## Introduction to DDL Commands

### Introduction

At the beginning of the course, we have mentioned that SQL performs some operations on the database such as retrieval, inserting, updating, creating. Of those, we have seen just SELECT statement for retrieval data operations on the database. As a side note, statements are also called "commands" in some sources.

Actually these statements are grouped into four main categories:

* DDL - Data Definition Language
* DML - Data Manipulation Language
* DCL - Data Control Language
* TCL - Transaction Control Language

[✱](https://www.toptal.com/designers/htmlarrows/symbols/heavy-asterisk/) Data Definition Language (DDL) specifies the database schema. The statements used in DDL are CREATE, ALTER, DROP. DDL statements are typically used to set up and configure a new database before we insert data.

[✱](https://www.toptal.com/designers/htmlarrows/symbols/heavy-asterisk/) Data Manipulation Language (DML) enables users to access or manipulate data. INSERT, UPDATE, DELETE, SELECT are the statements used in DML.

[✱](https://www.toptal.com/designers/htmlarrows/symbols/heavy-asterisk/) Data Control Language (DCL) is used to grant or revoke access control. Its statements are REVOKE and GRANT.

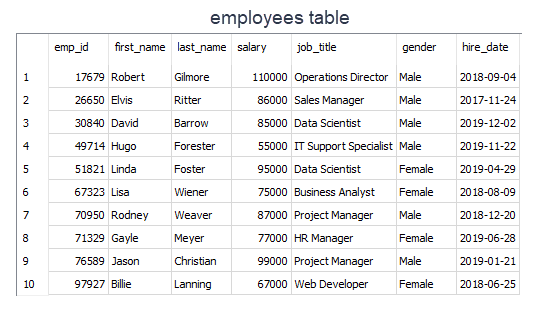
[✱](https://www.toptal.com/designers/htmlarrows/symbols/heavy-asterisk/) Transaction Control Language (TCL) controls the transactions of DML and DDL commands. Its statements are BEGIN and COMMIT.

DDL, DML, DCL, TCL are not separate languages. They compose parts of a single database language which is SQL. These are for categorical purposes. In this lesson, we'll learn DDL commands. As a side note, we are not going to cover other languages' commands in our course.

Before we dive into the DDL commands, we'll start with data types.

### Data Types

In SQL, data types define what type of data a column can contain or store. Let's find out the data types of the employees' columns. There is a special function typeof in SQLite to display the data types of the values stored in the columns.



query:

SELECT typeof(emp\_id),

typeof(first\_name),

typeof(hire\_date)

FROM employees

LIMIT 1;

output:

typeof(emp\_id) typeof(first\_name) typeof(hire\_date)

-------------- ------------------ -----------------

integer text text

As you see that emp\_id stores integer values, first\_name and hire\_date store textual values. Different database systems may use different data types though there are common ones and SQLite is no exception.  
  
In SQLite, there are five primitive data types. They are called **storage classes**. Here are the details of them:

* **NULL:**The value is a NULL value. NULL values mean missing information or unknown.
* **INTEGER:**The value is a whole number (positive or negative), stored in 1, 2, 3, 4, 6 or 8 bytes depending on the magnitude of the value.
* **REAL:** The value is a floating-point value, stored as an 8-byte float. They are real numbers with decimal values.
* **TEXT:** The value is a text string, stored using the database encoding (UTF-8, UTF-16BE or UTF-16LE).
* **BLOB:** BLOB stands for a binary large object that can store any kind of data. The value is a blob of data, stored exactly as it was input.

In SQLite, there is no separate Boolean storage class. Boolean values are stored as integers 0 (false) and 1 (true).  
SQLite doesn't have a storage class allocated for dates and/or times. Instead, you use the TEXT, INTEGER, or REAL to store the date and/or time values. 

**💡Tips:**

* A storage class is slightly more general than a data type. For example, the INTEGER storage class covers six different integer data types of different lengths. Most of the part, storage class is indistinguishable from data type and the two terms can be used interchangeably.

SQLite determines the data type of a value based on its data type in accordance with the following rules:

* If a value is enclosed by single or double quotes, it is assigned the TEXT storage class.
* If a value has no enclosing quotes and decimal point or exponent, it is assigned INTEGER storage class.
* If a value has no enclosing quotes, but decimal point or exponent, it is assigned REAL storage class.
* If a value is NULL without quotes, it is assigned NULL storage class.
* If a value has X'ABCD' or x'abcd', it is assigned BLOB storage class.

### Example

As we've learned from the previous part, we can use the typeof() function to check the storage class of value based on its format. Let's look at the examples below. As a side note, you don't always need to use FROM table\_name clause if not needed.

query:

SELECT

typeof('100'),

typeof(100),

typeof(10.0),

typeof(x'1000'),

typeof(NULL);

output:

typeof('100') typeof(100) typeof(10.0) typeof(x'1000') typeof(NULL)

------------- ----------- ------------ --------------- ------------

text integer real blob null

As you clearly see that the format of the values determines the storage class of the values. You may think that the result of typeof('100') is an integer. But there are enclosed single quotes in 100. Remember the rule that if a value is enclosed with single quote or double quotes, it is stored as text.  
  
Alright, it's time to start DDL commands. We'll start with CREATE.