHAR(n).

This type represents a string of symbols of varying lengths not longer than n. For example, one can insert the strings apple, plum, and peach into a column with the type VARCHAR(5). The strings orange and banana will exceed the length restriction and the system will either truncate them or generate an error if one tries to insert such long values.

**Boolean**

The BOOLEAN type represents boolean logic values: either TRUE or FALSE. This simple data type can be utilized for any attributes with flag semantics, for example, whether a client has visited a competitor's site.

**Who defines types and how?**

As a database user, you should just know the types of table columns you utilize to be able to process them correctly. However, as a software engineer, you should also know how to create a table and define the column types.

Let's consider an example of an SQL query that defines a table census with five columns: id of type INTEGER, name of type VARCHAR(20), birth\_place\_latitude of type REAL, year\_income of type DECIMAL(20,2), and is\_parent of type BOOLEAN.

CREATE TABLE census (

id INTEGER,

name VARCHAR(20),

birth\_place\_latitude REAL,

year\_income DECIMAL(20,2),

is\_parent BOOLEAN

);

One may see the following pattern:

CREATE TABLE table\_name (

column\_name\_1 column\_type\_1,

...,

column\_name\_n column\_type\_n

);

**Conclusion**

Data may be very diverse, and SQL supports an extensive set of data types to represent this diversity. We have discussed a basic subset of data types just to start with, yet there is more to the topic: type casting, compound types, special types for numeric data, text, timestamps, and so on.

Read more on this topic in [Relational vs. Non-relational Showdown](https://hyperskill.org/blog/post/relational-and-non-relational-databases) on Hyperskill Blog.

The table Car has the following columns:

* an integer identifier
* a textual description of a color
* an amount of the annual petrol usage
* the price in US dollars
* whether it is electric

Match the column names with the corresponding data types.

| **Column Name** | **Data Type** |
| --- | --- |
| **electricity** | ✅ **BOOLEAN** |
| **color** | ✅ **VARCHAR(20)** |
| **price** | ✅ **DECIMAL(20,2)** |
| **id** | ✅ **INTEGER** |
| **annual\_petrol\_use** | ✅ **REAL** |

How to define a column name where a string will be stored with a maximum length of 30 characters? name VARCHAR(30)

To restrict the set of values that can be stored in a column and define the set of possible operations on them, a software engineer should specify the column's...

The correct answer is:

✅ **type**

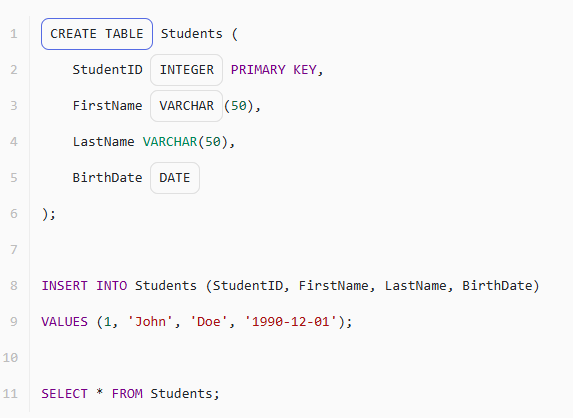
### Explanation:

The **column's type** (also known as data type) defines:

* The **kind of data** it can store (e.g., integer, text, boolean).
* The **set of valid values** and **operations** applicable to it.

Which data type would you choose for the column street address in the table Customer? VARCHAR(200)

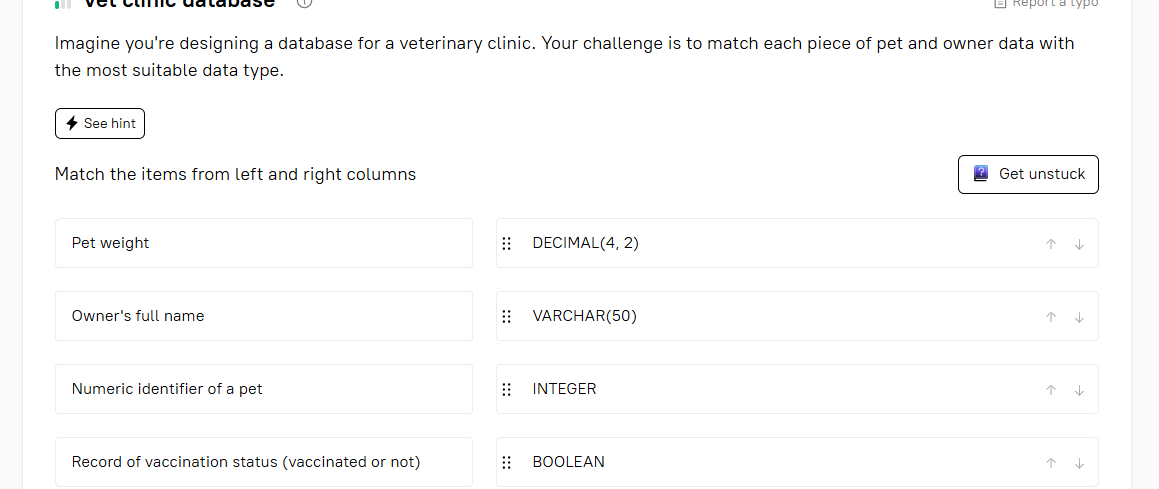
As the new database manager at a school, you will need to maintain a SQL database that contains a list of students. Each student has an ID, first name, last name, and birth date. Given the following SQL code with some missing parts (represented as ▭), your task is to fill in the blanks to define the correct SQL data types for the Students table's columns and then add a new student to this table with a SELECT statement returning all the students.



Which data type would you choose to store information about whether a client of an insurance company has a chronic disease? BOOLEAN

Which data type would you choose for a column with the number of clicks on a banner for a specific date (assuming there are not more than 1000 clicks per day)? INTEGER

Imagine you're designing a database for a veterinary clinic. Your challenge is to match each piece of pet and owner data with the most suitable data type.



Define the decimal type with 3 digits in the fractional part and 5 total digits.

**Hint**Decimal(Precision, Scale)- Precision: total count of digits- Scale: total count of digits to the right of the decimal point

To define a DECIMAL type with **5 total digits** and **3 digits in the fractional part**, use:

✅ **DECIMAL(5,3)**

### Explanation:

* **Precision (5)**: Total number of digits.
* **Scale (3)**: Number of digits after the decimal point.
* Example of a valid value: 12.345

What values can the boolean type take?

TRUE FALSE

## Fill in the blanks for SQL table creation

 Report a typo

Given an SQL query for creating a 'Students' table and inserting some data into it, there are a few crucial data types missing. Your task is to correctly fill in the blanks in the given SQL code to ensure it correctly creates the table with an 'ID' column of a type suitable for a unique identifier, a 'FullName' column for strings of characters, and a 'GPA' column for decimal numbers with precision and scale.

Fill in the gaps with the relevant elements

CREATE TABLE Students (

ID INT,

FullName VARCHAR(100),

Age INT,

GPA DECIMAL(3,2)

);

INSERT INTO Students (ID, FullName, Age, GPA)

VALUES (1, 'John Doe', 20, 3.5);

SELECT \* FROM Students;

## Difference between FLOAT and REAL

 Report a typo

What is the difference between the data type FLOAT and REAL ?

The correct answer is:

✅ **FLOAT defines the total number of digits by default but this is not the case with REAL**

### Explanation:

* **FLOAT**: You can specify precision (e.g., FLOAT(7)), which defines how many digits are stored.
* **REAL**: Typically has a **fixed** precision depending on the database system and doesn't allow specifying it.

reate a table called Patients with the following columns:

* patient\_id (integer)
* first\_name (30 characters or less)
* last\_name (30 characters or less)
* temperature (not higher than 100°C with no more than one digit after the decimal point)
* is\_discharged (indicates if a patient was discharged or not)

Use the DECIMAL() function to limit the number of digits in temperature.

Create table Patients(

patient\_id integer,

first\_name varchar(30),

last\_name varchar(30),

temperature Decimal(4,1) Check (temperature<=100),

is\_discharged boolean

)