

Trevor Pirone

9/9/17

Professor Labouseur

Lab #2

Query - CAP on postgres@localhost:5433

File Edit Query Favourites Macros View Help

SQL Editor | Graphical Query Builder

Previous queries

SELECT \*  
FROM Customers;

Scratch pad

Output pane

Data Output Explain Messages History

	cid	name	city	discountpct
	character(4)	text	text	numeric(5,2)
1	c001	Tyngrope	Duluth	0.00
2	c002	Tyrell	Dallas	12.00
3	c003	Eldon	Dallas	8.00
4	c004	ACME	Duluth	6.50
5	c005	Weyland	Rose	0.00
6	c006	ACME	Beijing	0.00

OK

Unix | Ln 2, Col 16, Ch 26

6 rows, 13 ms

Query - CAP on postgres@localhost:5433

File Edit Query Favourites Macros View Help

SQL Editor | Graphical Query Builder

Previous queries

SELECT \*  
FROM Orders;

Scratch pad

Output pane

Data Output Explain Messages History

	ordno	month	cid	aid	pid	quantity	totaland
	integer	character(3)	character(4)	character(3)	character(3)	integer	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	487.50
10	1022	Mar	c001	a05	p06	450	810.00
11	1023	Mar	c001	a04	p05	500	450.00
12	1024	Mar	c006	a06	p01	800	400.00
13	1025	Apr	c001	a05	p07	800	799.20
14	1026	May	c002	a05	p03	808	711.04

OK

Unix | Ln 2, Col 12, Ch 22

14 rows, 12 ms

Query - CAP on postgres@localhost:5433 \*

File Edit Query Favourites Macros View Help

SQL Editor | Graphical Query Builder

Previous queries

SELECT \*  
FROM Products;

Output pane

	pid	name	city	qty	pricesad
	character(3)	text	text	integer	numeric(10,2)
1	p01	Weissberg compensator	Dallas	111400	0.50
2	p02	universal translator	Bevark	203000	0.50
3	p03	Commodore PET	Duluth	150400	1.00
4	p04	ICARS module	Duluth	125300	1.00
5	p05	pencil	Dallas	221400	1.00
6	p06	trapper keeper	Dallas	123100	2.00
7	p07	fius capacitor	Bevark	100500	1.00
8	p08	HAL 9000 memory core	Bevark	200400	1.25

OK

Unix | Ln 2, Col 15, Ch 25

8 rows, 11 ms

Query - CAP on postgres@localhost:5433 \*

File Edit Query Favourites Macros View Help

SQL Editor | Graphical Query Builder

Previous queries

SELECT \*  
FROM Agents;

Output pane

	aid	name	city	commission
	character(3)	text	text	numeric(5,2)
1	a01	Smith	New York	5.60
2	a02	Jones	Bevark	6.00
3	a03	Perry	Bong Bong	7.00
4	a04	Gray	New York	6.00
5	a05	Oxani	Duluth	5.00
6	a06	Smith	Dallas	5.00
7	a08	Bond	London	7.07

OK

Unix | Ln 2, Col 12, Ch 22

7 rows, 12 ms

2. A primary key is a key that is unique to a record in the database. A primary key can be a unique identifier such as an id number and a relational database is only allowed to have one primary key. This means that there are no such things as relational databases with multiple primary keys. A candidate key is any column or columns that can be used as unique keys.

Candidate keys can identify records within a relational database without having to refer to any other type of data. Candidate keys exist in tables and there can be multiple candidate keys within a table. A superkey is used to uniquely identify the rows within the tables of a relational database.

3. A data type can be defined as the kind of value that a certain column in a table holds. There are many kinds of data types such as integer type, string type, etc. An example to visualize data types in action would be a table of books. The book table would have the following columns: Book ID, Book Name, Author, Price, and Quantity. Integers and strings can be null data types. In this situation, however, Book ID is an integer that will never be null since it will not be the primary key, but the number will be used to identify what that book is. Price and quantity will also be integers and those integer values will not be null in this situation because every item has some price even if it is 0 and some kind of quantity. Strings would represent book name and author and author could be null if the book written has no official author (i.e. The Bible).

4. The first normal form means that all fields rely on the primary key and there are no repeating groups within the table. In addition, the first normal form states that every component of every tuple is an atomic value and can only have a single value for each attribute. One real world example is an address book. If a person has more than one telephone number, for a relational database to comply with 1NF, it would be necessary to split each number into a separate entity or create two tables for customers and customer's phone numbers. The "access by content only" rule describes that a user cannot find content within a relational database through a column id or a row id. A user must type through the commands to find the content and information they are looking for. The "all rows must be unique" rule states that a table in a relational database cannot

contain multiple rows because then it is not unique. This means that the table would not be qualified as a relation.