

Eric Schmidt

1)

Data Output Explain Messages History					
<input type="checkbox"/>	cid character	name text	city text	discount numeric ...	
<input type="checkbox"/>	c001	Tiptop	Duluth	10	
<input type="checkbox"/>	c002	Tyrell	Dallas	12	
<input type="checkbox"/>	c003	Allied	Dallas	8	
<input type="checkbox"/>	c004	ACME	Duluth	8.5	
<input type="checkbox"/>	c005	Weyland	Risa	0	
<input type="checkbox"/>	c006	ACME	Kyoto	0	

Data Output Explain Messages History					
<input type="checkbox"/>	aid character	name text	city text	commissi... numeric ...	
<input type="checkbox"/>	a01	Smith	New York	6.5	
<input type="checkbox"/>	a02	Jones	Newark	6	
<input type="checkbox"/>	a03	Perry	Tokyo	7	
<input type="checkbox"/>	a04	Grey	New York	6	
<input type="checkbox"/>	a05	Otasi	Duluth	5	
<input type="checkbox"/>	a06	Smith	Dallas	5	
<input type="checkbox"/>	a08	Bond	London	7.07	

Data Output Explain Messages History						
<input type="checkbox"/>	pid character	name text	city text	quantity integer	priceusd numeric ...	
<input type="checkbox"/>	p01	comb	Dallas	111400	0.5	
<input type="checkbox"/>	p02	brush	Newark	203000	0.5	
<input type="checkbox"/>	p03	razor	Duluth	150600	1	
<input type="checkbox"/>	p04	pen	Duluth	125300	1	
<input type="checkbox"/>	p05	pencil	Dallas	221400	1	
<input type="checkbox"/>	p06	trapper	Dallas	123100	2	
<input type="checkbox"/>	p07	case	Newark	100500	1	
<input type="checkbox"/>	p08	eraser	Newark	200600	1.25	

Data Output Explain Messages History							
<input type="checkbox"/>	ordnumb... integer	month character	cid character	aid character	pid character	qty integer	totalusd 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

2) The best way to say the difference between the three key types is to explain the difference between sets of two.

-The difference between primary keys and super keys is that primary keys are chosen candidate keys that are not a collection of columns, like super keys.

- The difference between super keys and candidate keys is super keys are unique identifiers that have been chosen as a group of columns where candidate keys are just potential super keys that have not yet been chosen.

-The difference between candidate keys and primary keys is the primary key is a chosen candidate key. The candidate key is the first step in choosing which would be a primary key.

3) In eSports there are tons of reasons as why you would create a table. One would be teams in a major tournament. In this potential table, you would have table called Teams. In this table you may have a column that lists names of all the teams which would a string data type, a column for win rate percentages type from past tournaments, which would most likely be a int data, players1 – 5 (separate columns for each player), which would also be a string data type. They also may have extra player columns, on string data type, for benched players encase any player is sick or injured. For nullable columns, a very likely one to be null would be extra players, as every team doesn't have the resources or time to fill extra player slots. Team name would be null until a team has finally reached the tournament and filled in their team name in the registration.

4)The "first normal form" rule example:

TABLE_PRODUCT

Product ID	Color	Price
1	red, green	15.99
2	yellow	23.99
3	green	17.50
4	yellow, blue	9.99
5	red	29.99

In this example, the color table is not conforming to the first normal form rule because it has multiple sets of data on one column. To fix this issue you could make separate tables for colors.

This would be a better fix:

TABLE_PRODUCT_PRICE

Product ID	Price
1	15.99
2	23.99
3	17.50
4	9.99
5	29.99

TABLE_PRODUCT_COLOR

Product ID	Color
1	red
1	green
2	yellow
3	green
4	yellow
4	blue
5	red

The “access rows by content only” rule Example:

Electric Toothbrush Models

<u>Manufacturer</u>	<u>Model</u>	Model Full Name
Forte	X-Prime	Forte X-Prime
Forte	Ultraclean	Forte Ultraclean
Dent-o-Fresh	EZbrush	Dent-o-Fresh EZbrush
Kobayashi	ST-60	Kobayashi ST-60
Hoch	Toothmaster	Hoch Toothmaster
Hoch	X-Prime	Hoch X-Prime

If this was your original table and you ask to find column 1 and 2 exclusively.

Electric Toothbrush Manufacturers

<u>Manufacturer</u>	Manufacturer Country
Forte	Italy
Dent-o-Fresh	USA
Kobayashi	Japan
Hoch	Germany

If that same command to ask for only the first and second column, the table has now changed names and will now give you a different result for the same command.

All Rows must be Unique Rule:

If two columns had identical names it would cause tons of issues within your database. If there is two columns have the column name “name” it would bring up both columns if you ask to show the table for all contents in the “name” column even if you only meant for one of the columns to show.