

## Lab 2

Candidate Keys are single column in a table that all have to be the minimal set. However a primary key is a column or data type that will never be the same such as an orderID. A superkey is a data set that never changes no matter where it is and can be referenced anywhere in a database.

Data types are very important in a database, they are what separate different information and make it easier to sort things out. When creating a data table for the dining hall you would want the CWID as a data set since they swipe you in everytime you eat. The CWID field would be specified as numeric(8) and it would be not null. Another data set that you would have in that table would be the student's name, the student name would be specified as text and would be not null. These different data sets are then used by Marist Dining to charge and keep track of who comes in to eat.

- The "first normal form" rule is essentially the the record type for each value must be the same. If you have different record types for each value then you will have conflicting data.
- The "access rows by content only" rule means that you shouldn't look for where content is you should look for what the content is to access it so you are accessing it by its type instead of it's location.
- The "all rows must be unique" rule is pretty simple in that you can't have two rows with the same thing because then that would create redundancy.

## Customers

The screenshot shows a PostgreSQL SQL Editor window titled "Query - CAP2 on postgres@localhost:5432". The SQL Editor tab is active, displaying the following SQL code:

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

-- SQL statements for displaying example data into the CAP2 database
-- Connect to your Postgres server and set the active database to CAP2. Then . . .

select *
from customers;

select *
from agents;
```

The Output pane shows the results of the query "select \* from customers;". The output is a table with 6 rows and 4 columns: cid, name, city, and discount.

	cid	name	city	discount
1	c001	Tiptop	Duluth	10.00
2	c002	Basics	Dallas	12.00
3	c003	Allied	Dallas	8.00
4	c004	ACME	Duluth	8.00
5	c005	Weyland-Yutani	Acheron	0.00
6	c006	ACME	Kyoto	0.00

The status bar at the bottom indicates "OK", "Unix", "Ln 175, Col 16, Ch 5217", "24 chars", "6 rows.", and "18 msec".

## Agents

The screenshot shows a PostgreSQL SQL Editor window titled "Query - CAP2 on postgres@localhost:5432". The SQL Editor tab is active, displaying the following SQL code:

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

-- SQL statements for displaying example data into the CAP2 database
-- Connect to your Postgres server and set the active database to CAP2. Then . . .

select *
from customers;

select *
from agents;
```

The Output pane shows the results of the query "select \* from agents;". The output is a table with 7 rows and 4 columns: aid, name, city, and percent.

	aid	name	city	percent
1	a01	Smith	New York	6
2	a02	Jones	Newark	6
3	a03	Brown	Tokyo	7
4	a04	Gray	New York	6
5	a05	Otasi	Duluth	5
6	a06	Smith	Dallas	5
7	a08	Bond	London	7

The status bar at the bottom indicates "OK", "Unix", "Ln 177, Col 1, Ch 5219", "21 chars", "7 rows.", and "19 msec".

## Products

Query - CAP2 on postgres@localhost:5432 \*

SQL Editor Graphical Query Builder

Previous queries

```
select *  
from customers;  
  
select *  
from agents;  
  
select *  
from products;  
  
select *  
from orders;
```

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	folder	Dallas	123100	2.00
7	p07	case	Newark	100500	1.00
8	p08	clip	Newark	200600	1.25

OK. Unix Ln 180, Col 1, Ch 5242 23 chars 8 rows. 19 msec

## Orders

Query - CAP2 on postgres@localhost:5432 \*

SQL Editor Graphical Query Builder

Previous queries

```
select *  
from agents;  
  
select *  
from products;  
  
select *  
from orders;
```

Output pane

Data Output Explain Messages History

	ordno integer	mon character(3)	cid character(4)	aid character(3)	pid character(3)	qty integer	dollars 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. Unix Ln 183, Col 1, Ch 5267 21 chars 14 rows. 22 msec