### **Module 1 Homework**

### **Docker & SQL**

In this homework we'll prepare the environment and practice with Docker and SQL

# **Question 1. Knowing docker tags**

Run the command to get information on Docker

```
docker --help
```

Now run the command to get help on the "docker build" command:

```
docker build --help
```

Do the same for "docker run".

Which tag has the following text? - Automatically remove the container when it exits

- --delete
- --rc
- --rmc
- --rm

#### ANSWER: -RM = automatically remove the container when it exists

```
--restart string Restart policy to apply when a container exits (defau
--rm Automatically remove the container when it exits
--runtime string Runtime to use for this container
```

## Question 2. Understanding docker first run

Run docker with the python:3.9 image in an interactive mode and the entrypoint of bash. Now check the python modules that are installed ( use pip list ).

What is version of the package wheel?

- 0.42.0
- 1.0.0
- 23.0.1
- 58.1.0

**ANSWER:** wheel version = 0.42.0

```
☑ docker run -it --entrypoint=bash python:3.9
root@9714d5c9ec09:/# pip list
Package Version
-----
pip 23.0.1
setuptools 58.1.0
wheel 0.42.0
```

# **Prepare Postgres**

Run Postgres and load data as shown in the videos We'll use the green taxi trips from September 2019:

```
wget
```

```
https://github.com/DataTalksClub/nyc-tlc-data/releases/download/green/green_tripdata_2019-09.csv.gz
```

You will also need the dataset with zones:

```
wget https://s3.amazonaws.com/nyc-tlc/misc/taxi+_zone_lookup.csv
```

Download this data and put it into Postgres (with jupyter notebooks or with a pipeline)

### **Question 3. Count records**

How many taxi trips were totally made on September 18th 2019?

Tip: started and finished on 2019-09-18.

Remember that lpep\_pickup\_datetime and lpep\_dropoff\_datetime columns are in the format timestamp (date and hour+min+sec) and not in date.

- 15767
- 15612
- 15859
- 89009

**ANSWER: 15612** 

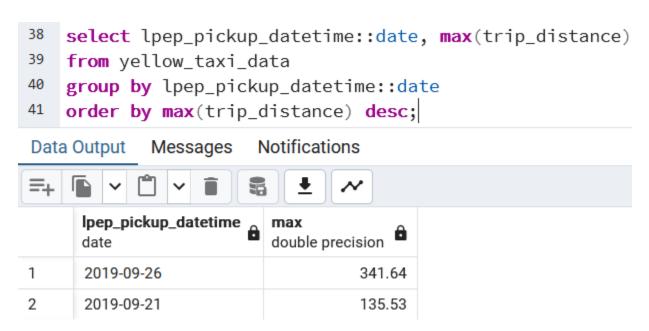
```
select count(1)
43
44
   from yellow_taxi_data
45
   where lpep_pickup_datetime::date = '2019-09-18'
   and lpep_dropoff_datetime::date = '2019-09-18';
46
Data Output
                       Notifications
            Messages
≡₊
     count
     bigint
1
         15612
```

## Question 4. Largest trip for each day

Which was the pick up day with the largest trip distance Use the pick up time for your calculations.

- 2019-09-18
- 2019-09-16
- 2019-09-26
- 2019-09-21

#### ANSWER: 2019-09-26



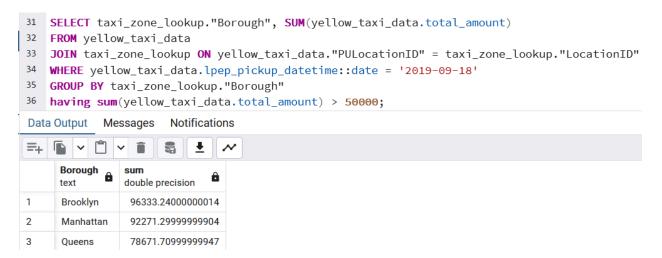
### Question 5. Three biggest pick up Boroughs

Consider Ipep\_pickup\_datetime in '2019-09-18' and ignoring Borough has Unknown

Which were the 3 pick up Boroughs that had a sum of total amount superior to 50000?

- "Brooklyn" "Manhattan" "Queens"
- "Bronx" "Brooklyn" "Manhattan"
- "Bronx" "Manhattan" "Queens"
- "Brooklyn" "Queens" "Staten Island"

#### Answer: Brooklyn, Manhattan, Queens



# Question 6. Largest tip

For the passengers picked up in September 2019 in the zone name Astoria which was the drop off zone that had the largest tip? We want the name of the zone, not the id.

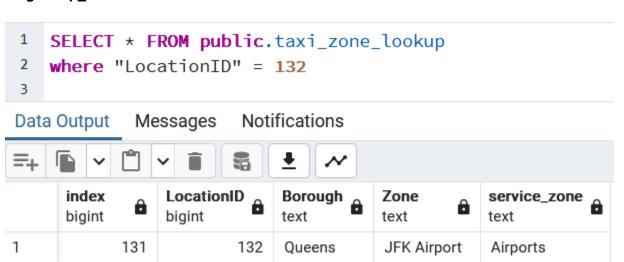
Note: it's not a typo, it's tip, not trip

- Central Park
- Jamaica
- JFK Airport
- Long Island City/Queens Plaza

**ANSWER: JFK airport** 

```
select yellow_taxi_data."DOLocationID", max(yellow_taxi_data.tip_amount)
24
   from yellow_taxi_data
   join taxi_zone_lookup
   on yellow_taxi_data."PULocationID" = taxi_zone_lookup."LocationID"
   where taxi_zone_lookup."Zone" = 'Astoria'
   group by yellow_taxi_data."DOLocationID"
28
   order by max(yellow_taxi_data.tip_amount) desc;
Data Output
            Messages
                      Notifications
=+
     DOLocationID
                  double precision
     bigint
1
              132
                            62.31
```

#### Largest tip\_amount came from DOLocationID of 132.



### **Terraform**

In this section homework we'll prepare the environment by creating resources in GCP with Terraform.

In your VM on GCP/Laptop/GitHub Codespace install Terraform. Copy the files from the course repo <a href="here">here</a> to your VM/Laptop/GitHub Codespace.

Modify the files as necessary to create a GCP Bucket and Big Query Dataset.

### **Question 7. Creating Resources**

After updating the main.tf and variable.tf files run:

terraform apply

Paste the output of this command into the homework submission form.

terraform apply

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:

+ create

Terraform will perform the following actions:

```
# google_bigquery_dataset.demo_dataset will be created
+ resource "google_bigquery_dataset" "demo_dataset" {
     + creation_time
                              = (known after apply)
                       = "demo_dataset"
     + dataset_id
     + default collation
                              = (known after apply)
     + delete_contents_on_destroy = false
     + effective_labels = (known after apply)
     + etag
                       = (known after apply)
     + id
                       = (known after apply)
     + is_case_insensitive
                             = (known after apply)
     + last_modified_time
                         = (known after apply)
                       = "US"
     + location
     + max_time_travel_hours = (known after apply)
```

```
+ project
                        = "rapid-will-412408"
     + self link
                        = (known after apply)
     + storage_billing_model = (known after apply)
     + terraform_labels
                              = (known after apply)
     }
# google_storage_bucket.demo-bucket will be created
+ resource "google_storage_bucket" "demo-bucket" {
     + effective_labels
                              = (known after apply)
     + force_destroy
                              = true
     + id
                        = (known after apply)
                        = "US"
     + location
                       = "rapid-will-412408-terra-bucket"
     + name
                        = (known after apply)
     + project
     + public_access_prevention
                                     = (known after apply)
     + self_link
                        = (known after apply)
                              = "STANDARD"
     + storage_class
     + terraform_labels
                              = (known after apply)
     + uniform_bucket_level_access = (known after apply)
                        = (known after apply)
     + url
     + lifecycle_rule {
     + action {
     + type = "AbortIncompleteMultipartUpload"
```

```
}
      + condition {
      + age
                          = 1
      + matches_prefix
                          = []
      + matches_storage_class = []
      + matches_suffix
                          = []
      + with_state
                          = (known after apply)
      }
      }
      }
Plan: 2 to add, 0 to change, 0 to destroy.
Do you want to perform these actions?
 Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.
 Enter a value: yes
google_bigquery_dataset.demo_dataset: Creating...
google_storage_bucket.demo-bucket: Creating...
google_bigquery_dataset.demo_dataset: Creation complete after 0s
[id=projects/rapid-will-412408/datasets/demo_dataset]
google_storage_bucket.demo-bucket: Creation complete after 1s
[id=rapid-will-412408-terra-bucket]
```

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.

# **Submitting the solutions**

- Form for submitting: <a href="https://courses.datatalks.club/de-zoomcamp-2024/homework/hw01">https://courses.datatalks.club/de-zoomcamp-2024/homework/hw01</a>
- You can submit your homework multiple times. In this case, only the last submission will be used.

Deadline: 29 January, 23:00 CET