TEXT TO SQL

ChatGPT4 - Attached the CAP
database file for each request
Shannon Maier
March 21, 2024

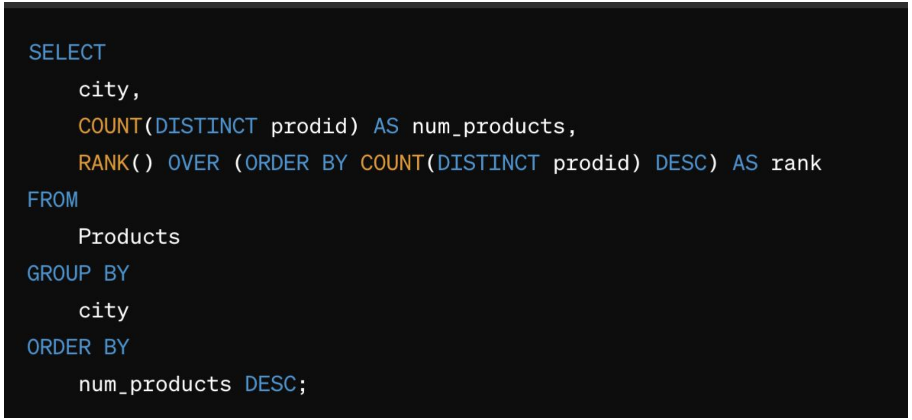
GRADING METHODOLOGY

- Each Question is Graded out of 5 points
- Each minor error deducts .5 point
- Major error deducts 1 point
- If the query is written completely wrong with major errors the grade will be 0/5

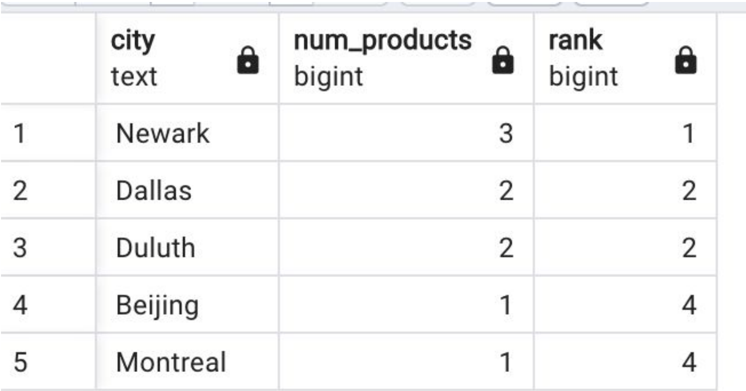
Query 1

Display the cities that makes the most different kinds of products. Experiment with the rank() function.

- My prompt: Using the CAP database provided in the attached document, write a query to Display the cities that make the most different kinds of products. Experiment with the rank() function.
- SQL Written by ChatGPT4: Result of Running:



```
SELECT
    city,
    COUNT(DISTINCT prodid) AS num_products,
    RANK() OVER (ORDER BY COUNT(DISTINCT prodid) DESC) AS rank
FROM
    Products
GROUP BY
    city
ORDER BY
    num_products DESC;
```

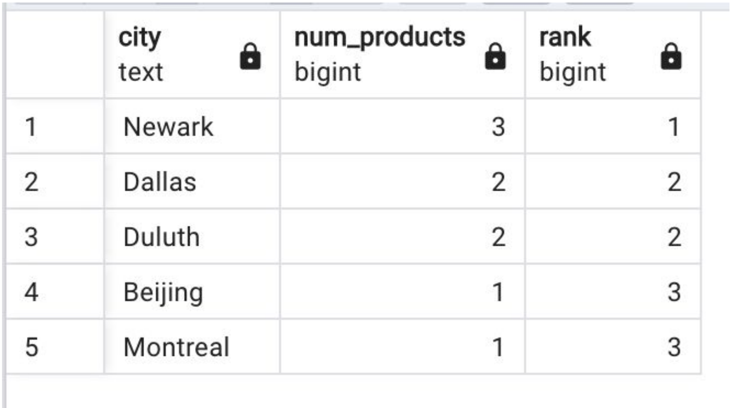


	city text	num_products bigint	rank bigint
1	Newark	3	1
2	Dallas	2	2
3	Duluth	2	2
4	Beijing	1	4
5	Montreal	1	4

- My Grade: 4.5/5 → rank skips #3
- My fixes: Change rank() to dense_rank() in order to make the rank not skip #3

Query 1 Cont.

- My Results:



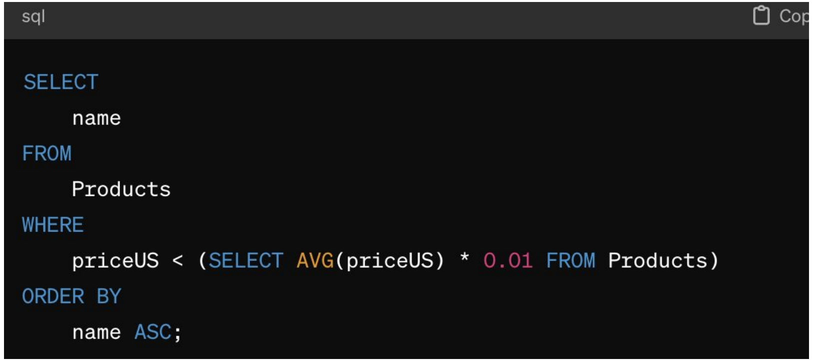
	city text	num_products bigint	rank bigint
1	Newark	3	1
2	Dallas	2	2
3	Duluth	2	2
4	Beijing	1	3
5	Montreal	1	3

- A measure of different they are (1-5 - 5 being the most different): 1 → only slight difference in the use of rank so the rank is shown as 4 and not 3.

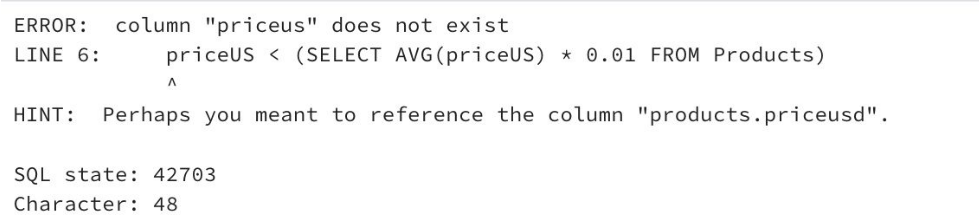
Query 2

Display the names of products whose priceUSD is less than 1% of the average priceUSD, in alphabetical order. From A to Z.

- My prompt: Using the CAP database provided in the attached document, write a query that Displays the names of products whose priceUSD is less than 1% of the average priceUSD, in alphabetical order. From A to Z.
- SQL Written by ChatGPT4: Result of Running:



```sql
SELECT
    name
FROM
    Products
WHERE
    priceUS < (SELECT AVG(priceUS) * 0.01 FROM Products)
ORDER BY
    name ASC;
```



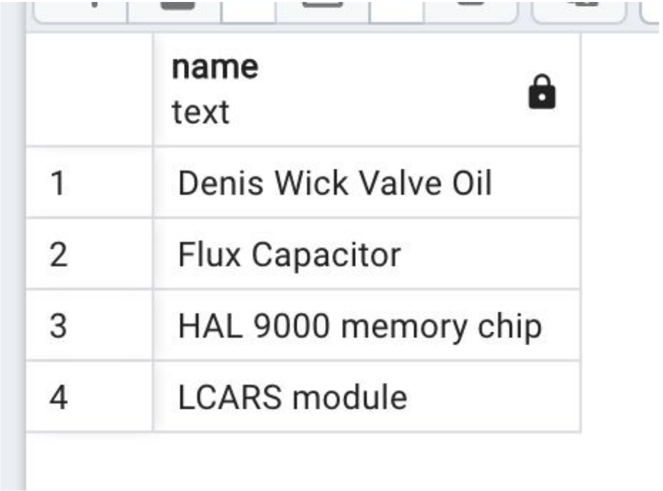
```
ERROR:  column "priceus" does not exist
LINE 6:     priceUS < (SELECT AVG(priceUS) * 0.01 FROM Products)
            ^
HINT:  Perhaps you meant to reference the column "products.priceusd".

SQL state: 42703
Character: 48
```

- My grade: 4.5/5
- My fixes: Change ‘priceUS’ to ‘priceUSD’

Query 2 Cont.

- My Results:



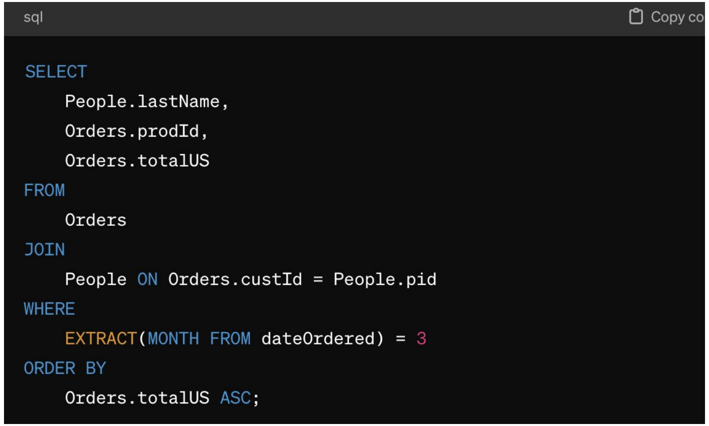
	name text
1	Denis Wick Valve Oil
2	Flux Capacitor
3	HAL 9000 memory chip
4	LCARS module

- A measure of different they are (1-5 - 5 being the most different): 1 → only had to change the mistake in syntax, slight difference.
-

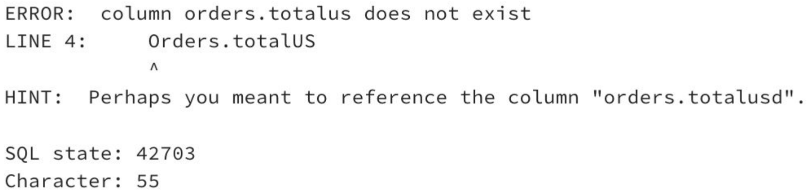
Query 3

Display the customer last name, product id ordered, and the totalUSD for all orders made in March of any year, sorted by totalUSD from low to high.

- My prompt: Using the CAP database provided in the attached document, write a query that Displays the customer last name, product id ordered, and the totalUSD for all orders made in March of any year, sorted by totalUSD from low to high.
- SQL Written by ChatGPT4: Result of Running:



```sql
SELECT
    People.lastName,
    Orders.prodId,
    Orders.totalUS
FROM
    Orders
JOIN
    People ON Orders.custId = People.pid
WHERE
    EXTRACT(MONTH FROM dateOrdered) = 3
ORDER BY
    Orders.totalUS ASC;
```



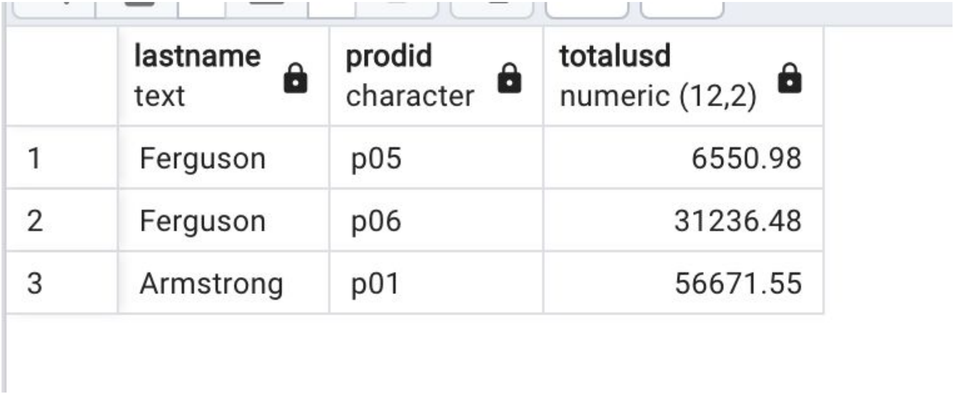
```
ERROR:  column orders.totalus does not exist
LINE 4:      Orders.totalUS
             ^
HINT:  Perhaps you meant to reference the column "orders.totalusd".

SQL state: 42703
Character: 55
```

- My grade: 4.5/5
- My fixes: Change ‘Orders.priceUS’ to ‘Orders.priceUSD’

Query 3 Cont

- My Results:



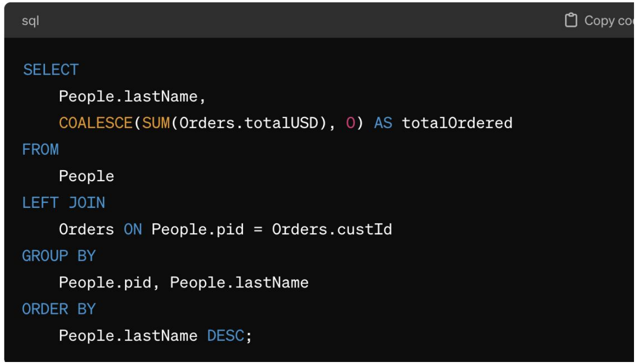
	lastname text	prodid character	totalusd numeric (12,2)
1	Ferguson	p05	6550.98
2	Ferguson	p06	31236.48
3	Armstrong	p01	56671.55

- A measure of different they are (1-5 - 5 being the most different): only had to change the mistake in syntax, slight difference. The difference between what I wrote though is slightly more because I used the date_part() function, while the AI used the extract function().The actual result was quite different because the AI result came out with an error.

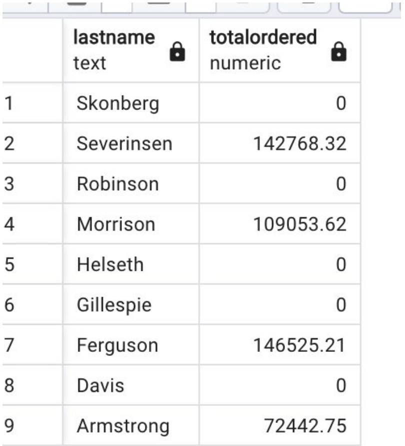
Query 4

Display the last name of all customers (in reverse alphabetical order) and their total ordered by customer, and nothing more. Use coalesce to avoid showing NULL totals.

- My prompt: Using the CAP database provided in the attached document, write a query to Display the last name of all customers (in reverse alphabetical order) and their total ordered by customer, and nothing more. Use coalesce to avoid showing NULL totals.
- SQL Written by ChatGPT4: Result of Running:



```sql
SELECT
    People.lastName,
    COALESCE(SUM(Orders.totalUSD), 0) AS totalOrdered
FROM
    People
LEFT JOIN
    Orders ON People.pid = Orders.custId
GROUP BY
    People.pid, People.lastName
ORDER BY
    People.lastName DESC;
```

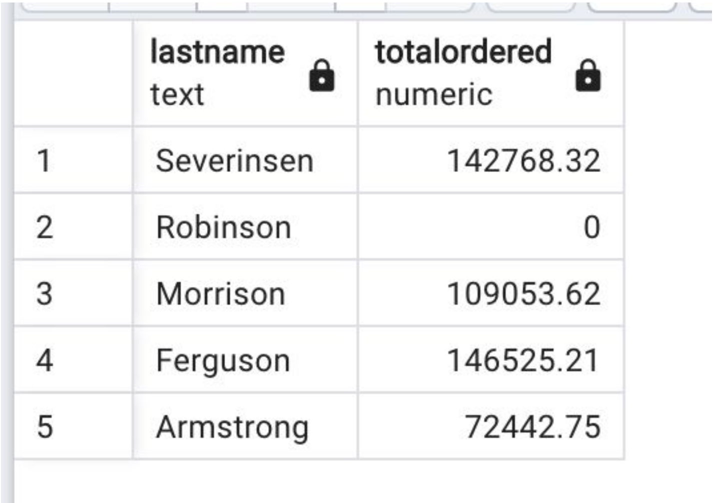


	lastname text	totalordered numeric
1	Skonberg	0
2	Severinsen	142768.32
3	Robinson	0
4	Morrison	109053.62
5	Helseth	0
6	Gillespie	0
7	Ferguson	146525.21
8	Davis	0
9	Armstrong	72442.75

- My grade: ⅘ → did not filter only customers, shows all peop
- My fixes: Add ‘inner join customers on people.pid = customers.pid” before the left join

Query 4 Cont.

- My Results:



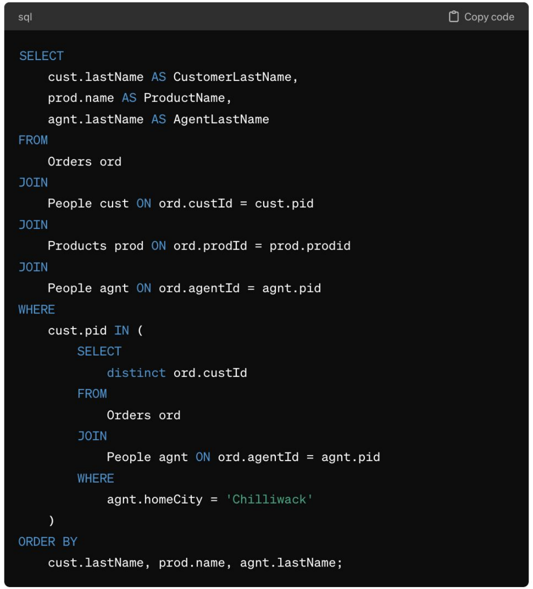
	lastname text	totalordered numeric
1	Severinsen	142768.32
2	Robinson	0
3	Morrison	109053.62
4	Ferguson	146525.21
5	Armstrong	72442.75

- A measure of different they are (1-5 - 5 being the most different): 2 → because it is a significant difference that changes the output of the query, but not a hard fix.

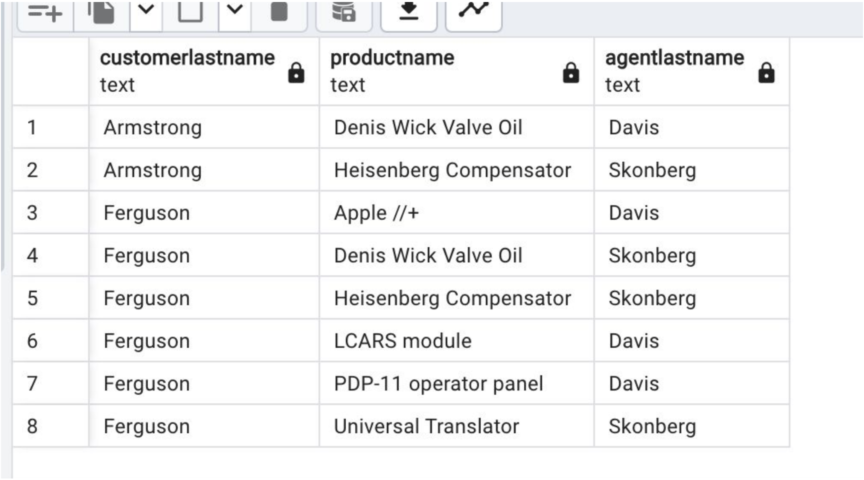
Query 5

Display the names of all customers who bought products from agents based in Chilliwack along with the names of the products they ordered, and the names of the agents who sold it to them.

- My prompt: Using the CAP database provided in the attached document, write a query to Display the names of all customers who bought products from agents based in Chilliwack along with the names of the products they ordered including the products not ordered from the agent in Chilliwack, and the names of the agents who sold it to them.
- SQL Written by ChatGPT4: Result of Running:



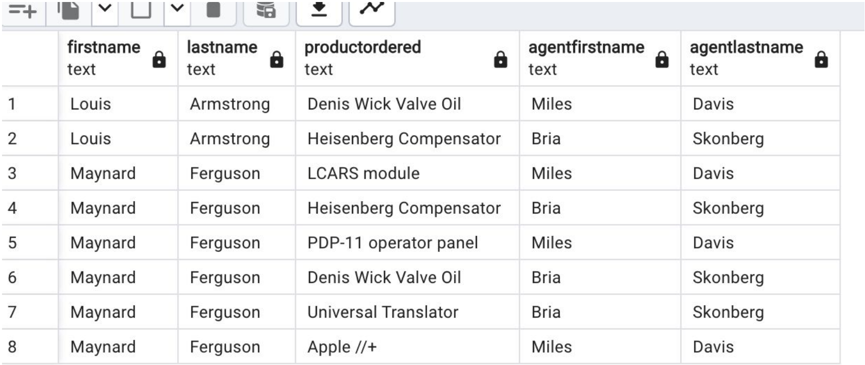
```sql
SELECT
    cust.lastName AS CustomerLastName,
    prod.name AS ProductName,
    agnt.lastName AS AgentLastName
FROM
    Orders ord
JOIN
    People cust ON ord.custId = cust.pid
JOIN
    Products prod ON ord.prodId = prod.prodid
JOIN
    People agnt ON ord.agentId = agnt.pid
WHERE
    cust.pid IN (
        SELECT
            distinct ord.custId
        FROM
            Orders ord
        JOIN
            People agnt ON ord.agentId = agnt.pid
        WHERE
            agnt.homeCity = 'Chilliwack'
    )
ORDER BY
    cust.lastName, prod.name, agnt.lastName;
```



	customerlastname text	productname text	agentlastname text
1	Armstrong	Denis Wick Valve Oil	Davis
2	Armstrong	Heisenberg Compensator	Skonberg
3	Ferguson	Apple //+	Davis
4	Ferguson	Denis Wick Valve Oil	Skonberg
5	Ferguson	Heisenberg Compensator	Skonberg
6	Ferguson	LCARS module	Davis
7	Ferguson	PDP-11 operator panel	Davis
8	Ferguson	Universal Translator	Skonberg

Query 5 Cont.

- My grade: 5/5 → did not need to change anything to have the correct output
- Result from how I wrote my query in Lab 6:



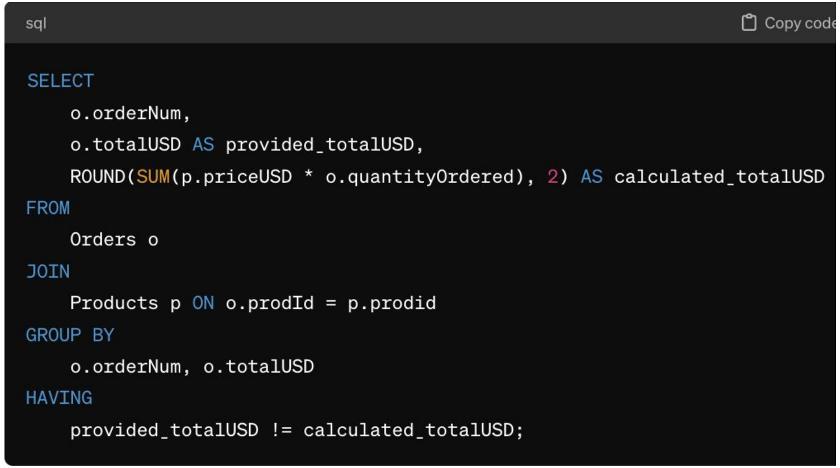
	firstname text	lastname text	productordered text	agentfirstname text	agentlastname text
1	Louis	Armstrong	Denis Wick Valve Oil	Miles	Davis
2	Louis	Armstrong	Heisenberg Compensator	Bria	Skonberg
3	Maynard	Ferguson	LCARS module	Miles	Davis
4	Maynard	Ferguson	Heisenberg Compensator	Bria	Skonberg
5	Maynard	Ferguson	PDP-11 operator panel	Miles	Davis
6	Maynard	Ferguson	Denis Wick Valve Oil	Bria	Skonberg
7	Maynard	Ferguson	Universal Translator	Bria	Skonberg
8	Maynard	Ferguson	Apple //+	Miles	Davis

- A measure of different they are (1-5 - 5 being the most different): 3 → Although the AI did write a correct query, there were some difference between how I implemented it in Lab 6. I did not order the results the same as CHATGPT, and I added the first and last names of the customers. In the actual writing of the query, I used a multileveled subquery, while the AI used a single leveled subquery with a join.

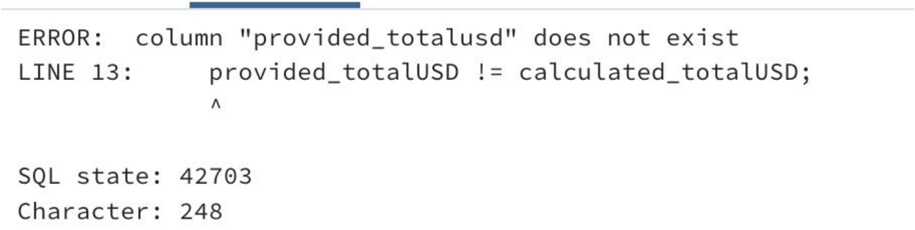
Query 6

Write a query to check the accuracy of the totalUSD column in the Orders table. This means calculating Orders.totalUSD from data in other tables and comparing those values to the values in Orders.totalUSD. Display all rows in Orders where Orders.totalUSD is incorrect, if any. If there are any incorrect values, explain why they are wrong. Round to exactly two decimal places.

- My prompt: Using the CAP database provided in the attached document,Write a query to check the accuracy of the totalUSD column in the Orders table. This means calculating Orders.totalUSD from data in other tables and comparing those values to the values in Orders.totalUSD. Display all rows in Orders where Orders.totalUSD is incorrect, if any. Round to exactly two decimal places.
- SQL Written by ChatGPT4: Result of Running:



```sql
SELECT
    o.orderNum,
    o.totalUSD AS provided_totalUSD,
    ROUND(SUM(p.priceUSD * o.quantityOrdered), 2) AS calculated_totalUSD
FROM
    Orders o
JOIN
    Products p ON o.prodId = p.prodid
GROUP BY
    o.orderNum, o.totalUSD
HAVING
    provided_totalUSD != calculated_totalUSD;
```

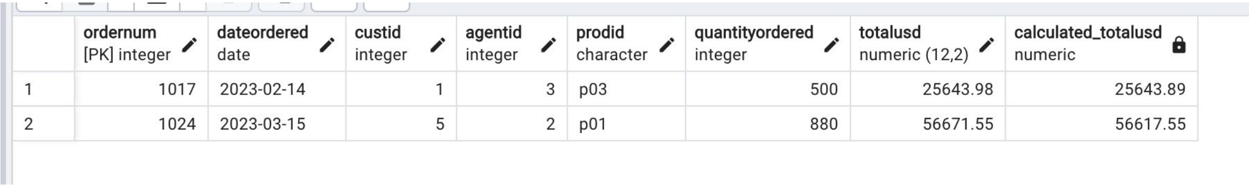


```
ERROR:  column "provided_totalusd" does not exist
LINE 13:     provided_totalUSD != calculated_totalUSD;
             ^

SQL state: 42703
Character: 248
```

Query 6 Cont.

- My grade: 0/5 → critical mistakes that caused the error messages and even when syntax was fixed to attempt to correct the error message, the logic was still incorrect, and the output shown had incorrect values and organization
- My fixes:
  - Changed the select statement:
    - orders.*
    - Deleted the Second Select line
    - Changed the Round Function to include all info needed to calculate the “calculated total”
  - Added two join statements to include the customers and agents table
  - Changed the HAVING to WHERE
- My Results:



	ordernum [PK] integer	dateordered date	custid integer	agentid integer	prodid character	quantityordered integer	totalusd numeric (12,2)	calculated_totalusd numeric
1	1017	2023-02-14	1	3	p03	500	25643.98	25643.89
2	1024	2023-03-15	5	2	p01	880	56671.55	56617.55

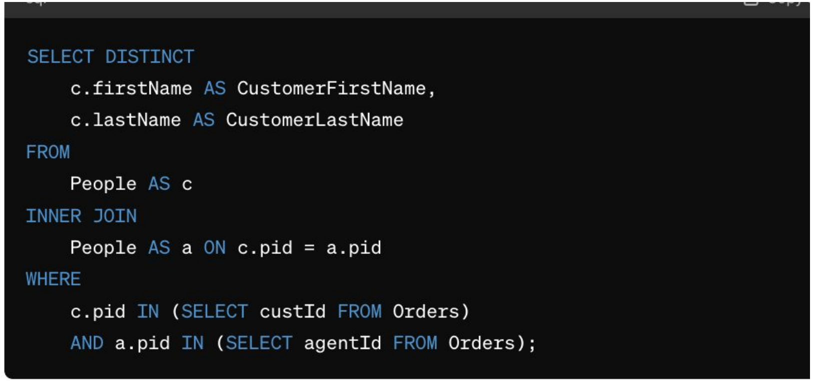
- A measure of how different they are (1-5 - 5 being the most different): 5 → They had many significant differences as the original AI code produced errors and calculated the totals wrong so every calculated total was shown as different from the proposed total.

Query 7

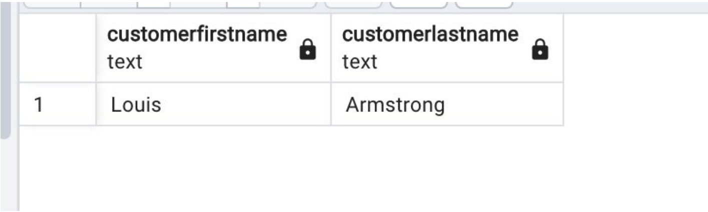
Display the first and last name of all customers who are also agents.

- My prompt: Using the CAP database provided in the attached document, Write a query to Display the first and last name of all customers who are also agents.
- SQL Written by ChatGPT4:

Result of Running:



```
SELECT DISTINCT
    c.firstName AS CustomerFirstName,
    c.lastName AS CustomerLastName
FROM
    People AS c
INNER JOIN
    People AS a ON c.pid = a.pid
WHERE
    c.pid IN (SELECT custId FROM Orders)
    AND a.pid IN (SELECT agentId FROM Orders);
```

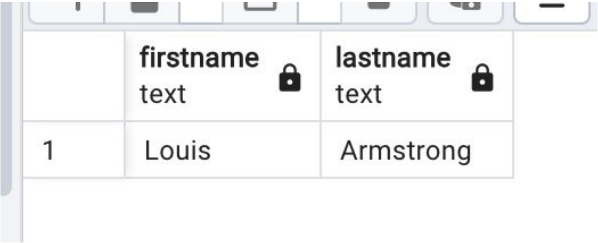


	customerfirstname text	customerlastname text
1	Louis	Armstrong

- My grade: 5/5 → did not need to change anything to have the correct output

Query 7 Cont.

- Result from how I wrote my query in Lab 6:



	firstname text	lastname text
1	Louis	Armstrong

- A measure of different they are (1-5 - 5 being the most different): 3 → Although the AI did write a correct query, there were some difference between how I implemented it in Lab 6. The way I wrote my code displayed slight different because I did not show the column names as customerfirstname and customerlastname → only difference, not an error. I also implemented through subqueries, not joins

Conclusion

For the most part, this experiment worked very well for the query requests I used in ChatGPT4. As the queries became more complex (like query 6), it was obvious that the AI did not understand the requests as well as the simpler ones. For the most part though, ChatGPT understood the basis of what I was asking, and most of the queries only needed minor tweaking to be done correctly. I think with enough manipulation of wording and rewording, ChatGPT4 can do a very good job of writing efficient queries.