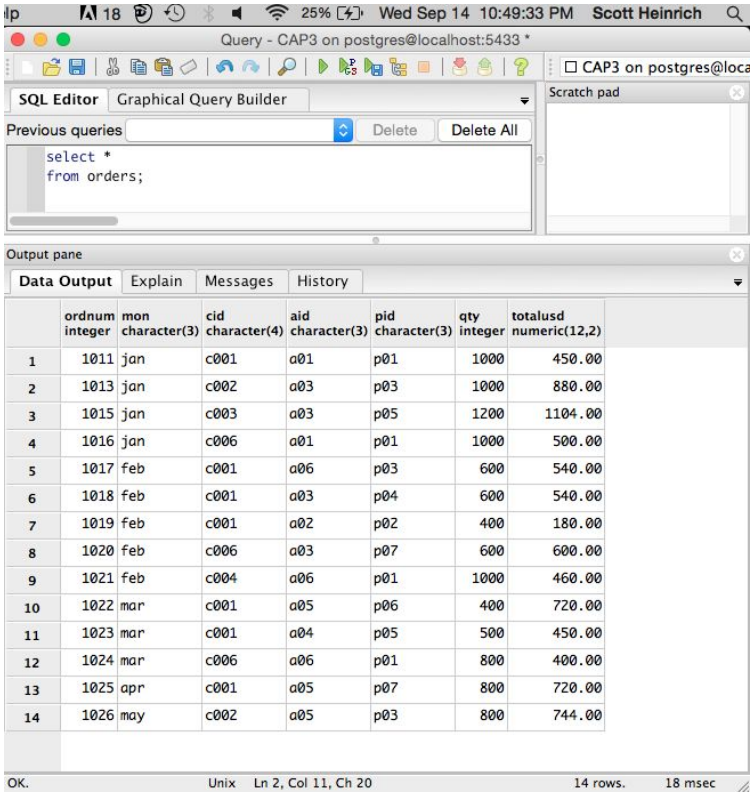
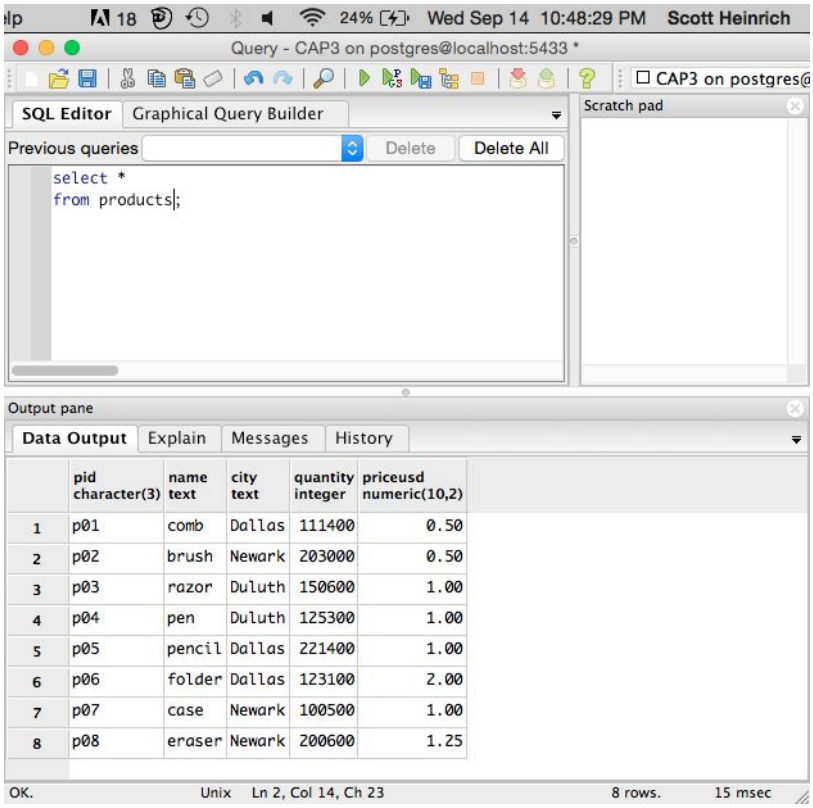
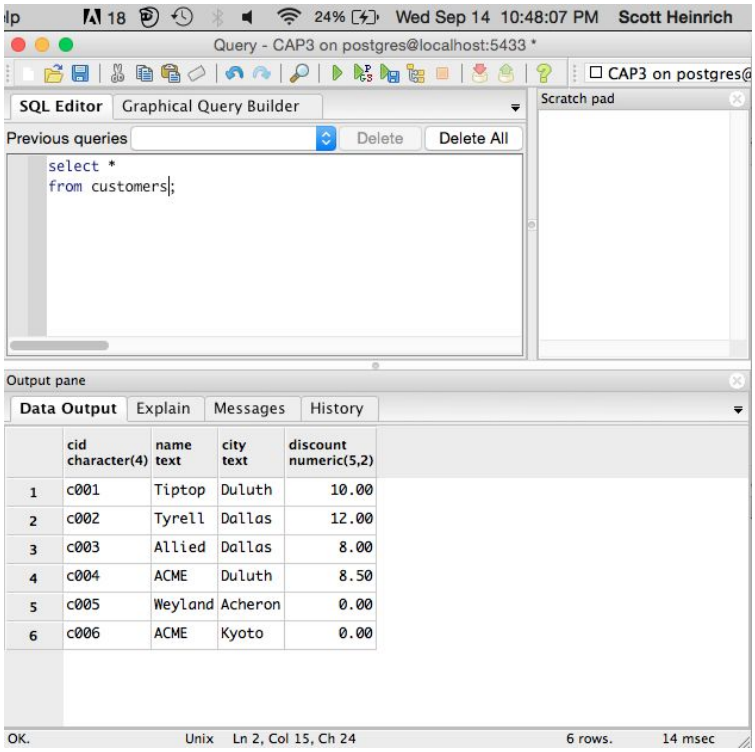
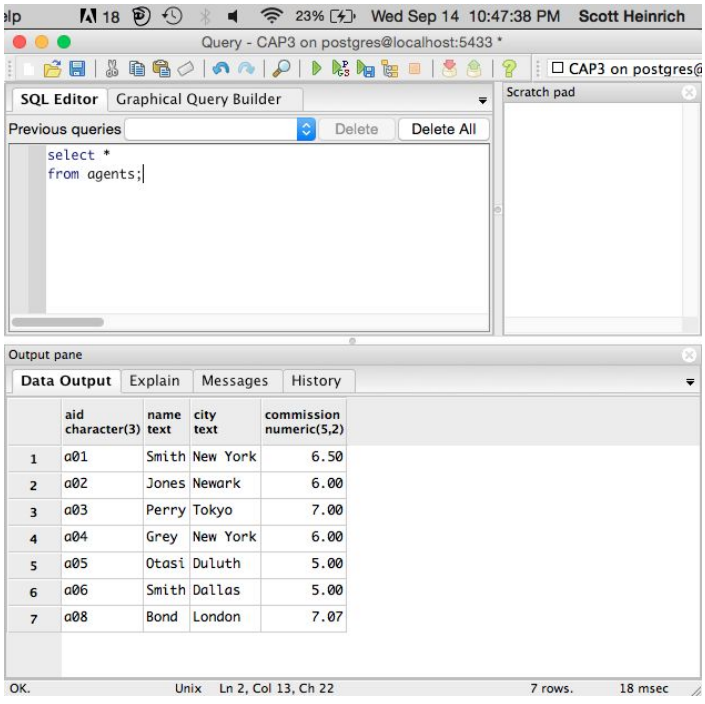

Wed Sep 14 10:49:33 PM Scott Heinrich
Query - CAP3 on postgres@localhost:5433 *
SQL Editor
Graphical Query Builder
Previous queries
Delete
Delete All
select *
from orders;
Scratch pad
Output pane
Data Output
Explain
Messages
History
ordnum integer
mon character(3)
cid character(4)
aid character(3)
pid character(3)
qty integer
totalusd numeric(12,2)
1 1011 jan c001 a01 p01 1000 450.00
2 1013 jan c002 a03 p03 1000 880.00
3 1015 jan c003 a03 p05 1200 1104.00
4 1016 jan c006 a01 p01 1000 500.00
5 1017 feb c001 a06 p03 600 540.00
6 1018 feb c001 a03 p04 600 540.00
7 1019 feb c001 a02 p02 400 180.00
8 1020 feb c006 a03 p07 600 600.00
9 1021 feb c004 a06 p01 1000 460.00
10 1022 mar c001 a05 p06 400 720.00
11 1023 mar c001 a04 p05 500 450.00
12 1024 mar c006 a06 p01 800 400.00
13 1025 apr c001 a05 p07 800 720.00
14 1026 may c002 a05 p03 800 744.00
OK.
Unix Ln 2, Col 11, Ch 20
14 rows.
18 msec


Wed Sep 14 10:48:29 PM Scott Heinrich
Query - CAP3 on postgres@localhost:5433 *
SQL Editor
Graphical Query Builder
Previous queries
Delete
Delete All
select *
from products;
Scratch pad
Output pane
Data Output
Explain
Messages
History
pid character(3)
name text
city text
quantity integer
priceusd numeric(10,2)
1 p01 comb Dallas 111400 0.50
2 p02 brush Newark 203000 0.50
3 p03 razor Duluth 150600 1.00
4 p04 pen Duluth 125300 1.00
5 p05 pencil Dallas 221400 1.00
6 p06 folder Dallas 123100 2.00
7 p07 case Newark 100500 1.00
8 p08 eraser Newark 200600 1.25
OK.
Unix Ln 2, Col 14, Ch 23
8 rows.
15 msec


Wed Sep 14 10:48:07 PM Scott Heinrich
Query - CAP3 on postgres@localhost:5433 *
SQL Editor
Graphical Query Builder
Previous queries
Delete
Delete All
select *
from customers;
Scratch pad
Output pane
Data Output
Explain
Messages
History
cid character(4)
name text
city text
discount numeric(5,2)
1 c001 Tiptop Duluth 10.00
2 c002 Tyrell Dallas 12.00
3 c003 Allied Dallas 8.00
4 c004 ACME Duluth 8.50
5 c005 Weyland Acheron 0.00
6 c006 ACME Kyoto 0.00
OK.
Unix
Ln 2, Col 15, Ch 24
6 rows.
14 msec


Wed Sep 14 10:47:38 PM Scott Heinrich
Query - CAP3 on postgres@localhost:5433 *
SQL Editor
Graphical Query Builder
Previous queries
Delete
Delete All
select *
from agents;
Scratch pad
Output pane
Data Output
Explain
Messages
History
aid character(3)
name text
city text
commission numeric(5,2)
1 a01 Smith New York 6.50
2 a02 Jones Newark 6.00
3 a03 Perry Tokyo 7.00
4 a04 Grey New York 6.00
5 a05 Otasi Duluth 5.00
6 a06 Smith Dallas 5.00
7 a08 Bond London 7.07
OK.
Unix
Ln 2, Col 13, Ch 22
7 rows.
18 msec

Explain the distinctions among the terms primary key, candidate key, and superkey.

A primary key a column that can identify all table records. The key must contain a unique value for each data row.

A candidate key is a column or set of columns that can unqiuely identify any record in the database without referring to any other data. A primary key can be considered a special candidate key

A superkey is a combination of columns that uniquely identify any row within a database table.

A popular reason to use tables would be for student records. There are various datatypes that could be used when dealing with a student's record. A few examples are character, Integer, and Timestamp, and Decimal. To explain this, imagine a table with the title student records. It has 4 columns, which read “Student Name”, “Age”, “Date of Birth”, and GPA. Their datatypes would be Character, Integer, Timestamp, and Decimal respectively. You could get creative and add boolean as a data type to this table. An easy example would be a column for transfer students; if they have transferred, it would return true, if not it would return false. There are a wide variety of data types that can be used for tables. Student name would most likely be the primary key, organizing the table in alphabetical order, so in this case, Character can not be null.

Explain the following relational “rules” with examples and reasons why they are important.

A. The “first normal form” rule

The fields should be atomic. What that means is, that in each column and row there should be a single value. This is important because it would be hard to search for content when it has multiple items.

B. The “access rows by content only” rule

Access by what, never where. You should always search by content, rather than location because the data isn’t necessarily stored in an organized way. Just because the table we are viewing is organized, doesn’t mean the data is.

C. The “all rows must be unique” rule

All rows should must be unique because if it wasn’t unique, you couldn’t access a specific row. If there were two of the same rows, primary keys would be useless.