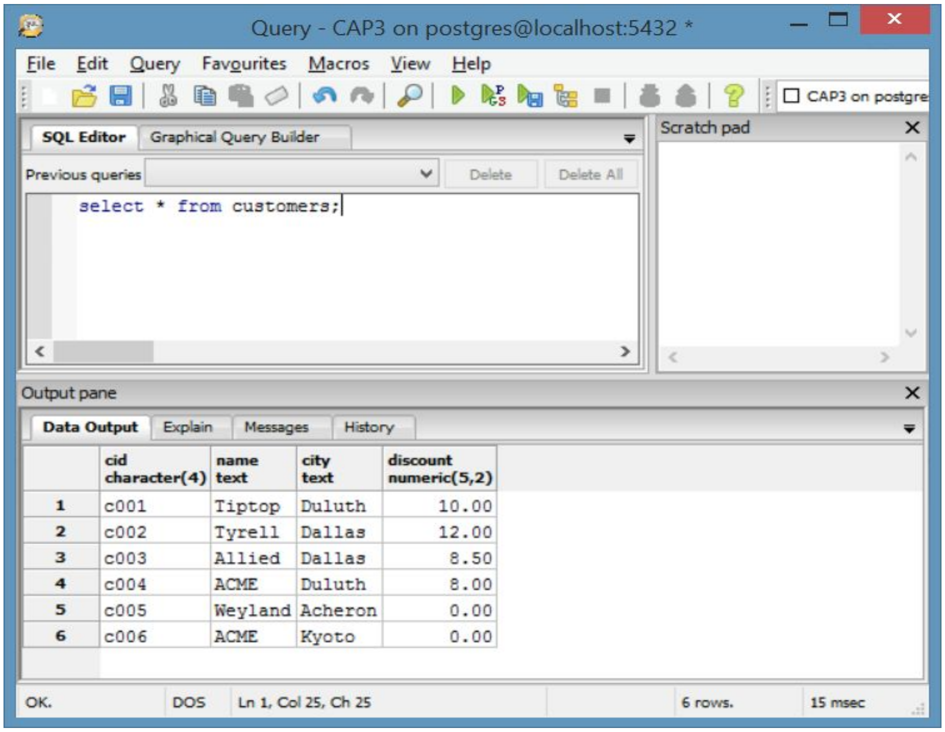
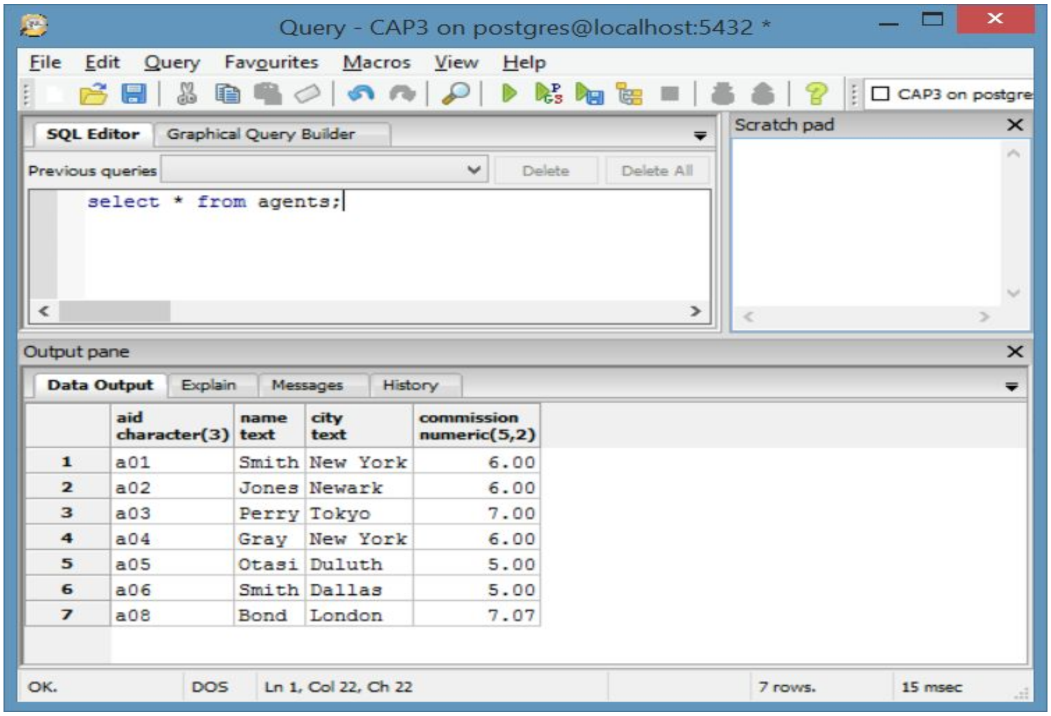
Iglika Hadjiyska
Database Management
Labouseur
February 1st, 2016
Lab 2: CAP Database



```
select * from customers;
```

	cid character(4)	name text	city text	discount numeric(5,2)
1	c001	Tiptop	Duluth	10.00
2	c002	Tyrell	Dallas	12.00
3	c003	Allied	Dallas	8.50
4	c004	ACME	Duluth	8.00
5	c005	Weyland	Acheron	0.00
6	c006	ACME	Kyoto	0.00

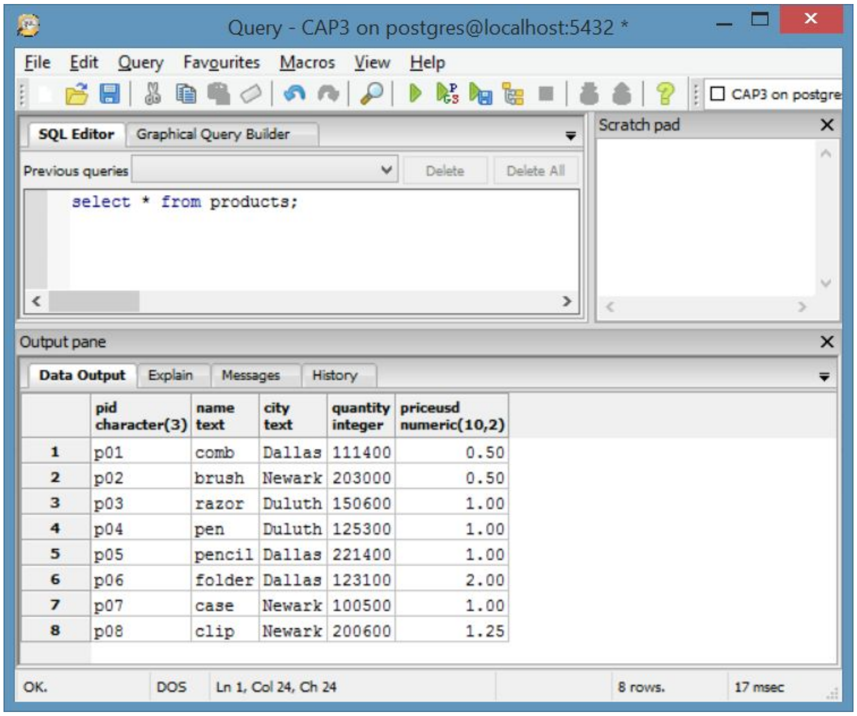
OK. DOS Ln 1, Col 25, Ch 25 6 rows. 15 msec



```
select * from agents;
```

	aid character(3)	name text	city text	commission numeric(5,2)
1	a01	Smith	New York	6.00
2	a02	Jones	Newark	6.00
3	a03	Perry	Tokyo	7.00
4	a04	Gray	New York	6.00
5	a05	Otasi	Duluth	5.00
6	a06	Smith	Dallas	5.00
7	a08	Bond	London	7.07

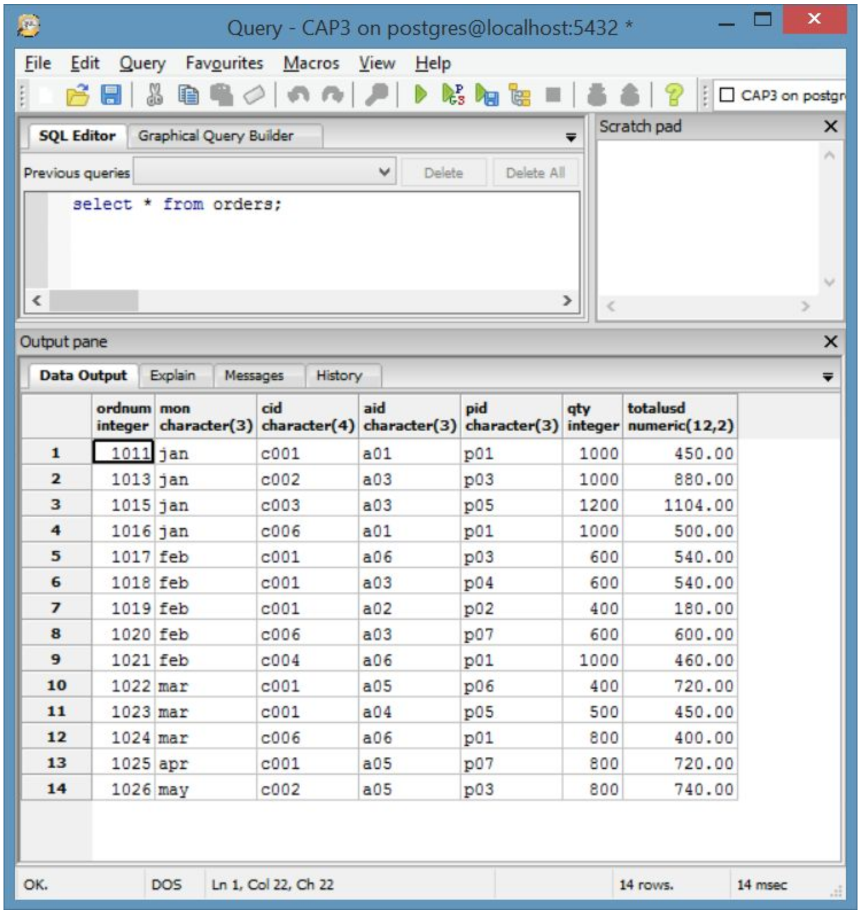
OK. DOS Ln 1, Col 22, Ch 22 7 rows. 15 msec



```
select * from products;
```

	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	clip	Newark	200600	1.25

OK. DOS Ln 1, Col 24, Ch 24 8 rows. 17 msec



```
select * from orders;
```

	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	740.00

OK. DOS Ln 1, Col 22, Ch 22 14 rows. 14 msec

Keys

The primary, candidate and superkey are all related in their definitions, most specifically because some are subsets of others. Both primary and candidate keys are superkeys, although they are more specific types. A *superkey* is defined as any field or set of fields that uniquely identifies every row or record in the table. For example, a table about employee information could have two columns, Employee ID and Employee Name. Both of the column names, Employee ID and Employee Name, would be the superkey of the table because they could both identify each row in the table since every employee should have an ID and a name. The candidate key, however, would be more specific in this example. The *candidate key* is the minimal superkey, with the least amount of columns inside of it. Within the example of the employee table, both Employee ID and Employee Name would be candidate keys because not only are they part of the super key but they are the most basic units of information. Employee ID only has ID numbers while Employee Name only has the name of the employee, rather than any other additional information. Furthermore, the *primary key* is a candidate key that is picked to be the reference key for the table and must have unique and not null values, and must uniquely identify every record in the table. Primary keys are often used to create relationships between different tables and will be found in other tables as well.

Data Types

All attributes in a database must have a data type, or must be otherwise null. There are six primitive data types that are supported by SQL systems. One of the six is the character data type that consists of strings of different lengths with different characters. Examples of where this type would be used are for employee names or store locations. The bit data type is similar, consisting of strings of varying sizes but with bits instead of characters. Another data type is the boolean, which is for an attribute whose value must be logical, such as TRUE, FALSE or UNKNOWN. This data type would be used for yes and no categories, such as whether an employee has opted into the company's retirement plan. The integer data type consists of number values, which can be used for price columns or discount percentages. Furthermore, the floating point data type also consists of numeric values although they are represented differently, either as an approximated real number or a specific number with fixed decimal point. This data type would be used for calculations and approximations. Finally, the last data type is the data and time type, and its use is self explanatory. This data type uses strings with special formats to express data and time.

To use a real example of these data types, a table written about employee information can be used. The entire table is named Employee Info and each column helps create a profile about each of the employees in a sample company. The first column would be the employee ID, which is also the primary key, and its data type is an integer or character type, depending on whether the ID includes letters and other special characters. Next to that column would be the column describing the employee's name and that data type would most definitely be a character type.

These two columns cannot have types that are null. This is because one column is used as the primary key and must have a value if it is used to create relationships with other tables, and because employees must have a name so they can be readily identified. The next column would be a record of their most recent punch into work, which would be a date and time data type. This value also cannot be nullable unless an employee has been fired. Another column would be a simple informational column about whether an employee is part time or full time, which could be a boolean or bit data type. Since this column answers a simple, logical yes or no question, it can use the boolean data type. If more efficiency is needed, a bit data type can be used to represent the same thing, with 0 representing FALSE and 1 representing TRUE. This column could possibly be null if that information is not yet received but the column could also contain UNKNOWN as well. The column next to this would be displaying the amount of break time that each employee receives, represented by a date and time type or a floating data type. The floating data type would represent the break time simply. This column could also be null if this information is still unknown and is not pertinent to the structure of the table.

Relational Rules

First Normal Rule

This rule applies to the structure of the columns. It states that a column cannot have multiple values. For example, a column about book authors cannot include more than one author in one specific row and column intersection. If there is another author, then another column must be made just for that author. This rule is important because it makes the table simpler to search and read since the user would not have to iterate through multiple values in each column.

Access Rows by Content Only Rule

This rule is applied to accessing and searching through rows. It specifies that reference IDs for rows cannot be used. For example, a database cannot be searched by “Row 6” but can be searched by the row that contains “006”. This ensures that there is no particular order to rows and columns. The importance of this disorder is that it allows free addition to the database without worry of updating the numerical system of the rows, i.e. if a value is added, the database does not have to worry about whether things in “Row 6” will be able to be access just like before.

All Rows Must Be Unique Rows Rule

This rule was created to mitigate duplication in tables. It states that all rows must be unique, meaning that a value in one column cannot be identical to the value in another column in the same row. This rule effectively stops duplication and aids search through the database because it does not display two identical results.