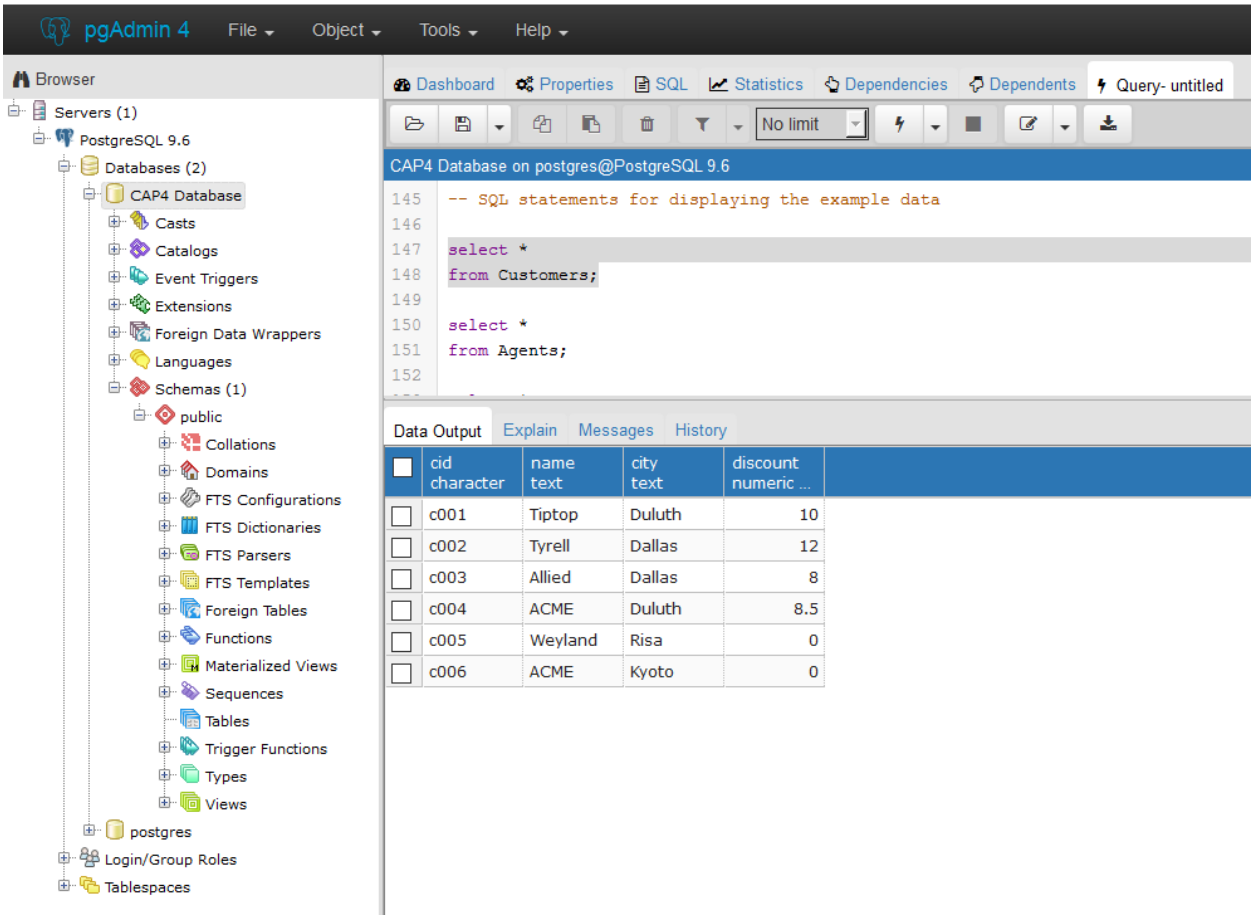
Clayton Szelestey

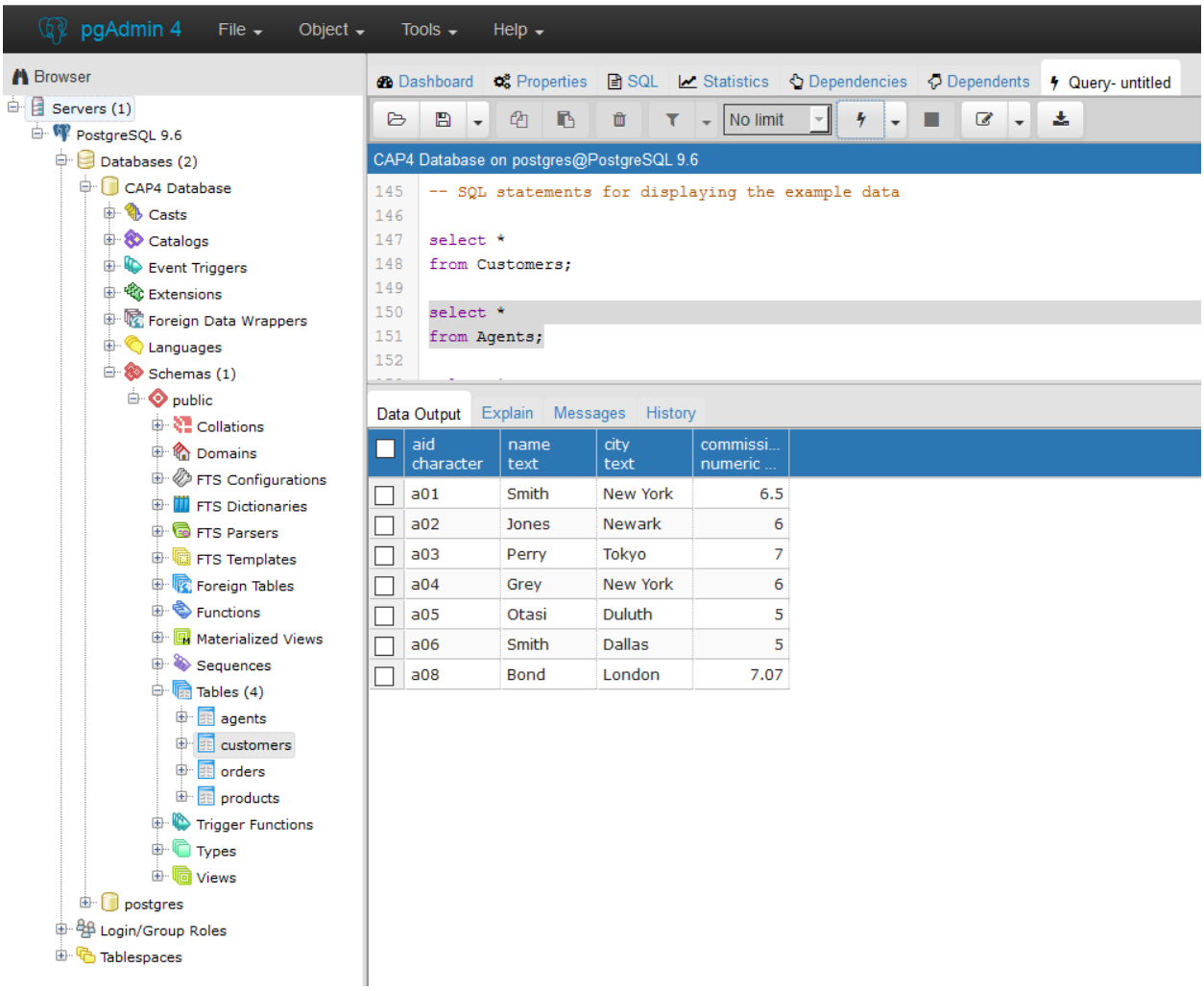
1/31/17

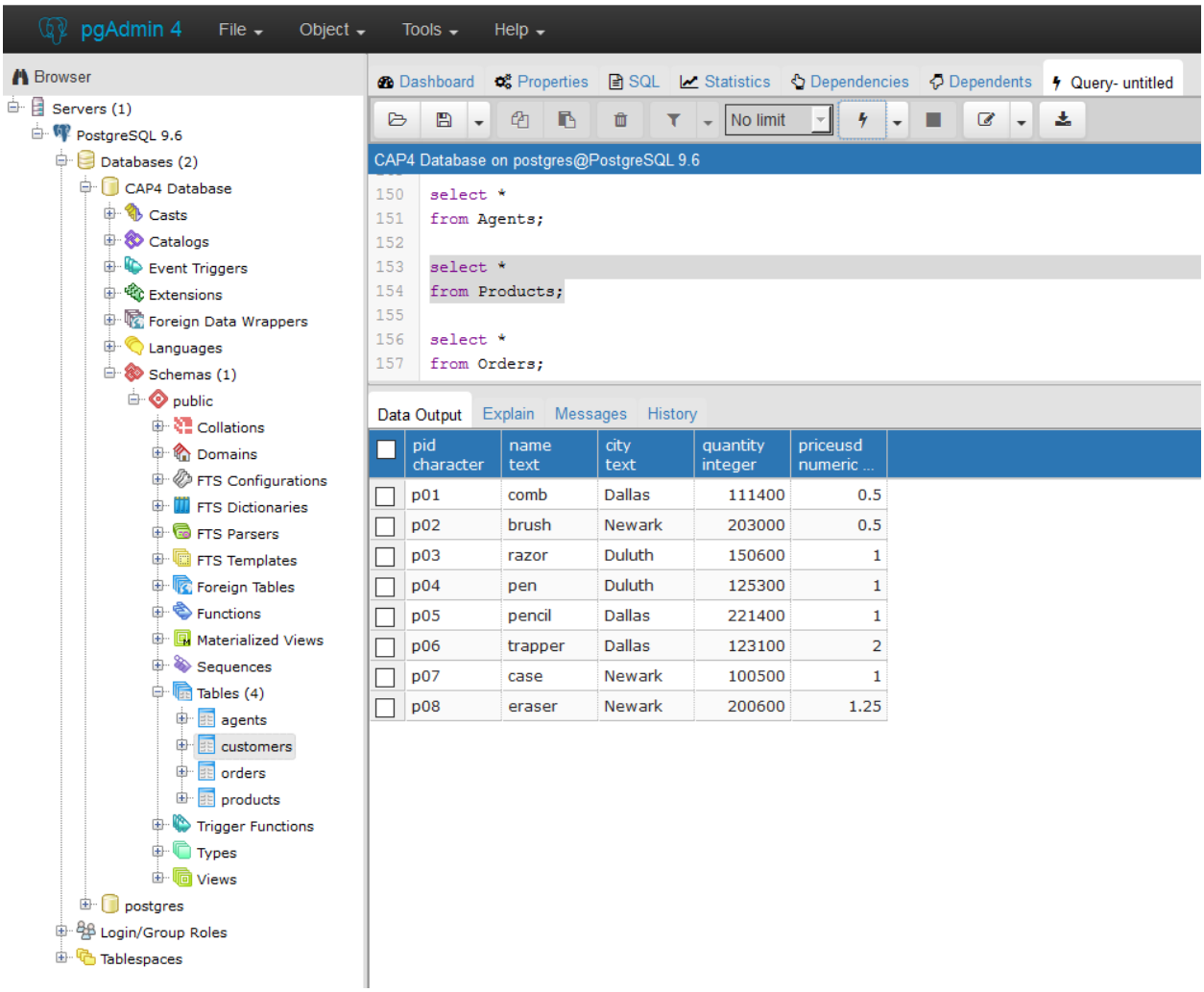
Database Management

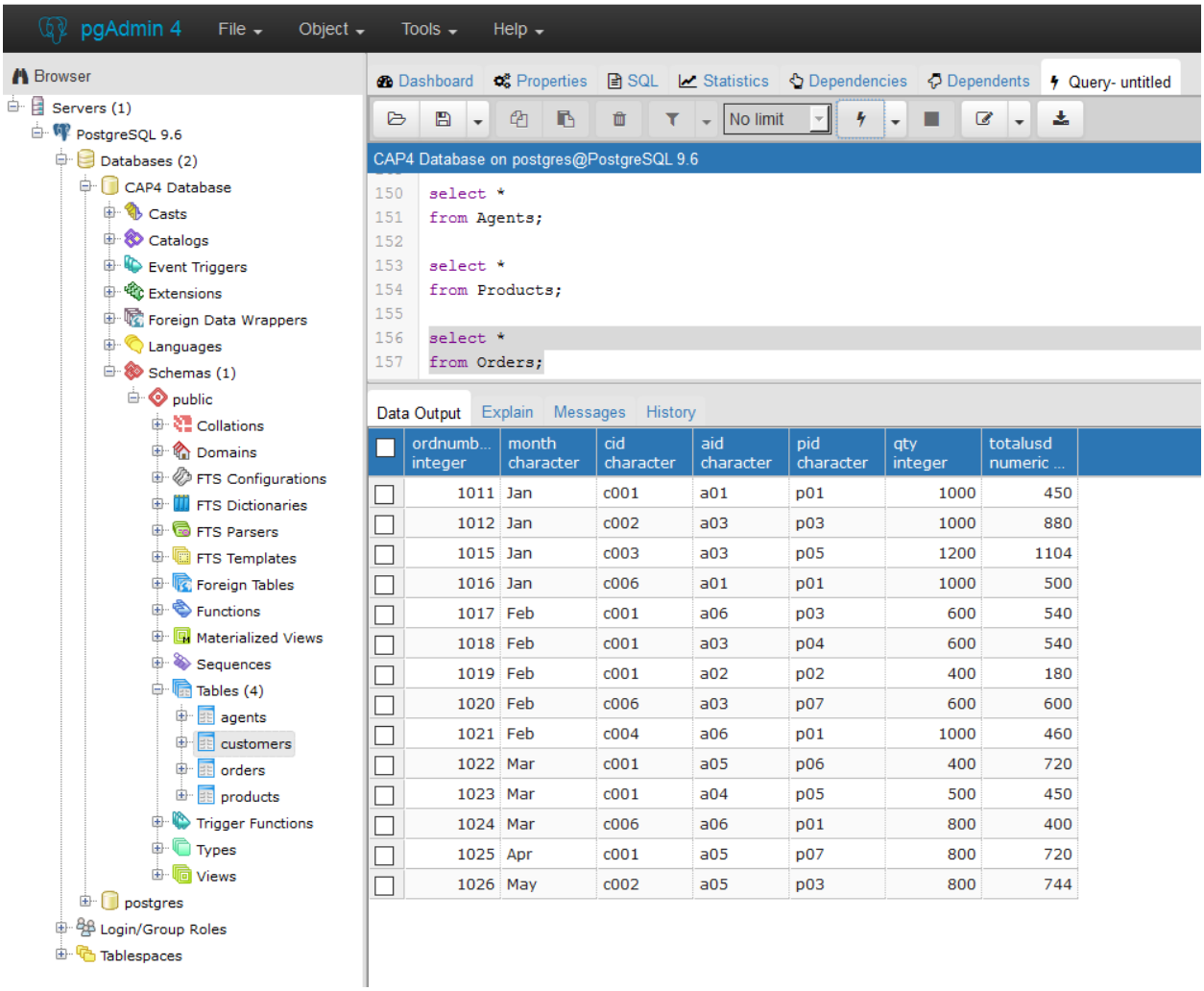
Lab 2

1.)




pgAdmin 4
File
Object
Tools
Help
Browser
Servers (1)
PostgreSQL 9.6
Databases (2)
CAP4 Database
Casts
Catalogs
Event Triggers
Extensions
Foreign Data Wrappers
Languages
Schemas (1)
public
Collations
Domains
FTS Configurations
FTS Dictionaries
FTS Parsers
FTS Templates
Foreign Tables
Functions
Materialized Views
Sequences
Tables (4)
agents
customers
orders
products
Trigger Functions
Types
Views
postgres
Login/Group Roles
Tablespaces
Dashboard
Properties
SQL
Statistics
Dependencies
Dependents
Query- untitled
No limit
CAP4 Database on postgres@PostgreSQL 9.6
145 -- SQL statements for displaying the example data
146
147 select *
148 from Customers;
149
150 select *
151 from Agents;
152
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
File
Object
Tools
Help
Browser
Servers (1)
PostgreSQL 9.6
Databases (2)
CAP4 Database
Casts
Catalogs
Event Triggers
Extensions
Foreign Data Wrappers
Languages
Schemas (1)
public
Collations
Domains
FTS Configurations
FTS Dictionaries
FTS Parsers
FTS Templates
Foreign Tables
Functions
Materialized Views
Sequences
Tables (4)
agents
customers
orders
products
Trigger Functions
Types
Views
postgres
Login/Group Roles
Tablespaces
Dashboard
Properties
SQL
Statistics
Dependencies
Dependents
Query- untitled
No limit
CAP4 Database on postgres@PostgreSQL 9.6
150 select *
151 from Agents;
152
153 select *
154 from Products;
155
156 select *
157 from Orders;
Data 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 trapper Dallas 123100 2
p07 case Newark 100500 1
p08 eraser Newark 200600 1.25



2.) A super key is a column or set of columns that ensures that every row will be unique. A candidate key is a super key with the minimum number of columns. A primary key is simply the candidate key that you choose to use.

3.) An example topic for which one might create a table would be inventory in a video game store. One could have a table labeled “Inventory” and its fields could contain headings such as Game Title, Price in USD New, Price in USD Used, Quantity New, Quantity Used, Release Date, Number of DLCs, etc. Game Title would be a String data type, Price in USD New would be a Decimal data type, Price in USD Used would be a Decimal data type, Quantity New would

be an Integer data type, Quantity Used would be an integer data type, Release Date would be a Date data type, and Number of DLCs would be an Integer data type. Price in USD Used may be nullable because the store may not have received any used copies of a new game and therefore not have determined a price for the used game.

4.) The “first normal form” rule states that all intersections must be atomic, or not multivalued or structured. This means that for every cell in a table, there must be only one value within the cell. One cannot have a list within a cell.

The “access rows by content only” rule states that one has to look up rows based on content only, not the order of the rows. The order is completely irrelevant to the computer, so the way rows must be accessed based on their unique values.

The “all rows must be unique” rule states that no two rows can have the same exact content. If two rows did have the exact same content, the computer would not be able to distinguish the two. If two rows have the exact same content, then how does one know which row is meant to be accessed or changed? And if one is changed, then is the older one now incorrect? Should both have been changed? This depends on the situation, of course, but a problem like this could arise.