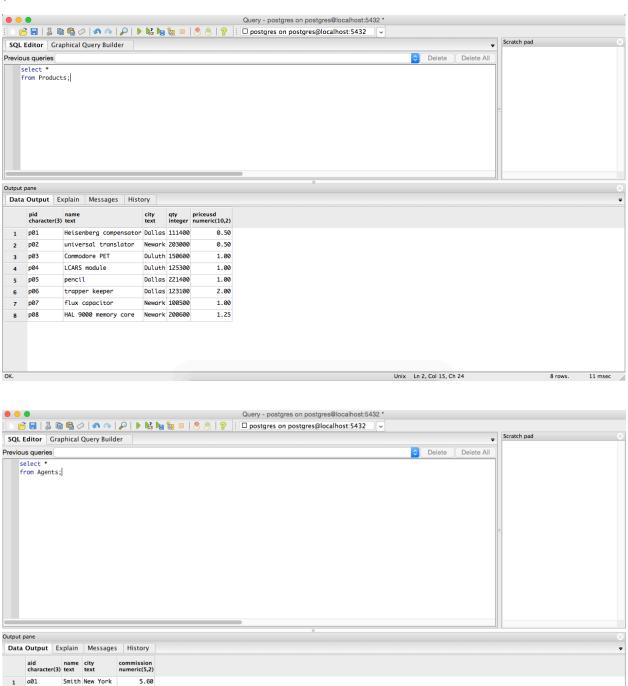
## 1.)



Unix Ln 2, Col 13, Ch 22 7 rows. 19 msec

6.00

7.00

6.00

5.00

5.00

7.07

a02

a04

5 a05

2 a03

6 a06 7 a08

Jones Newark

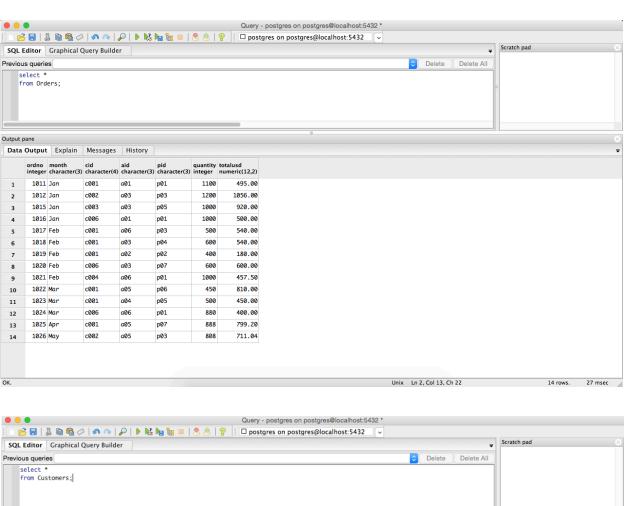
Perry Hong Kong

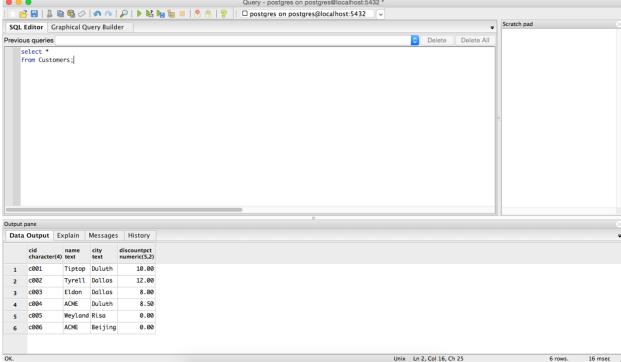
Gray New York

Otasi Duluth

Smith Dallas

Bond London





2.)

A primary key is a unique identifier assigned to each row in a table, to distinguish among the rows within the relation. A super key is an attribute or set of attributes that uniquely identify a row within a relation. Finally, candidate keys are the different columns in a table that account for the uniqueness in the row. Primary key is an example of a candidate key.

3.)

There are numerous data types that are supported by SQL. There are CHAR and VARCHAR, BIT and BIT VARYING, BOOLEAN, INT or INTEGER, FLOAT or REAL, NUMERIC, and DATE and TIME. CHAR and VARCHAR denote fixed-length strings of up to n characters. BIT and BIT VARYING are like CHAR and VARCHAR in that they represent strings of fixed and varying lengths. However, instead of characters, these data types denote bit strings. BOOLEAN signifies a value that is logical. For example, the value could be TRUE, FALSE, or UNKOWN. INT involves typical integer values. FLOAT or REAL are associated with classic floating point numbers (eg. 0123.45). NUMERIC values are any that can be measured. Finally, DATE and TIME are just strings of a special form. These values can be represented as string types, or may be designated as data and time, depending on what makes sense.

An example of a table that may be created could be a student table, containing data regarding different students in a high school math class. The name of this table is Mrs. Dunn's 11<sup>th</sup> Grade Math. Some of the columns could be student ID number, student name, HW1, HW2, Project1, Lab1, Midterm, and Final. The student ID number would be of the INT data type, and

cannot be null. It is the primary key in this table and must have a value for all new rows. Student name would be a CHAR data type because it contains all characters, and cannot be null. Every student in each row must have a student ID and student name. The rest of the values are NUMERIC, and some may be null. This is because a student may not take a test, so their row would be null for that test.

4.)

The "first normal form" rule basically states that all values must be in a database table, with rows and columns. These rows must contain atomic values only. This means that the values cannot be divided (eg. separated by semicolons, commas, etc.) For example, if the column held values of locations of business, the row cannot contain a value of Chicago: New York City: California. It must only contain one value. One column could be Location1 and could contain Chicago. Another could be Location2 and contain New York City, and so on. For database normalization to occur, the first normal form is imperative and is a minimum requirement to improve the database structure.

The access rows by content only rule implies that you cannot refer to a row in a location such as "third one down to the right" because the rows do not actually have order. They are called upon by the value that they contain. This is incredibly important when querying because there is no way in the language to ask to retrieve the third row down because that does not exist. It would have to ask for a value that is in a row to call from that spot.

Finally, the "all rows must be unique" rule states that two rows cannot be identical in every attribute. This would make them the same row, and leads to duplicate values. This is important because in the querying language there must be a way to distinguish two rows from

each other, so their values cannot be the same for all values. A way to resolve this is by ensuring that there are primary keys in place that create this uniqueness between the rows, and help to differentiate between information.