Megan O’Halloran

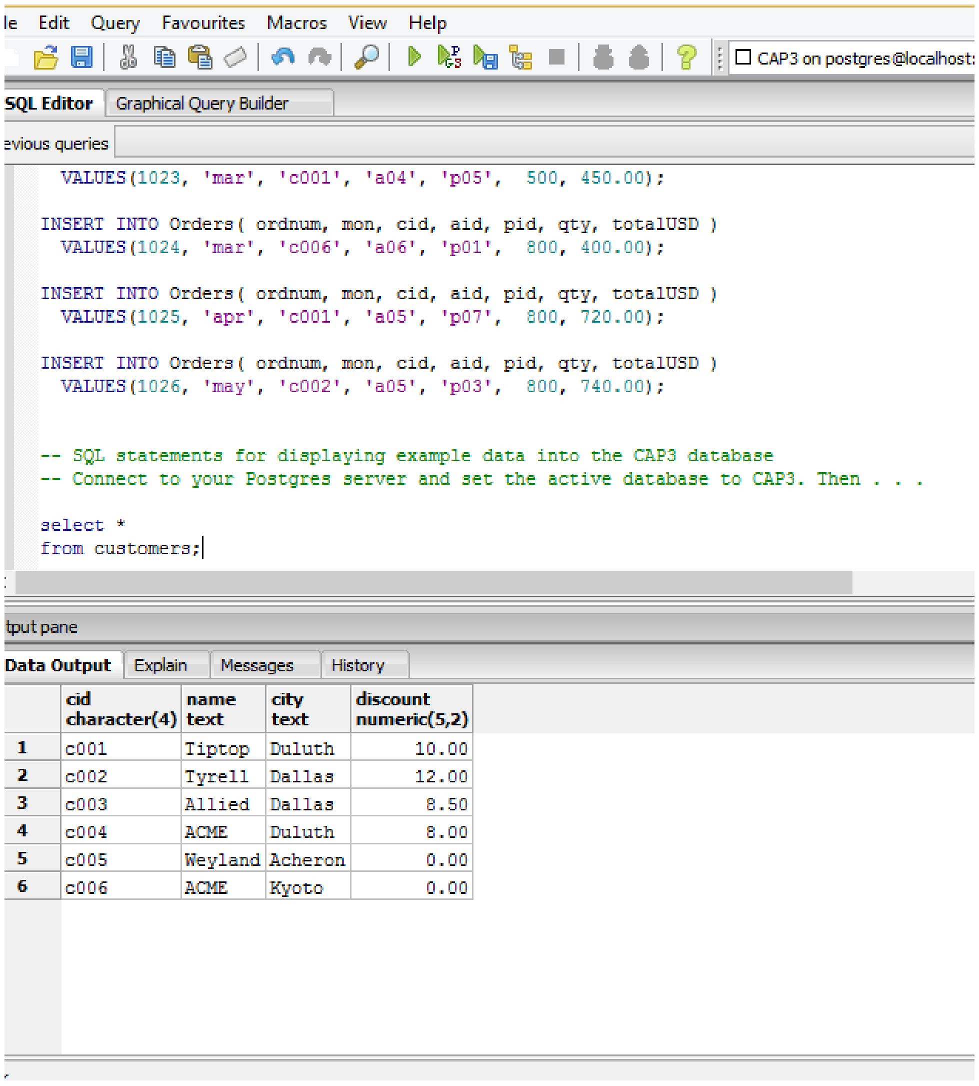
CAP Database

January 27, 2016

1. Screenshots:

   Select *

   Customers;

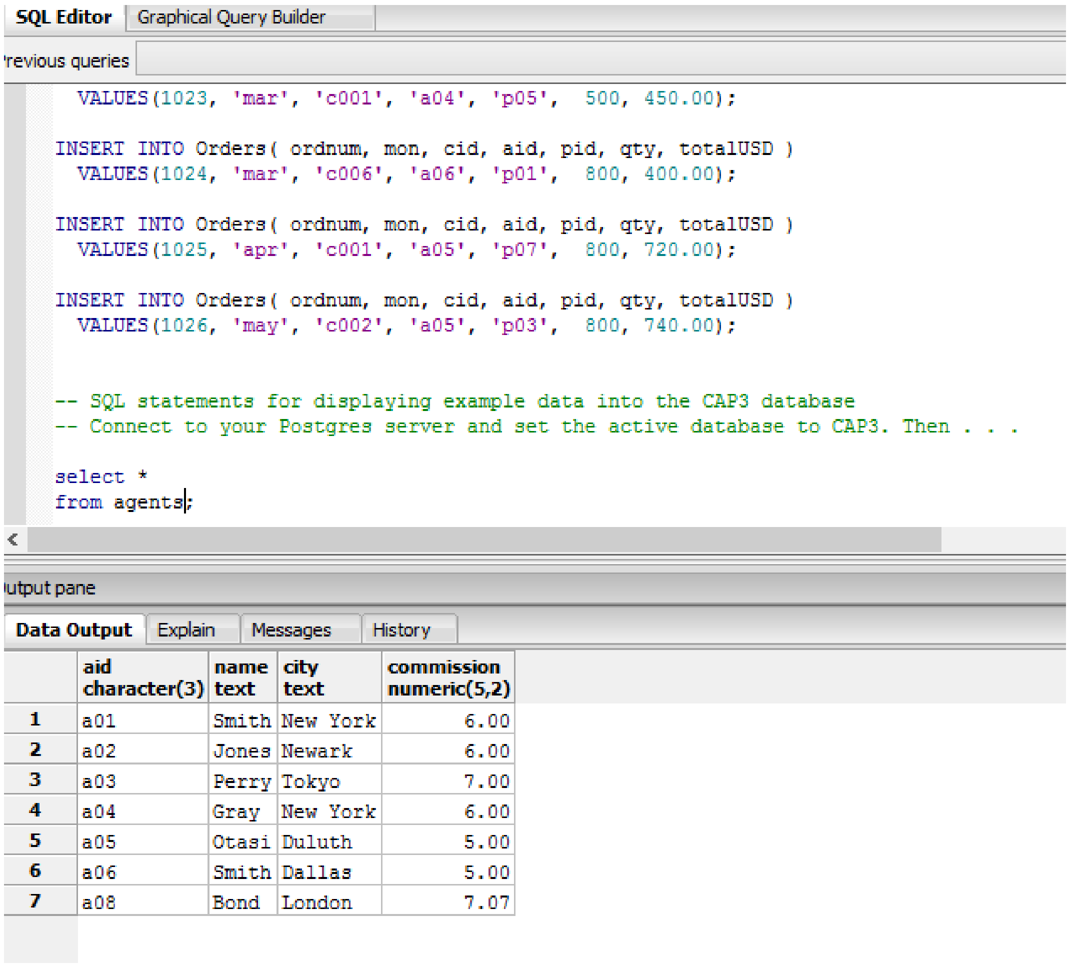


```
  VALUES(1023, 'mar', 'c001', 'a04', 'p05',  500, 450.00);

INSERT INTO Orders( ordnum, mon, cid, aid, pid, qty, totalUSD )
  VALUES(1024, 'mar', 'c006', 'a06', 'p01',  800, 400.00);

INSERT INTO Orders( ordnum, mon, cid, aid, pid, qty, totalUSD )
  VALUES(1025, 'apr', 'c001', 'a05', 'p07',  800, 720.00);

INSERT INTO Orders( ordnum, mon, cid, aid, pid, qty, totalUSD )
  VALUES(1026, 'may', 'c002', 'a05', 'p03',  800, 740.00);


-- SQL statements for displaying example data into the CAP3 database
-- Connect to your Postgres server and set the active database to CAP3. Then . . .

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

select *

from agents;

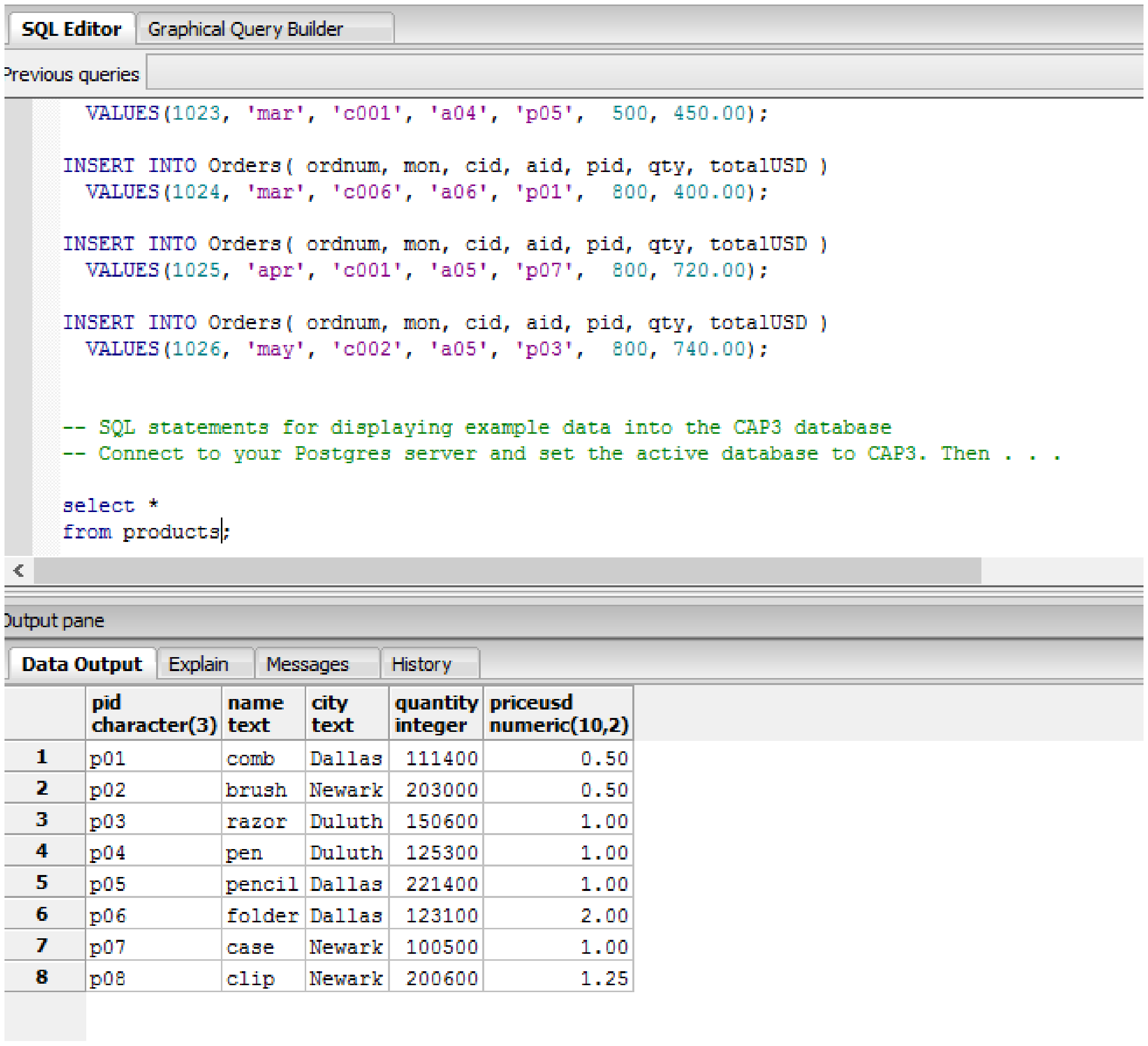


```
  VALUES(1023, 'mar', 'c001', 'a04', 'p05',  500, 450.00);

INSERT INTO Orders( ordnum, mon, cid, aid, pid, qty, totalUSD )
  VALUES(1024, 'mar', 'c006', 'a06', 'p01',  800, 400.00);

INSERT INTO Orders( ordnum, mon, cid, aid, pid, qty, totalUSD )
  VALUES(1025, 'apr', 'c001', 'a05', 'p07',  800, 720.00);

INSERT INTO Orders( ordnum, mon, cid, aid, pid, qty, totalUSD )
  VALUES(1026, 'may', 'c002', 'a05', 'p03',  800, 740.00);


-- SQL statements for displaying example data into the CAP3 database
-- Connect to your Postgres server and set the active database to CAP3. Then . . .

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

select *

from products;

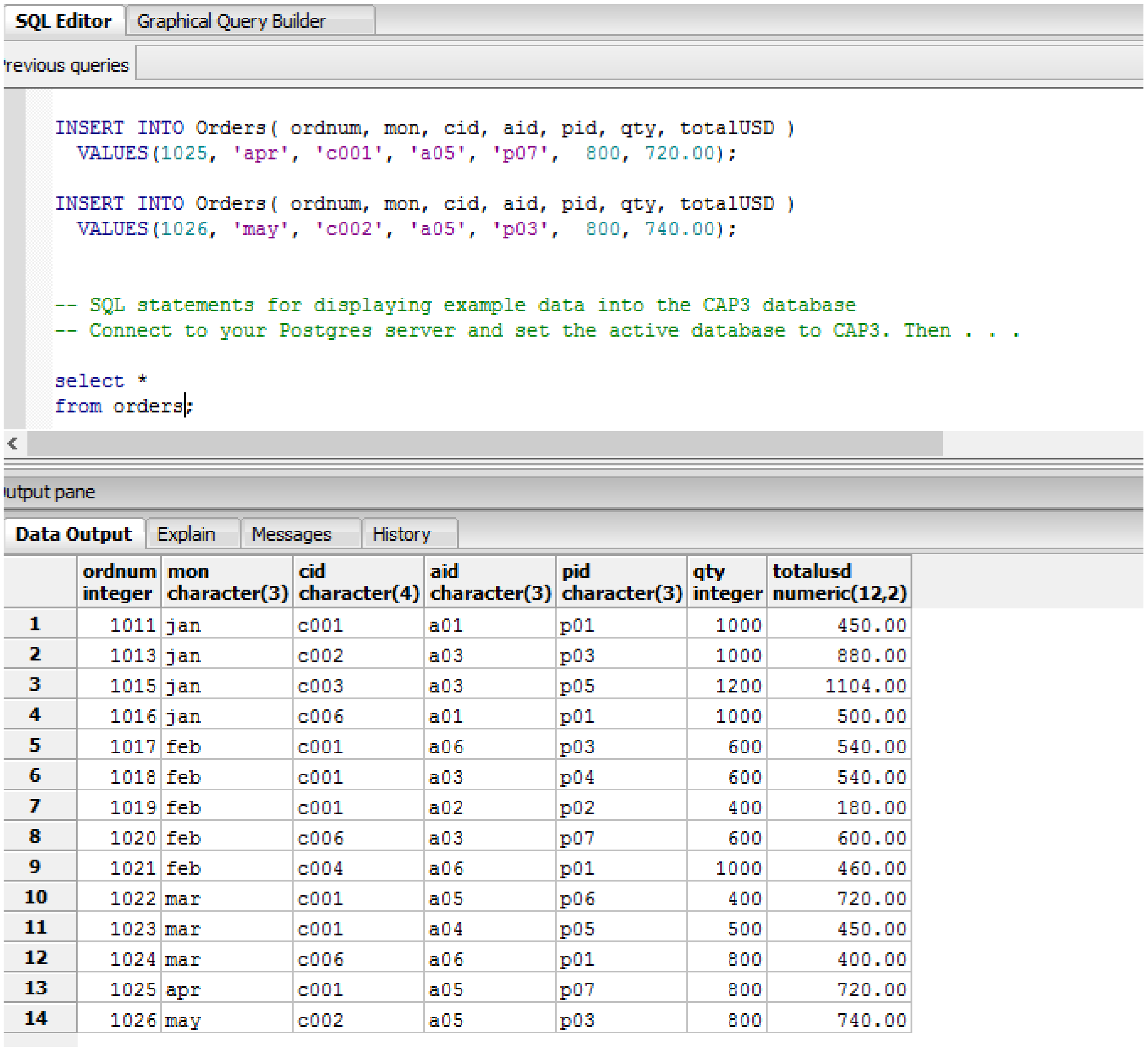


```
  VALUES(1023, 'mar', 'c001', 'a04', 'p05',  500, 450.00);

INSERT INTO Orders( ordnum, mon, cid, aid, pid, qty, totalUSD )
  VALUES(1024, 'mar', 'c006', 'a06', 'p01',  800, 400.00);

INSERT INTO Orders( ordnum, mon, cid, aid, pid, qty, totalUSD )
  VALUES(1025, 'apr', 'c001', 'a05', 'p07',  800, 720.00);

INSERT INTO Orders( ordnum, mon, cid, aid, pid, qty, totalUSD )
  VALUES(1026, 'may', 'c002', 'a05', 'p03',  800, 740.00);


-- SQL statements for displaying example data into the CAP3 database
-- Connect to your Postgres server and set the active database to CAP3. Then . . .

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

select *

from orders;



```
INSERT INTO Orders( ordnum, mon, cid, aid, pid, qty, totalUSD )
  VALUES(1025, 'apr', 'c001', 'a05', 'p07',  800, 720.00);

INSERT INTO Orders( ordnum, mon, cid, aid, pid, qty, totalUSD )
  VALUES(1026, 'may', 'c002', 'a05', 'p03',  800, 740.00);


-- SQL statements for displaying example data into the CAP3 database
-- Connect to your Postgres server and set the active database to CAP3. Then . . .

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

2. The primary key is a column that has a unique identifier for its records. A database must always have one primary key. A candidate key can be any column, or combination of columns, that can be a unique key in a database. There can be more than one candidate keys. A candidate key can also be a primary key. The superkey is a set of columns in a table for which there are no two rows that will share the same combination of values. So, the superkey is unique for each and every row in the table. A superkey can also be just a single column.

3. A table could be created for a small business which contains their employees and their information. The name of the table would be MO Employees. The fields of this database would be the following: first name, last name, employee ID, date of birth, social security number, phone number, address, date they were employed, department, and title. The fields date of birth and date they were employed would be the data type datetime, and they are both not nullable. The social security number, phone number and address fields are all char data type and are all not nullable except for the social security number field because an employee could be on a work visa or something else, making it nullable. Finally, the fields labeled first name, last name, department, and title all have the data type varchar and are all not nullable.

4.
    a. The “first normal form” rule: You must define the data. The data must be looked at, organized into columns and defining their data type, and putting it into the related column. An example of this would be if a business had information about their employees, and creating a table to input all of that information. This rule is important because it states that data must be named and organized, making it easier to access information.
    b. The “access rows by content only” rule: This states that the only way we can retrieve the information we want is by the content in the row. Some people do not obey this rule and will look things up by the row number ID. However this rule is important because, depending on what you are looking for, things could be out of order. An outsider doesn’t know how the database is organized, it could be organized by order number, or employee ID, so when they look up the information they are looking for, they could get the wrong thing, or numerous things. An example of this rule would be by getting information on a certain product; they would have to find the product number or name.
    c. The “all rows must be unique” rule: This rule is exactly what it sounds like. The rows in a database cannot be exactly the same. This rule is important because it prevents the problem of having repeated information on the same thing. An

example of this rule would be the employee ID number in a business, which could also be the primary key. This gives every row a unique column, making each row unique.