

"Hallo homie, hope you be pretty good and keep yourself as I left you."

Topics

- "Hallo homie, hope you be pretty good and keep yourself as I left you."
 - Topics
 - Important websites
 - Important Notes
 - Commands & Functions

Important websites

1. [Mockaroo](#)
2. [PostgreSQL](#)

Important Notes

- **help** can be used only when = exist
- THE DIFFERENCE BETWEEN = & - \ & / just you can imagine the commands in the boxes or groups there is a key to open the whole box \ & = and there is a different one which opens the smaller one / & - ,if you want to exit from one group press ;
- ▶ **Foto**
- **COUNT(*)** is a fun from aggregation functions there is a lot of them you can visit this site to know more [postgres](#) in this like you may face this fun like **max()** , **min()** , **avr()** , **round** note aggregate function calls can't be nested
- you have only one **ELSE** and **END** per **CASE** but multi **WHEN**
- If you want to delete a row in sub-table you must at first delete any record has this foreign key this will be done by 3 ways the first is to delete the record ,second is to update these columns by NULL value third is to make cascade method but this is a bad practice because it will delete every record that has this foreign key
- A nice trick when the foreign key and primary key has the same name it is better to use **USING**
- **EXTENSIONS** to know what is the new function that has been added press \df we used uuid instead of original id with only integer number because it is so difficult for attackers to guess the exact id and it is possible to migrate two databases without any clashes because all id is different not like just integers 1,2,3,4,5
- ▶ **Foto**
- to export your table to a CSV file you type
- ▶ **Foto**
- to know what is the last id you have or the last variable that type is bigserial is you can do this by or if you want to increment these bigserial just use the entire function under default after SELECT
- ▶ **Foto**

Commands & Functions

- **\! CLS**** TO CLEAR SCREEN

- \c to connect to another database you can connect to specific things not all database use commands below ,but \c is easy and common

► Foto

- CREATE DATABASE "NAME" & DROP DATABASE "NAME" create and remove whole database just like DELETE FROM "NAME" but this delete only table or column if you specific this

► Foto

- CREATE TABLE **men** to create table in my database **person** but first must be connect to this database , you have to know the [data types](#) \dt show details about tables in your current database

► Foto

- to insert new record use enter the data between ' ' note that if you enter date you must enter DATE '2002-03-30'

► Foto

to insert multi records use \i command and put the path of your SQL file note that / in your path not \if you want any date visit Mockaroo

► Foto

- To Read from tables use SELECT

► Foto

- to sort cloums use ORDER BY with DESC or ASC (however ASC is the defult) , this command is the last command we write

► Foto

- to remove duplicateied date and to only know what is the unique date that we have and how many use DISTINCT

► Foto

- you can also add conditions by using WHERE keyword you can inaddtion use AND OR to have specific condition ,you have a compersion operator like < ,> , =,<> ,<=,>= and it will return t or f or return the query that provides these conditions

► Foto

- to get range of some number you can use BETWEEN that have the same functionality

► Foto

- you can know how many this data apper or duplictied by using GROUP BY key word but you should use also COUNT()

► Foto

- to make any condtions on grop we use HAVING and put the condition in COUNT(*)

► Foto

- there is a fun to bring data -used with **WHERE** -that is like to what you looking for is called **LIKE** or **ILIKE** there are not pretty diffrenet between them only **ILIKE** is not case sensitive... you can add a ouch if underscores to indicate how many letters you want to be found only in your query

► Foto

- **AS** key word change the name of the column that you have added or any variable or long name you have to make it easy to operate with them and we write it at the end of name of any column

► Foto

- **COALESCE** key word used to fill null values with a specific character ,**Note that** don't forget ' '

► Foto

- **NULLIF** key word used to check if the two passing numbers are equal so it returns **NULL** or return the very first value it is good way to handle divide by zero instead of get error message you get null or zero if you use **COALESCE**

► Foto

- **NOW()** function return an actual date with time and time zone if you want a piece of it just type **::** , if you want a specific part you can use **EXTRACT**

► Foto

, if you want to add or subtract date from each other use **INTERVAL** ' ' and you can write any time between periods 1 YEAR, 6 MONTH ,10 DAY , there is also a special fun from **INTERVAL** called **AGE('ending','starting')**

► Foto

- if you want to add a **PRIMARY KEY** constraint[for rows] or **UNIQUE** constraint [for column] use **ALTER TABLE**

► Foto

to remove any of them use **ALTER TABLE 'name' DROP**

► Foto

CHECK(condition) that can be drive to table by the same way to declare any constraint it used to put a specific constraint depends on the condition--- to remove a specific rows use **DELETE FROM men WHERE** to remove all table use **DROP TABLE men ;** or **DELETE FROM men ;**

► Foto

- **CASE** keyword is the correspondent(equivalent) of **IF** in the another language However PostgreSQL has its **IF** conditions too it only return **0** or **1**

► Foto

- **UPDATE** to update(edit or add) a column or more in one record or more depend on **WHERE** Clause record

► Foto

- to handel duplicate key conflict which comes throw because of constraints by using **ON CONFLICT** (name of any column has a constraint either primray key or unique constraint) **DO NOTHING** this is useful when database catch a conflict it will continue and go on it will only do nothing ,, but if you want to udate some date or client enter two diffrent data you must pick up throw lastest one so you want to do UPDATE this is allowed using this **ON CONFLICT (id) DO UPDATE SET email= EXCLUDED.email;** **NOTE that** the first **email** referece to the **email** which stored and **.email** referece to **email** that just pass or want to insert you can add columns as you want to change but remmber to change it first when you insert these using **INSERT**

► Foto

- **Relation** between tables in postgresQL it will be by using Foreign key it is a primary key that refrece to another primary key in another table and be wrriten in table details when be created in the first time or be added like that **car_id BIGINT REFERENCES car(id), UNIQUE (car_id) ;** which **id** referece to **id** in another table and **car_id** referece to th actual table the most important thing you must write sub-tables in the first then write the main one which has a relations with them

► Foto

- **inner joins*** we can grab data from table and its relations by using **JOIN** like that ,after JOIN we write what is the point or column that makes relation between these table ,, write it like Objects
- **LEFT JOIN** is like **JOIN** but in **JOIN** it discards the data that has NULL value in the relation column or the foregin key however **LEFT JOIN** grab all data in the main table with the intersections too ,like **JOIN**

► Foto