

Gary Coltrane

Lab 2

1/30/17

1.

```
1 -- Database: CAP4
2 SELECT * FROM ORDERS;
```

<div>MessagesData OutputExplainHistory</div>								
<input type="checkbox"/>	ordnumber 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	

```
1 -- Database: CAP4|
2 SELECT * FROM PRODUCTS;
```

Data Output History Explain Messages

<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	

1	-- Database: CAP4
2	SELECT * FROM Customers;

Data Output				
	cid	name	city	discount
	character	text	text	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

2.

Keys are very important in relational databases, since they represent a unique row inside of a table. Looking at super keys, candidate keys, primary keys, and foreign keys, one can create a clear distinction of each key. Super keys represent a column or a set of columns that will uniquely display a row. Any large amount of keys are considered to be a super key. Candidate keys are basically superkeys to the minimal extent, since it displays a unique row while using the a small amount of keys. The most important key are primary keys, since they simply use one single key that uniquely identifies each record in a table. Last but not least; a foreign key is a key that basically serves as a primary key in another table.

3.

When constructing a database, one must explicitly establish a data type for each specific column. The reason for this is to allow any user to know what each column represents and what type of data should be there. Without data types, it would be highly difficult to identify what belongs in a column, and it also leaves room for inconsistent data types to fill fields. In a sample database called Library, there are various data types representing many columns. In this database there is a table called Books, which contains its own columns and data types to be examined.

In the Books table there exists several columns named: bld, title, author, publisher, ISBN, and publish date. For bld, the data type would be integer since an ID can only be a number, and it would be not null. In addition, bld would serve as the primary key for the table. Title, author, and publisher would be text and not null also since every book would need to have a title, author, and publisher in order to be considered as a valid book. ISBN would be an integer and not nullable either, since this is considered to be a key like bld and would have its own unique values. Lastly, publish date would be considered to be a text data type and it would be nullable.

Ultimately, it is required to specify a data type for a column, since it allows the database creator and any user to identify what type of data would be entered for each field. It is required in every SQL server to specify a data type for each column. Thus, in order to build efficient effective database systems one must know each data type and when to use it.

4.

- a. The “first normal form” rule expresses that a table should have properly defined columns, a primary key, and non duplicated data. The reason for this is to ensure that everything is placed in its proper fields, and each field is unique. Imagine a table called Persons that have several rows with the same primary key, however with different first name and last names. This violates the first normal form because it doesn’t express a primary key that uniquely defines a specific field.
- b. The “access rows by content only” rule represents that it is highly preferred to access a row from its column name, not by its column number. A reason why this is important because accessing a row from a column number doesn’t exactly identify what the column represents, while stating its name exactly shows what is represented and expected. An example can be seen in the table called Library, and stating that a user wants to select all values from column 2 would violate this rule.
- c. “All rows must be unique” rule expresses that each row in a table must have unique fields in each column. This rule is expressed because it would be completely redundant and unnecessary to have multiple rows containing the same column information. The more unique and less redundant proves more efficiency in a database. An example of this in the Books table would be multiple rows containing the same book information.