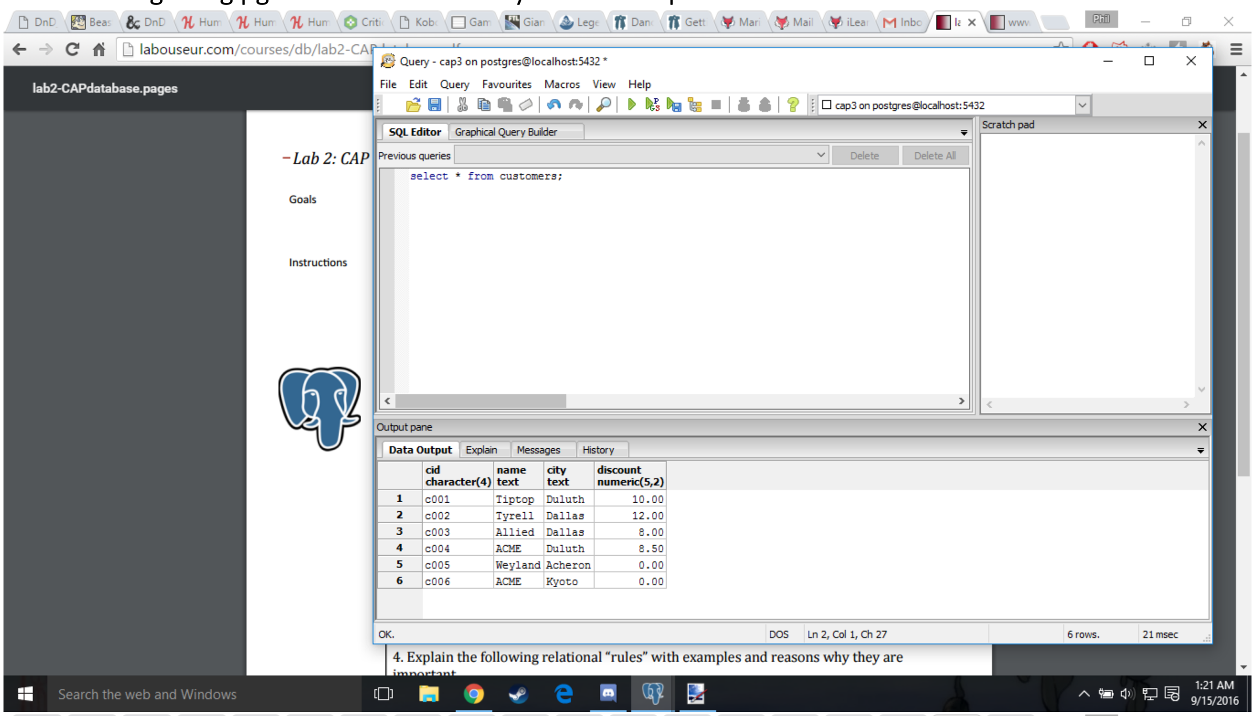
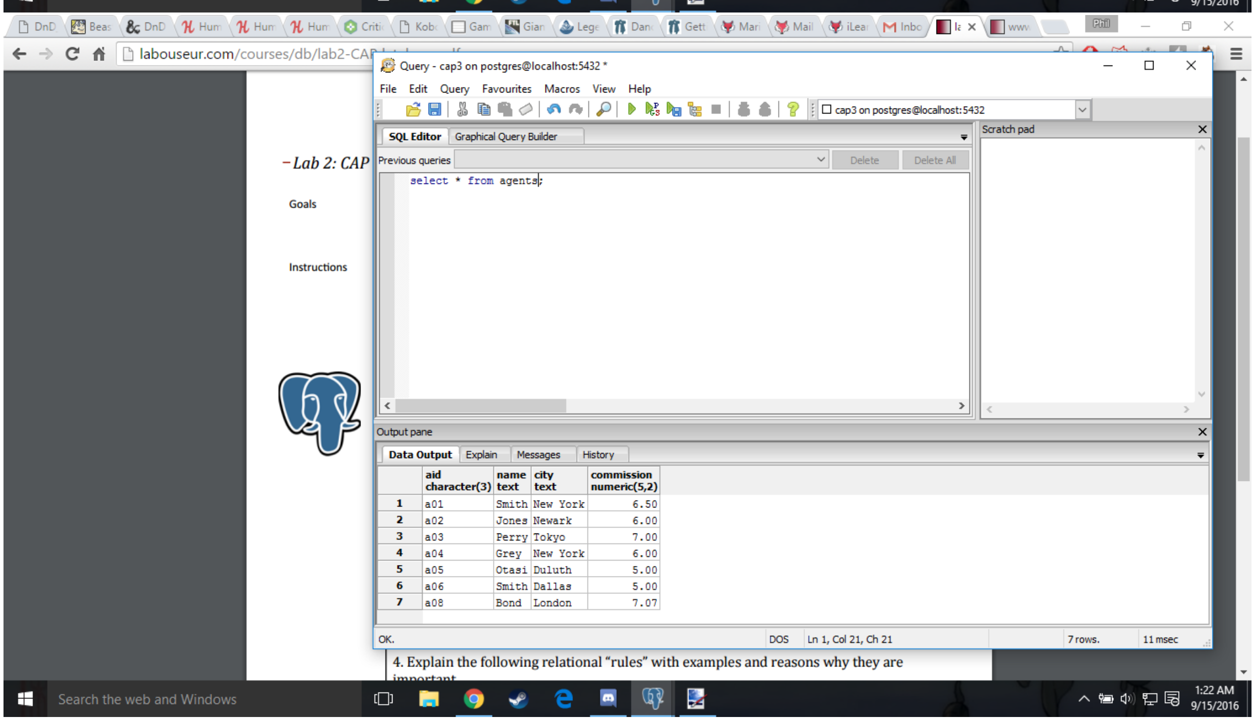
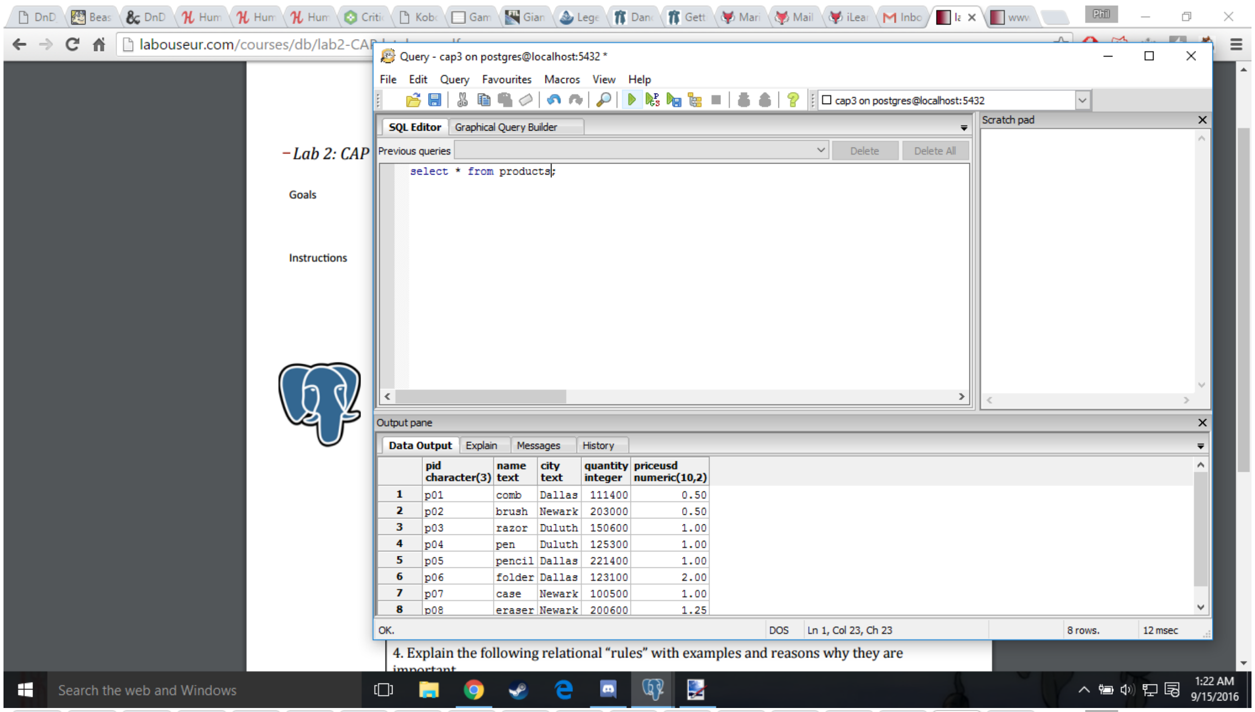
Philip Kirwin

Lab 2

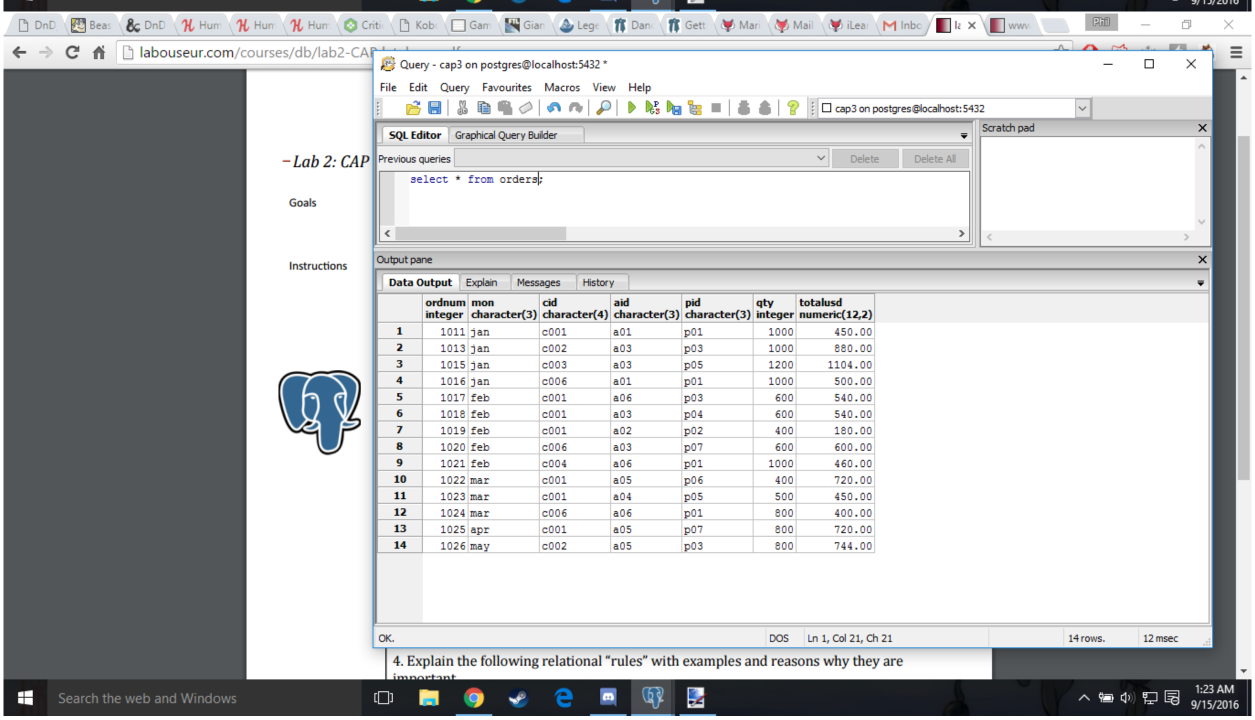
1. For these screenshots, please excuse the Windows 10, I had to use my other computer since I had trouble getting pgAdmin to work on my school computer.






| | 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 |




| | 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 |

2. In a relational database, the function of a primary key is to distinguish a specific row in a table among all other rows such that other tables may identify and use data from that specific row. Alternatively, a candidate key can also be used to identify a specific row in a table, however, a single table may contain multiple candidate keys while it may only have a single primary key. A super key is a type of candidate key, as is a primary key, however it does not require that all associated fields be filled in in every row, as a primary key does.

3. An example of where a database table may be needed would be to store the data of a library catalog. Some of the fields that could be included in such a table would be the ISBN of every book, its title, the author's name, the copyright date, and the number of pages, among others. The ISBN would be an integer, being a number, and would not be nullable due to the fact that this number acts as an identifier for any book, and thus may be used a primary key in this case. The book title and author would be text fields, given that either can use any number and type of characters, and could be nullable due to the fact that the ISBN is enough information to identify any given book. The copyright date of the book would be of the datetime data type and would also be nullable because it is not as important as the ISBN in identifying a book. Lastly, the number of pages in a book would be of the integer data type and would also be nullable because it, too would not be necessary in identifying a specific book.
4. The first relational rule is that all fields must be atomic. This means that all intersections between a row and a column cannot have any further structure and thus may only contain one piece of data. This rule is important for the proper function of a database because if a given field was allowed to have further structure and multiple pieces of data, it would become very difficult to access any one specific piece of data due to the need to work around the extra structure. For example, if in the library table, all of a single book's information was stored within one field, it would be very difficult to find the title of a given book due to the amount of unnecessary data that is stored alongside the title that you are looking for. The second relational rule is that information may only be accessed by what it is, rather than where it is on the table. This is because the actual database is stored in sets, which have no intrinsic order despite the tabular form that we imply databases have. This rule is important because one cannot query the database for the data in the third row and the second column, for example, because rows and columns are simply not stored and numbered in this way. Instead, one must query the database for the information that each row and column represents. For example, when querying the library database, one must search for a specific ISBN, title, or any other specific piece of information rather than a location on a theoretical table. The third relational rule is that all rows must be unique in at least one column. This rule is important because the relational model is based on set theory, and according to this theory, a single set may not contain duplicate values. So in the library example, any two books stored in the table may not be the same in every single column and while this does not inherently break the database, it violates the theory that the database is built upon and is therefore poor design.