

SQL

January 31, 2018

1 SQL Overview

Guía completa: <http://www.postgresqltutorial.com>

1.1 Requisitos:

- Docker (Windows y OSX: <http://kitematic.com/>)
- pgAdmin <https://pgadmin.org>
- Python (modulos: sqlalchemy, psycopg2)
- Sample database: <http://www.postgresqltutorial.com/download/dvd-rental-sample-database/?wpdmdl=969>

1.2 Descripción de la DB a usar

-DVD Rental database - 15 tablas - 1 trigger - 7 vistas - 8 funciones - 1 dominio - 13 secuencias

1.3 Modelo de Entidad-Relación

1.4 Tablas

1. actor - datos de actor (primer nombre y apellido)
2. film - datos propios de filmes: título, año de publicación, duración, rating, etc
3. film_actor - relaciones entre files y actores
4. category - categorías para filmes
5. film_category - relaciones entre filmes y categorías
6. store - datos de tiendas, empleados y dirección
7. inventory - datos de inventario
8. rental - datos de rentas efectuadas
9. payment - datos sobre pagos de clientes
10. staff - datos del personal en tienda
11. customer - datos de los clientes
12. address - direcciones de personal en tienda y clientes
13. city - nombres de ciudades
14. country - nombres de países
15. language - lenguajes disponibles para filmes

1.5 Cargar DB en PostgreSQL usando pgAdmin

- Crear usuario hackerschool con password datasciclass
- Cargar DB y asignarla al usuario default (postgres)
- Explorar la DB con pgAdmin

```
In [48]: # Load dependencies
import psycopg2, pandas

# Set up connection parameters
db_connection = psycopg2.connect(host="192.168.99.100", port="32768",
                                user="postgres", password="",
                                dbname="dvdrental")

# Connect to the database server
db_cursor = db_connection.cursor()
```

1.6 CRUD - Create, Read, Update & Delete

1.6.1 Read (SELECT)

```
SELECT column_1,
       column_2,
       ...
FROM table_name
WHERE conditions
ORDER BY
  column_1 ASC,
  column_2 DESC;
```

```
In [49]: # Basic select
sql_query = "SELECT first_name, last_name, email FROM customer;"

dataframe_from_query = pandas.read_sql_query(sql_query, db_connection)
dataframe_from_query
```

```
Out[49]:
```

	first_name	last_name	email
0	Jared	Ely	jared.ely@sakilacustomer.org
1	Mary	Smith	mary.smith@sakilacustomer.org
2	Patricia	Johnson	patricia.johnson@sakilacustomer.org
3	Linda	Williams	linda.williams@sakilacustomer.org
4	Barbara	Jones	barbara.jones@sakilacustomer.org
5	Elizabeth	Brown	elizabeth.brown@sakilacustomer.org
6	Jennifer	Davis	jennifer.davis@sakilacustomer.org
7	Maria	Miller	maria.miller@sakilacustomer.org
8	Susan	Wilson	susan.wilson@sakilacustomer.org
9	Margaret	Moore	margaret.moore@sakilacustomer.org
10	Dorothy	Taylor	dorothy.taylor@sakilacustomer.org
11	Lisa	Anderson	lisa.anderson@sakilacustomer.org

12	Nancy	Thomas	nancy.thomas@sakilacustomer.org
13	Karen	Jackson	karen.jackson@sakilacustomer.org
14	Betty	White	betty.white@sakilacustomer.org
15	Helen	Harris	helen.harris@sakilacustomer.org
16	Sandra	Martin	sandra.martin@sakilacustomer.org
17	Donna	Thompson	donna.thompson@sakilacustomer.org
18	Carol	Garcia	carol.garcia@sakilacustomer.org
19	Ruth	Martinez	ruth.martinez@sakilacustomer.org
20	Sharon	Robinson	sharon.robinson@sakilacustomer.org
21	Michelle	Clark	michelle.clark@sakilacustomer.org
22	Laura	Rodriguez	laura.rodriguez@sakilacustomer.org
23	Sarah	Lewis	sarah.lewis@sakilacustomer.org
24	Kimberly	Lee	kimberly.lee@sakilacustomer.org
25	Deborah	Walker	deborah.walker@sakilacustomer.org
26	Jessica	Hall	jessica.hall@sakilacustomer.org
27	Shirley	Allen	shirley.allen@sakilacustomer.org
28	Cynthia	Young	cynthia.young@sakilacustomer.org
29	Angela	Hernandez	angela.hernandez@sakilacustomer.org
..
569	Ivan	Cromwell	ivan.cromwell@sakilacustomer.org
570	Johnnie	Chisholm	johnnie.chisholm@sakilacustomer.org
571	Sidney	Burleson	sidney.burleson@sakilacustomer.org
572	Byron	Box	byron.box@sakilacustomer.org
573	Julian	Vest	julian.vest@sakilacustomer.org
574	Isaac	Oglesby	isaac.oglesby@sakilacustomer.org
575	Morris	Mccarter	morris.mccarter@sakilacustomer.org
576	Clifton	Malcolm	clifton.malcolm@sakilacustomer.org
577	Willard	Lumpkin	willard.lumpkin@sakilacustomer.org
578	Daryl	Larue	daryl.larue@sakilacustomer.org
579	Ross	Grey	ross.grey@sakilacustomer.org
580	Virgil	Wofford	virgil.wofford@sakilacustomer.org
581	Andy	Vanhorn	andy.vanhorn@sakilacustomer.org
582	Marshall	Thorn	marshall.thorn@sakilacustomer.org
583	Salvador	Teel	salvador.teel@sakilacustomer.org
584	Perry	Swafford	perry.swafford@sakilacustomer.org
585	Kirk	Stclair	kirk.stclair@sakilacustomer.org
586	Sergio	Stanfield	sergio.stanfield@sakilacustomer.org
587	Marion	Ocampo	marion.ocampo@sakilacustomer.org
588	Tracy	Herrmann	tracy.herrmann@sakilacustomer.org
589	Seth	Hannon	seth.hannon@sakilacustomer.org
590	Kent	Arsenault	kent.arsenault@sakilacustomer.org
591	Terrance	Roush	terrance.roush@sakilacustomer.org
592	Rene	Mcalister	rene.mcalister@sakilacustomer.org
593	Eduardo	Hiatt	eduardo.hiatt@sakilacustomer.org
594	Terrence	Gunderson	terrence.gunderson@sakilacustomer.org
595	Enrique	Forsythe	enrique.forsythe@sakilacustomer.org
596	Freddie	Duggan	freddie.duggan@sakilacustomer.org
597	Wade	Delvalle	wade.delvalle@sakilacustomer.org

```
598      Austin      Cintron      austin.cintron@sakilacustomer.org
```

```
[599 rows x 3 columns]
```

```
In [50]: # Basic select 2
```

```
sql_query = "SELECT first_name, last_name FROM customer WHERE first_name = 'Amanda' "
```

```
dataframe_from_query = pandas.read_sql_query(sql_query, db_connection)
dataframe_from_query
```

```
Out[50]:   first_name last_name
0      Amanda      Carter
1         Amy      Lopez
```

```
In [ ]: ### Ejercicios:
```

```
- Seleccionar de la guía la query adecuada a cada sentencia
```

```
In [ ]: # ORDER BY
```

```
# http://www.postgresqltutorial.com/postgresql-order-by/
```

```
sql_query = ""
```

```
dataframe_from_query = pandas.read_sql_query(sql_query,db_connection)
```

```
In [ ]: # WHERE
```

```
# http://www.postgresqltutorial.com/postgresql-where/
```

```
sql_query = ""
```

```
dataframe_from_query = pandas.read_sql_query(sql_query,db_connection)
```

```
In [ ]: # LIMIT
```

```
# http://www.postgresqltutorial.com/postgresql-limit/
```

```
sql_query = ""
```

```
dataframe_from_query = pandas.read_sql_query(sql_query,db_connection)
```

```
In [ ]: # ORDER BY
```

```
# http://www.postgresqltutorial.com/postgresql-order-by/
```

```
sql_query = ""
```

```
dataframe_from_query = pandas.read_sql_query(sql_query,db_connection)
```

```
In [ ]: # INNER JOIN
```

```
# http://www.postgresqltutorial.com/postgresql-inner-join/
```

```
sql_query = ""
```

```
dataframe_from_query = pandas.read_sql_query(sql_query,db_connection)
```

```
In [ ]: # LEFT JOIN
        # http://www.postgresqltutorial.com/postgresql-left-join/
        sql_query = ""

        dataframe_from_query = pandas.read_sql_query(sql_query,db_connection)

In [47]: # Close connection
        db_connection.close()
```

1.6.2 Futura exploración:

- <http://www.postgresqltutorial.com/postgresql-insert/>
- <http://www.postgresqltutorial.com/postgresql-update/>
- <http://www.postgresqltutorial.com/postgresql-delete/>