

lab2-CAPdatabase.p... Microsoft Word - T... www.lab...
pgAdmin 4
Servers (1)
PostgreSQL 9.6
Databases (2)
CAP3
Catalogs
Event Triggers
Extensions
Foreign Data Wrappers
Languages
Schemas (1)
public
Collations
Domains
FTS Configurations
FTS Dictionaries
FTS Parsers
FTS Templates
Foreign Tables
Functions
Materialized Views
Sequences
Tables (4)
agents
customers
orders
products
Trigger Functions
Types
Views
postgres
Login/Group Roles
Tablespaces

Dashboard Properties SQL Statistics Dependencies Dependents Query: untitled

CAP3 on postgres@PostgreSQL 9.6

```
1 select *
2 from customers;
```

Data Output Explain Messages History

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

Ask me anything 12:08 PM 9/8/2016

lab2-CAPdatabase.p... Microsoft Word - T... www.lab...
pgAdmin 4
Servers (1)
PostgreSQL 9.6
Databases (2)
CAP3
Catalogs
Event Triggers
Extensions
Foreign Data Wrappers
Languages
Schemas (1)
public
Collations
Domains
FTS Configurations
FTS Dictionaries
FTS Parsers
FTS Templates
Foreign Tables
Functions
Materialized Views
Sequences
Tables (4)
agents
customers
orders
products
Trigger Functions
Types
Views
postgres
Login/Group Roles
Tablespaces

Dashboard Properties SQL Statistics Dependencies Dependents Query: untitled

CAP3 on postgres@PostgreSQL 9.6

```
1 select *
2 from agents;
```

Data Output Explain Messages History

aid	name	city	commissi...	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	

Fn Lock: On Ask me anything 12:08 PM 9/8/2016

lab2-CAPdatabase.p... Microsoft Word - T... www.labouseur.com... pgAdmin 4

Fn Lock: On

- Lab 2

Goals

Instruct

pgAdmin 4

Database: CAP3 on postgres@PostgreSQL 9.6

```
1 select *
2 from orders;
```

oid	name	pid	qty	totalcost
1011	jan	c001	a01	1000
1013	jan	c002	a03	1000
1015	jan	c003	a03	1200
1016	jan	c006	a01	1000
1017	feb	c001	a06	600
1018	feb	c001	a03	600
1019	feb	c001	a02	400
1020	feb	c006	a03	600
1021	feb	c004	a06	1000
1022	mar	c001	a05	400
1023	mar	c001	a04	500
1024	mar	c006	a06	800
1025	apr	c001	a05	800
1026	may	c002	a05	800

Data query runtime: 494 msec
14 rows retrieved

1208 PM 9/8/2016

lab2-CAPdatabase.p... Microsoft Word - T... www.labouseur.com... pgAdmin 4

Fn Lock: On

- Lab 2

Goals

Instruct

pgAdmin 4

Database: CAP3 on postgres@PostgreSQL 9.6

```
1 select *
2 from products;
```

pid	name	qty	quantity	pricecost
p01	comb	Dallas	111400	0.5
p02	brush	Newark	203000	0.5
p03	razor	Duluth	159600	1
p04	pen	Duluth	125300	1
p05	pencil	Dallas	221400	1
p06	folder	Dallas	123100	2
p07	case	Newark	100500	1
p08	eraser	Newark	200600	1.25

Data query runtime: 426 msec
8 rows retrieved

1209 PM 9/8/2016

2. A candidate key refers to a column or a set of columns in an entity. For example, a candidate key would be studentDOB or firstName and lastName. A primary key is a singular unique candidate key that is the main key for an entity, such as studentID. A superkey is a set of columns that identify a specific set. For example, a superkey would identify the name of someone by using the columns with the ID firstName and lastName.
3. A data type is an element that is organized within a table. Data types can be names, ages, phone numbers, addresses, ID's, or position. Each data type must be unique and atomic. Data types describe a specific subject and must relate to each other. For example, we need data to keep track of the people who have annual tickets to a theme park. The data types for this would include first and last name, DOB, address, email address, number of tickets purchased, total purchase and pass type. The table would be called "Theme Park Ticket Holders". None of the values would be nullable since the people in the data would only contain people who have contained passes into the theme park.

firstName	lastName	DOB	Address	emailAddress	ticketsPurchased	passType	totalPurchase
String	String	Date	String	String	Integer	String	Real
String	String	Date	String	String	Integer	String	Real
String	String	Date	String	String	Integer	String	Real

4. Relational rule 1 declares that all the fields must be atomic. For example, in a field every value must be singular such as in the age field there can only be one number. A field is the intersection of a row and a column and has one value stored into it. Relational rule 2 states that the data types must be chosen by what they are not where. The data table cannot be chosen using a pointer but with the combination of the name of the table, primary key and attribute name. For example, to select the date of birth of a student first the table name must be chosen, such as student contact info. Then the primary key, which is either the last name or first name of the student. Lastly the attribute name must be chosen, in this case it would have to be the date of birth column. The last relational rule is that all rows must be unique. Multiple rows cannot have the same information for a person. If one row has the same information for the same person, then the data would be distorted. It is possible for someone to have similar data values but it should not be possible for them to have all the same data.