# Loading Taxi Data into Google Cloud SQL

# Setting(prepare environment,creating cloud Sql)

**Activate Google Cloud Shell**

**Preparing your Environment**

Create environment variables that will be used later in the lab for your project ID and the storage bucket that will contain your data:

```
export PROJECT_ID=$(gcloud info --format='value(config.project)')
export BUCKET=${PROJECT_ID}-ml
```

**Create a Cloud SQL instance**

Enter the following commands to create a Cloud SQL instance:

```
gcloud sql instances create taxi \
   --tier=db-n1-standard-1 --activation-policy=ALWAYS
```

.Set a **root password** for the Cloud SQL instance:

```
gcloud sql users set-password root --host % --instance taxi \
 --password Passw0rd
```

When prompted for the password type Passw0rd and press enter this will update root password.

Now create an environment variable with the **IP address** of the Cloud Shell:

```
export ADDRESS=$(wget -qO - http://ipecho.net/plain)/32
```

Whitelist the Cloud Shell instance for **management access** to your SQL instance.

```
gcloud sql instances patch taxi --authorized-networks $ADDRESS
```

# Setting(IP adress, management access ,create table)

When prompted for the password type Passw0rd and press enter this will update root password.

Now create an environment variable with the **IP address** of the Cloud Shell:

```
export ADDRESS=$(wget -qO - http://ipecho.net/plain)/32
```

Whitelist the Cloud Shell instance for **management access** to your SQL instance.

```
gcloud sql instances patch taxi --authorized-networks $ADDRESS
```

**Getting the IP address** of your Cloud SQL instance by running:

```
MYSQLIP=$(gcloud sql instances describe \
taxi --format="value(ipAddresses.ipAddress)")
```

Check the variable MYSQLIP:

```
echo $MYSQLIP
```

you should get an IP address as an output.

**Create** the taxi trips **table** by logging into the mysql command line interface

```
mysql --host=$MYSQLIP --user=root \
      --password --verbose
```

# Create schema

When prompted for a password enter Passw0rd. Paste the following content into the command line to **create** the **schema** for the trips table:

```
create database if not exists bts;
use bts;
drop table if exists trips;
create table trips (
  vendor_id VARCHAR(16),
  pickup_datetime DATETIME,
  dropoff_datetime DATETIME,
  passenger_count INT,
  trip_distance FLOAT,
  rate_code VARCHAR(16),
  store_and_fwd_flag VARCHAR(16),
  payment_type VARCHAR(16),
  fare_amount FLOAT,
  extra FLOAT,
  mta_tax FLOAT,
  tip_amount FLOAT,
  tolls_amount FLOAT,
  imp_surcharge FLOAT,
  total_amount FLOAT,
  pickup_location_id VARCHAR(16),
  dropoff_location_id VARCHAR(16)
);
```

# Add data to Cloud SQL instance

Now you'll copy the New York City taxi trips CSV files stored on Cloud Storage locally. To keep resource usage low, you'll only be working with a subset of the data (~20,000 rows).
Run the following in the command line:

```
gsutil cp gs://cloud-training/OCBL013/nyc_tlc_yellow_trips_2018_subset_1.csv trips.csv-1
gsutil cp gs://cloud-training/OCBL013/nyc_tlc_yellow_trips_2018_subset_2.csv trips.csv-2
```

Import the CSV file data into Cloud SQL using mysql:

```
mysqlimport --local --host=$MYSQLIP --user=root --password \
--ignore-lines=1 --fields-terminated-by=',' bts trips.csv-*
```

When prompted for a password enter Passw0rd.
Connect to the mysql interactive console:

```
mysql --host=$MYSQLIP --user=root  --password
```

When prompted for a password enter Passw0rd.

# Queries SQL (data integrity , explore data)

In the mysql command line interface check the import by entering the following commands:

```
describe trips;
```

Query the trips table:

```
select distinct(pickup_location_id) from trips;
```

**Range of data**

-
  ```
  select
          max(trip_distance),
          min(trip_distance)
      from
          trips;
  ```

-
  ```
  select count(*) from trips where trip_distance = 0;
  ```

-
  ```
  select count(*) from trips where fare_amount < 0;
  ```

-
  ```
  select
           payment_type,
          count(*)
      from
          trips
     group by
          payment_type;
  ```

- #payment type = 1 has 13863 rows
- #payment type = 2 has 6016 rows
- #payment type = 3 has 113 rows
- #payment type = 4 has 32 row

Exit the mysql interactive console:

```
exit
```