

Kyle Wright

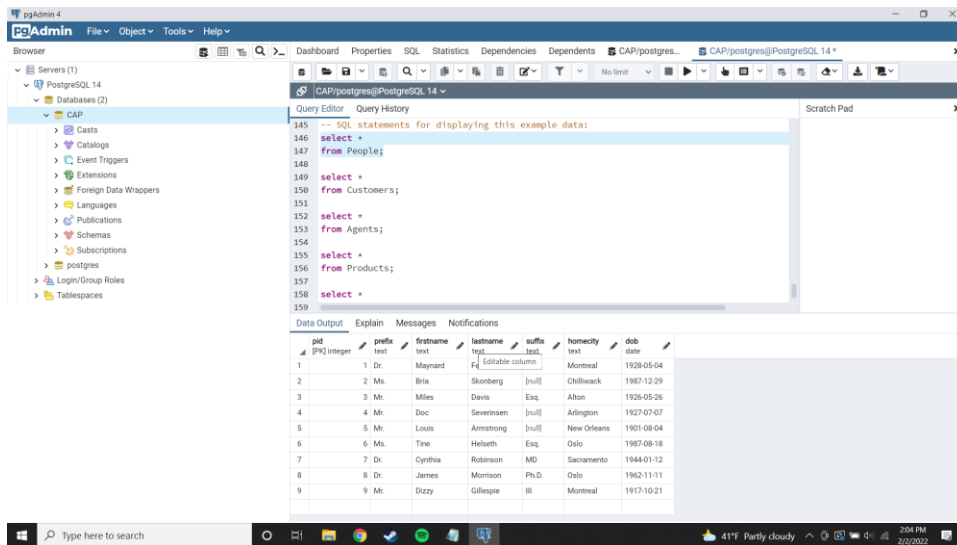
CMPT 308-111

Labouseur

2/2/2022

## Lab 2: CAP Database

1. *Execute the following queries(one at a time)from pgAdmin's SQL Tool:*

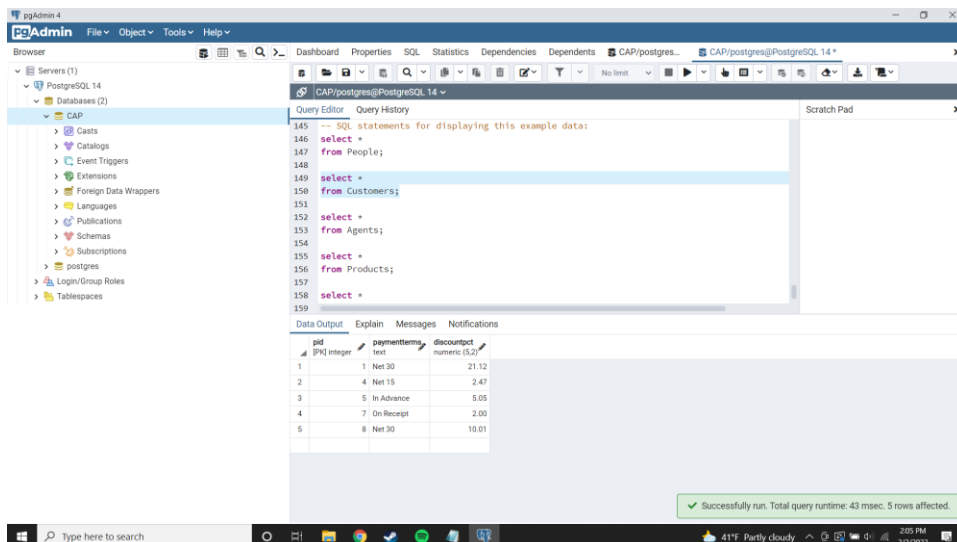


Query Editor

```
-- SQL statements for displaying this example data:
145
146 select *
147 from People;
148
149 select *
150 from Customers;
151
152 select *
153 from Agents;
154
155 select *
156 from Products;
157
158 select *
159
```

Data Output

pid	prefix	firstname	lastname	suffix	homecity	dob
1	Dr.	Maynard			Montreal	1928-05-04
2	Ms.	Bria	Stenberg	[null]	Chilliwack	1963-12-29
3	Mr.	Miles	Davis	Esq.	Albion	1926-05-26
4	Mr.	Doc	Sevenson	[null]	Arlington	1927-07-07
5	Mr.	Louis	Armstrong	[null]	New Orleans	1901-08-04
6	Ms.	Tine	Helath	Esq.	Oslo	1967-08-19
7	Dr.	Cynthia	Robinson	MD	Sacramento	1944-01-12
8	Dr.	James	Morrison	Ph.D.	Oslo	1962-11-11
9	Mr.	Dizzy	Gillespie	III	Montreal	1917-10-21



Query Editor

```
-- SQL statements for displaying this example data:
145
146 select *
147 from People;
148
149 select *
150 from Customers;
151
152 select *
153 from Agents;
154
155 select *
156 from Products;
157
158 select *
159
```

Data Output

pid	paymentterms	discountpct
1	Net 30	21.12
2	Net 15	2.47
3	In Advance	5.05
4	On Receipt	2.00
5	Net 30	10.01

Successfully run. Total query runtime: 43 msec. 5 rows affected.

pgAdmin 4

File Object Tools Help

Browser

- Servers (1)
  - PostgreSQL 14
    - Databases (2)
      - CAP
        - Costs
        - Catalogs
        - Event Triggers
        - Extensions
        - Foreign Data Wrappers
        - Languages
        - Publications
        - Schemas
        - Subscriptions
        - postgres
        - Login/Group Roles
        - Tablespaces

Query Editor Query History

SQL statements for displaying this example data:

```

145 -- SQL statements for displaying this example data:
146 select *
147 from People;
148
149 select *
150 from Customers;
151
152 select *
153 from Agents;
154
155 select *
156 from Products;
157
158 select *
159

```

Data Output Explain Messages Notifications

pid	paymentterms	commissioned
[PK] integer	text	numeric(5,2)
1	2 Quarterly	5.00
2	3 Annually	10.00
3	5 Monthly	1.00
4	6 Weekly	2.00

Successfully run. Total query runtime: 46 msec. 4 rows affected.

pgAdmin 4

File Object Tools Help

Browser

- Servers (1)
  - PostgreSQL 14
    - Databases (2)
      - CAP
        - Costs
        - Catalogs
        - Event Triggers
        - Extensions
        - Foreign Data Wrappers
        - Languages
        - Publications
        - Schemas
        - Subscriptions
        - postgres
        - Login/Group Roles
        - Tablespaces

Query Editor Query History

```

146 select *
147 from People;
148
149 select *
150 from Customers;
151
152 select *
153 from Agents;
154
155 select *
156 from Products;
157
158 select *
159 from Orders;
160

```

Data Output Explain Messages Notifications

prodid	name	city	qtyonhand	priceusd
[PK] character(2)	text	text	integer	numeric(10,2)
1 p01	Hesenberg Compensator	Dallas	47	67.50
2 p02	Universal Translator	Newark	2399	5.50
3 p03	Apple //+	Duluth	1979	65.02
4 p04	LCARS module	Duluth	3	47.00
5 p05	Dens Wick Valve Oil	Dallas	8675309	16.61
6 p06	Trapper Keeper	Dallas	1982	2.00
7 p07	Flux Capacitor	Newark	1007	1.00
8 p08	HAL 9000 memory core	Newark	200	1.25
9 p09	Bach Stradivarius 37	Montreal	1	37900.47

Successfully run. Total query runtime: 45 msec. 9 rows affected.

pgAdmin 4

File Object Tools Help

Browser

- Servers (1)
  - PostgreSQL 14
    - Databases (2)
      - CAP
        - Costs
        - Catalogs
        - Event Triggers
        - Extensions
        - Foreign Data Wrappers
        - Languages
        - Publications
        - Schemas
        - Subscriptions
        - postgres
        - Login/Group Roles
        - Tablespaces

Query Editor Query History

```

130 ('1815', '2021-01-23', 005, 003, 'p05', 10
131 ('1816', '2021-01-23', 008, 003, 'p03', 10
132 ('1817', '2021-02-14', 001, 003, 'p03', 5
133 ('1818', '2021-02-14', 001, 003, 'p04', 6
134 ('1819', '2021-02-14', 001, 002, 'p02', 4
135 ('1820', '2021-02-14', 004, 005, 'p07', 6
136 ('1821', '2021-02-14', 004, 005, 'p01', 10
137 ('1822', '2021-03-15', 001, 003, 'p06', 4
138 ('1823', '2021-03-15', 001, 002, 'p05', 5
139 ('1824', '2021-03-15', 005, 002, 'p01', 8
140 ('1825', '2020-04-01', 008, 003, 'p07', 8
141 ('1826', '2021-05-04', 008, 005, 'p03', 8
142 ;
143
144 -- SQL statements for displaying this ex
145
146 select *
147 from People;
148
149 select *
150 from Customers;
151
152 select *
153 from Agents;
154
155 select *
156 from Products;
157
158 select *
159 from Orders;
160

```

Data Output

orderid	dateordered	custid	agentid	prodid	quantityordered	totalusd
[PK] integer	date	integer	integer	character(2)	integer	numeric(12,2)
1	1011	2021-01-23	1	2 p01	1100	58566.40
2	1012	2021-01-23	4	3 p03	1200	76096.81
3	1015	2021-01-23	5	3 p05	1000	15771.20
4	1016	2021-01-23	8	3 p01	1000	60743.25
5	1017	2021-02-14	1	3 p03	500	25643.88
6	1018	2021-02-14	1	3 p04	600	22244.16
7	1019	2021-02-14	1	2 p02	400	1735.36
8	1020	2021-02-14	4	5 p07	600	585.18
9	1021	2021-02-14	4	5 p01	1000	65382.75
10	1022	2021-03-15	1	3 p06	450	709.92
11	1023	2021-03-15	1	2 p05	500	6550.98
12	1024	2021-03-15	5	2 p01	880	56400.30
13	1025	2020-04-01	8	3 p07	888	799.11
14	1026	2021-05-04	8	5 p03	808	47277.29

Data Output Explain Messages Notifications

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

A superkey is a column in the database which has a unique value for each row of the database, thus allowing a row to be identified by using the superkey. A candidate key is any column which satisfies the requirements of a superkey and therefore is a candidate to be used as the table key. The primary key is the candidate key which was chosen to be the key for the table. This is an important process to go through because not all superkeys should be used as the primary key. Price, for example, is a column which can easily satisfy the requirements of a superkey, but is a poor primary key because there can be duplicates as the database grows.

3. *Short essay on data types*

Data types are used in databases (and computers as a whole) to allow for more efficient use of memory, and to help promote uniformity in inputs and outputs. For example, if a specific query is performed on a column "PriceUSD", we can expect a floating-point value with two decimal places to be returned for every row that is returned. This is uniform, and allows us to use this data more directly and efficiently. "PriceUSD" would be considered a floating-point field, and would not accept input that does not comply with this. This is vital because the string "Twenty-two dollars and sixty-seven cents" is also a price in USD but cannot be used by a program to perform arithmetic.

**A table of cheeses**

- "NumID" - integer (not nullable)
- "Name" - String (not nullable)
- "Country of origin" - String (not nullable)
- "PriceUSD/pound" - float (nullable)
- "Pounds on hand" - float (nullable)
- "Years aged" - integer (nullable)
- "Best wine pairing" - String (nullable)

4. *Three relational rules*

The first relational rule is the "First Normal Form" rule and states that each cell of a table must be "atomic" or have only one value. If a category that is intended for use as a column would not satisfy this rule, another table can be made that exhibits it. This rule is important for ease of use, as having multiple values come up in a single location destroys the uniformity of the database and any queries performed on it. The second rule is "Access rows by content only" and is rather self-explanatory. Databases may not be stored sequentially in memory, so referencing them by order of entry, for example, may not yield the desired result. However, this rule is easily followed by using a primary key to reference distinct entries. The third rule is that "all rows must be unique." This rule also can be followed easily using a primary key, but it is important as to be sure that every entry can be accessed.