

Evan Mastriano

The CAP4 Database

Customers

cid	name	city	discount
c001	Tiptop	Duluth	10.00
c002	Tyrell	Dallas	12.00
c003	Allied	Dallas	8.00
c004	ACME	Duluth	8.50
c005	Weyland	Risa	0.00
c006	ACME	Kyoto	0.00

Orders

aid	name	city	commission
a01	Smith	New York	6.50
a02	Jones	Newark	6.00
a03	Perry	Tokyo	7.00
a04	Gray	New York	6.00
a05	Otasi	Duluth	5.00
a06	Smith	Dallas	5.00
a08	Bond	London	7.07

pid	name	city	quantity	priceUSD
-----	------	------	----------	----------

Products

pid	name	city	quantity	priceUSD
p01	comb	Dallas	111,400	0.50
p02	brush	Newark	203,000	0.50
p03	razor	Duluth	150,600	1.00
p04	pen	Duluth	125,300	1.00
p05	pencil	Dallas	221,400	1.00
p06	trapper	Dallas	123,100	2.00
p07	case	Newark	100,500	1.00
p08	eraser	Newark	200,600	1.25

The screenshot shows the PostgreSQL GUI interface. On the left, a tree view displays the database structure, including 'postgres' and 'Tablespaces'. The main window shows a SQL query: `select * from Customers;`. Below the query editor, the 'Data Output' tab is active, displaying a table with 5 columns: 'cid', 'name', 'city', 'discount', and 'numeric ...'. The results show 6 rows of customer data.

cid	name	city	discount	numeric ...
c001	Tiptop	Duluth	10	
c002	Tyrell	Dallas	12	
c003	Allied	Dallas	8	
c004	ACME	Duluth	8.5	
c005	Weyland	Risa	0	
c006	ACME	Kyoto	0	

The screenshot shows the PostgreSQL GUI interface. The SQL query is: `select * from Agents;`. The 'Data Output' tab displays a table with 5 columns: 'aid', 'name', 'city', 'commisi...', and 'numeric ...'. The results show 8 rows of agent data.

aid	name	city	commisi...	numeric ...
a01	Smith	New York	6.5	
a02	Jones	Newark	6	
a03	Perry	Tokyo	7	
a04	Grey	New York	6	
a05	Otasi	Duluth	5	
a06	Smith	Dallas	5	
a08	Bond	London	7.07	

The screenshot shows the PostgreSQL GUI interface. The SQL query is: `select * from Products;`. The 'Data Output' tab displays a table with 6 columns: 'pid', 'name', 'city', 'quantity', 'priceusd', and 'numeric ...'. The results show 8 rows of product data.

pid	name	city	quantity	priceusd	numeric ...
p01	comb	Dallas	111400	0.5	
p02	brush	Newark	203000	0.5	
p03	razor	Duluth	150600	1	
p04	pen	Duluth	125300	1	
p05	pencil	Dallas	221400	1	
p06	trapper	Dallas	123100	2	
p07	case	Newark	100500	1	
p08	eraser	Newark	200600	1.25	

Orders

ordNumber	month	cid	aid	pid	qty	totalUSD
1011	Jan	c001	a01	p01	1000	450.00
1012	Jan	c002	a03	p03	1000	880.00
1015	Jan	c003	a03	p05	1200	1104.00
1016	Jan	c006	a01	p01	1000	500.00
1017	Feb	c001	a06	p03	600	540.00
1018	Feb	c001	a03	p04	600	540.00
1019	Feb	c001	a02	p02	400	180.00
1020	Feb	c006	a03	p07	600	600.00
1021	Feb	c004	a06	p01	1000	460.00
1022	Mar	c001	a05	p06	400	720.00
1023	Mar	c001	a04	p05	500	450.00
1024	Mar	c006	a06	p01	800	400.00
1025	Apr	c001	a05	p07	800	720.00
1026	May	c002	a05	p03	800	744.00

Originally from Database Principles, Prog

```

138 select *
139 from Orders;

```

Data Output								Explain	Messages	History	
<input type="checkbox"/>	ordnumb...	month	cid	aid	pid	qty	totalusd				
	integer	character	character	character	character	integer	numeric				
<input type="checkbox"/>	1011	Jan	c001	a01	p01	1000	450				
<input type="checkbox"/>	1012	Jan	c002	a03	p03	1000	880				
<input type="checkbox"/>	1015	Jan	c003	a03	p05	1200	1104				
<input type="checkbox"/>	1016	Jan	c006	a01	p01	1000	500				
<input type="checkbox"/>	1017	Feb	c001	a06	p03	600	540				
<input type="checkbox"/>	1018	Feb	c001	a03	p04	600	540				
<input type="checkbox"/>	1019	Feb	c001	a02	p02	400	180				
<input type="checkbox"/>	1020	Feb	c006	a03	p07	600	600				
<input type="checkbox"/>	1021	Feb	c004	a06	p01	1000	460				
<input type="checkbox"/>	1022	Mar	c001	a05	p06	400	720				
<input type="checkbox"/>	1023	Mar	c001	a04	p05	500	450				
<input type="checkbox"/>	1024	Mar	c006	a06	p01	800	400				
<input type="checkbox"/>	1025	Apr	c001	a05	p07	800	720				
<input type="checkbox"/>	1026	May	c002	a05	p03	800	744				

Part 2: Key Distinction

A super key is the column or collection of columns that ensure every row will be unique. A candidate key uniquely identifies every row in the fewest number of columns, so it is sometimes referred to as a minimal super key. A primary key is the chosen candidate key. This key is the most important, and establishes the relationships between tables. So, in short, the super key sets each row apart in any way. A candidate key is the shortest way to create a super key. And the primary key is one row the sets the table apart and links the table to others as a foreign key elsewhere.

Part 3: Data Types

Data types are extremely important. Data types include strings, integers, floating points, booleans, etc. Keeping columns locked to one data type attempts to keep data appropriate and uniform throughout the table. For example, if I wanted to create a table called Wrestlers. I could have columns such as FirstName, LastName, Gender, and NumTitlesWon. FirstName would be text and not null. LastName would also be text, but could be null (such as with Sting, he has no second name. Gender could be an int, 0 for male, 1 for female and cannot be null. NumTitlesWon would be an int. It could be null, or it could range from 0-100. Null would only be allowed when a field isn't required, such as having multiple names or owning certain things.

Part 4: Relational "Rules" of Nature

The first relational rule, or 1NF, states that at the intersection of a row and column, you cannot have anything that has it's own structure. This means it must be atomic. This prevents any confusion in table creation and relations between tables, or complicate data conversion. This also helps with data duplication, preventing something from being stored multiple times. For example, storing "Shawn Michaels" under name is difficult, and should be split into a first name and a last name column.

The second rule explains how you can only search for a row by what's in it. This is because on different systems or with different users, the same row might be in a different location on that table. This requires the use of SQL commands such as "select * from Wrestlers where name = 'Bayley'" to find certain rows.

The third rule states that for each table, all rows must be unique. This arises from set theory, and prevents confusion when searching for rows. Because of the second rule, we can only search by content. So if two rows have the same exact content, there would be literally no way to distinguish the two or find one row.