Week-1 Solutions and Output

Q1. Install Google Cloud SDK. What's the version you have? (To get the version, run gcloud --version)

Command:

gcloud --version

Output:

Google Cloud SDK 369.0.0

Q2. Question 2: terraform apply

Command:

terraform apply

Output:

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.dataset will be created
+ resource "google_bigquery_dataset" "dataset" {
    + creation_time             = (known after apply)
    + dataset_id                = "trips_data_all"
    + delete_contents_on_destroy = false
    + etag                      = (known after apply)
    + id                        = (known after apply)
    + last_modified_time        = (known after apply)
```

```
    + location                  = "europe-west6"
    + project                   = "datasciene-338718"
    + self_link                 = (known after apply)

    + access {
        + domain        = (known after apply)
        + group_by_email = (known after apply)
        + role          = (known after apply)
        + special_group  = (known after apply)
        + user_by_email  = (known after apply)

        + view {
            + dataset_id = (known after apply)
            + project_id = (known after apply)
            + table_id   = (known after apply)
          }
      }
  }

# google_storage_bucket.data-lake-bucket will be created
+ resource "google_storage_bucket" "data-lake-bucket" {
    + force_destroy             = true
    + id                        = (known after apply)
    + location                  = "EUROPE-WEST6"
    + name                      = "dtc_data_lake_datasciene-338718"
    + project                   = (known after apply)
    + self_link                 = (known after apply)
```

```
      + storage_class              = "STANDARD"
      + uniform_bucket_level_access = true
      + url                        = (known after apply)

      + lifecycle_rule {
          + action {
              + type = "Delete"
            }

          + condition {
              + age                   = 30
              + matches_storage_class = []
              + with_state            = (known after apply)
            }
        }

      + versioning {
          + enabled = true
        }
    }

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.dataset: Creating...

google_storage_bucket.data-lake-bucket: Creating...

google_bigquery_dataset.dataset: Creation complete after 3s [id=projects/datasciene-338718/datasets/trips_data_all]

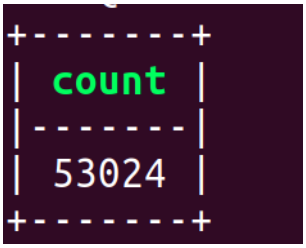
google_storage_bucket.data-lake-bucket: Creation complete after 7s [id=dtc_data_lake_datasciene-338718]

Q3: How many taxi trips were there on January 15?

Command:

```
select count(*) from yellow_taxi_data where extract(month from tpep_pickup_datetime)=1 and extract(day from tpep_pickup_datetime)=15
```

Output:



Q4: On which day it was the largest tip in January? (note: it's not a typo, it's "tip", not "trip")

Command:

```
select tip_amount,tpep_pickup_datetime::date from yellow_taxi_data where extract(month from tpep_pickup_datetime)=1 order by tip_amount desc limit 1
```

Output:

tip_amount	tpep_pickup_datetime
1140.44	2021-01-20

Q5: What was the most popular destination for passengers picked up in central park on January 14? Enter the zone name (not id). If the zone name is unknown (missing), write "Unknown"

Command:

```
select count(y."DOLocationID") as frequency,"DOLocationID", t2."Zone" from yellow_taxi_data y, taxi_zones t1, taxi_zones t2 where y."PULocationID"=t1."LocationID" and y."DOLocationID"=t2."LocationID" and extract (day from y."tpep_pickup_datetime")=14 and extract(month from y."tpep_pickup_datetime")=1 and t1."Zone"='Central Park' group by y."DOLocationID", t2."Zone" order by frequency desc limit 1
```

Output:

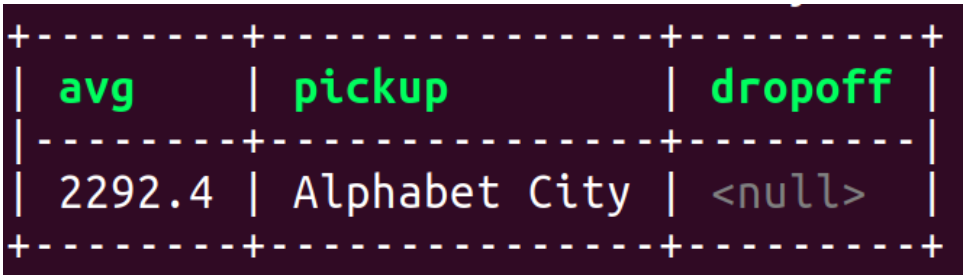
frequency	DOLocationID	Zone
97	237	Upper East Side South

Q6: What's the pickup-dropoff pair with the largest average price for a ride (calculated based on total_amount)? Enter two zone names separated by a slashFor example:"Jamaica Bay / Clinton East"If any of the zone names are unknown (missing), write "Unknown". For example, "Unknown / Clinton East".

Command:

```
select avg(y.total_amount) as avg, t1."Zone" as pickup, t2."Zone" as dropoff from yellow_taxi_data y, taxi_zones t1, taxi_zones t2 where y."PULocationID"=t1."LocationID" and y."DOLocationID"=t2."LocationID" group by t1."Zone",t2."Zone" order by avg desc limit 1
```

Output



avg	pickup	dropoff
2292.4	Alphabet City	<null>