

Hannah Youssef

Dr. Alan Labouseur

Database Management

11 September 2017

Lab 2 – CAP Database

1. pgAdmin3 Queries screenshots

Customers

The screenshot shows the pgAdmin3 Query Editor interface. The top toolbar includes icons for file operations, query execution, and window management. The main window is titled "Query - postgres on postgres@localhost:5432 *". The "SQL Editor" tab is active, displaying a query that selects data from the Customers table. The query is as follows:

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

select *
from Agents;

select *
from Products;

select *
from Orders;
```

The "Output pane" at the bottom shows the results of the query. It includes tabs for "Data Output", "Explain", "Messages", and "History". The "Data Output" tab is selected, showing a table with 6 rows and 5 columns: cid, name, city, discount, and pct. The data is as follows:

	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	Eldon	Dallas	8.00
4	c004	ACME	Duluth	8.50
5	c005	Weyland	Risa	0.00
6	c006	ACME	Beijing	0.00

The status bar at the bottom indicates "OK", "Unix", "Ln 147, Col 1, Ch 4576", "24 chars", "6 rows", and "19 msec".

Agents

The screenshot shows a PostgreSQL SQL Editor window titled "Query - postgres on postgres@localhost:5432 *". The window has a menu bar, a toolbar, and a tab bar with "SQL Editor" and "Graphical Query Builder". The "SQL Editor" tab is active, showing a list of previous queries and a scratch pad. The previous queries list contains 12 queries, each with a unique ID and a list of parameters. The scratch pad is empty. The output pane at the bottom shows the results of the last query, which is a table with 5 columns: aid, name, city, and commission. The table contains 7 rows of data.

Previous queries

- (1017, 'Feb', 'c001', 'a06', 'p03', 500, 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, 457.50),
- (1022, 'Mar', 'c001', 'a05', 'p06', 450, 810.00),
- (1023, 'Mar', 'c001', 'a04', 'p05', 500, 450.00),
- (1024, 'Mar', 'c006', 'a06', 'p01', 880, 400.00),
- (1025, 'Apr', 'c001', 'a05', 'p07', 888, 799.20),
- (1026, 'May', 'c002', 'a05', 'p03', 808, 711.04);

-- SQL statements for displaying the example data

```
select *
from Customers;

select *
from Agents;

select *
from Products;

select *
from Orders;
```

Output pane

	aid character(3)	name text	city text	commission numeric(5,2)
1	a01	Smith	New York	5.60
2	a02	Jones	Newark	6.00
3	a03	Perry	Hong Kong	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. Unix Ln 150, Col 1, Ch 4602 21 chars 7 rows. 19 msec

Products

The screenshot shows a PostgreSQL query editor window titled "Query - postgres on postgres@localhost:5432 *". The window has a menu bar, a toolbar, and a tabbed interface with "SQL Editor" and "Graphical Query Builder". The "SQL Editor" tab is active, displaying a list of products and their details. The "Previous queries" section shows a list of products with their IDs, names, cities, quantities, and prices. The "Scratch pad" is empty. The "Output pane" is at the bottom, showing the "Data Output" tab with a table of product details.

Previous queries

```
(1017, 'Feb', 'c001', 'a06', 'p03', 500, 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, 457.50),
(1022, 'Mar', 'c001', 'a05', 'p06', 450, 810.00),
(1023, 'Mar', 'c001', 'a04', 'p05', 500, 450.00),
(1024, 'Mar', 'c006', 'a06', 'p01', 880, 400.00),
(1025, 'Apr', 'c001', 'a05', 'p07', 888, 799.20),
(1026, 'May', 'c002', 'a05', 'p03', 808, 711.04);

-- SQL statements for displaying the example data

select *
from Customers;

select *
from Agents;

select *
from Products;

select *
from Orders;
```

Output pane

Data Output

	pid character(3)	name text	city text	qty integer	priceusd numeric(10,2)
1	p01	Heisenberg compensator	Dallas	111400	0.50
2	p02	universal translator	Newark	203000	0.50
3	p03	Commodore PET	Duluth	150600	1.00
4	p04	LCARS module	Duluth	125300	1.00
5	p05	pencil	Dallas	221400	1.00
6	p06	trapper keeper	Dallas	123100	2.00
7	p07	flux capacitor	Newark	100500	1.00
8	p08	HAL 9000 memory core	Newark	200600	1.25

OK. Unix Ln 153, Col 1, Ch 4625 23 chars 8 rows. 18 msec

Orders

The screenshot shows a PostgreSQL query editor window titled "Query - postgres on postgres@localhost:5432 *". The window has a menu bar, a toolbar, and a tab labeled "SQL Editor". Below the tab, there's a "Previous queries" section with a dropdown and "Delete" and "Delete All" buttons. The main editor area contains SQL code for displaying example data. To the right of the editor is a "Scratch pad" area. Below the editor is an "Output pane" with tabs for "Data Output", "Explain", "Messages", and "History". The "Data Output" tab is active, showing a table with 14 rows and 8 columns. The status bar at the bottom indicates "OK.", "Unix", "Ln 156, Col 1, Ch 4650", "21 chars", "14 rows.", and "14 msec".

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

select *
from Customers;

select *
from Agents;

select *
from Products;

select *
from Orders;
```

	ordno integer	month character(3)	cid character(4)	aid character(3)	pid character(3)	quantity integer	totalusd numeric(12,2)
1	1011	Jan	c001	a01	p01	1100	495.00
2	1012	Jan	c002	a03	p03	1200	1056.00
3	1015	Jan	c003	a03	p05	1000	920.00
4	1016	Jan	c006	a01	p01	1000	500.00
5	1017	Feb	c001	a06	p03	500	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	457.50
10	1022	Mar	c001	a05	p06	450	810.00
11	1023	Mar	c001	a04	p05	500	450.00
12	1024	Mar	c006	a06	p01	880	400.00
13	1025	Apr	c001	a05	p07	888	799.20
14	1026	May	c002	a05	p03	808	711.04

OK. Unix Ln 156, Col 1, Ch 4650 21 chars 14 rows. 14 msec

2. Primary key, Candidate key, and Super key

A primary key is the attribute or column that uniquely identifies a row in a database. A table can only have one primary key. The candidate key is the attribute or column in a table that qualifies for uniqueness of each row. One table can have more than one candidate keys. A primary key is actually a candidate key that is the most suited to maintain uniqueness in a table at the row level.

If you add any other column to a candidate key then it becomes a super key. A super key is a combination of attributes that can be uniquely used to identify a database record. A table can have many super keys. Candidate keys are a special subset of super keys that do not have an extraneous information in them.

3. Data types

This table example is called “Students in Database Management”.

The fields this table has are:

- firstName
 - data type: text
 - Nullable
- lastName
 - data type: text
 - Nullable
- Cwid
 - data type: int
 - Nullable

- classYear
 - data type: smallint
 - Nullable
- classStartDate
 - data type: datetime
 - Not nullable

4. Relational rules

- a. The first normal form (1NF) rule is where data in each cell must be atomic and unstructured, meaning it cannot be broken down into more parts. An example of this demonstrating how this is important is if there is a field name called “fullName”. This field name can be broken down into “firstName” and “lastName” making the data atomic.
- b. The access rows by content only rule means whenever addressing the contents of a table, you have to ask *what* content is there, rather than *where* it is. A table is a set, and elements of a set have no intrinsic or ordinal values. This is important because if someone says, “tell me the content of row 3”, it might be something one day, and then something completely different another day, because the content can move around a lot. Rather than asking where it is, someone should ask, “tell me the content of the row where xxx is”.
- c. The all rows must be unique rule means that all the field names must be different and unique. This is important because it will make it hard to search for a specific piece of data in a field if there are more than one field with the same name.