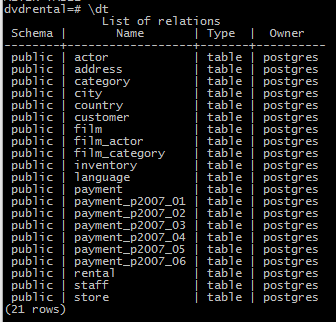
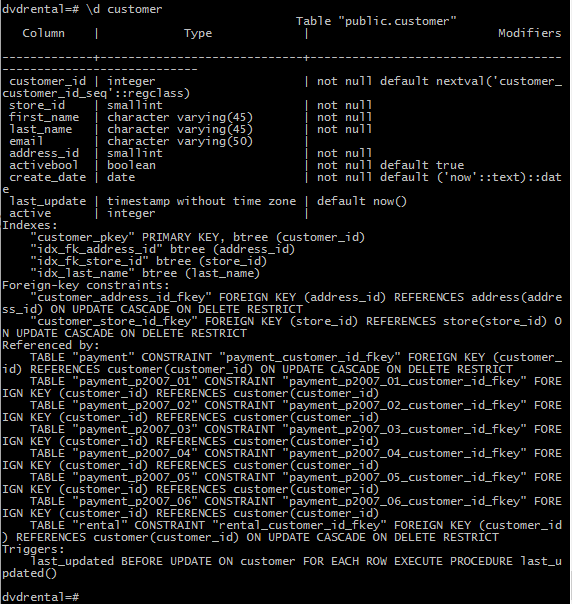
***Djamal Pullom***

**Lab 5 – PostgresSQL Introduction**

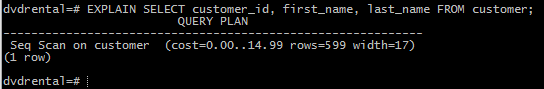
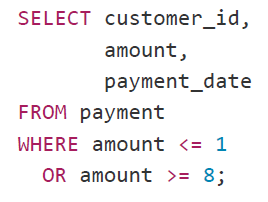
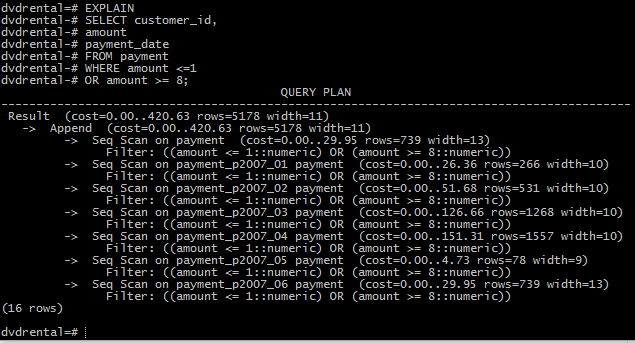
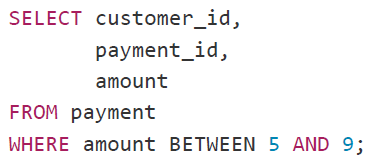
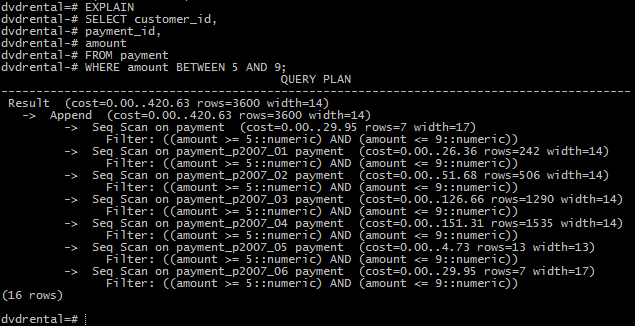
Question 1: What is the output of \dt?

* The output of “\dt” is the summary of all tables in the database, otherwise known as the database schema
* 

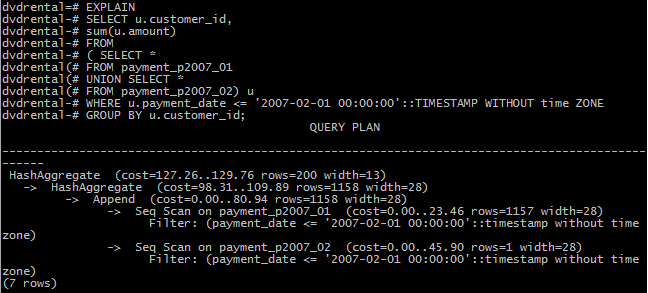
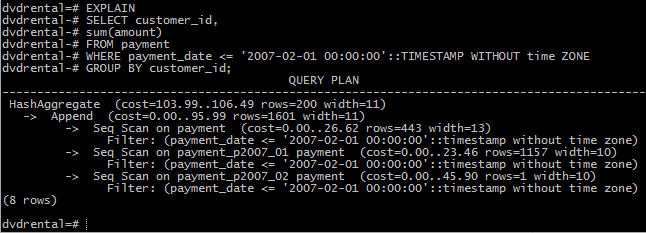
Question 2: What is the schema for the customer table?

* The customer table schema includes the fields customer\_id, store\_id, first\_name, last\_name, email, address\_id, activebool, create\_date, last\_update, and active.
* Customer\_id is the primary key
* Foreign keys include address\_id and store\_id
* All fields have the constraint “not null,” meaning they must have a value that fits their data type, except for email and active.
* Customer\_id fills in with a default of assigning the next sequential number, while create\_date and last\_update both fill in a default of “now”. Activebool fills in a default of true.
* 

Question 3: What similarities do you see in the explain plains for these 3 queries?

* #1: Table of customer id, first and last name
  + 
  + 
* #2 Table of customer id, amount, and payments dates for amounts less than or equal to 1 or greater than or equal to 8.
  + 
  + 
* #3 Table of customer id, payment id, and amount for amounts between 5 and 9.
  + 
  + 
* All three of these are queries on a customer database, so the differences in the explain data are due to the difference in the type of data return that is desired and what table to data is returned from. Since the second two are from the payment table, there are many more entries in that table than there are entries in the customer table, so the explain query shows many more rows for the second two. The widths are mostly similar, but again, it depends on the question being asked (or the desired query return). Query #1 produced 599 rows of width 17, Query #2 produced 5178 rows of width 11, and Query #3 produced 3600 rows of width 17.

Question 4: What is the difference between the plans for the Partitioned table and the union query? Why do you think this difference exists?

* The cost seems to be less for the partitioned version because it doesn’t have to do as many data manipulations to execute the query. Instead of constructing the union and doing that comparison, it just has to examine the logic for each entry, which is computationally less expensive.
  + 
  + 

Question 5: What join algorithm is used for the inner join?

* The join algorithm takes selected data fields (customer id, first name, last name, email, amount, and payment date) from the customer table and joins these entries to corresponding customer id’s from the payment table