

LAB 2

1. AGENT QUERY

The screenshot shows a PostgreSQL query editor interface. The top toolbar includes buttons for SQL, Dashboard, Properties, Statistics, Dependencies, and Dependents. The query editor displays the following SQL query:

```
1 select * -- cid, name, city, discount
2 from agents
3
```

Below the query editor, the 'Data Output' tab is active, showing the results of the query. The results are displayed in a table with the following columns: aid, character, name, city, and commissi... (commission). The table contains 8 rows of data:

aid	character	name	city	commissi...
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	

CUSTOMER QUERY

SQLDashboardPropertiesStatisticsDependenciesDependentsQuery-untitledQuery-untitled

No limit

CAP on postgres@PostgreSQL 9.6

```
1 select * -- cid, name, city, discount
2 from customers
3
```

Data OutputExplainMessagesHistory

	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

ORDER QUERY

SQL
Dashboard
Properties
Statistics
Dependencies
Dependents
Query-untitled
Query-untitled

CAP on postgres@PostgreSQL 9.6

```

1 select * -- cid, name, city, discount
2 from orders
3
4

```

Data Output
[Explain](#)
[Messages](#)
[History](#)

	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	

PRODUCT QUERY

The screenshot shows a PostgreSQL query editor interface. The top menu bar includes options like SQL, Dashboard, Properties, Statistics, Dependencies, and Query-untitled. Below the menu is a toolbar with icons for file operations and query execution. The main editor area contains a SQL query:

```
1 select * -- cid, name, city, discount
2 from products
3
4
```

Below the query editor, the 'Output' tab is active, displaying the results of the query in a table format. The table has five columns: pid, name, city, quantity, and priceusd. The results are as follows:

pid	name	city	quantity	priceusd
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

2.KEYS

Primary key: A Primary Key is the main key in locating a specific row or record. It can contain one or more attributes. It is unique for every row and therefore optimal for sorting data. For example in a table of workers with a column "EMPLOYEEID" would make a good primary key as it makes the data easily accessible.

Candidate key: A Candidate Key is very similar to a primary key in all aspects except that it is simply not the primary key. It meets all the same requirements and could be a primary key if it was needed. It is also unique to every row

Superkey: A Superkey is a Primary Key that has more than one column or attribute assigned to it. I almost imagine it as two primary keys combined into one.

3.SHORT ESSAY: DATATYPES

If I were to create a database right now I think it is within capability to create a lost and found database. In this database people would be able to input many different types of data and information. They could enter many things: The name of the item, a description, where it was found OR where it was lost. The date that it was lost or found. The contact number for the owner if the item is found. All of these possibilities would need to be the appropriate data type. There would be VARCHAR, INTEGER(P), BOOLEAN, and ARRAY.

4.RELATIONAL RULES

"First normal form" : This is the rule that is usually a standard. It is the rule that if make a query to a specific row and column in a database that there will only be one return value. For example in a table containing people's contact information and a query was made for someone's address, every person has only 1 address assigned to them

"Access rows by content only": Applying this rule to a database prohibits any order to the rows or columns in a database table that are not content related. For example, in a table for a database of lost and found items there would not be a column sorting the objects by "ObjectID" or anything like that.

"All rows must be unique": As this rule is aptly named, this rule disallows any 2 rows to be identical in a table.