

Stephen Bohner

Dr. Rivas

Database Management

3 February 2016

Lab #2

SQL Editor	Graphical Query Builder
Previous queries	
<pre>SELECT * FROM customers;_</pre>	

Data Output	Explain	Messages	History	
	cid character(4)	name text	city text	discount numeric(5,2)
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

SQL Editor	Graphical Query Builder
Previous queries	
<pre>SELECT * FROM agents;_</pre>	

Data Output	Explain	Messages	History	
	aid character(3)	name text	city text	percent real
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

SQL Editor	Graphical Query Builder
Previous queries	
<pre>SELECT * FROM products;_</pre>	

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

SQL Editor	Graphical Query Builder
Previous queries	
<pre>SELECT * FROM orders;</pre>	

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

There are differences between the terms, primary key, candidate key, and superkey. A primary key is a data type (column) in a database that is used as a unique identifier. An example of this would be something such as a student ID here at Marist. The ID is a unique identifier of every student that attends Marist. No other Marist student can have the same ID as another. A candidate key is similar to a primary key in the sense that it creates uniqueness. A candidate key is uses the minimum set of columns to create a unique row. An example of this would be a set of two columns such as a name and a phone number, because two people in a database might have the same name, but will not have the same phone number as well. Lastly we have a super key. A super key is like a candidate key because it creates a unique row. The only difference is that a super key is any set of columns that creates a unique row. This could vary from two to five or more columns. It does not matter how many columns it takes to create a super key, they are usually abundant.

There are various different data types in SQL. A few examples of data types are VARCHAR, INT, FLOAT, and BOOLEAN. These data types are relatively self explanatory. A data type is set for the type of data you will be storing in your tables. A topic I might use to create a table in SQL would be a user table. This table would have the columns: uid, username, password and email. The data types for these columns would be: INT PRIMARY KEY, VARCHAR, VARCHAR, and VARCHAR respectively. This table could be as an example to insert user accounts that have been created. All of the fields in this example theoretically could be nullable, except for the primary key. Although the data types are in fact nullable, you should declare them as not null so the user must enter in their necessary information.

There are several reasons why rules in a relational database are very important. These three rules are “first normal form” rule, the “access rows by content only” rule, and the “all rows must be unique” rule. The first rule of a relational database is that tables must be in “normal form”. This means that when creating tables you must have effective rows and columns that reduce redundancy, while allowing them to relate to other tables. This rule is important because if you have redundancy in your tables, relating them will not be easy. The second important rule of relational databases is access rows by content only. This means you should not ask where all of the information exactly is located, rather you should ask what data is stored to find that information. This helps when you are trying to locate information in an efficient, but effective way using the data you currently have. The last rule that is absolutely essential when dealing with a relational database is that all rows must be unique. This was mentioned before because it is so important, each row in a table must have a element in the row that can distinguish it from the others. This is necessary because in order if a query were to execute, it would return more than one row which would result in errors and confusion. All three of these rules are required when you are creating a relational database. If any one of these rules are not met, you will not create a successful and efficient relational database.