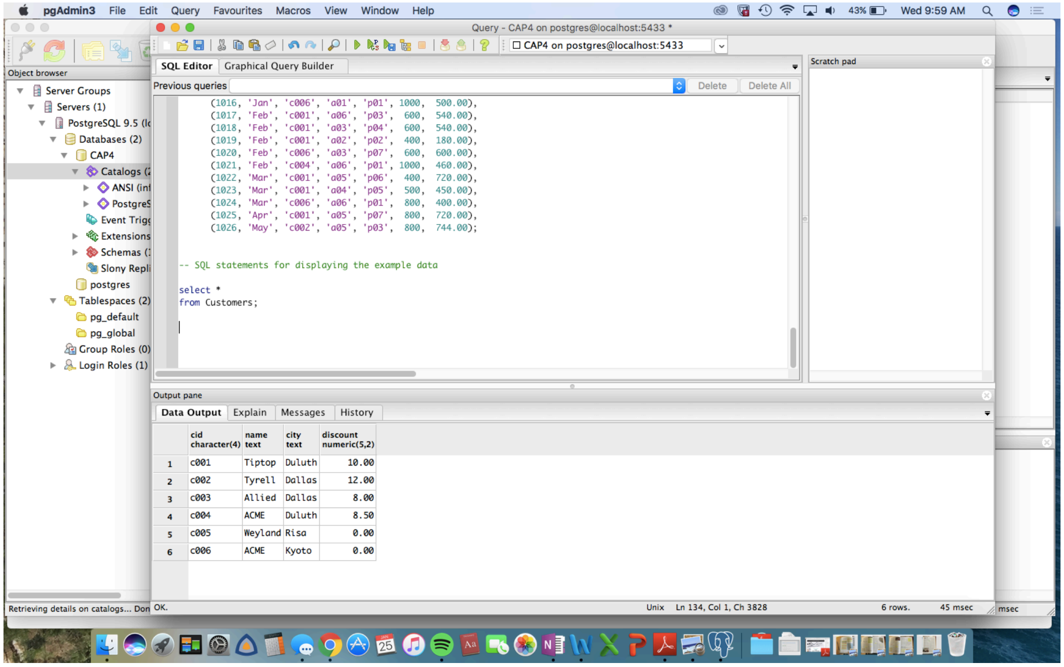
Katelyn Boylan
CMPT 308: Database Systems
Lab 2: CAP Database
Tuesday, January 31st, 2017

1. Execute each query from CAP4 SQL
Customers:

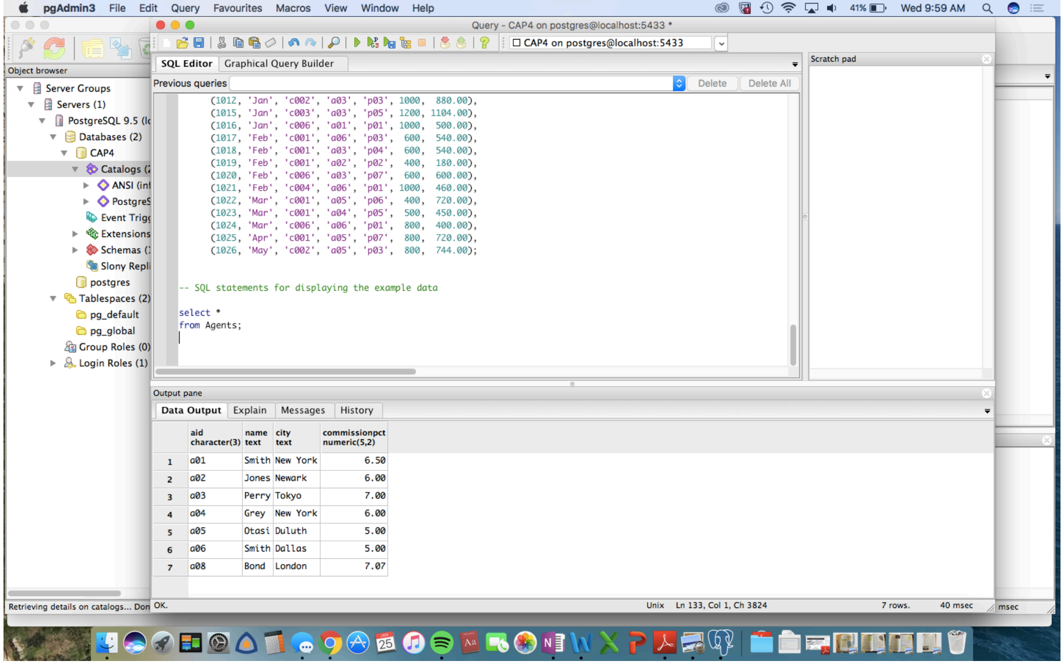

pgAdmin3 File Edit Query Favourites Macros View Window Help
Wed 9:59 AM
Query - CAP4 on postgres@localhost:5433 *
CAP4 on postgres@localhost:5433
SQL Editor
Graphical Query Builder
Previous queries
Delete
Delete All
Scratch pad
Output pane
Data Output
Explain
Messages
History

```
    (1016, 'Jan', 'c006', 'a01', 'p01', 1000,  500.00),
    (1017, 'Feb', 'c001', 'a06', 'p03',  600,  540.00),
    (1018, 'Feb', 'c001', 'a03', 'p04',  600,  540.00),
    (1019, 'Feb', 'c001', 'a02', 'p02',  400,  180.00),
    (1020, 'Feb', 'c006', 'a03', 'p07',  600,  600.00),
    (1021, 'Feb', 'c004', 'a06', 'p01', 1000,  460.00),
    (1022, 'Mar', 'c001', 'a05', 'p06',  400,  720.00),
    (1023, 'Mar', 'c001', 'a04', 'p05',  500,  450.00),
    (1024, 'Mar', 'c006', 'a06', 'p01',  800,  400.00),
    (1025, 'Apr', 'c001', 'a05', 'p07',  800,  720.00),
    (1026, 'May', 'c002', 'a05', 'p03',  800,  744.00);


-- SQL statements for displaying the example data

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	8.00
4	c004	ACME	Duluth	8.50
5	c005	Weyland	Risa	0.00
6	c006	ACME	Kyoto	0.00

```
OK.    Unix    Ln 134, Col 1, Ch 3828    6 rows.    45 msec
```

Agents:

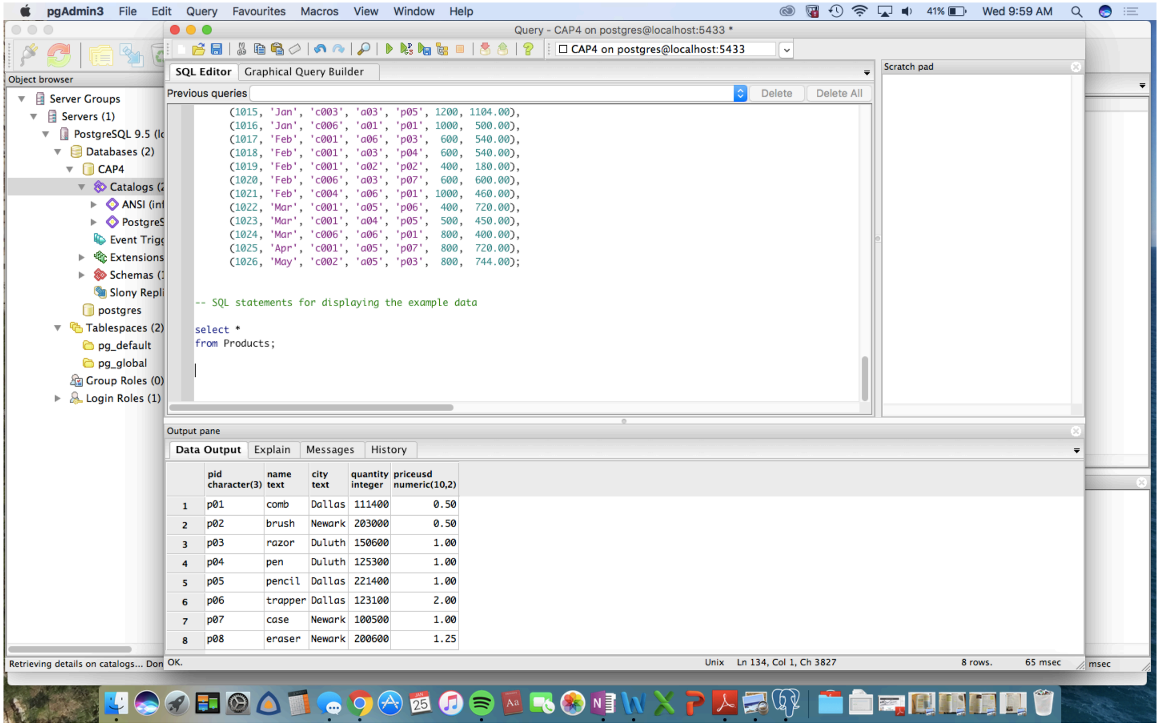

pgAdmin3 File Edit Query Favourites Macros View Window Help
Wed 9:59 AM
Query - CAP4 on postgres@localhost:5433 *
CAP4 on postgres@localhost:5433
SQL Editor
Graphical Query Builder
Previous queries
Delete
Delete All
Scratch pad
Output pane
Data Output
Explain
Messages
History

```
    (1012, 'Jan', 'c002', 'a03', 'p03', 1000,  880.00),
    (1015, 'Jan', 'c003', 'a03', 'p05', 1200, 1104.00),
    (1016, 'Jan', 'c006', 'a01', 'p01', 1000,  500.00),
    (1017, 'Feb', 'c001', 'a06', 'p03',  600,  540.00),
    (1018, 'Feb', 'c001', 'a03', 'p04',  600,  540.00),
    (1019, 'Feb', 'c001', 'a02', 'p02',  400,  180.00),
    (1020, 'Feb', 'c006', 'a03', 'p07',  600,  600.00),
    (1021, 'Feb', 'c004', 'a06', 'p01', 1000,  460.00),
    (1022, 'Mar', 'c001', 'a05', 'p06',  400,  720.00),
    (1023, 'Mar', 'c001', 'a04', 'p05',  500,  450.00),
    (1024, 'Mar', 'c006', 'a06', 'p01',  800,  400.00),
    (1025, 'Apr', 'c001', 'a05', 'p07',  800,  720.00),
    (1026, 'May', 'c002', 'a05', 'p03',  800,  744.00);


-- SQL statements for displaying the example data

select *
from Agents;
```

	aid character(3)	name text	city text	commissionpct numeric(5,2)
1	a01	Smith	New York	6.50
2	a02	Jones	Newark	6.00
3	a03	Perry	Tokyo	7.00
4	a04	Grey	New York	6.00
5	a05	Otasi	Duluth	5.00
6	a06	Smith	Dallas	5.00
7	a08	Bond	London	7.07

```
OK.    Unix    Ln 133, Col 1, Ch 3824    7 rows.    40 msec
```

Products:

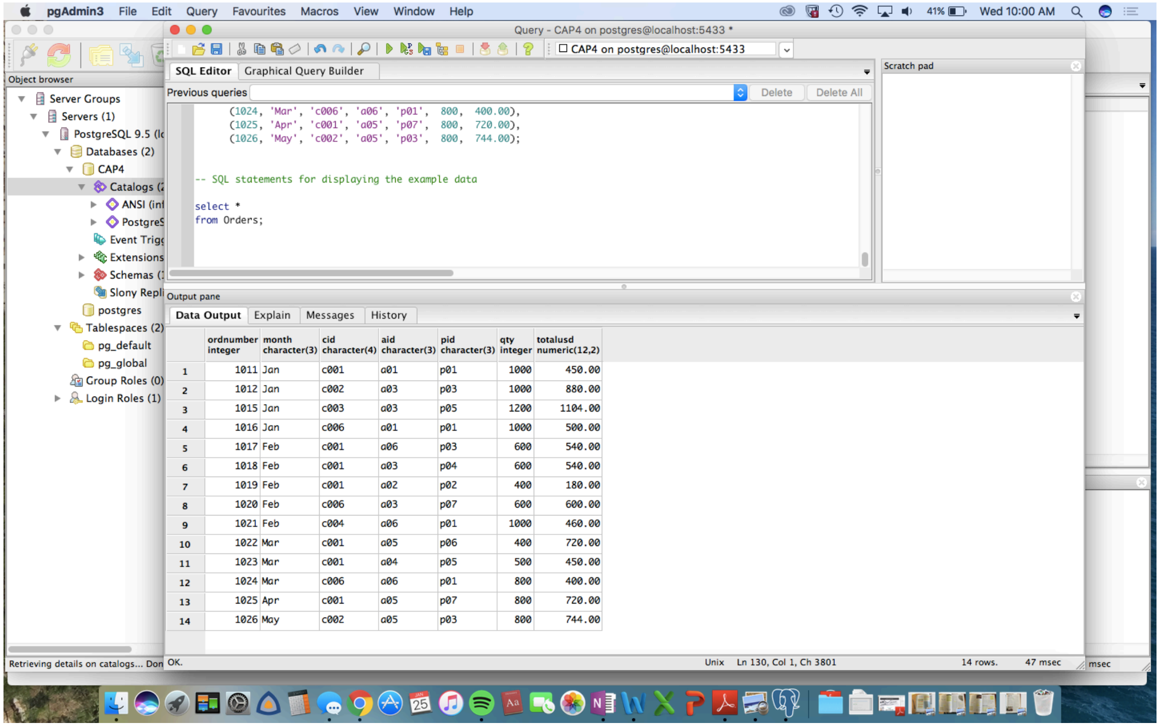


```
    (1015, 'Jan', 'c003', 'a03', 'p05', 1200, 1104.00),
    (1016, 'Jan', 'c006', 'a01', 'p01', 1000,  500.00),
    (1017, 'Feb', 'c001', 'a06', 'p03',  600,  540.00),
    (1018, 'Feb', 'c001', 'a03', 'p04',  600,  540.00),
    (1019, 'Feb', 'c001', 'a02', 'p02',  400,  180.00),
    (1020, 'Feb', 'c006', 'a03', 'p07',  600,  600.00),
    (1021, 'Feb', 'c004', 'a06', 'p01', 1000,  460.00),
    (1022, 'Mar', 'c001', 'a05', 'p06',  400,  720.00),
    (1023, 'Mar', 'c001', 'a04', 'p05',  500,  450.00),
    (1024, 'Mar', 'c006', 'a06', 'p01',  800,  400.00),
    (1025, 'Apr', 'c001', 'a05', 'p07',  800,  720.00),
    (1026, 'May', 'c002', 'a05', 'p03',  800,  744.00);


-- SQL statements for displaying the example data

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	trapper	Dallas	123100	2.00
7	p07	case	Newark	100500	1.00
8	p08	eraser	Newark	200600	1.25

OK.

Orders:

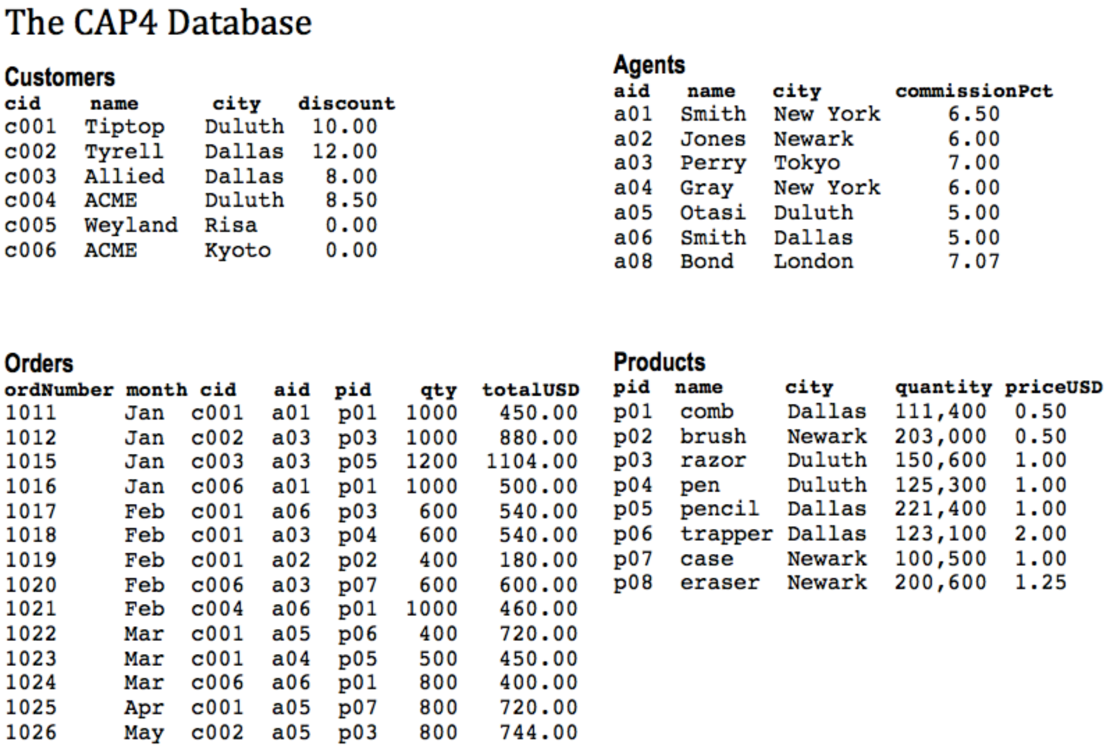


```
    (1024, 'Mar', 'c006', 'a06', 'p01',  800,  400.00),
    (1025, 'Apr', 'c001', 'a05', 'p07',  800,  720.00),
    (1026, 'May', 'c002', 'a05', 'p03',  800,  744.00);


-- SQL statements for displaying the example data

select *
from Orders;
```

	ordnumber integer	month 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	1012	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

OK.

CAP snapshot:



The CAP4 Database

Customers

cid	name	city	discount
c001	Tiptop	Duluth	10.00
c002	Tyrell	Dallas	12.00
c003	Allied	Dallas	8.00
c004	ACME	Duluth	8.50
c005	Weyland	Risa	0.00
c006	ACME	Kyoto	0.00

Agents

aid	name	city	commissionPct
a01	Smith	New York	6.50
a02	Jones	Newark	6.00
a03	Perry	Tokyo	7.00
a04	Gray	New York	6.00
a05	Otasi	Duluth	5.00
a06	Smith	Dallas	5.00
a08	Bond	London	7.07

Orders

ordNumber	month	cid	aid	pid	qty	totalUSD
1011	Jan	c001	a01	p01	1000	450.00
1012	Jan	c002	a03	p03	1000	880.00
1015	Jan	c003	a03	p05	1200	1104.00
1016	Jan	c006	a01	p01	1000	500.00
1017	Feb	c001	a06	p03	600	540.00
1018	Feb	c001	a03	p04	600	540.00
1019	Feb	c001	a02	p02	400	180.00
1020	Feb	c006	a03	p07	600	600.00
1021	Feb	c004	a06	p01	1000	460.00
1022	Mar	c001	a05	p06	400	720.00
1023	Mar	c001	a04	p05	500	450.00
1024	Mar	c006	a06	p01	800	400.00
1025	Apr	c001	a05	p07	800	720.00
1026	May	c002	a05	p03	800	744.00

Products

pid	name	city	quantity	priceUSD
p01	comb	Dallas	111,400	0.50
p02	brush	Newark	203,000	0.50
p03	razor	Duluth	150,600	1.00
p04	pen	Duluth	125,300	1.00
p05	pencil	Dallas	221,400	1.00
p06	trapper	Dallas	123,100	2.00
p07	case	Newark	100,500	1.00
p08	eraser	Newark	200,600	1.25

All queries executed on Postgres are identical to the CAP data tables above.

2. In each relational database, there is a superkey, primary key, and candidate key. A superkey is a column or set of columns where every row is unique and there is no duplication. For example, Name is a superkey in the "Products" data table in the CAP database. priceUSD is not because there is duplicates. However, name and priceUSD together is a superkey. When deciding on a candidate key, the database management team must keep in mind the context of the project and the information they are looking for from this data. The candidate key is the minimal superkey that uniquely identifies each row. Name can be the candidate key of the "Products" table because it specifies each of the other columns in the set. Lastly, the primary key is the chosen candidate key. Pid in "Products" would be the best choice for the primary key because it is consistent with the other primary keys in the "Customers" data table. In addition to this, the PID is unique to each row and unlike the name column, the pid number will not repeat and will be unique to each product.

3. Data types are defined by the values it can take. Examples of data types are text, number, and date of birth. The user cannot type a number into a text column such as name. When writing SQL code, programmers must be as specific as possible in order for the user and analyst to have no confusion in data type, therefore being able to put context to the project to pull information and conclusions from the data. A good example of this is cost. It must be specified in currency, otherwise analysts do not know what to conclude from their database.

An example of a database is a group of team members. The members must specify their name, date of birth, and age. The table name will be "Group 1." The name column must be text, date of birth will be dated, and age will be a number. Data is "nullable" when the data in the column does not fit the data type specified. In the case of the "Group 1" database, each row of data should not be NULL. This is because the data types are specific and common.

4. The relational database "rules" was written by Edgar Codd, the father of relational databases. The three rules are key to a clean and specific database. The first rule, otherwise known as the First Normal form rule or 1NF, says that you can't have anything that has its own structure in a data table. In other words, this means that there can't be more than one data type in a row or column. An example of this is having multiple pieces of information in one column and row. A solution to this problem is to create another table with a primary key to connect your information to. Data must be atomic and not multivalued. This is another reason why having specific and different column names are important for each data type.

The next relational database rule says that access to rows must be by content only. This is the "What, not where" rule. A data table is not specified by row, instead by the data in that row. What's in row 5 one day may be in row 6 the next day. A user would run into trouble if they were to ask for data from row 5 when they really the data from row 6. Sets don't have an intrinsic order.

Lastly, Codd's third rule says that for each table, all rows must be unique. There can be now duplicates of rows because users may double count data, which would lead to the wrong conclusions about information. By each data table being unique, the database is consistent and clean. With all three rules and context, information can be pulled from every relational database.