

2. Explain the distinctions among the terms primary key, candidate key, and superkey.

SUPER KEY (any set of columns that uniquely identify a record)

CANDIDATE KEY (a minimal SUPER KEY - a set with smallest number of columns)

PRIMARY KEY (may not have NULL value component)

A candidate key that is selected / designated as primary

Must be unique!

May be declared as a column / field attribute or in the schema item list

FOREIGN KEY (must match another table's PRIMARY KEY)

Component to ensure "Referential Integrity"

COMPOSITE KEY - [xxx composed with yyy]

3. Write a short essay on data types. Select a topic for which you might create a table. Name the table and list its fields (columns). For each field, give its data type and whether or not it is nullable.

Data types include text, integer, date/time, Boolean, and auto number; and variations upon the basic types such as long, short, byte, single, and double. Often there is more than one particular data type that might be used in a particular instance. For example, world-class database systems such as Lotus Notes might offer zipcode as an alternative to using an integer data type - I don't know for sure, I'm not worthy to check. I return to the Red Hook Alumni Association database for my topic table. Here is the main table:

Field Name	Data Type	Nullable?
Record_Number	AutoNumber	No
Last_Update	Date/Time	No
Updated_By	Text	No
Affiliation	Text	No
Grad_Year	Integer	Yes
Mailing_Code	Text	Yes
Address_Status	Yes/No	No
Deceased	Yes/No	No
Honored_Year	Integer	Yes
Prefix_Now	Text	Yes
First_Name	Text	No
Middle_Initial	Text	Yes
Last_Name_Grad	Text	No
Suffix_Grad	Text	Yes
Last_Name_Now	Text	No
Suffix_Now	Text	Yes
Address_1	Text	Yes
Address_2	Text	Yes
City	Text	Yes
State	Text	Yes
Zip	Text	Yes
Postal_Code	Text	Yes
Country	Text	Yes
Phone	Text	Yes
Email	Text	Yes
Payment_Date	Date/Time	Yes
Payment_Amount	Currency	Yes

4. Explain the following relational "rules" with examples and reasons why they are important.

The normal form rules build upon each other. Our text states that each normal form rule incorporates or relies upon the prior rules. That much I understood. The remainder of the book's discussion of normal form rules made my head hurt.

a. The "first normal form" rule

FIRST: You do not talk about the first normal form rule. You may write about the first normal rule, but you must use comic sans. You shall write that no multi-value attributes or data types are permitted for any columns or field.

b. The "access rows by content only" rule

SECOND: Select by "What" not by "Where." Sets have no ordinal value. I.e., data is stored and queried by type / schema / attribute, not by physical location.

c. The "all rows must be unique" rule

THIRD: All rows must be unique. If the proposed schema for the data tends to produce rows or sets that are not unique, it is likely that additional relationships / tables are indicated. In the sad event that the basic schema is sound, yet multiple rows might contain the same data set, it will be necessary to introduce an artificial primary key to accommodate this rule.

7. Ignoring any of the normal form rules is CARDINAL SIN.

The screenshot shows the PostgreSQL SQL Editor window titled "Query - postgres on postgres@localhost:5432 *". The interface includes a menu bar (File, Edit, Query, Favurites, Macros, View, Help) and a toolbar with various icons. The main SQL Editor pane contains the following SQL statements:

```
-- SQL statements for displaying example data into the
-- Connect to your Postgres server and set the active d

select *
from customers;

select *
from agents;

select *
from products;

select *
from orders;
```

To the right of the SQL Editor is a "Scratch pad" pane. Below the SQL Editor is the "Output pane", which has tabs for "Data Output", "Explain", "Messages", and "History". The "Data Output" tab is active, displaying a table with 6 rows and 5 columns:

	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

At the bottom of the window, a status bar shows "OK.", "Unix", "Ln 188, Col 1, Ch 5810", "24 chars", "6 rows.", and "31 ms".

The screenshot shows the PostgreSQL Query Editor window titled "Query - postgres on postgres@localhost:5432 *". The window has a menu bar (File, Edit, Query, Favurites, Macros, View, Help) and a toolbar. The main area is the "SQL Editor" with a "Graphical Query Builder" tab. It contains the following SQL queries:

```
-- SQL statements for displaying example data into the  
-- Connect to your Postgres server and set the active d  
  
select *  
from customers;  
  
select *  
from agents;  
  
select *  
from products;  
  
select *  
from orders;
```

The "Output pane" at the bottom shows the results of the last query in a table with 7 rows. The table has columns: aid, name, city, and percent. The data is as follows:

	aid	name	city	percent
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

The status bar at the bottom indicates "OK.", "Unix", "Ln 192, Col 13, Ch 5857", "21 chars", "7 rows.", and "31 ms".

The screenshot shows the PostgreSQL Query Editor window titled "Query - postgres on postgres@localhost:5432 *". The window has a menu bar (File, Edit, Query, Favurites, Macros, View, Help) and a toolbar. The main area is the "SQL Editor" with a "Graphical Query Builder" tab. It contains the following SQL queries:

```
-- SQL statements for displaying example data into the
-- Connect to your Postgres server and set the active d

select *
from customers;

select *
from agents;

select *
from products;

select *
from orders;
```

The "Output pane" at the bottom shows the "Data Output" tab with a table of results. The table has 6 columns: pid, name, city, quantity, priceusd, and priceusd. The data is as follows:

	pid	name	city	quantity	priceusd	priceusd
	character(3)	text	text	integer	numeric(10,2)	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	

The status bar at the bottom shows "OK.", "Unix", "Ln 195, Col 15, Ch 5882", "23 chars", "8 rows.", and "31 ms".

The screenshot shows the PostgreSQL Query Editor window titled "Query - postgres on postgres@localhost:5432 *". The window has a menu bar (File, Edit, Query, Favurites, Macros, View, Help) and a toolbar. The main area is the "SQL Editor" tab, which contains the following SQL queries:

```
-- SQL statements for displaying example data into the
-- Connect to your Postgres server and set the active d

select *
from customers;

select *
from agents;

select *
from products;

select *
from orders;
```

The "Output pane" at the bottom shows the "Data Output" tab with a table of results. The table has 8 columns: ordno, mon, cid, aid, pid, qty, and dollars. The data is as follows:

	ordno	mon	cid	aid	pid	qty	dollars
	integer	character(3)	character(4)	character(3)	character(3)	integer	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

The status bar at the bottom shows "OK.", "Unix", "Ln 198, Col 13, Ch 5905", "21 chars", "14 rows.", and "31 ms".