

Raymond Mattingly

Professor Alan Labouseur

Database Management

11 September 2016

The screenshot shows the pgAdmin III interface. The SQL Editor contains the query: `select * from customers;`. The Output pane displays the results of this query in a table format. The table has four columns: `cid` (character(4)), `name` (text), `city` (text), and `discount` (numeric(5,2)). There are 6 rows of data.

	cid	name	city	discount
1	c001	Tiptop	Duluth	10.00
2	c002	Tyrell	Dallas	12.00
3	c003	Allied	Dallas	8.00
4	c004	ACME	Duluth	8.80
5	c005	Weyland	Acheron	0.00
6	c006	ACME	Ryoto	0.00

The screenshot shows the pgAdmin III interface. The SQL Editor contains the query: `select * from agents;`. The Output pane displays the results of this query in a table format. The table has five columns: `aid` (character(3)), `name` (text), `city` (text), `commission` (numeric(5,2)), and an unlabeled column (text). There are 7 rows of data.

	aid	name	city	commission	
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	O'neal	Duluth	5.00	
6	a06	Smith	Dallas	5.00	
7	a08	Bond	London	7.07	

Raymond Mattingly

Professor Alan Labouseur

Database Management

11 September 2016

The screenshot shows the pgAdmin III interface with a query window titled "Query - postgres on postgres@localhost:5432". The SQL Editor contains the query: `select * from products;`. The Output pane displays the results in a table with 8 rows and 6 columns: `pid`, `name`, `city`, `quantity`, `priceused`, and `numeric(10,2)`. The status bar at the bottom indicates "8 rows, 56 msec".

	pid	name	city	quantity	priceused
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	eraser	Newark	200600	1.25

The screenshot shows the pgAdmin III interface with a query window titled "Query - postgres on postgres@localhost:5432". The SQL Editor contains the query: `select * from orders;`. The Output pane displays the results in a table with 13 rows and 8 columns: `ordnum`, `mon`, `cid`, `aid`, `pid`, `qty`, and `totalused`. The status bar at the bottom indicates "14 rows, 69 msec".

	ordnum	mon	cid	aid	pid	qty	totalused
1	1011	jan	e001	a01	p01	1000	450.00
2	1013	jan	e002	a03	p03	1000	880.00
3	1015	jan	e003	a03	p05	1200	1104.00
4	1016	jan	e006	a01	p01	1000	500.00
5	1017	feb	e001	a06	p03	600	540.00
6	1018	feb	e001	a03	p04	600	540.00
7	1019	feb	e001	a02	p02	400	180.00
8	1020	feb	e006	a03	p07	600	600.00
9	1021	feb	e004	a06	p01	1000	460.00
10	1022	mar	e001	a05	p06	400	720.00
11	1023	mar	e001	a04	p05	500	450.00
12	1024	mar	e006	a06	p01	800	400.00
13	1025	apr	e001	a05	p07	800	720.00

Raymond Mattingly

Professor Alan Labouseur

Database Management

11 September 2016

2. Your primary key identifies the column that will maintain uniqueness throughout one's table; in other words, it refers to the ID column. A candidate key refers to a column in a table wherein all rows would be unique. A superkey is an additional column added to one's primary key, so if I wanted to return ordnum and mon then mon would be my superkey.

3. Data types are the various ways to store data entries so as to most effectively use said data. A simple login system would require a data table with multiple data types. One would want the columns ID, username, password, and email. The data type for ID would work best as integers (int), whereas one would want to use a string data type (text or varchar) for the username, password, and email entries. In a login system all fields would need to be non-nullable because the system requiring login wouldn't want users with missing information.

4. The first relational rule specifies that all cells should be atomic. In other words, you don't want to be storing multiple data types or multiple entries within one cell. This rule is why Professor Labouseur doesn't like array data types in relational databases. The second relational rule says that one must access fields by content rather than location within the table because location isn't necessarily fixed and one would like to preserve the longevity of a query by instead searching content, which should be much more constant. The third relational rule is that all rows must be unique; this means that each row must have at least one unique element within it. That's one reason why a good

Raymond Mattingly

Professor Alan Labouseur

Database Management

11 September 2016

database will have a primary key within it to preserve uniqueness throughout the table regardless of what data is entered.