

Exercise 3.9

Chrisie Hunter

Step 1: Find the average amount paid by the top 5 customers.

The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database structure for 'Rockbuster' and 'postgres'. The main pane shows a SQL query in the 'Query' tab:

```
1 With payment_amt_cte (customer_id, amount) AS
2 (Select A.customer_id,
3   A.amount,
4   B.first_name,
5   B.last_name
6 From payment A
7 Inner Join customer B ON A.customer_id = B.customer_id)
8 Select
9   customer_id,
10  first_name,
11  last_name,
12  Avg(amount) AS average_amount_paid
13 From payment_amt_cte
14 Group by customer_id,
15 first_name,
16 last_name
17 Order by average_amount_paid DESC
18 Limit 5;
```

The 'Data Output' tab shows the results of the query:

customer_id	first_name	last_name	average_amount_paid
187	Brittany	Riley	5.6196296296296296
321	Kevin	Schuler	5.5194117647058824
19	Ruth	Martinez	5.4900000000000000
3	Linda	Williams	5.4483333333333333
311	Paul	Trout	5.3900000000000000

Total rows: 5 of 5 Query complete 00:00:00.260 Ln 15, Col 4

The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database structure for 'Rockbuster' and 'postgres'. The main pane shows a SQL query in the 'Query' tab:

```
1 With payment_amt_cte (customer_id, amount, first_name, last_name, country) AS
2 (Select A.customer_id,
3   A.amount,
4   B.first_name,
5   B.last_name,
6   E.country
7 From payment A
8 Inner Join customer B ON A.customer_id = B.customer_id
9 Inner Join address C ON B.address_id = C.address_id
10 Inner Join city D ON C.city_id = D.city_id
11 Inner Join country E ON D.country_id = E.country_id)
12 Select
13   customer_id,
14   first_name,
15   last_name,
16   Avg(amount) AS average_amount_paid,
17   country
18 From payment_amt_cte
19 Group by customer_id,
20 first_name,
21 last_name,
22 country
23 Order by average_amount_paid DESC
24 Limit 5;
```

The 'Data Output' tab shows the results of the query:

customer_id	first_name	last_name	average_amount_paid	country
187	Brittany	Riley	5.6196296296296296	Ukraine
321	Kevin	Schuler	5.5194117647058824	Nepal
19	Ruth	Martinez	5.4900000000000000	South Africa
3	Linda	Williams	5.4483333333333333	Greece
311	Paul	Trout	5.3900000000000000	Russian Federation

Total rows: 5 of 5 Query complete 00:00:00.130 Ln 5, Col 15

Step 2.

pgAdmin 4

File Object Tools Help

Browsers

- Servers (1)
 - PostgreSQL 15
 - Databases (2)
 - Rockbuster
 - Cast
 - Catalogs
 - Event Triggers
 - Extensions
 - Foreign Data Wrappers
 - Languages
 - Publications
 - Schemas
 - Subscriptions
 - postgres
 - Login/Group Roles
 - Tablespaces

Dashboard Properties SQL Statistics Dependencies Dependents Processes 3.8 step 2.sql Rockbuster/postgres@PostgreSQL 15*

Query Query History

```
1 Select Avg(payment_amt) AS average_payment
2 From
3 (Select A.customer_id,
4  A.first_name,
5  A.last_name,
6  C.city,
7  D.country,
8  Sum(E.amount) As payment_amt
9 From customer A
10 Inner Join address B On A.address_id = B.address_id
11 Inner Join city C On B.city_id = C.city_id
12 Inner Join country D On C.country_id = D.country_id
13 Inner Join payment E on A.customer_id = E.customer_id
14 Where C.city IN ('Aurora',
15  'Atlixco',
16  'Pingxiang',
17  'Adoni',
18  'Dhule(Dhulia)',
19  'Kirashiki',
20  'Xintai',
21  'Sivas',
22  'Celaya',
23  'Nezahualcoyotli')
24 Group By A.customer_id,
25  A.first_name,
26  A.last_name,
27  C.city,
28  D.country
29 Order by payment_amt DESC
30 Limit 5) AS total_amount_paid;
```

Data Output Messages Notifications

average_payment
107.35400000000000000000

Total rows: 1 of 1 Query complete 00:00:00.153 Ln 3, Col 22

Russell 2000 -1.29%

12:08 PM 4/14/2023

Exercise 3.8.1

pgAdmin 4

File Object Tools Help

Browsers

- Servers (1)
 - PostgreSQL 15
 - Databases (2)
 - Rockbuster
 - Cast
 - Catalogs
 - Event Triggers
 - Extensions
 - Foreign Data Wrappers
 - Languages
 - Publications
 - Schemas
 - Subscriptions
 - postgres
 - Login/Group Roles
 - Tablespaces

Dashboard Properties SQL Statistics Dependencies Dependents Processes 3.8 step 2.sql Rockbuster/postgres@PostgreSQL 15*

Query Query History

```
1 Explain
2 Select Avg(payment_amt) AS average_payment
3 From
4 (Select A.customer_id,
5  A.first_name,
6  A.last_name,
7  C.city,
8  D.country,
9  Sum(E.amount) As payment_amt
10 From customer A
11 Inner Join address B On A.address_id = B.address_id
12 Inner Join city C On B.city_id = C.city_id
13 Inner Join country D On C.country_id = D.country_id
14 Inner Join payment E on A.customer_id = E.customer_id
15 Where C.city IN ('Aurora',
16  'Atlixco',
17  'Pingxiang',
18  'Adoni',
19  'Dhule(Dhulia)',
20  'Kirashiki',
21  'Xintai',
22  'Sivas',
23  'Celaya',
24  'Nezahualcoyotli')
25 Group By A.customer_id,
26  A.first_name,
27  A.last_name,
28  C.city,
29  D.country
30 Order by payment_amt DESC
31 Limit 5)
```

Data Output Messages Notifications

QUERY PLAN

```
1 Aggregate (cost=64.40..64.50 rows=1 width=32)
2  -> Limit (cost=64.00..64.42 rows=5 width=67)
3  -> Sort (cost=64.41..65.02 rows=244 width=67)
4  Sort Key: (sum(e.amount)) DESC
5  -> HashAggregate (cost=57.31..60.36 rows=244 width=67)
6  Group Key: a.customer_id, c.city, d.country
7  -> Nested Loop (cost=18.16..54.87 rows=244 width=41)
8  -> Hash Join (cost=17.88..37.14 rows=10 width=35)
9  Hash Cond: (c.country_id = d.country_id)
10 -> Nested Loop (cost=14.43..33.66 rows=10 width=28)
11 -> Hash Join (cost=14.15..29.77 rows=10 width=15)
12 Hash Cond: (b.city_id = c.city_id)
13 -> Seq Scan on address b (cost=0.00..14.03 rows=603 width=4)
14 -> Hash (cost=14.03..14.03 rows=10 width=15)
15 -> Seq Scan on city c (cost=0.00..14.03 rows=10 width=15)
16 Filter: ((city)::text = ANY ('Aurora,Atlixco,Pingxiang,Adoni,Dhule(Dhulia),Kirashiki,Xintai,Sivas,Celaya,Nezahualcoyotli')::text))
17 -> Index Scan using idx_fk_address_id on customer a (cost=0.28..0.38 rows=1 width=19)
18 Index Cond: (address_id = b.address_id)
19 -> Hash (cost=2.09..2.09 rows=109 width=13)
20 -> Seq Scan on country d (cost=0.00..2.09 rows=109 width=13)
21 -> Index Scan using idx_fk_customer_id on payment e (cost=0.29..1.53 rows=24 width=8)
22 Index Cond: (customer_id = a.customer_id)
```

Total rows: 22 of 22 Query complete 00:00:00.123 Ln 1, Col 8

76°F Partly sunny

12:09 PM 4/14/2023

Exercise 3.8.1

pgAdmin 4

File Object Tools Help

Browsers

- Servers (1)
 - PostgreSQL 15
 - Databases (1)
 - Rockbuster
 - Casts
 - Catalogs
 - Event Triggers
 - Extensions
 - Foreign Data Wrappers
 - Languages
 - Publications
 - Schemas
 - Subscriptions
 - postgres
 - Login/Group Roles
 - Tablespaces

Dashboard Properties SQL Statistics Dependencies Dependents Processes Rockbuster/postgres@PostgreSQL 15*

Query Query History

```
1 with payment_amt_cte (customer_id, amount) AS
2 (select A.customer_id,
3    A.amount,
4    B.first_name,
5    B.last_name
6   from payment A
7   inner join customer B ON A.customer_id = B.customer_id)
8 select
9   customer_id,
10  first_name,
11  last_name,
12  Avg(amount) AS average_amount_paid
13   from payment_amt_cte
14  group by customer_id,
15         first_name,
16         last_name
17  order by average_amount_paid DESC
18  limit 5;
```

Data Output Messages Notifications

customer_id	first_name	last_name	average_amount_paid
107	Bethany	May	5.0179027942794238
121	Karen	Schuler	5.5174117164708824
19	Ruth	Marlowe	5.4800000000000000
5	Linda	Williams	5.4493333333333330
211	Paul	Tsai	5.2800000000000000

Total rows: 5 of 5 Query complete 38:00:00.265 Ln 18, Col 4

76°F Mostly sunny 11:05 AM 4/14/2023

Exercise 3.9.1

pgAdmin 4

File Object Tools Help

Browsers

- Servers (1)
 - PostgreSQL 15
 - Databases (2)
 - Rockbuster
 - Casts
 - Catalogs
 - Event Triggers
 - Extensions
 - Foreign Data Wrappers
 - Languages
 - Publications
 - Schemas
 - Subscriptions
 - postgres
 - Login/Group Roles
 - Tablespaces
 - postgres

Dashboard Properties SQL Statistics Dependencies Dependents Processes Rockbuster/postgres@PostgreSQL 15*

Query Query History

```
1 Explain
2 With payment_amt_cte (customer_id, amount) AS
3 (select A.customer_id,
4    A.amount,
5    B.first_name,
6    B.last_name
7   from payment A
8   inner join customer B ON A.customer_id = B.customer_id)
9 select
10   customer_id,
11   first_name,
12   last_name,
13   Avg(amount) AS average_amount_paid
14   from payment_amt_cte
15  group by customer_id,
16         first_name,
17         last_name
18  order by average_amount_paid DESC
19  limit 5;
```

Data Output Messages Notifications

QUERY PLAN

text

```
1 Limit (cost=885.86..885.88 rows=5 width=47)
2  -> Sort (cost=885.86..922.35 rows=14596 width=47)
3    Sort Key: (avg(a.amount)) DESC
4    -> HashAggregate (cost=460.98..643.43 rows=14596 width=47)
5      Group Key: a.customer_id, b.first_name, b.last_name
6      -> Hash Join (cost=22.48..315.02 rows=14596 width=21)
7        Hash Cond: (a.customer_id = b.customer_id)
8        -> Seq Scan on payment a (cost=0.00..253.96 rows=14596 width=8)
9        -> Hash (cost=14.99..14.99 rows=599 width=17)
10       -> Seq Scan on customer b (cost=0.00..14.99 rows=599 width=17)
```

Total rows: 10 of 10 Query complete 00:00:00.188 Ln 18, Col 9

76°F Mostly sunny 11:06 AM 4/14/2023

Exercise 3.9.1

pgAdmin 4

File Object Tools Help

Browsers

- Servers (1)
 - PostgreSQL 15
 - Databases (2)
 - Rockbuster
 - casts
 - Catalogs
 - Event Triggers
 - Extensions
 - Foreign Data Wrappers
 - Languages
 - Publications
 - Schemas
 - Subscriptions
 - postgres
 - Login/Group Roles
 - Tablespaces

Dashboard Properties SQL Statistics Dependencies Dependents Processes **3.8 step 2.sql**

Query Query History

```

1 select distinct count(A.customer_id) AS all_customer_count,
2 D.country,
3 count(top_5_customers) AS top_customer_count
4 from customer A
5 inner join address B on A.address_id = B.address_id
6 inner join city C on B.city_id = C.city_id
7 inner join country D on C.city_id = D.country_id
8 left join
9 (select A.customer_id,
10 A.first_name,
11 A.last_name,
12 C.city,
13 D.country,
14 sum(e.amount) As payment_amt
15 from customer A
16 inner join address B on A.address_id = B.address_id
17 inner join city C on B.city_id = C.city_id
18 inner join country D on C.country_id = D.country_id
19 inner join payment E on A.customer_id = E.customer_id
20 where C.city IN ('Aurora',
21 'Atlixco',
22 'Pingziang',
23 'Adoni',
24 'Ouhle (Ouhlia)',
25 'Kirashiki',
26 'Xintal',
27 'Sivas',
28 'Celaya',
29 'Nezahualcyotl')
30
31 group by A.customer_id,
32 A.first_name,
33 A.last_name,
34 C.city,
35 D.country
36 order by payment_amt Desc
37 limit 5) AS top_5_customers
38 on A.customer_id = top_5_customers.customer_id
39 group by D.country
40 having count (top_5_customers)>0

```

Data Output Messages Notifications

all_customer_count	country	top_customer_count
1	Hong Kong	1
2	Australia	1
3	Sweden	1
4	Virgin Islands	1

Total rows: 4 of 4 Query complete 00:00:00.457 Ln 1, Col 1

Exercise 3.8.2

pgAdmin 4

File Object Tools Help

Browsers

- Servers (1)
 - PostgreSQL 15
 - Databases (2)
 - Rockbuster
 - casts
 - Catalogs
 - Event Triggers
 - Extensions
 - Foreign Data Wrappers
 - Languages
 - Publications
 - Schemas
 - Subscriptions
 - postgres
 - Login/Group Roles
 - Tablespaces

Dashboard Properties SQL Statistics Dependencies Dependents Processes **3.8 step 2.sql**

Query Query History

```

1 Explain
2 select distinct count(A.customer_id) AS all_customer_count,
3 D.country,
4 count(top_5_customers) AS top_customer_count
5 from customer A
6 inner join address B on A.address_id = B.address_id
7 inner join city C on B.city_id = C.city_id
8 inner join country D on C.city_id = D.country_id
9 left join
10 (select A.customer_id,
11 A.first_name,
12 A.last_name,
13 C.city,
14 D.country,
15 sum(e.amount) As payment_amt
16 from customer A
17 inner join address B on A.address_id = B.address_id
18 inner join city C on B.city_id = C.city_id
19 inner join country D on C.country_id = D.country_id
20 inner join payment E on A.customer_id = E.customer_id
21 where C.city IN ('Aurora',
22 'Atlixco',
23 'Pingziang',
24 'Adoni',
25 'Ouhle (Ouhlia)',
26 'Kirashiki',
27 'Xintal',
28 'Sivas',
29 'Celaya',
30 'Nezahualcyotl')
31 group by A.customer_id,
32 A.first_name,
33 A.last_name,
34 C.city,
35 D.country
36 order by payment_amt Desc
37 limit 5) AS top_5_customers
38 on A.customer_id = top_5_customers.customer_id
39 group by D.country
40 having count (top_5_customers)>0

```

Data Output Messages Notifications

QUERY PLAN

```

1 Unique (cost=120.48..120.84 rows=36 width=25)
2  -> Sort (cost=120.48..120.57 rows=36 width=25)
3    Sort Key: (count(top_5_customers *)), (count(a.customer_id)) DESC, d.country
4    -> HashAggregate (cost=118.19..119.55 rows=36 width=25)
5      Group Key: d.country
6      Filter: (count(top_5_customers) > 0)
7      -> Hash Left Join (cost=98.62..117.37 rows=109 width=72)
8        Hash Cond: (a.customer_id = top_5_customers.customer_id)
9        -> Hash Join (cost=34.08..52.41 rows=109 width=13)
10          Hash Cond: (a.address_id = b.address_id)
11          -> Seq Scan on customer a (cost=0.00..14.99 rows=599 width=6)
12          -> Hash (cost=32.71..32.71 rows=110 width=13)
13            -> Hash Join (cost=15.32..32.71 rows=110 width=13)
14              Hash Cond: (b.city_id = c.city_id)
15              -> Seq Scan on address b (cost=0.00..14.03 rows=603 width=6)
16              -> Hash (cost=13.96..13.96 rows=109 width=17)
17                Merge Join (cost=6.05..13.96 rows=109 width=17)
18                  Merge Cond: (c.city_id = d.country_id)
19                  -> Index Only Scan using city_pk on city c (cost=0.28..33.27 rows=600 width=4)
20                  -> Sort (cost=5.78..6.05 rows=109 width=13)
21                    Sort Key: d.country_id
22                    -> Seq Scan on country d (cost=0.00..2.09 rows=109 width=13)
23                    -> Hash (cost=64.48..64.48 rows=5 width=63)
24                      Subquery Scan on top_5_customers (cost=64.41..64.48 rows=5 width=63)
25                        Limit (cost=64.41..64.43 rows=5 width=67)
26                        -> Sort (cost=64.41..65.02 rows=244 width=67)
27                          Sort Key: (sum(e.amount)) DESC

```

Total rows: 45 of 45 Query complete 00:00:00.122 Ln 38, Col 47

Exercise 3.8.2

pgAdmin 4

File Object Tools Help

Browsers

- Servers (1)
 - PostgreSQL 15
 - Databases (2)
 - Rockbuster
 - Cast
 - Catalogs
 - Event Triggers
 - Extensions
 - Foreign Data Wrappers
 - Languages
 - Publications
 - Schemas
 - Subscriptions
 - postgres
 - Login/Group Roles
 - Tablespaces

Dashboard Properties SQL Statistics Dependencies Dependents Processes 3.8 step 2.sql*

Query Query History

```
Count(top_5_customers) AS top_customer_count
From customer A
Inner Join address B ON A.address_id = B.address_id
Inner Join city C ON B.city_id = C.city_id
Inner Join country D ON C.city_id = D.country_id
Left Join
(Select A.customer_id,
A.first_name,
A.last_name,
C.city,
D.country,
sum(e.amount) AS payment_amt
From customer A
Inner Join address B ON A.address_id = B.address_id
Inner Join city C ON B.city_id = C.city_id
Inner Join country D ON C.country_id = D.country_id
Inner Join payment E ON A.customer_id = E.customer_id
Where C.city IN ('Aurora',
'Atlixco',
'Pingtang',
'Adoni',
'Dhule (Dhulia)',
'Kirashiki',
'Xintai',
'Sivas',
'Celaya',
'Nezahualcotli'))
Group by A.customer_id,
A.first_name,
A.last_name,
C.city,
D.country
Order by payment_amt Desc
Limit 5) AS top_5_customers
On A.customer_id = top_5_customers.customer_id
Group By D. Country
Having Count (top_5_customers)>0
Order by count(top_5_customers)
count(A.customer_id) DESC
```

Query Plan

19 -> Index Only Scan using city_pk on city c (cost=0.28..33.27 rows=600 width=4)

20 -> Sort (cost=5.78..6.05 rows=109 width=13)

21 Sort Key: d.country_id

22 -> Seq Scan on country d (cost=0.00..2.09 rows=109 width=13)

23 -> Hash (cost=64.48..64.48 rows=5 width=63)

24 -> Subquery Scan on top_5_customers (cost=64.41..64.48 rows=5 width=63)

25 -> Limit (cost=64.41..64.43 rows=5 width=67)

26 -> Sort (cost=64.41..65.02 rows=244 width=67)

27 Sort Key: (sum(e.amount)) DESC

28 -> HashAggregate (cost=57.31..60.36 rows=244 width=67)

29 Group Key: a_1.customer_id, c_1.city, d_1.country

30 -> Nested Loop (cost=18.16..37.14 rows=10 width=41)

31 -> Hash Join (cost=17.88..37.14 rows=10 width=35)

32 Hash Cond: (c_1.country_id = d_1.country_id)

33 -> Nested Loop (cost=14.43..33.66 rows=10 width=28)

34 -> Hash Join (cost=14.15..29.77 rows=10 width=15)

35 Hash Cond: (b_1.city_id = c_1.city_id)

36 -> Seq Scan on address b_1 (cost=0.00..14.03 rows=603 width=6)

37 -> Hash (cost=14.03..14.03 rows=10 width=15)

38 -> Seq Scan on city c_1 (cost=0.03..14.03 rows=10 width=15)

39 Filter: ((city).text = ANY ('Aurora,Atlixco,Pingtang,Adoni,Dhule (Dhulia),Kirashiki,Xintai,Sivas,Celaya,Nezahualcotli'))

40 -> Index Scan using idx_fk_address_id on customer a_1 (cost=0.28..0.38 rows=1 width=19)

41 Index Cond: (address_id = b_1.address_id)

42 -> Hash (cost=2.09..2.09 rows=109 width=13)

43 -> Seq Scan on country d_1 (cost=0.00..2.09 rows=109 width=13)

44 -> Index Scan using idx_fk_customer_id on payment e (cost=0.29..1.53 rows=24 width=8)

45 Index Cond: (customer_id = a_1.customer_id)

Total rows: 45 of 45 Query complete 00:00:00.122

Ln 38, Col 47

74°F Mostly sunny

Search

11:53 AM 4/14/2023

Exercise 3.8.2

pgAdmin 4

File Object Tools Help

Browsers

- Servers (1)
 - PostgreSQL 15
 - Databases (2)
 - Rockbuster
 - Cast
 - Catalogs
 - Event Triggers
 - Extensions
 - Foreign Data Wrappers
 - Languages
 - Publications
 - Schemas
 - Subscriptions
 - postgres
 - Login/Group Roles
 - Tablespaces

Dashboard Properties SQL Statistics Dependencies Dependents Processes 3.9.2.sql*

Query Query History

```
With payment_amt_cte (customer_id, amount, first_name, last_name, country) AS
(Select A.customer_id,
A.amount,
B.first_name,
B.last_name,
E.country
From payment A
Inner Join customer B ON A.customer_id = B.customer_id
Inner Join address C ON B.address_id = C.address_id
Inner Join city D ON C.city_id = D.city_id
Inner Join country E ON D.country_id = E.country_id)
Select
customer_id,
first_name,
last_name,
Avg(amount) AS average_amount_paid,
country
From payment_amt_cte
Group by
customer_id,
first_name,
last_name,
country
Order by average_amount_paid DESC
Limit 5;
```

Data Output

customer_id	first_name	last_name	average_amount_paid	country
1	187	Brittany	Riley	Ukraine
2	321	Kevin	Schuler	Nepal
3	19	Ruth	Martinez	South Africa
4	3	Linda	Williams	Greece
5	311	Paul	Trout	Russian Federation

Total rows: 5 of 5 Query complete 00:00:00.130

Ln 5, Col 15

74°F Mostly sunny

Search

11:34 AM 4/14/2023

Exercise 3.9.2

pgAdmin 4

File Object Tools Help

Dashboard Properties SQL Statistics Dependencies Dependents Processes 3.9.2.sql*

Rockbuster/postgres@PostgreSQL 15

Query Query History

```
1 Explain
2 With payment_amt_cte (customer_id, amount, first_name, last_name, country) AS
3 (Select A.customer_id,
4      A.amount,
5      A.first_name,
6      A.last_name,
7      A.country
8   From Rockbuster.payment_amt A)
9 Select *
10 From payment_amt_cte
```

Data Output Messages Notifications

QUERY PLAN

text

```
1 Limit (cost=1082.84..1082.85 rows=5 width=56)
2  -> Sort (cost=1082.84..1119.33 rows=14596 width=56)
3    Sort Key: (avg(a.amount)) DESC
4    -> HashAggregate (cost=657.96..840.41 rows=14596 width=56)
5      Group Key: a.customer_id, b.first_name, b.last_name, e.country
6      -> Hash Join (cost=66.00..475.51 rows=14596 width=30)
7        Hash Cond: (d.country_id = e.country_id)
8        -> Hash Join (cost=62.55..432.25 rows=14596 width=23)
9          Hash Cond: (c.city_id = d.city_id)
10          -> Hash Join (cost=44.05..375.17 rows=14596 width=23)
11            Hash Cond: (b.address_id = c.address_id)
12            -> Hash Join (cost=22.48..315.02 rows=14596 width=23)
13              Hash Cond: (a.customer_id = b.customer_id)
14              -> Seq Scan on payment a (cost=0.00..253.96 rows=14596 width=8)
15                -> Hash (cost=14.99..14.99 rows=599 width=19)
16                -> Seq Scan on customer b (cost=0.00..14.99 rows=599 width=19)
17                -> Hash (cost=14.03..14.03 rows=603 width=6)
18                -> Seq Scan on address c (cost=0.00..14.03 rows=603 width=6)
19                -> Hash (cost=11.00..11.00 rows=600 width=6)
20                -> Seq Scan on city d (cost=0.00..11.00 rows=600 width=6)
21                -> Hash (cost=2.09..2.09 rows=109 width=13)
22                -> Seq Scan on country e (cost=0.00..2.09 rows=109 width=13)
```

Total rows: 22 of 22 Query complete 00:00:00.155 Ln 24, Col 9

74°F Mostly sunny 11:39 AM 4/14/2023

Exercise 3.9.2