

Danny Puckett

02/06/17

Lab 2 – Database CMPT 308N

- 1.) After running the code from, <http://www.labouseur.com/courses/db/cap4.txt>, and running the following queries... (at one time):
select * from Customers;

I have the following output from said queries:

Query - postgres on postgres@localhost:5432 - [C:\Users\Danny Puckett\Desktop\Databases repo\second lab.sql] *

File Edit Query Favourites Macros View Help

SQL Editor Graphical Query Builder

Previous queries

```
-- SQL statements for displaying the example data

select *
from Customers;

--select *
--from Agents;

--select *
--from Products;

--select *
--from Orders;
```

Output pane

Data Output Explain Messages History

	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. DOS Ln 154, Col 3, C

select * from Agents;

Query - postgres on postgres@localhost:5432 - [C:\Users\Danny Puckett\Desktop\Databases repo\second lab.sql] *

File Edit Query Favurites Macros View Help

postgres on postgres@localhost:5432

SQL Editor Graphical Query Builder

Previous queries Delete Delete All

```
-- SQL statements for displaying the example data  
  
--select *  
--from Customers;  
  
select *  
from Agents;  
  
--select *  
--from Products;  
  
--select *  
--from Orders;
```

Scratch pad

Output pane

Data Output Explain Messages History

	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. DOS Ln 148, Col 3, Ch 4665 7 rows. 169 msec

select * from Products;

Query - postgres on postgres@localhost:5432 *

File Edit Query Favurites Macros View Help

postgres on postgres@localhost:5432

SQL Editor Graphical Query Builder

Previous queries Delete Delete All

```
(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;  
  
--select *  
--from Agents;  
  
select *  
from Products;  
  
--select *  
--from Orders;
```

Scratch pad

Output pane

Data Output Explain Messages History

	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. DOS Ln 157, Col 3, Ch 4756 8 rows. 216 msec

select * from Orders;

The screenshot shows a PostgreSQL SQL Editor window titled "Query - postgres on postgres@localhost:5432". The SQL Editor pane contains the following query:

```
--from Customers;
--select *
--from Agents;
--select *
--from Products;
select *
from Orders;
```

The Output pane displays the results of the query in a table format. The table has 8 columns: **ordnumber** (integer), **month** (character(3)), **cid** (character(4)), **aid** (character(3)), **pid** (character(3)), **qty** (integer), and **totalusd** (numeric(12,2)). The results are as follows:

ordnumber	month	cid	aid	pid	qty	totalusd
1	1011	Jan	c001	a01	p01	1000
2	1012	Jan	c002	a03	p03	1000
3	1015	Jan	c003	a03	p05	1200
4	1016	Jan	c006	a01	p01	1000
5	1017	Feb	c001	a06	p03	600
6	1018	Feb	c001	a03	p04	600
7	1019	Feb	c001	a02	p02	400
8	1020	Feb	c006	a03	p07	600
9	1021	Feb	c004	a06	p01	1000
10	1022	Mar	c001	a05	p06	400
11	1023	Mar	c001	a04	p05	500
12	1024	Mar	c006	a06	p01	800
13	1025	Apr	c001	a05	p07	800
14	1026	May	c002	a05	p03	800

The status bar at the bottom indicates "OK", "DOS", "Ln 153, Col 3, Ch 4716", "14 rows.", and "185 msec".

2.)

Explain the distinction among the terms primary keys, candidate key, and super key.

A super key is a column or set of columns within a data table that insures every row will be unique.

A candidate key is a minimized super key, to the point that its is at its most reduced state and at its best utilization.

A primary key is a chosen candidate key for each data table for correlational purposes with the un-remarked upon foreign key...

To be noted, the super key and candidate key are said to be unique to their primary tables...

3.)Write a brief essay on data on data types. Select a topic for which you might create a table. Name the table and list its fields (columns). For each field, give its data type and whether or not it is nullable.

By google definition a data type is: a particular kind of data item, as defined by the values it can take, the programming language used, or the operations that can be performed on it. (google.com) A few examples of data types are calendars, time of day, or a restaurant menu... select a data type: I'll select a Calendar. Its columns (fields) would be the days of the week; Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, and Saturday. And as a whole none of these are nullable in totality, yet every day of the week is considered nullable for its first week value except for Saturday, and all are considered nullable for the last week of the month except for Sunday... to that degree they are all nullable in specific cases depending on the "way the days fall".

4.) Explain the following relational “rules” with examples and reasons why they are important.

- a. The “First normal form” rule
- b. The “access rows by content only” rule
- c. The “all rows must be unique” rule

a. The first normal form “rule” is that all data is to be at its atomical value, its most concise value.

Example: 2 is 2, not 2.00 unless needed for uniform reasoning...

b. The access rows by content only “rule” is Codd’s second Relational Rule, to which data is to be accessed by content not value.. you don’t select elements of a data table by row and column, you do so with select, from, and where in relation to the entire data construct to be evaluated.

Example: not to be, selecting row 5 column 2. To be, select * from Orders where month = Feb and ordNumber = 1017....

c. The all rows must be unique “rule” is just that, all rows in your data construct must be unique. If they aren’t you are either repeating yourself, which causes redundancy, or the rows aren’t unique they just hazily seem to be. Same content, different field reasoning....

Example: in a data table of a calendar for this example... you cant have a Sunday February 5th 2017 and also have a Monday February 5th 2017, its redundant, and it doesn’t happen. You have to wait 6 years for a next occurring incidence of Monday to fall on the 5th in February.