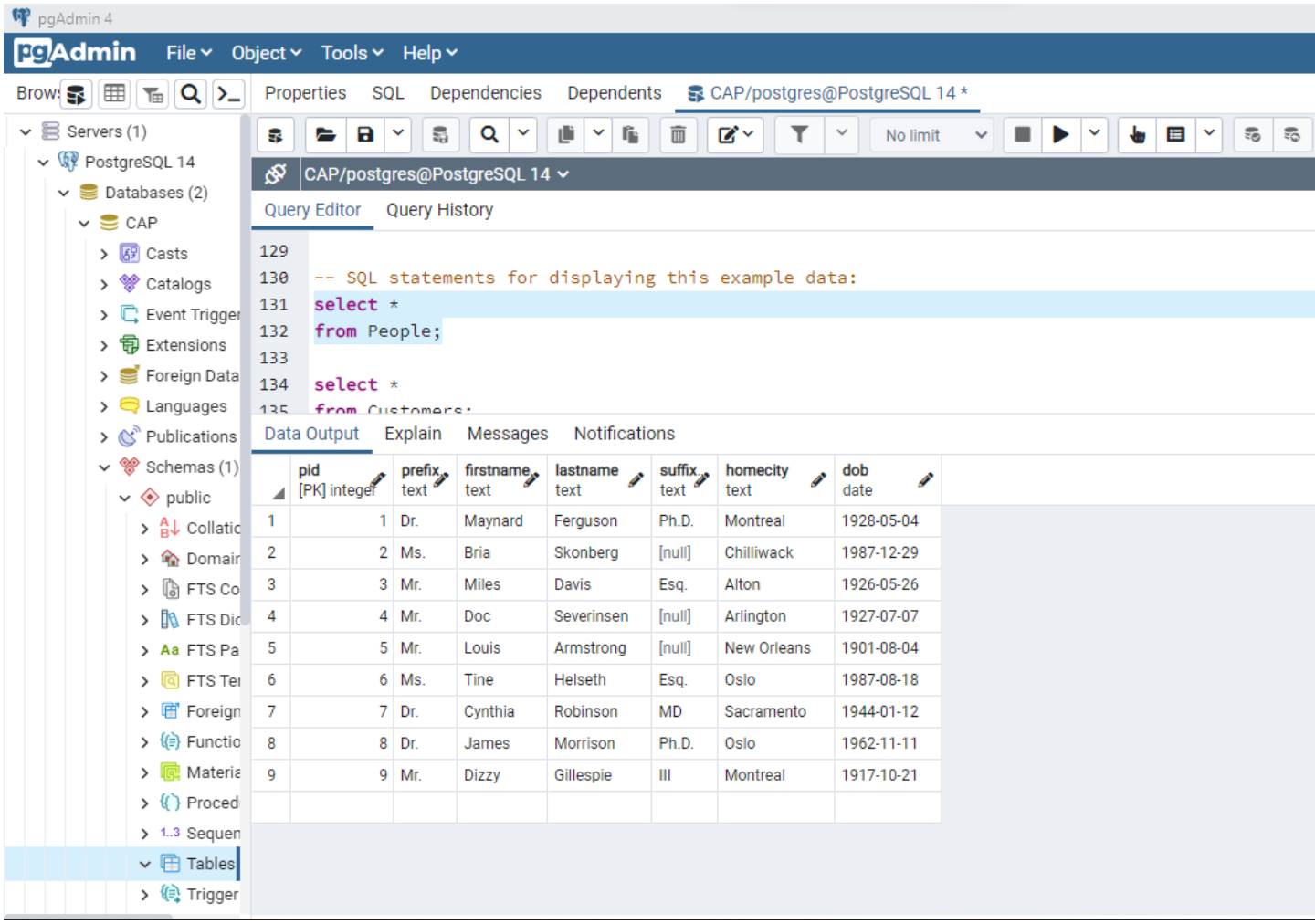
Raymond Tomo
01/27/21

# Lab 2: The  CAP Database

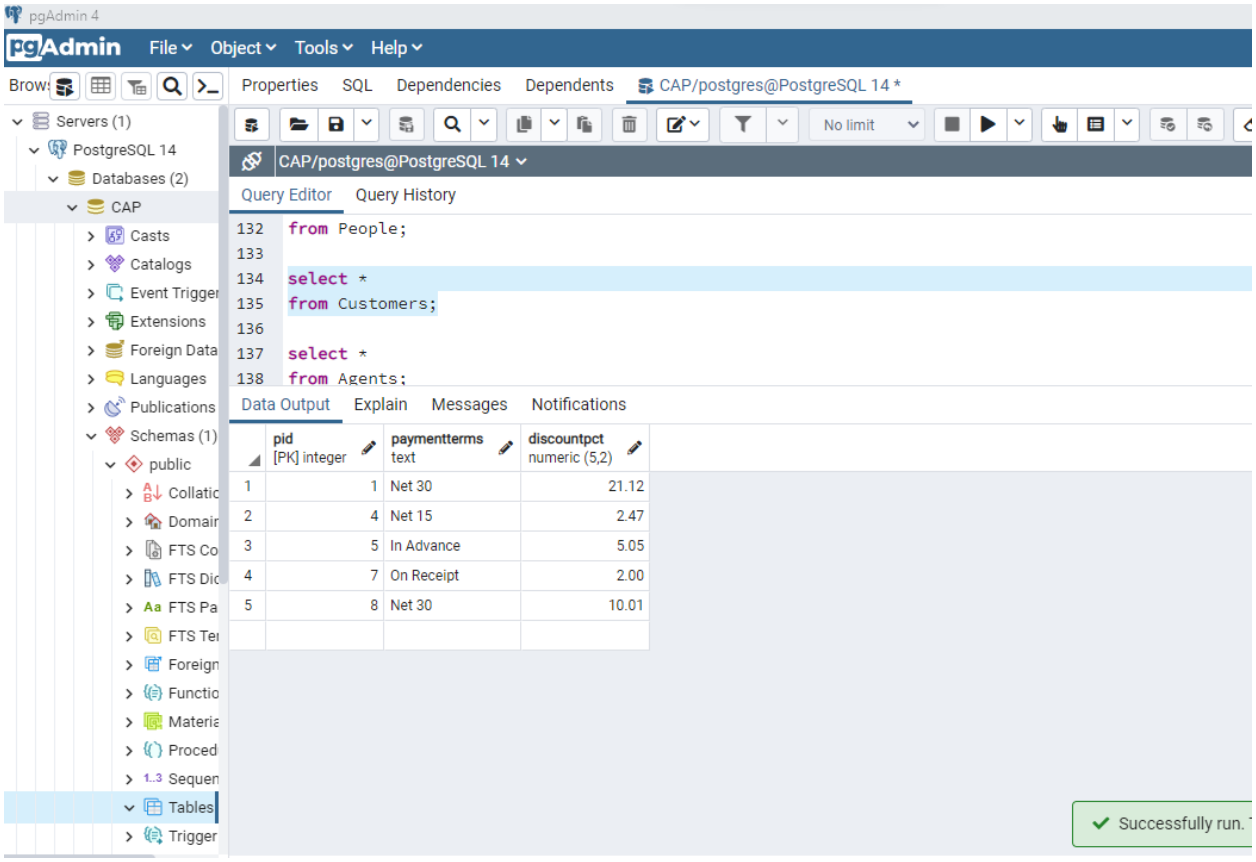
## Question #1

### Snapshots of Each Command

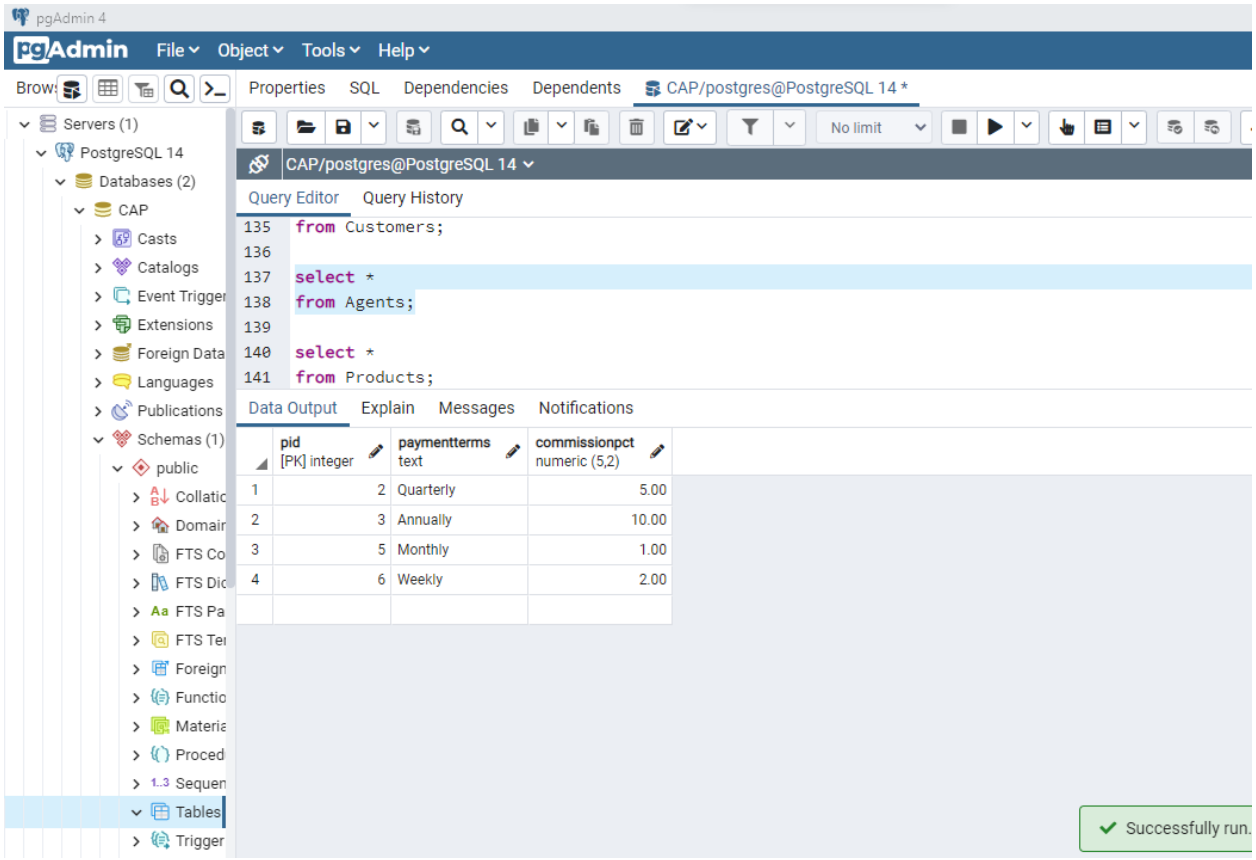
From People



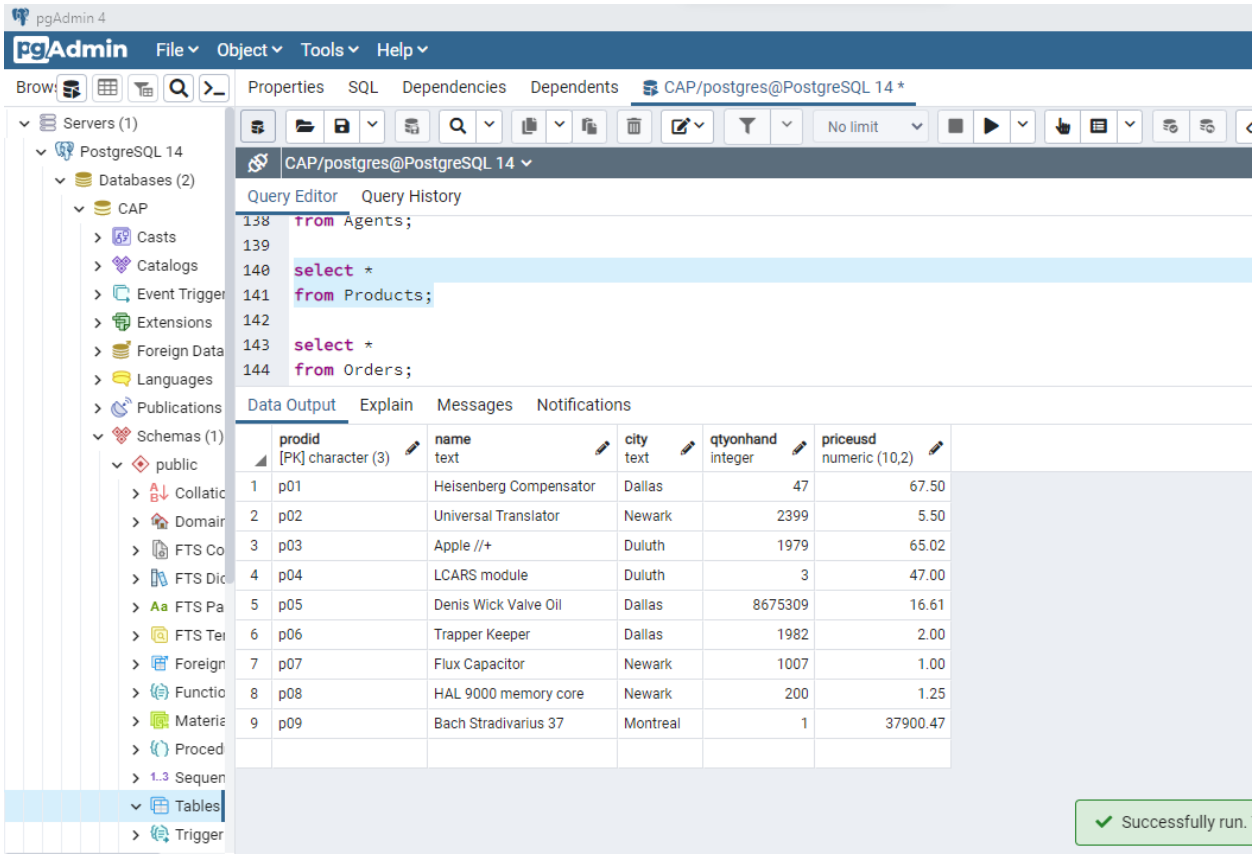

From Customers


pgAdmin 4
pgAdmin
File
Object
Tools
Help
Properties
SQL
Dependencies
Dependents
CAP/postgres@PostgreSQL 14 *
No limit
Servers (1)
PostgreSQL 14
Databases (2)
CAP
Casts
Catalogs
Event Trigger
Extensions
Foreign Data
Languages
Publications
Schemas (1)
public
Tables
Trigger
CAP/postgres@PostgreSQL 14
Query Editor
Query History
132 from People;
133
134 select *
135 from Customers;
136
137 select *
138 from Agents;
Data Output
Explain
Messages
Notifications
pid [PK] integer
paymentterms text
discountpct numeric (5,2)
1 1 Net 30 21.12
2 4 Net 15 2.47
3 5 In Advance 5.05
4 7 On Receipt 2.00
5 8 Net 30 10.01
Successfully run.

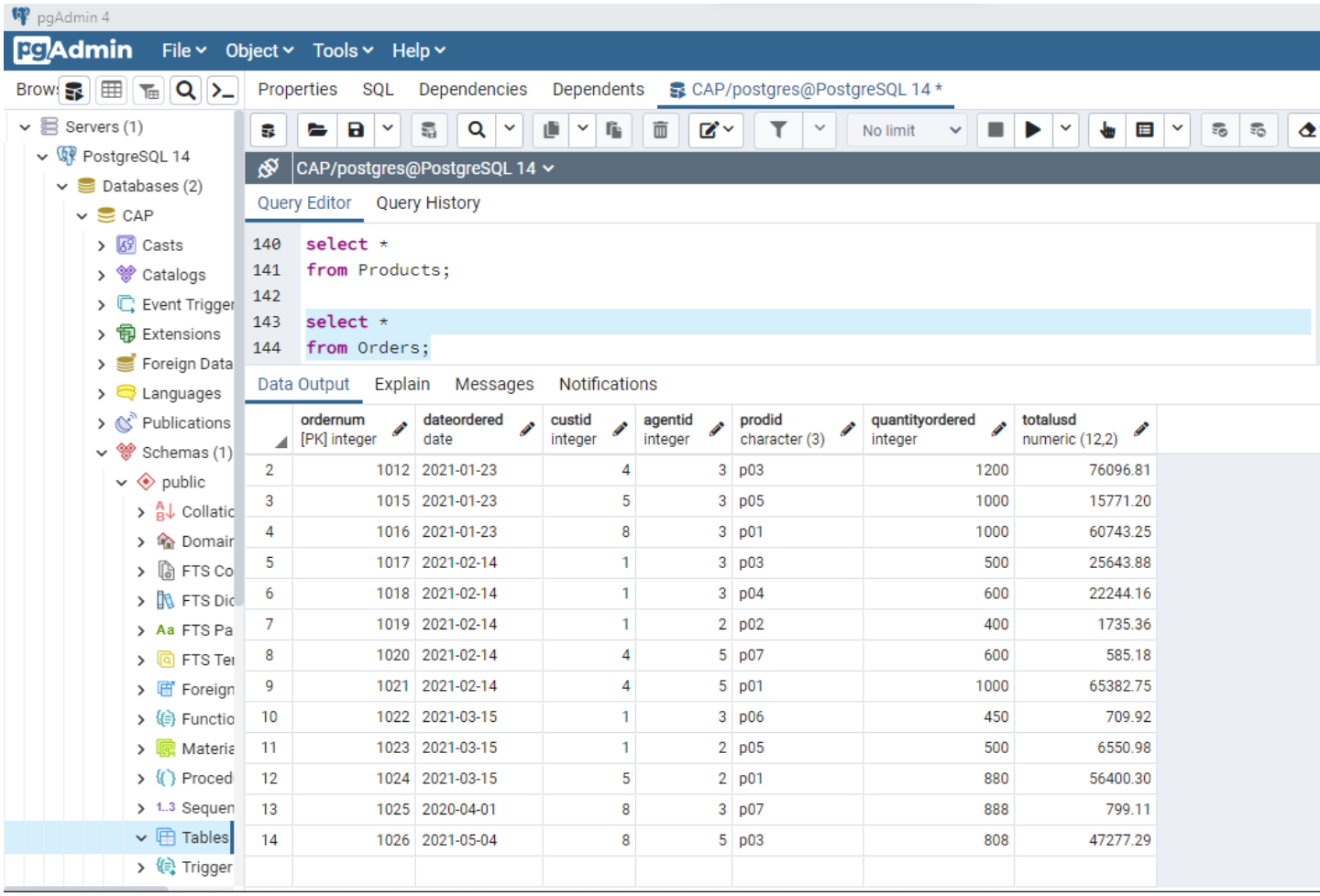
from Agents




from Products


pgAdmin 4
pgAdmin
File
Object
Tools
Help
Properties
SQL
Dependencies
Dependents
CAP/postgres@PostgreSQL 14 *
No limit
Servers (1)
PostgreSQL 14
Databases (2)
CAP
Casts
Catalogs
Event Trigger
Extensions
Foreign Data
Languages
Publications
Schemas (1)
public
Tables
Trigger
CAP/postgres@PostgreSQL 14
Query Editor
Query History
from Agents;
select *
from Products;
select *
from Orders;
Data Output
Explain
Messages
Notifications
prodid [PK] character (3)
name text
city text
qtyonhand integer
priceusd numeric (10,2)
1 p01 Heisenberg 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 Denis 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.

From Orders




| | ordernum [PK] integer | dateordered date | custid integer | agentid integer | prodid character (3) | quantityordered integer | totalusd numeric (12,2) |
|---|---|---|---|---|---|---|---|
| 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 |

In comparison to the snapshot on the website:

From People-

People

| pid | prefix | firstname | lastname | suffix | homecity | dob |
|---|---|---|---|---|---|---|
| 1 | Dr. | Maynard | Ferguson | Ph.D. | Montreal | 1928-05-04 |
| 2 | Ms. | Bria | Skonberg | | Chilliwack | 1987-12-29 |
| 3 | Mr. | Miles | Davis | Esq. | Alton | 1926-05-26 |
| 4 | Mr. | Doc | Severinsen | | Arlington | 1927-07-07 |
| 5 | Mr. | Louis | Armstrong | | New Orleans | 1901-08-04 |
| 6 | Ms. | Tine | Helseth | Esq. | Oslo | 1987-08-18 |
| 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 |

Data Output Explain Messages Notifications

| | pid [PK] integer | prefix text | firstname text | lastname text | suffix text | homecity text | dob date |
|---|---|---|---|---|---|---|---|
| 1 | 1 | Dr. | Maynard | Ferguson | Ph.D. | Montreal | 1928-05-04 |
| 2 | 2 | Ms. | Bria | Skonberg | [null] | Chilliwack | 1987-12-29 |
| 3 | 3 | Mr. | Miles | Davis | Esq. | Alton | 1926-05-26 |
| 4 | 4 | Mr. | Doc | Severinsen | [null] | Arlington | 1927-07-07 |
| 5 | 5 | Mr. | Louis | Armstrong | [null] | New Orleans | 1901-08-04 |
| 6 | 6 | Ms. | Tine | Helseth | Esq. | Oslo | 1987-08-18 |
| 7 | 7 | Dr. | Cynthia | Robinson | MD | Sacramento | 1944-01-12 |
| 8 | 8 | Dr. | James | Morrison | Ph.D. | Oslo | 1962-11-11 |
| 9 | 9 | Mr. | Dizzy | Gillespie | III | Montreal | 1917-10-21 |

All of the data matches!

From Customers-

Customers

```
 pid | paymentterms | discountpct
-----+--------------+-------------
   1 | Net 30       |       21.12
   4 | Net 15       |        2.47
   5 | In Advance   |        5.05
   7 | On Receipt   |        2.00
   8 | Net 30       |       10.01
```

Data Output Explain Messages Notifications

| | pid [PK] integer | paymentterms text | discountpct numeric (5,2) |
|---|---|---|---|
| 1 | 1 | Net 30 | 21.12 |
| 2 | 4 | Net 15 | 2.47 |
| 3 | 5 | In Advance | 5.05 |
| 4 | 7 | On Receipt | 2.00 |
| 5 | 8 | Net 30 | 10.01 |

All of the data matches!

From Agents-

## Agents

| pid | paymentterms | commissionpct |
|---|---|---|
| 2 | Quarterly | 5.00 |
| 3 | Annually | 10.00 |
| 5 | Monthly | 1.00 |
| 6 | Weekly | 2.00 |

Data Output  Explain  Messages  Notifications

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

All of the data matches!

From Products-

## Products

| prodid | name | city | qtyonhand | priceusd |
|---|---|---|---|---|
| p01 | Heisenberg Compensator | Dallas | 47 | 67.50 |
| p02 | Universal Translator | Newark | 2399 | 5.50 |
| p03 | Apple //+ | Duluth | 1979 | 65.02 |
| p04 | LCARS module | Duluth | 3 | 47.00 |
| p05 | Denis Wick Valve Oil | Dallas | 8675309 | 16.61 |
| p06 | Trapper Keeper | Dallas | 1982 | 2.00 |
| p07 | Flux Capacitor | Newark | 1007 | 1.00 |
| p08 | HAL 9000 memory core | Newark | 200 | 1.25 |
| p09 | Bach Stradivarius 37 | Montreal | 1 | 37900.47 |

Data Output | Explain | Messages | Notifications

| | prodid [PK] character (3) | name text | city text | qtyonhand integer | priceusd numeric (10,2) |
|---|---|---|---|---|---|
| 1 | p01 | Heisenberg 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 | Denis 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 |

All the data matches!

From Orders-

## Orders

| ordernum | dateordered | custid | agentid | prodid | quantityordered | totalusd |
|---|---|---|---|---|---|---|
| 1011 | 2021-01-23 | 1 | 2 | p01 | 1100 | 58568.40 |
| 1012 | 2021-01-23 | 4 | 3 | p03 | 1200 | 76096.81 |
| 1015 | 2021-01-23 | 5 | 3 | p05 | 1000 | 15771.20 |
| 1016 | 2021-01-23 | 8 | 3 | p01 | 1000 | 60743.25 |
| 1017 | 2021-02-14 | 1 | 3 | p03 | 500 | 25643.88 |
| 1018 | 2021-02-14 | 1 | 3 | p04 | 600 | 22244.16 |
| 1019 | 2021-02-14 | 1 | 2 | p02 | 400 | 1735.36 |
| 1020 | 2021-02-14 | 4 | 5 | p07 | 600 | 585.18 |
| 1021 | 2021-02-14 | 4 | 5 | p01 | 1000 | 65382.75 |
| 1022 | 2021-03-15 | 1 | 3 | p06 | 450 | 709.92 |
| 1023 | 2021-03-15 | 1 | 2 | p05 | 500 | 6550.98 |
| 1024 | 2021-03-15 | 5 | 2 | p01 | 880 | 56400.30 |
| 1025 | 2020-04-01 | 8 | 3 | p07 | 888 | 799.11 |
| 1026 | 2021-05-04 | 8 | 5 | p03 | 808 | 47277.29 |

Data Output Explain Messages Notifications

| | ordernum [PK] integer | dateordered date | custid integer | agentid integer | prodid character (3) | quantityordered integer | totalusd numeric (12,2) |
|---|---|---|---|---|---|---|---|
| 1 | 1011 | 2021-01-23 | 1 | 2 | p01 | 1100 | 58568.40 |
| 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 |

All of the data matches

# Question #2

Explain the distinctions between primary key, candidate key, and super key.

A super key is **any field** or a set of fields that can uniquely identify each row on the table. For instance, a field that gives each row a unqiue order number in a table that keeps track of orders would be a super key on that table. A column that simply checks whether or not an order has been recieved with either yes or no as potential data options would not be considered a key.

A **candidate key** is a *minimal* super key. By minimal, I mean that it only affects one field. This means unlike a super key, it cannot encompass more than one column/field of data by definition. So while a candidate key is a super key, a super key is not always a candidate key. Thus, is it a type of superkey in the same sense that a square is a rectangle but not all rectangles are squares,

A **primary key** is a selected candidate key to serve a purpose of the database. While there can be **many** candidate keys, there can only be one of them per relational database, following good database design. Like the previous example, a primary key is a candidate key, but not all candidate keys are primary keys. This key will usually be used in multiple tables within the

database, such as an ID for a specific customer that might be in a customers table and a orders table to identify them within the database.

# Question #3

There are many datatypes available within SQL, and a great deal of interesting native data types for the server. One of the dataypes that can be used is a numeric data type.This includes datatypes such as the an integer, which represents a basic whole number. Other variants of this data are decimal, smallint and bigint, which represent decimal values, small integers, and large integers. For text-based values there are character types, such as text (unlimited character values), varchar(n) (which limits the the characters to the value of n) and other types that allow you to insert a text based value. Lastly, there are other more unique data types such as date, which allows a user to insert a specific date as a value in a table.

For the example of a table, imagine there is a list of students registered for a club called CLUB LIST. There are four columns for the person that they can use. First is the student ID column, which is a dataype int column that keeps them organized by their id number and is not nullable. Second, is the First names column, which is of type varchar (50) that allows a person to insert their first name. The last name columns works in the same way. Both of them are not nullable Lastly, the final column is of the date type, which lists their date of birth, and is not nullable.

# Question #4

The **first normal form rule** of database design dictates that there can be no multiple valued attributes (fields) within a table. For instance, if there was a table called courses where each row represents a student, there would be many values that could be entered into that column, making it ineffective and confusing. So long as a field cannot be subdivided it will abide by this rule of database design.

The **access rows by content only** dictates that we can only ask for data by what's there, not where the data is. What this means is that we can ask for the name of a specific element through SQL for instance. But we cannot make SQL tell us the name in the first row of a database. Something vague such as that will not work and is generally considered bad database design if it is permitted, as it goes against the ordered and structured methods used to create the relational model in the first place.

The **all rows must be unique rule** dictates that there cannot be any duplicate rows of information. Since there are no intrinsic orders within the set a table represents,a  duplicate row inherently goes against this logical principle, making it confusing and leading to a confusionof the information provided by the table. This is considered a failure of database design if this occurs, as data with bad or unreliable context is utterly useless. T