Udemy Training Course

Started: 4/24/2023

Completed:

[Course Link](https://avicado.udemy.com/course/postgresqlmasterclass/learn/lecture/22230284#overview)

**PostgreSQL Bootcamp: Go From Beginner to Advanced, 60+ hours**

**Introduction to PostgreSQL**

Incoming calls port number: 5432

In order to connect to a server in PG Admin, click the server to the left hand side of the screen and enter your password, you will now be connected to the server.

You can create a user within the ‘Login/Group Roles’ section. When creating that user, PostgreSQL actually adds an entry in the database using a CREATE ROLE statement that characterizes the user privileges based on information entered in the GUI

**Query Order of Execution:**

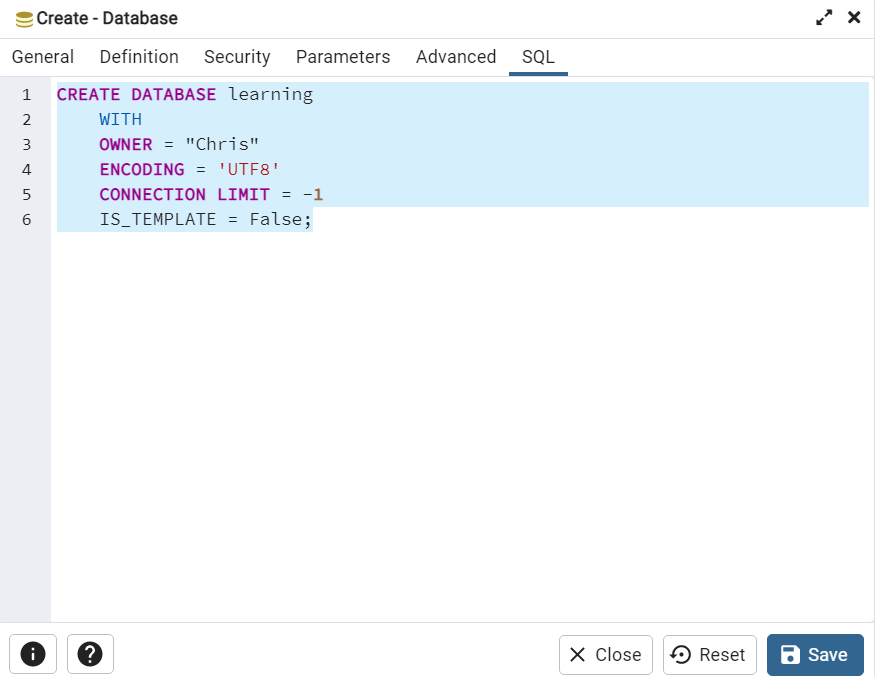
SQL executes the WHERE clause after the FROM clause but before the SELECT and ORDER BY clauses. Generally, the sequence goes something like:

FROM -> WHERE -> SELECT -> ORDER BY

**Random Facts:**

1. Table names and keywords are not case sensitive
2. || will concatenate data from different fields
3. When querying data from a table, the data is not returned in a specific order

**Creating a database:**

In SQL, the command looks like:  
  


When creating a database, set encoding to UTF8, which is an industry standard for storing non-ASCII characters. UTF-8 allows flexibility for the types of languages stored within the database, Chinese and Arabic, for example.

**UTF-8**:  
UTF-8 is a widely used character encoding that can represent virtually every character in the Unicode standard. Here are some pros and cons of using UTF-8:

Pros:

Compatibility: UTF-8 is widely used and supported, making it compatible with a wide range of software and systems. It is also the default encoding for many programming languages and web standards.

Multilingual support: UTF-8 can represent characters from multiple scripts and languages, including Latin, Cyrillic, Greek, Arabic, Chinese, Japanese, and Korean, among others. This makes it a good choice for applications that need to support multiple languages.

Compact representation: UTF-8 uses a variable-length encoding scheme that can represent any Unicode character using one to four bytes. This means that it is more efficient in terms of storage space than some other encodings that use a fixed-length scheme.

Cons:

Complexity: UTF-8 is more complex than some other encodings, such as ASCII, because it uses a variable-length scheme. This can make it more difficult to work with in some cases.

Processing overhead: Because UTF-8 uses a variable-length encoding scheme, it can require more processing power to manipulate than some other encodings.

Misinterpretation: If a file or document is not properly marked as being encoded in UTF-8, it can be misinterpreted by software and display incorrectly.

In general, UTF-8 is a good choice for applications that need to support multiple languages or scripts. Its compatibility, multilingual support, and efficient use of storage space make it a popular choice for web applications and other software systems. However, it may not be the best choice for all applications, especially those that require very high performance or that need to work with legacy systems that use different encodings.

**Database Schema:**

In PostgreSQL, a database schema is a named collection of database objects, including tables, views, indexes, sequences, and other schema-level constructs. A schema provides a way to organize database objects into logical groups and namespaces, making it easier to manage and understand a database's structure.

Here are some of the functions and relevance of a database schema in PostgreSQL:

Namespace management: A schema provides a way to manage the namespace of database objects. By organizing objects into separate schemas, you can avoid naming conflicts and make it easier to find and reference specific objects.

Access control: A schema can be used to control access to specific objects or groups of objects. By setting permissions on a schema, you can control who can view or modify the objects it contains.

Logical organization: By grouping related objects into a schema, you can create a logical organization of the database that reflects the structure of the data itself. This can make it easier to understand and maintain the database.

Multi-tenancy support: PostgreSQL supports multi-tenancy through the use of schemas. By creating a separate schema for each tenant, you can isolate their data and avoid conflicts between different tenants.

Extension management: PostgreSQL extensions often define their own database objects, such as tables, views, and functions. By creating a separate schema for an extension, you can isolate its objects and avoid naming conflicts with other extensions or with user-defined objects.

In summary, a database schema in PostgreSQL provides a way to organize database objects into logical groups and namespaces, making it easier to manage and understand the database's structure. It also supports access control, logical organization, multi-tenancy, and extension management.

**Dropping a database:**

Only super users can drop a database.

In order to drop a database, you may need to terminate/disconnect other users. To see connections, use:  
select \* from pg\_stat\_activity;

Then get the PID and end the session using:

SELECT pg\_terminate\_backend(<PID Number>);

Finally, you can use:

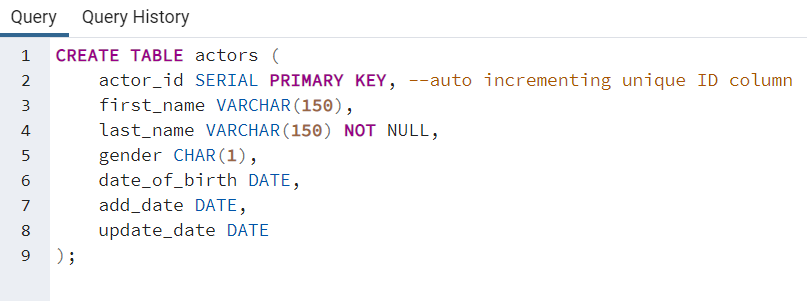
DROP DATABASE <database name>;

**Creating a table:**

CREATE TABLE syntax is very similar to MySQL, but when creating a unique, auto populating ID field, the term ‘SERIAL’ is used instead of ‘AUTO INCREMENT’.

CHAR - CHAR is a fixed-length character string data type in PostgreSQL. It means that when you define a column as CHAR(10), it will always use 10 bytes of storage space to store the value, even if the actual value is shorter than 10 characters. If the value is shorter than the defined length, it will be padded with spaces to fill up the remaining space. This data type is useful when you need to store values of a fixed length, such as postal codes or phone numbers.

VARCHAR - VARCHAR is a variable-length character string data type in PostgreSQL. It means that the length of the value stored in the column can vary up to the maximum length specified when the column is defined. If the actual value is shorter than the defined length, it will only use the necessary storage space to store the value without any padding. This data type is useful when you need to store values of varying lengths, such as names or addresses. As the VARCHAR data type is variable-length, it uses only the necessary storage space to store the actual values without any padding.



**Changing Column Data Types**

Column data types can be changed in the PG Admin GUI, but only to ‘adjacent’ data types. For instance, a ‘char’ could be changed to character, character varying, or text, but not integer. To change column types to unrelated types, you must use the query editor.

**Dealing with Quotes**

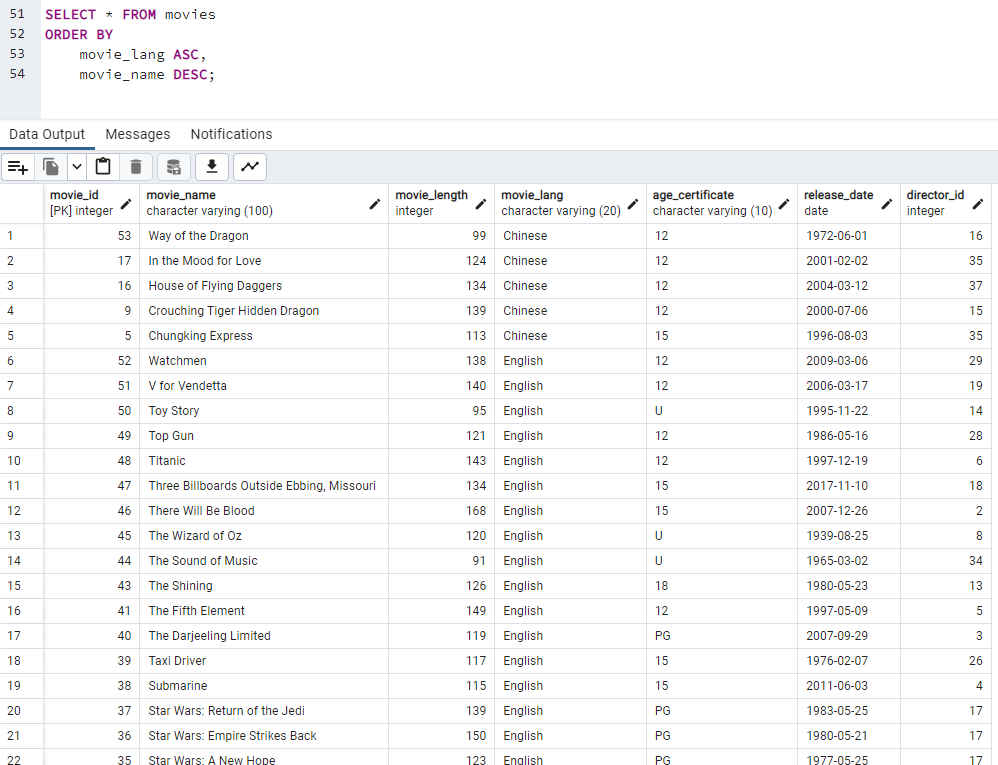
When inserting text data that uses a single quote, just add another single quote directly after the one that should appear in the text. Appears to act as an escape character since it isn’t wrapping the single quote in another pair of quotes. Example, if we wanted to INSERT INTO the name Bill O’Sullivan, we would do: ‘Bill O’’Sullivan.

**Updating values in the table**

Updating column values in postgres is very similar to MySQL, but you can set a uniform value for all records without needing to write a dummy where clause, which is nice.

**Sorting**

Data can be sorted by multiple variables in postgres, to do so simply add multiple fields to the ORDER BY clause:



**Operators**

Operators in SQL are keywords that are used in conjunction with clauses to:

1. Compare values between fields
2. Select subsets of fields
3. Perform arithmetic operations

More generally, in computer science, an operator is a symbol or function that performs an operation on one or more operands, which are values or variables that the operator operates on. The operation performed by an operator can be mathematical, logical, or bitwise.

In programming languages, operators are used to manipulate data and perform various calculations or comparisons. They can be used to assign values to variables, compare values, perform arithmetic calculations, and more. For example, the addition operator (+) is used to add two numbers together, and the assignment operator (=) is used to assign a value to a variable.

Types:

1. Comparison
   1. Equal To =
   2. Greater Than >
   3. Less Than <
   4. Greater Than or Equal To >=
   5. Less Than or Equal To <=
   6. Not Equal To <>, !=
2. Logical
   1. AND
   2. OR
   3. LIKE
   4. IN
   5. BETWEEN
3. Arithmetic
   1. Addition +
   2. Subtraction -
   3. Division /
   4. Multiply \*
   5. Modulo %

When used in the same SELECT statement, the AND operator is processed first and the OR operator is processed second.