Peter Antonaros
Math 290 Homework 3

Exercise 1: (Complete)

Filled out the google spreadsheet

ERROR: could not open file "" for reading: Permission denied

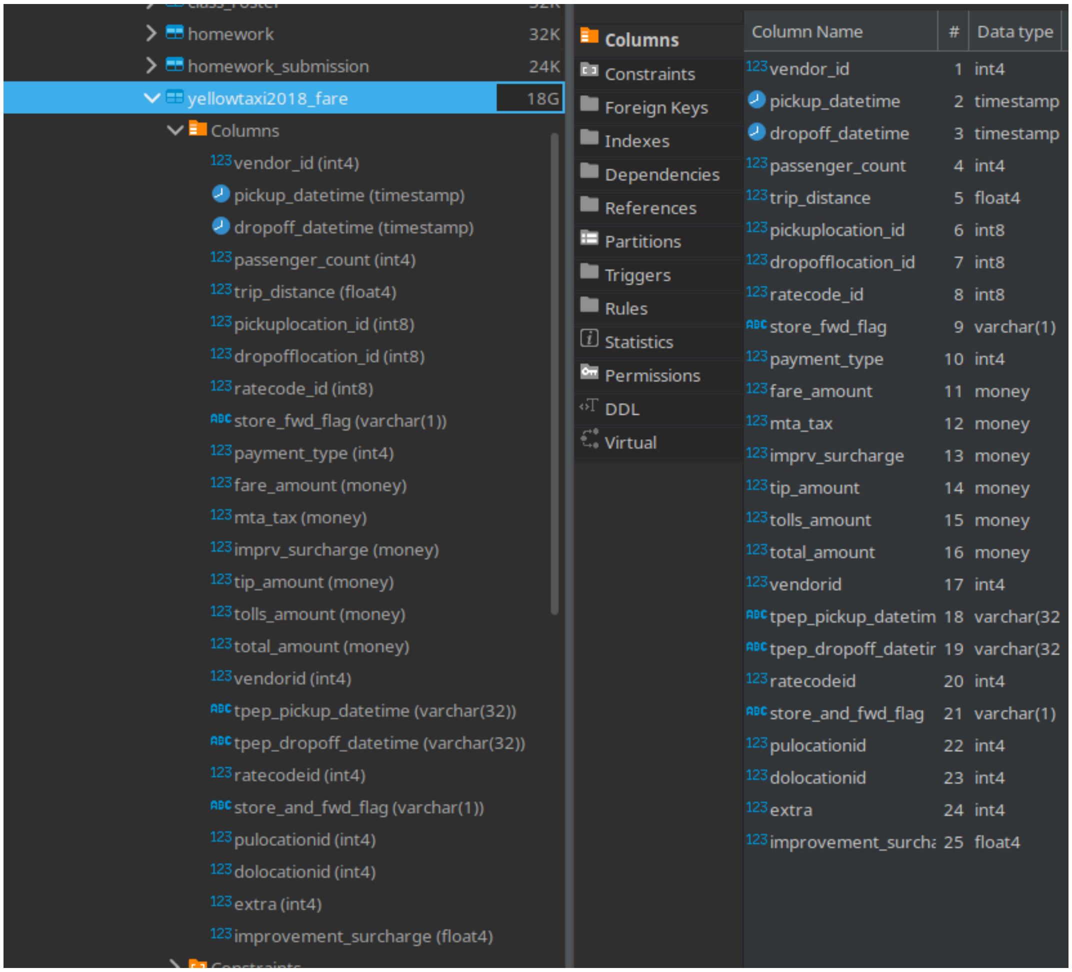
This was solved by changing the linux file permissions to the CSV taxi data set. I did this with

chmod 764 taxiDataset.csv

Here the 7 represents (READ+WRITE+EXECUTE) for the OWNER
6 (READ+WRITE) for the USERGROUP
4(READ) for the WORLD

My copy statement in Dbeaver then worked how I expected it to

Exercise 2: (Complete)



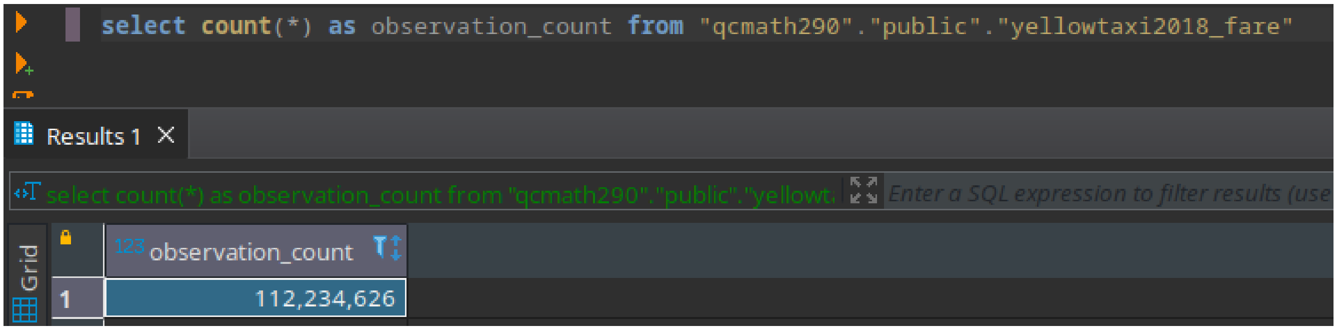
```
> homework                        32K
> homework_submission             24K
v yellowtaxi2018_fare             18G
  v Columns
      vendor_id (int4)
      pickup_datetime (timestamp)
      dropoff_datetime (timestamp)
      passenger_count (int4)
      trip_distance (float4)
      pickuplocation_id (int8)
      dropofflocation_id (int8)
      ratecode_id (int8)
      store_fwd_flag (varchar(1))
      payment_type (int4)
      fare_amount (money)
      mta_tax (money)
      imprv_surcharge (money)
      tip_amount (money)
      tolls_amount (money)
      total_amount (money)
      vendorid (int4)
      tpep_pickup_datetime (varchar(32))
      tpep_dropoff_datetime (varchar(32))
      ratecodeid (int4)
      store_and_fwd_flag (varchar(1))
      pulocationid (int4)
      dolocationid (int4)
      extra (int4)
      improvement_surcharge (float4)
```

Column Name	#	Data type
vendor_id	1	int4
pickup_datetime	2	timestamp
dropoff_datetime	3	timestamp
passenger_count	4	int4
trip_distance	5	float4
pickuplocation_id	6	int8
dropofflocation_id	7	int8
ratecode_id	8	int8
store_fwd_flag	9	varchar(1)
payment_type	10	int4
fare_amount	11	money
mta_tax	12	money
imprv_surcharge	13	money
tip_amount	14	money
tolls_amount	15	money
total_amount	16	money
vendorid	17	int4
tpep_pickup_datetim	18	varchar(32
tpep_dropoff_datetir	19	varchar(32
ratecodeid	20	int4
store_and_fwd_flag	21	varchar(1)
pulocationid	22	int4
dolocationid	23	int4
extra	24	int4
improvement_surchi	25	float4

Peter Antonaros
Math 290 Homework 3

Exercise 3: (Complete)

select count(*) as observation_count from "qcmath290"."public"."yellowtaxi2018_fare"
This statement took 59s to execute



```
select count(*) as observation_count from "qcmath290"."public"."yellowtaxi2018_fare"
```

Results 1
select count(*) as observation_count from "qcmath290"."public"."yellowt
Enter a SQL expression to filter results (use
Grid

	observation_count
1	112,234,626

This matches the number of rows provided by the website from where the dateset originated from

Exercise 4: (Complete)

select count(distinct "vendor_id") from "qcmath290"."public"."yellowtaxi2018_fare.vendor_id"
This statement took 57s to execute

There were only 3 distinct numbers.

Exercise 5: (Complete)

Given the fact that there are only 3 distinct numbers in vendor_id, it would be a bad idea to use this as a primary key. For a primary key we want something that is distinct to that we can efficiently retrieve data. With only 3 distinct numbers, our 112 million rows would be broken into 3 parts each of which would then need a “WHERE” statement to narrow down the 3 row groups into a single row. This would be extremely slow since each “row group” has about 37m rows in it.

Exercise 6: (Complete)

My first impression was the perhaps the datetime would be a reasonable guess for a primary key since it has the greatest level of granularity, thus the highest chance of being unique.

select count(distinct "pickup_datetime") from "qcmath290"."public"."yellowtaxi2018_fare.vendor_id"

select count(distinct "dropoff_datetime") from "qcmath290"."public"."yellowtaxi2018_fare.vendor_id"

Peter Antonaros
Math 290 Homework 3

Neither of these columns had ALL distinct values which in hindsight makes sense, since with so many rides per day I guess our time measurement is bound to have conflict.

At this point rather than trying more columns which were obviously not going to be distinct it would be best to add our own primary key that is unique to each row!

Exercise 7: (Complete)

None of the rows in the vendor_id column are null, which is good from the perspective that we don’t need to drop rows without a vendor_id.

Select * from “qcmath290"."public"."yellowtaxi2018_fare” where (COLUMS HERE) is null

I used this statement which is essentially saying get all the rows where there is a null value in any of the columns. I excluded all the column names in this document as it is a wall of text, but once that is inserted we get our result of NO NULL VALUES IN THE COLUMNS