Dietrich Mosel
Prof. Labouseur
Database Mgmt.
9/11/17

# Lab 2: CAP Database

## Distinctions Among Terms:

There are many differences between primary, candidate and superkeys. A candidate key is a column, or set, contained in a table that can recognize any database record without having to call upon any other data. The primary key, on the other hand, is the candidate key but it calls upon the column that is the best fit for maintaining the uniqueness within the table. Lastly, a superkey is a combination of different columns that can identify any row within a table. It is more complex than the other two because it's purpose is to reduce time spent calling upon many columns. They all build off the foundation that the primary key provides and cannot stand without one another.

## Data Types:

In SQL, there are three main data types. They are text, number and date data types. Within text data types, there are a multitude of functions that refer to the number of characters, letters or numbers in each word with maximum or minimum strings. Number data types contain ranges of digits (positive and negative) and commands that deal with floating decimal points with large or small numbers. Lastly, there are date data types which refer to dates in a range of formats, timestamps, time (in terms of hours, minutes and seconds) and years as well. For my specific table, I chose for it to consist of running statistics. The title would be "College Runners" and the columns or fields would be "Name," "Best Event," "Time" and "School." A person's name would be classified as a text data type (specifically TINYTEXT) because it contains less than 255 characters. Best Event would be dealing with numbers/distances like 5,000 meters or 10,000 meters but it also contains words as well. This would be another text data type (specifically CHAR(size)) because it can hold a fixed length string with letters, numbers or special characters. Time on the other hand is solely composed of numbers but it would not be a number data type. Instead, it would be a date data type (specifically TIME()) because it has a format of HH:MI:SS which is perfect for running times and can support a wide range large enough for this purpose. Finally, School would be sorted as another text data type (specifically TINYTEXT) because it only contains letters. These fields would not be nullable because everyone has a name and I'm not specifying a certain running event so people would be required to fill that field out along with time and school since they are college students.

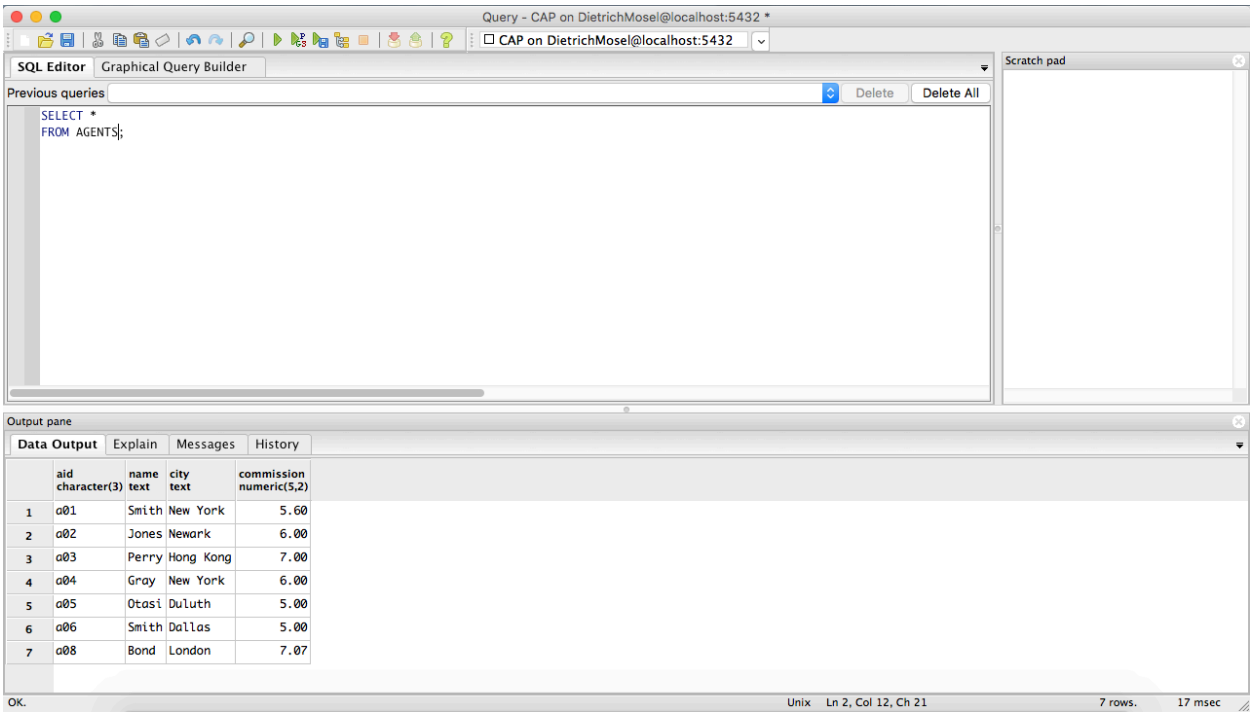
## Relational "Rules":

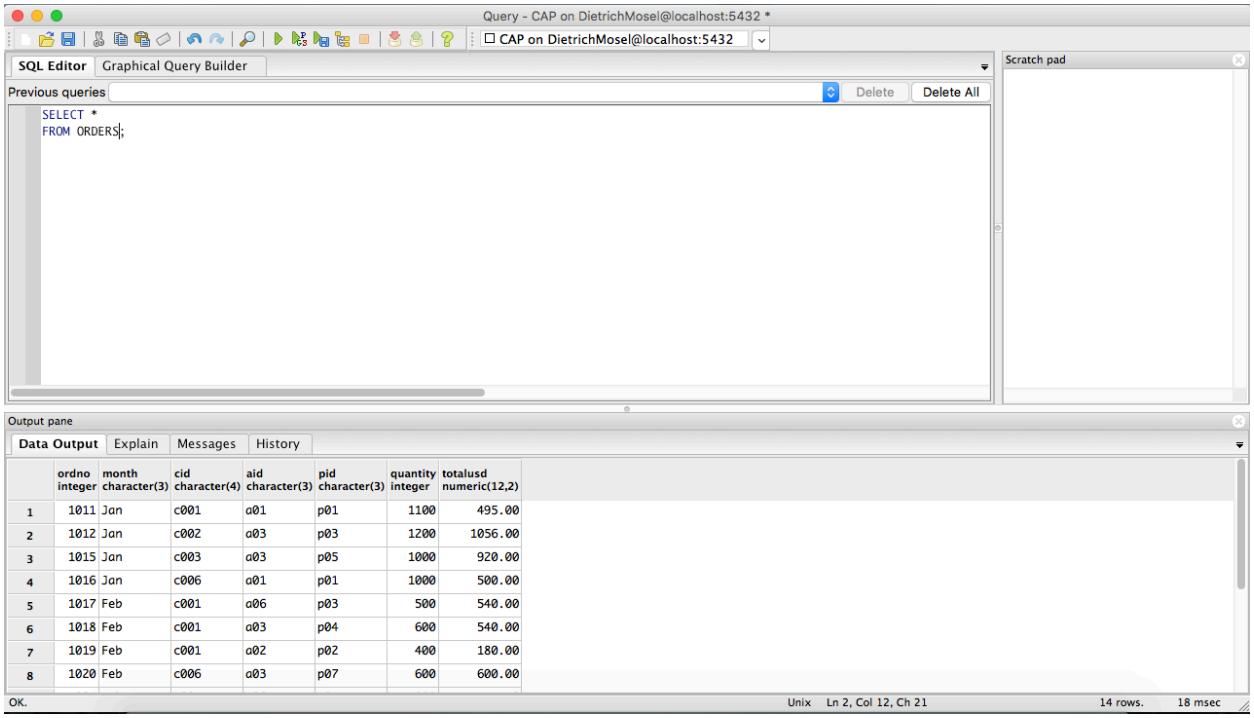
a. The "first normal form" rule – Otherwise known as 1NF, the first normal form rule is crucial to assembling any SQL table because it makes searching, sorting and filtering information easier. The data has to be in a database table that stores information in

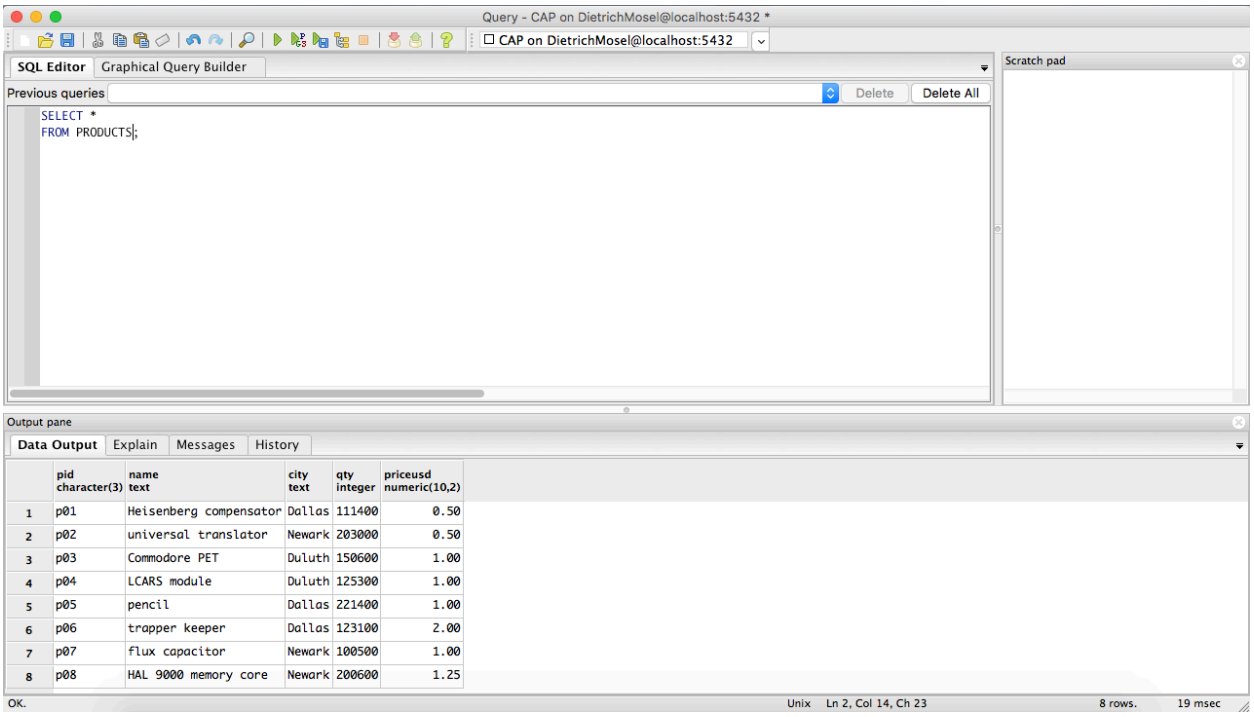
columns and rows with each column containing groups of columns that are not repeating. For example, a table with a product id, color and price that contains multiple colors in that column is not in first normal form. In order to convert it, you would create two separate tables, one with the product id and price and the other with product id and color.

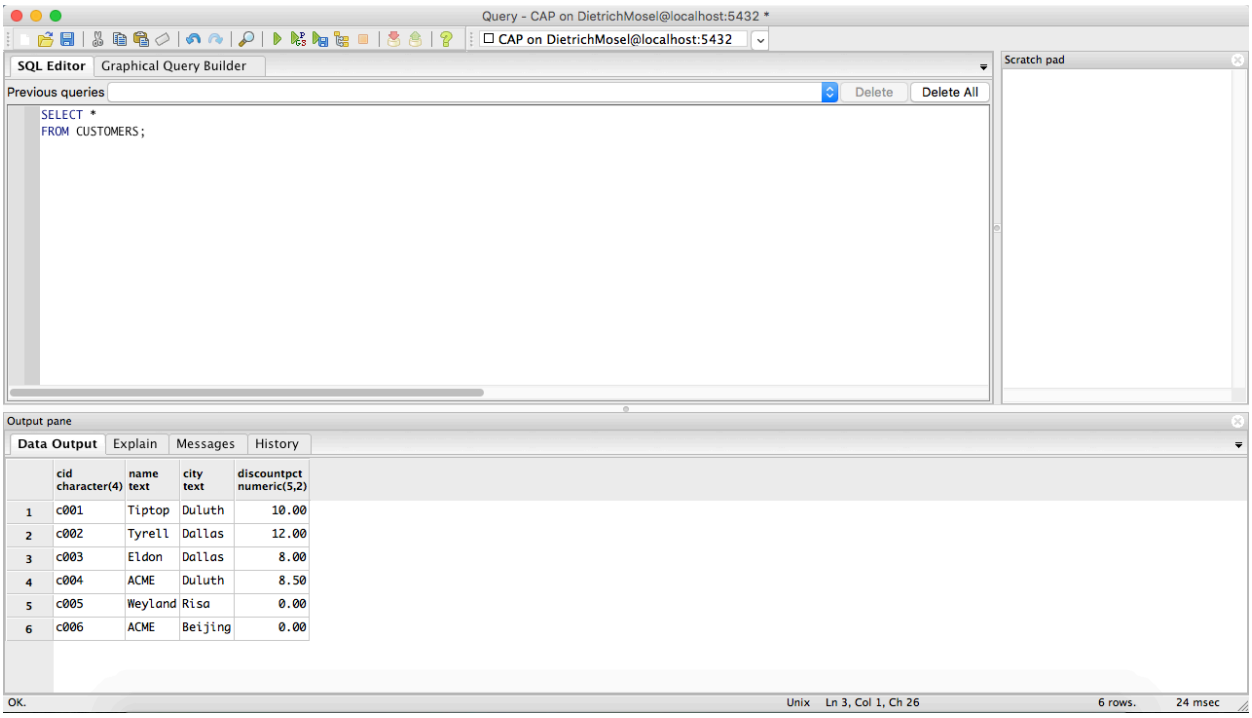
b. The "access rows by content only" rule – The second relational rule that simply states what it does. It accesses by content and accesses rows by the value of their columns. You cannot be vague with this rule by saying "second row from the bottom," because there is no order to the columns or rows. For example, Select * From Agents where Agent # = 5. This is important because it adds order to the searching process. By allowing the user to access rows based on content, it eliminates useless time spent on filtering out unnecessary rows that do not contain the proper information.

c. The "all rows must be unique" rule – This third and final relational rule is similar to the second rule and also relates to the primary key in terms of being unique and necessary. There can be no duplicate tuples so that they cannot be identical in all column values at the same time. This is not necessarily a bad thing for tables containing such information as temperature readings where you will have a lot of repetition. However, it is still important for tables that contain similar (but not for beneficial purposes) rows which someone would like to distinctly sort out.




Query - CAP on DietrichMosel@localhost:5432 *
CAP on DietrichMosel@localhost:5432
SQL Editor
Graphical Query Builder
Previous queries
Delete
Delete All
SELECT *
FROM ORDERS;
Scratch pad
Output pane
Data Output
Explain
Messages
History
ordno integer
month character(3)
cid character(4)
aid character(3)
pid character(3)
quantity integer
totalusd numeric(12,2)
1 1011 Jan c001 a01 p01 1100 495.00
2 1012 Jan c002 a03 p03 1200 1056.00
3 1015 Jan c003 a03 p05 1000 920.00
4 1016 Jan c006 a01 p01 1000 500.00
5 1017 Feb c001 a06 p03 500 540.00
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
OK.
Unix Ln 2, Col 12, Ch 21
14 rows.
18 msec


Query - CAP on DietrichMosel@localhost:5432 *
CAP on DietrichMosel@localhost:5432
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
qty integer
priceusd numeric(10,2)
1 p01 Heisenberg compensator Dallas 111400 0.50
2 p02 universal translator Newark 203000 0.50
3 p03 Commodore PET Duluth 150600 1.00
4 p04 LCARS module Duluth 125300 1.00
5 p05 pencil Dallas 221400 1.00
6 p06 trapper keeper Dallas 123100 2.00
7 p07 flux capacitor Newark 100500 1.00
8 p08 HAL 9000 memory core Newark 200600 1.25
OK.
Unix Ln 2, Col 14, Ch 23
8 rows.
19 msec


Query - CAP on DietrichMosel@localhost:5432 *
CAP on DietrichMosel@localhost:5432
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
discountpct numeric(5,2)
1 c001 Tiptop Duluth 10.00
2 c002 Tyrell Dallas 12.00
3 c003 Eldon Dallas 8.00
4 c004 ACME Duluth 8.50
5 c005 Weyland Risa 0.00
6 c006 ACME Beijing 0.00
OK.
Unix Ln 3, Col 1, Ch 26
6 rows.
24 msec