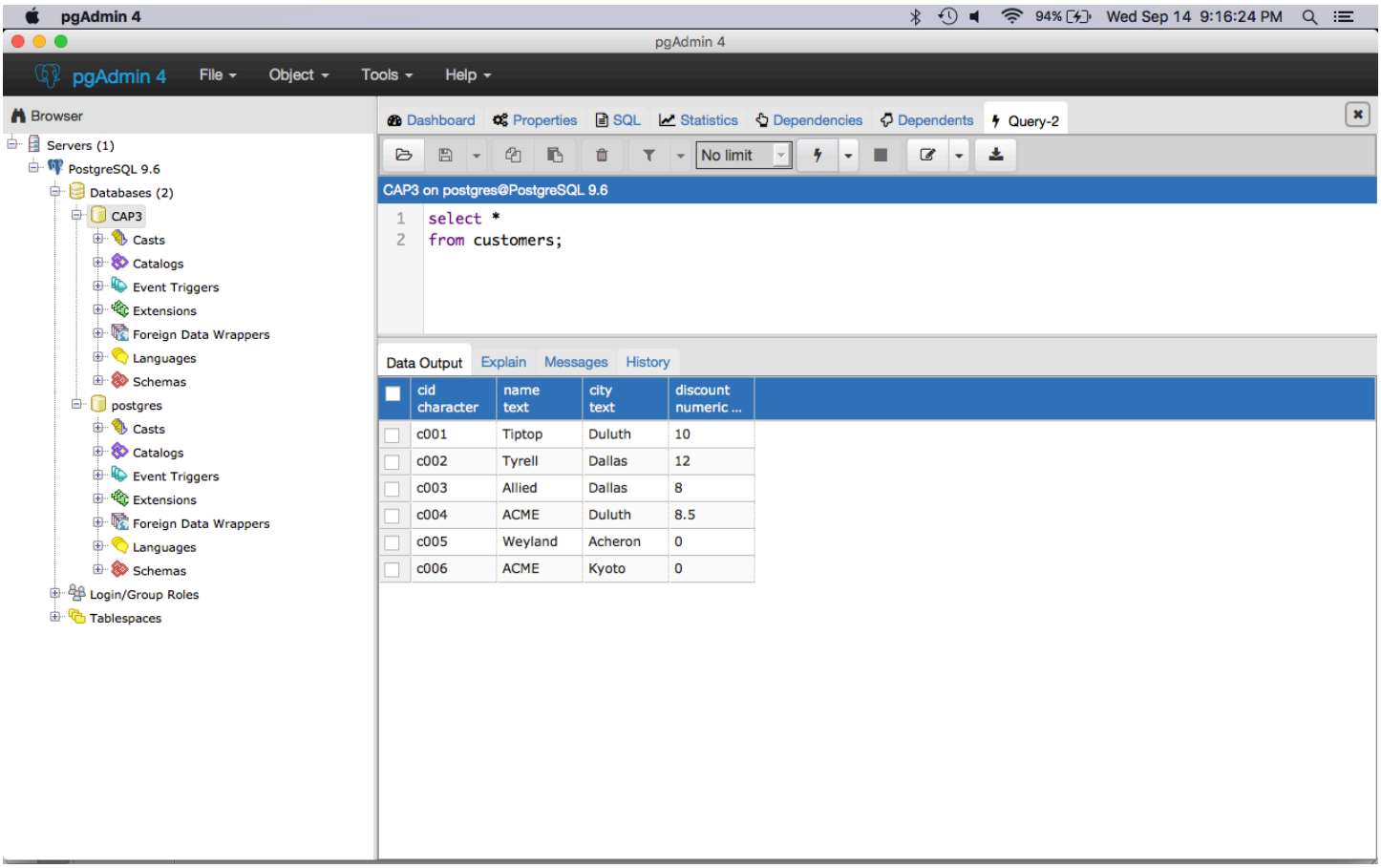
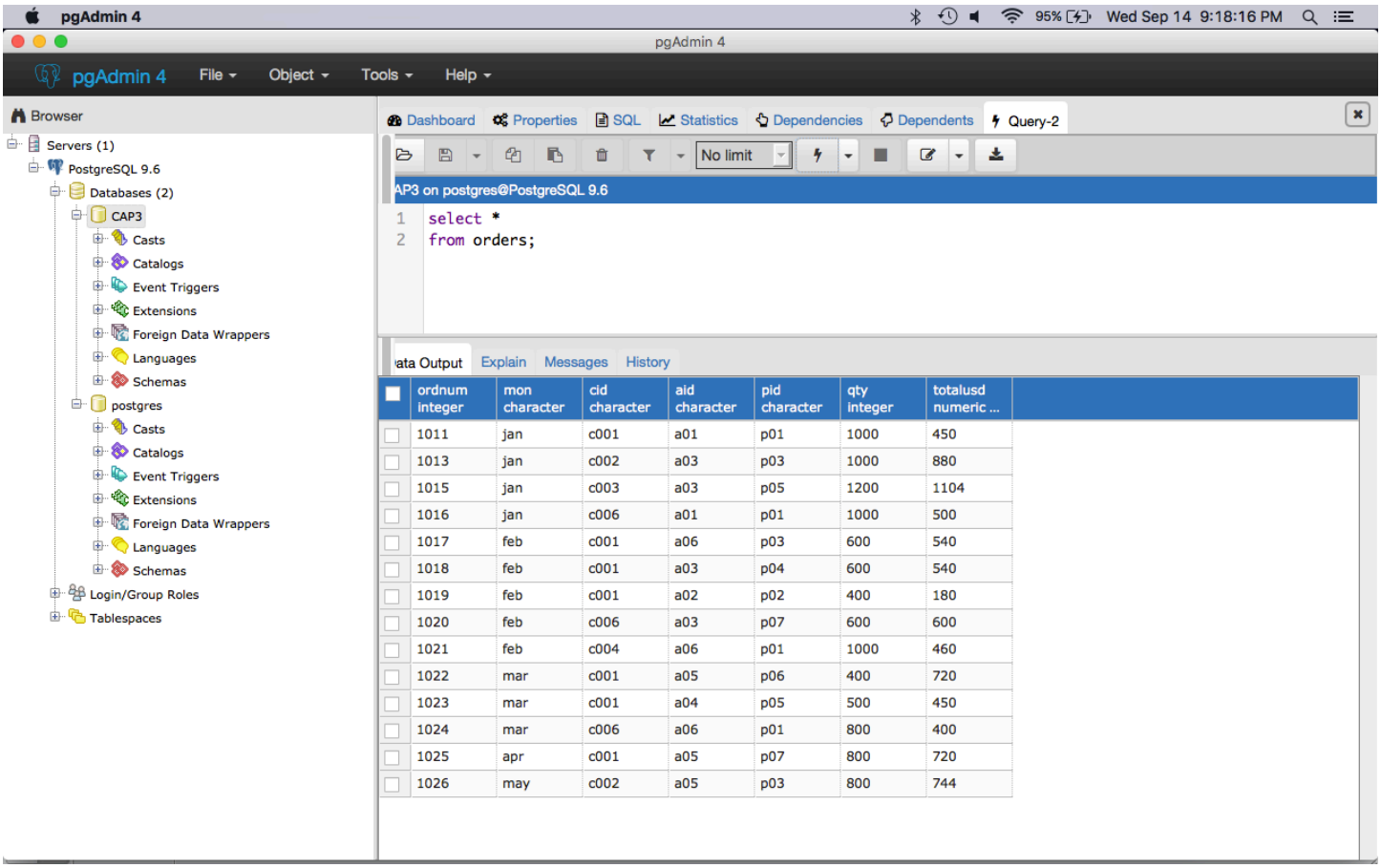
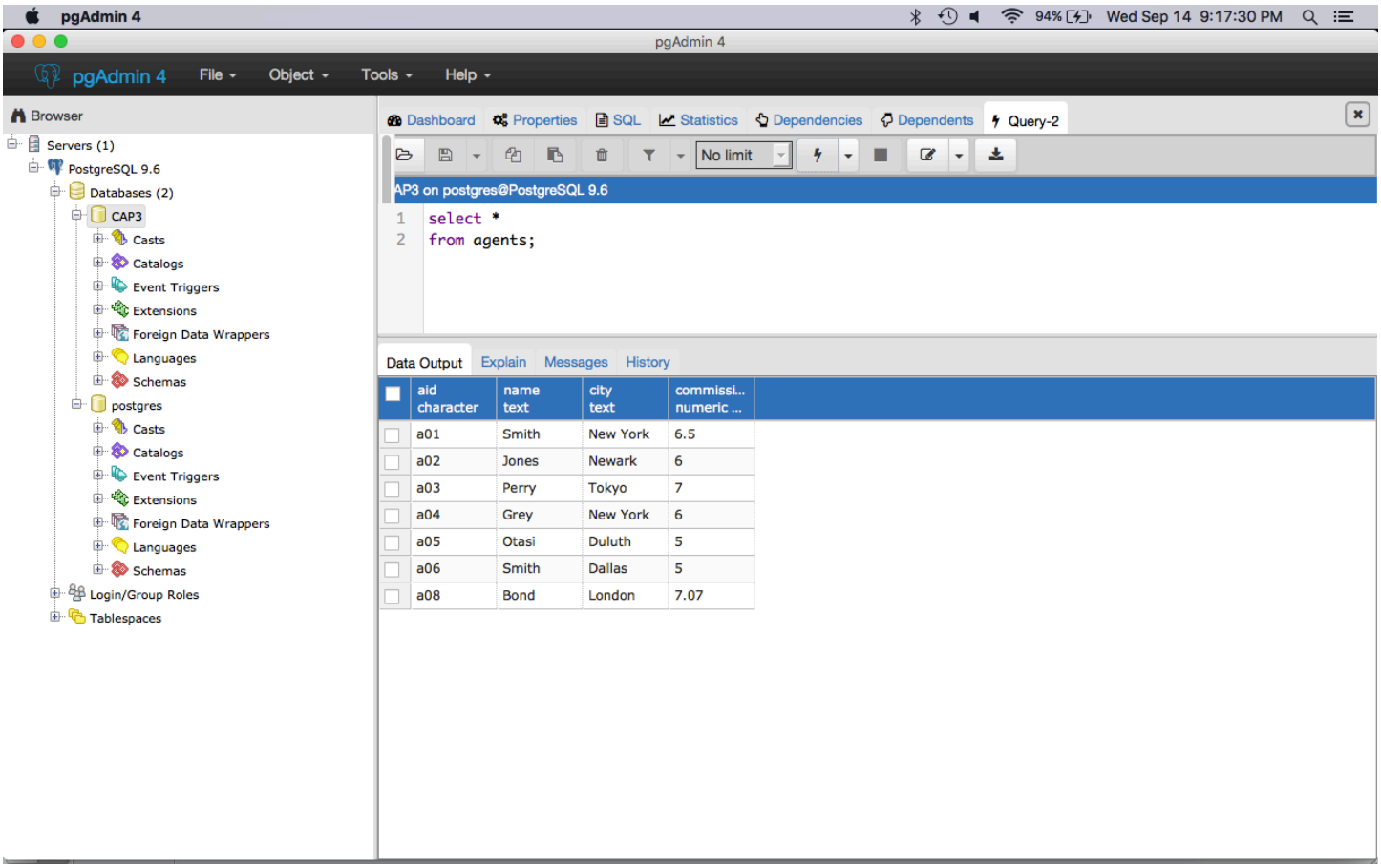
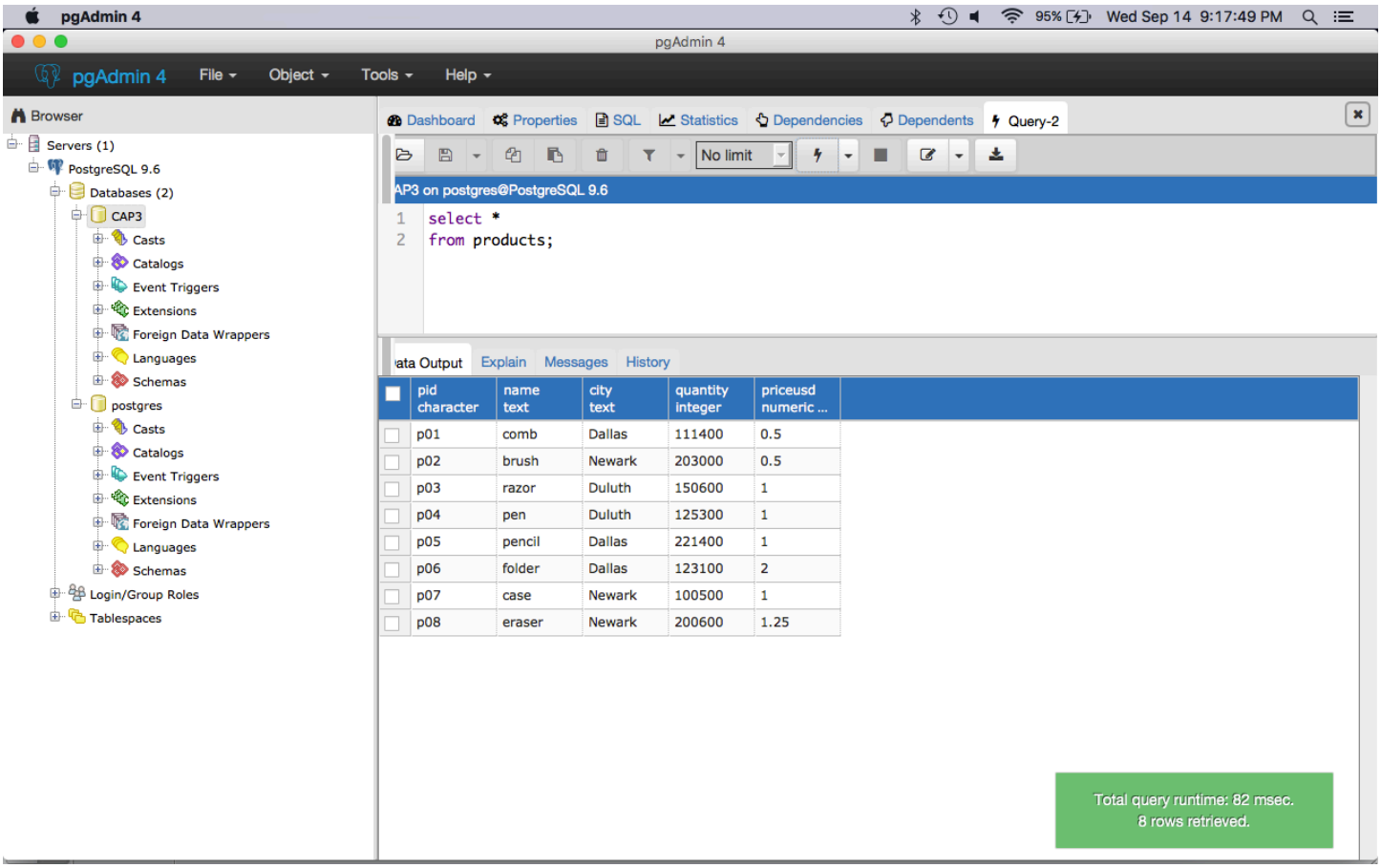

pgAdmin 4
94% Wed Sep 14 9:16:24 PM
pgAdmin 4
pgAdmin 4
File
Object
Tools
Help
Browser
Servers (1)
PostgreSQL 9.6
Databases (2)
CAP3
Casts
Catalogs
Event Triggers
Extensions
Foreign Data Wrappers
Languages
Schemas
postgres
Casts
Catalogs
Event Triggers
Extensions
Foreign Data Wrappers
Languages
Schemas
Login/Group Roles
Tablespaces
Dashboard
Properties
SQL
Statistics
Dependencies
Dependents
Query-2
No limit
CAP3 on postgres@PostgreSQL 9.6
1 select *
2 from customers;
Data Output
Explain
Messages
History
cid character
name text
city text
discount numeric ...
c001 Tiptop Duluth 10
c002 Tyrell Dallas 12
c003 Allied Dallas 8
c004 ACME Duluth 8.5
c005 Weyland Acheron 0
c006 ACME Kyoto 0


pgAdmin 4
95% Wed Sep 14 9:18:16 PM
pgAdmin 4
pgAdmin 4
File
Object
Tools
Help
Browser
Servers (1)
PostgreSQL 9.6
Databases (2)
CAP3
Casts
Catalogs
Event Triggers
Extensions
Foreign Data Wrappers
Languages
Schemas
postgres
Casts
Catalogs
Event Triggers
Extensions
Foreign Data Wrappers
Languages
Schemas
Login/Group Roles
Tablespaces
Dashboard
Properties
SQL
Statistics
Dependencies
Dependents
Query-2
No limit
AP3 on postgres@PostgreSQL 9.6
1 select *
2 from orders;
ata Output
Explain
Messages
History
ordnum integer
mon character
cid character
aid character
pid character
qty integer
totalusd numeric ...
1011 jan c001 a01 p01 1000 450
1013 jan c002 a03 p03 1000 880
1015 jan c003 a03 p05 1200 1104
1016 jan c006 a01 p01 1000 500
1017 feb c001 a06 p03 600 540
1018 feb c001 a03 p04 600 540
1019 feb c001 a02 p02 400 180
1020 feb c006 a03 p07 600 600
1021 feb c004 a06 p01 1000 460
1022 mar c001 a05 p06 400 720
1023 mar c001 a04 p05 500 450
1024 mar c006 a06 p01 800 400
1025 apr c001 a05 p07 800 720
1026 may c002 a05 p03 800 744


pgAdmin 4
94% Wed Sep 14 9:17:30 PM
pgAdmin 4
pgAdmin 4
File
Object
Tools
Help
Browser
Servers (1)
PostgreSQL 9.6
Databases (2)
CAP3
Casts
Catalogs
Event Triggers
Extensions
Foreign Data Wrappers
Languages
Schemas
postgres
Casts
Catalogs
Event Triggers
Extensions
Foreign Data Wrappers
Languages
Schemas
Login/Group Roles
Tablespaces
Dashboard
Properties
SQL
Statistics
Dependencies
Dependents
Query-2
No limit
AP3 on postgres@PostgreSQL 9.6
1 select *
2 from agents;
Data Output
Explain
Messages
History
aid character
name text
city text
commissi... numeric ...
a01 Smith New York 6.5
a02 Jones Newark 6
a03 Perry Tokyo 7
a04 Grey New York 6
a05 Otasi Duluth 5
a06 Smith Dallas 5
a08 Bond London 7.07


pgAdmin 4
95% Wed Sep 14 9:17:49 PM
pgAdmin 4
pgAdmin 4
File
Object
Tools
Help
Browser
Servers (1)
PostgreSQL 9.6
Databases (2)
CAP3
Casts
Catalogs
Event Triggers
Extensions
Foreign Data Wrappers
Languages
Schemas
postgres
Casts
Catalogs
Event Triggers
Extensions
Foreign Data Wrappers
Languages
Schemas
Login/Group Roles
Tablespaces
Dashboard
Properties
SQL
Statistics
Dependencies
Dependents
Query-2
No limit
AP3 on postgres@PostgreSQL 9.6
1 select *
2 from products;
ata Output
Explain
Messages
History
pid character
name text
city text
quantity integer
priceusd numeric ...
p01 comb Dallas 111400 0.5
p02 brush Newark 203000 0.5
p03 razor Duluth 150600 1
p04 pen Duluth 125300 1
p05 pencil Dallas 221400 1
p06 folder Dallas 123100 2
p07 case Newark 100500 1
p08 eraser Newark 200600 1.25
Total query runtime: 82 msec.
8 rows retrieved.

2. A primary key in a relational database is a table column designated to uniquely identify each record in the table. It must contain a unique value for each row of data and can't contain null values. A primary key can be either an existing table column or a column that is specifically generated by the database that is based on a defined sequence. When querying data, utilizing the primary key uniqueness feature guarantees one result.

   A candidate key is a one or a combination of attributes or columns that can be uniquely used to identify any database record without referring to any other data. Each table can have one or more candidate keys, but one of those candidate keys, must be the primary key. All primary keys are candidate keys but not all candidate keys are primary keys.

   A super key is a set of attributes, where there are no two identical records that have the same values for the attributes in that set. A candidate key is a minimal super key in that it is the minimal set of attributes necessary to identify a unique record.

3. Data types are essential in database design in order to give data meaning and turn it into information. Either using data types to determine what type of type is stored in a certain column or designating the data type in attribute headers provide meaning to data in columns. It creates context that can work with other columns of data that also has data types to provide information about the records in the table and the database as a whole. For example a online magazine emailing list that lists information about customers in order to provide this particular magazine service. In this customer information table there are fields such as customer id, first name, last name, age, birthdate, subscription plan id, and email. For customer id, it is a primary key for this table. It will have a data type of characters with a given length to accommodate the size of the customer base. The first name is type text with a reasonable size limit. The last name is also type text except it might have a longer size limit. Age and birthdate are related with the age determined by the birthdate. Age would be an integer data type. Birthdate would have a date data type. The subscription plan would be an integer or character type according to how the id is set up. The email would be text with a certain size limit. The customer id can't be null. The first and last name could be nullable but contextually that would not be an option for customers. Birthdate could be optional contextually so that could also be nullable. Subscription id could be null if the customer is on the mailing list but does not have a subscription but most likely just being on the email list could be considered a type of plan in the database. The email could also be null but contextually the email is needed to have a magazine subscription.

4. Relational Rules

   a. First normal form rule sets the basic rules for an organized database. The first part of this rule is that the data items must be defined. This means taking data and organizing it into columns, giving the columns

a certain data type, and grouping related columns together to form tables. The second part is making sure that there are no repeatable groups of data. Lets say there is a list of customers with their id, name, address, and item that they ordered. If a particular customer ordered a lot of items, their basic information would be repeated with the only change being the item ordered. To avoid this, one would create a table for just orders with each order just referring to the customer id. The third part is to create a primary key for each table, which helps link tables to each other as well as make every record unique in a table.

b. The "access rows by content only" rule is the idea that tables change by number of records or order of attributes so in order to access certain information one must use the attributes and the data under those attributes to find particular content. Using the location 10 rows down and 5 columns across is against the rule because the particular data that is pointed to in this case could change by the order of columns or the addition of records and data.

c. The "all rows must be unique" rule can be satisfied by creating primary keys, which would be different for each row in the table, or creating additional tables. This makes sure that no record would be identical to another. In referencing the example in a. lets say the customer, in the orders table, ordered multiples of a certain item. Listing the customer, and the item multiple times would be repeating itself and creating identical rows. By adding quantity to the orders table, the multiple identical rows can be avoided. Another option which might help in this example follow this rule would to create and item table and in orders, reference an item id instead of the item name.