

Answer of SQL and Phyton Exericse (Week 2)

This Notebook will answer all assignment questions that was a part of IYKRA Data Science Fellowship Submission.  
Part 1 : SQL Question

Part 2 : Phyton

Datasets from <http://www.postgresqltutorial.com/postgresql-sample-database/>

In order to connect postgresQL into Jupyter, there are several steps that we need to follow:

- 1. Download and install **iphyton-sql**, **psycpg2-binary** , and **sqlalchemy** (By using !pip install)

```
**iphyton-sql** --> use for SQL Magic (%% SQL)

**sqlalchemy** --> help SQL function save in pandas phyton
```

In [18]: `#!pip install ipython-sql
#!pip install psycpg2-binary
#!pip install sqlalchemy`

- 1. %load\_ext sql to load sql function in notebook

In [5]: `%load_ext sql`

- 1. From sqlalchemy import create\_engine for make datasets into available in notebook

In [6]: `from sqlalchemy import create_engine`

- 1. Connect PostGreSQL database with Notebook with formula : %sql dialect+driver://username:password@host:port/database

In [7]: `%sql postgresql://postgres:Dedebotak2305@localhost/dvdrental`

- 1. Connect database with sqlalchemy with engine function. Formula = create\_engine('dialect+driver://username:password@host:port/database')

In [8]: `engine = create_engine('postgresql://postgres:Dedebotak2305@localhost/dvdrental')`

Thanks to Muti Siahaan (<https://github.com/moowtea/iykra/blob/master/SQL%20and%20Python%20Exercise.ipynb>) and Andrei Teleron (<https://medium.com/analytics-vidhya/postgresql-integration-with-jupyter-notebook-deb97579a38d>) who helped me to understand about how to import SQL to Jupyter Notebook

Part 1 : SQL

A customer wants to know the films about **"Astronout"**. How many recommendations could you give for him?

In [10]: `%%sql

select count(*) as Total
from film f
where f.description @@ to_tsquery('astronaut');

* postgresql://postgres:***@localhost/dvdrental
1 rows affected.`

Out[10]: 

| total |
|-------|
| 78    |

How many film have a rating of 'R' and a replacement cost between \$5 to \$15?

In [11]: `%%sql

select count(*) as Total
from film
where rating = 'R' and replacement_cost between 5 and 15;

* postgresql://postgres:***@localhost/dvdrental
1 rows affected.`

Out[11]: 

| total |
|-------|
| 52    |

How **many payment** did each staff member handle?  
How **\*\*much total amount\*\*** processed by each staff member?

In [12]: `%%sql

select concat(s.first_name, ' ',s.last_name) as Name, s.staff_id as id, count(*),sum(p2.amoun
t)
from staff s
inner join payment p2
on s.staff_id = p2.staff_id
group by s.staff_id
order by 3 desc;

* postgresql://postgres:***@localhost/dvdrental
2 rows affected.`

Out[12]: 

|  | name         | id | count | sum      |
|--|--------------|----|-------|----------|
|  | Jon Stephens | 2  | 7304  | 31059.92 |
|  | Mike Hillyer | 1  | 7292  | 30252.12 |

Final Answer = **Jon Stephens (ID : 2)** should get a bonus.

**Average Replacement Cost By Rating:**

In [13]: `%%sql

select rating, avg(replacement_cost) as average from film
group by rating
order by 1 asc;

* postgresql://postgres:***@localhost/dvdrental
5 rows affected.`

Out[13]: 

| rating | average             |
|--------|---------------------|
| G      | 20.1248314606741573 |
| PG     | 18.9590721649484536 |
| PG-13  | 20.4025560538116592 |
| R      | 20.2310256410256410 |
| NC-17  | 20.1376190476190476 |

5 customers who spend **most amount** of money:

In [14]: `%%sql

select concat(c2.first_name, ' ',c2.last_name) as Name, c2.email, sum(p2.amount) as Total
from customer c2
inner join payment p2
on c2.customer_id = p2.customer_id
group by 1,2
order by 3 desc
limit 5;

* postgresql://postgres:***@localhost/dvdrental
5 rows affected.`

Out[14]: 

|  | name           | email                             | total  |
|--|----------------|-----------------------------------|--------|
|  | Eleanor Hunt   | eleanor.hunt@sakilacustomer.org   | 211.55 |
|  | Karl Seal      | karl.seal@sakilacustomer.org      | 208.58 |
|  | Marion Snyder  | marion.snyder@sakilacustomer.org  | 194.61 |
|  | Rhonda Kennedy | rhonda.kennedy@sakilacustomer.org | 191.62 |
|  | Clara Shaw     | clara.shaw@sakilacustomer.org     | 189.60 |

How **many copies** of each movie in each store do we have?  
(To see all answers, please erase the **\*\*limit\*\***)

In [15]: `%%sql

select i.film_id, f.title, a2.address,count(*) as Total
from (inventory i
inner join film f on i.film_id = f.film_id)
inner join store s2 on i.store_id = s2.store_id
inner join address a2 on s2.address_id = a2.address_id
group by 1,2,3
order by 1,2,3
Limit 10;

* postgresql://postgres:***@localhost/dvdrental
10 rows affected.`

Out[15]: 

| film_id | title            | address            | total |
|---------|------------------|--------------------|-------|
| 1       | Academy Dinosaur | 28 MySQL Boulevard | 4     |
| 1       | Academy Dinosaur | 47 MySakila Drive  | 4     |
| 2       | Ace Goldfinger   | 28 MySQL Boulevard | 3     |
| 3       | Adaptation Holes | 28 MySQL Boulevard | 4     |
| 4       | Affair Prejudice | 28 MySQL Boulevard | 3     |
| 4       | Affair Prejudice | 47 MySakila Drive  | 4     |
| 5       | African Egg      | 28 MySQL Boulevard | 3     |
| 6       | Agent Truman     | 28 MySQL Boulevard | 3     |
| 6       | Agent Truman     | 47 MySakila Drive  | 3     |
| 7       | Airplane Sierra  | 28 MySQL Boulevard | 3     |

Customers who eligible for the credit card (**minimum 40 transactions**) :

In [16]: `%%sql

select concat(c2.first_name, ' ',c2.last_name) as Name, c2.email, count(*) as Total
from customer c2
inner join payment p2
on c2.customer_id = p2.customer_id
group by 1,2
having count(*) >= 40
order by 3;`

Out[16]: 

|  | name         | email                           | total |
|--|--------------|---------------------------------|-------|
|  | Clara Shaw   | clara.shaw@sakilacustomer.org   | 40    |
|  | Karl Seal    | karl.seal@sakilacustomer.org    | 42    |
|  | Eleanor Hunt | eleanor.hunt@sakilacustomer.org | 45    |

Part 2 : Phyton

Please make a phyton function that introduces your name, address, DOB, and print them into one sentence!

In [17]: `name = input('Input Name: ')
address = input('Input Address: ')
dob = input('Input D.O.B: ')

print('My name is ' + str(name) + ', I live in ' + str(address) + ', I was born on ' + str(do
b) +'.')`

Input Name: Ravelto Wangistu  
Input Address: Tangerang  
Input D.O.B: May 23,1998  
My name is Ravelto Wangistu, I live in Tangerang, I was born on May 23,1998.

Thank You and Have a nice day! For any suggestion, please chat me through github.

In [ ] :