Ryan Rendeiro

Pablo Rivas

Database Management

2/3/2016

Data Types

A new company is started, how will the company keep track of its employees, by creating a table in its database for tacking current employees. First the table will need fields that describe what is needed to keep accurate records of employees. The employees table will have fields for employee ID, employee name, employee position, and employee salary. Employee ID would need to be either the numeric or the varchar datatype given how the company wants to specify Employee IDs and those entries in to the database would be not null because the employee ID would be both the table primary key and to differentiate between two employees of the same name and position. Employee name would be the char datatype and this to would be not null because knowing the employee's name is important, because if the boss or other employees don't know each other's names it will be very awkward and demoralizing by addressing everyone one by their employee ID. Employee position would be the char datatype because no numbers are needed to describe an employee's position in the company and this to would be not null because if an employee doesn't have a position in the company then they don't fill the requirements to be an employee at the company because you need to be employed to be an employee. Employee salary would be the money datatype because you don't pay anyone anything besides money for doing their job, and this to would be not null because if an employee isn't being paid they will either leave or sue the company for not paying them.

Ryan Rendeiro

Pablo Rivas

Database Management

2/3/2016

The differences between primary key, candidate key, and super key are that first the super keys are a set of fields in a table such that you can make the table have unique rows. Candidate keys are the minimal set of super keys, and primary key or keys are the field or fields of a table that are used as a unique identifier for the table.

The first normal form rule is that all data must be atomic which means they cannot be broken down in to any further piece of data, the data could be an integer or a string but not an array or a set. The access row by content only rule is where database users and programmers only supposed to ask for what the data is and not where the data is supposed to be in the database. Asking for all entries about cars with spoilers is okay but, asking for all the entries on the C: drive of the database server is not okay. The all rows must be unique rule is that no row can be a duplicate of another row because duplicates while violating this rule also violate the previous rule because after you get the results for the original entry and its duplicate you then have to ask for either one or the other and then you are asking for location which isn't allowed.

```
select *
from customers;

select *
from agents;

select *
from products;

select *
from orders;
```

Output pane

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

```
-- Connect to your Postgres server and set the active data

select *
from customers;

select *
from agents;

select *
from products;

select *
from orders;
```

Output pane

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

```
select *
from customers;

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

```
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