

Object Explorer Servers (1) PostgresSQL 16 Databases (4)

> Demo demo greencycles Casts Catalogs Event Triggers Extensions Foreign Data Wrappers Languages Publications Schemas (1) public

greencycles/postgres@PostgreSQL 16

greencycles/postgres@... demo/postgres@... greencycles/postgres@PostgreSQL 16*

Dependencies Dependents Processes greencycles/postgres@PostgreSQL 16*

Query Query History

1 /*Question 1: Level: Simple
2 Topic: DISTINCT
3 Task: Create a list of all the different (distinct) replacement costs of the films.
4 Question: What's the lowest replacement cost?*/
5
6 SELECT DISTINCT replacement_cost
7 FROM film
8 ORDER BY 1

Scratch Pad

Data Output Messages Notifications

replacement_cost

	replacement_cost
1	9.99
2	10.99
3	11.99
4	12.99
5	13.99
6	14.99
7	15.99
8	16.99

Total rows: 21 of 21 Query complete 00:00:01 508

File Object Tools Help

Line 8, Col 11

9:13 AM 3/11/2022

File Object Tools Help

Object Explorer Servers (1) PostgreSQL 16 Databases (4)

Demo demo greencycles/postgres@PostgreSQL 16* greencycles/postgres@... greencycles/postgres@PostgreSQL 16* greencycles/postgres@... greencycles/postgres@PostgreSQL 16*

Dependencies X Dependents X Processes X greencycles/postgres@PostgreSQL 16* X No limit ▾

Query History

Scratch Pad X

```

1 /*Question 2:
2 Level: Moderate, Topic: CASE + GROUP BY
3 Task: Write a query that gives an overview of how many films have replacement costs
4 low: 9.99 - 19.99, medium: 20.00 - 24.99, high: 25.00 - 29.99
5 Question: How many films have a replacement cost in the "low" group?*/
6 SELECT
7 CASE
8 WHEN replacement_cost BETWEEN 9.99 AND 19.99
9 THEN 'low' WHEN replacement_cost BETWEEN 20 AND 24.99
10 THEN 'medium' ELSE 'high' END as cost_range,
11 COUNT(*) FROM film GROUP BY cost_range

```

Data Output Messages Notifications

	cost_range	count
1	medium	250
2	high	236
3	low	514

Total rows: 3 of 3 Query complete 00:00:00 2023

Object Explorer

:

- ✓ Servers (1)
- ✓ PostgreSQL 16
- ✓ Databases (4)

- > Demo
- > demo

- > greencycles
- > Casts
- > Catalogs
- > Event Triggers
- > Extensions
- > Foreign Data Wrappers
- > Languages
- > Publications
- > Schemas (1)

Query Query History

greencycles/postgres@PostgreSQL 16

No limit

Scratch Pad

```

1
2 /*Level: Moderate
3 Topic: JOIN & GROUP BY
4 Task: Create an overview of how many movies (titles) there are in each category (name)
5 Question: Which category (name) is the most common among the films?*/
6
7 SELECT
8     name, COUNT(title) FROM film f
9 INNER JOIN film_category fc ON f.film_id=fc.film_id
10 INNER JOIN category c ON c.category_id=fc.category_id
11 GROUP BY name ORDER BY 2 DESC

```

Dependencies X

Dependents X

Processes X

greencycles/postgres@PostgreSQL 16*

demo/postgres@... X

greencycles/postgres@PostgreSQL 16* X

greencycles/postgres@PostgreSQL 16* X


Data Output Messages Notifications

≡+ ↗ ↘ ↕ ↛ ↝ ↞

	name	count
	text	bigint

- > Aggregates
- > Collations
- > Domains
- > FTS Configuration
- > FTS Dictionaries
- > FTS Parsers
- > FTS Templates
- > Foreign Tables
- > Functions
- > Materialized View
- > Operators
- > Procedures
- > Sequences

Total rows: 16 of 16 Query complete 00:00:00.000 256



File Object Tools Help

Object Explorer

- Servers (1)
 - PostgreSQL 16
- Databases (4)
 - Demo
 - demo
 - greencycles
 - public
- Catalogs
- Event Triggers
- Extensions
- Foreign Data Wrappers
- Languages
- Publications
- Schemas (1)

Total rows: 4 of 4 Query complete 00:00:00.500

Dependencies X

Dependents X

Processes X

greencycles/postgres@PostgreSQL 16*

greencycles/postgres@PostgreSQL 16* demo/postgres@... greencycles/postgres@PostgreSQL 16* demo/postgres@...

Query History

```

1 /*Question 6:
2 Level: Moderate, Topic: LEFT JOIN & FILTERING
3 Task: Create an overview of the addresses that are not associated to any customer.
4 Question: How many addresses are that?*/
5
6 SELECT * FROM address a
7 LEFT JOIN customer c
8 ON c.address_id = a.address_id
9 WHERE c.first_name is null
  
```

Scratch Pad X

Data Output Messages Notifications

	address_id	address	address2	district	city_id	postal_code	phone	last_update	customer_id	store_id
1	2	28 MySQL Bouleva...	[null]	QLD	576	6172235589	2020-02-15 04:45:30-05	2020-02-15 04:45:30-05	1	4
2	4	1411 Lillydale Drive	[null]	QLD	576	14033335568	2020-02-15 04:45:30-05	2020-02-15 04:45:30-05	1	3
3	1	47 MySakila Drive	[null]	Alberta	300	14033335568	2020-02-15 04:45:30-05	2020-02-15 04:45:30-05	2	3
4	3	23 Workhaven Lane	[null]	Alberta	300	14033335568	2020-02-15 04:45:30-05	2020-02-15 04:45:30-05	3	4

38°F

Operators

Procedures

Sequences

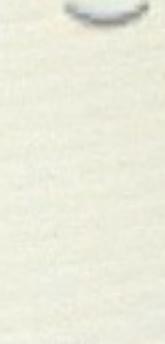
Materialized View

9:38 AM
3/11/2024

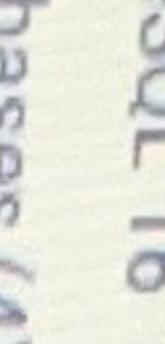
Object Explorer



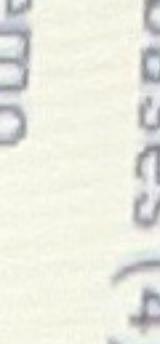
Servers (1)



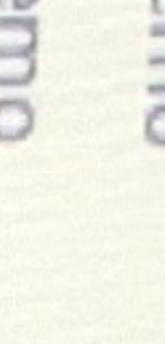
PostgreSQL 16



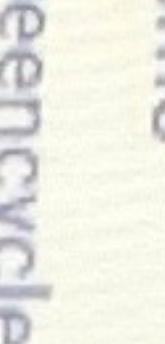
Databases (4)



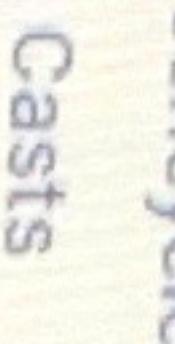
Demo



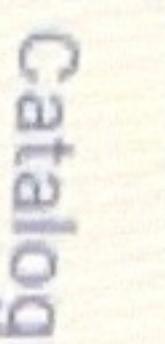
demo



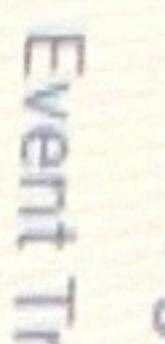
greencycles



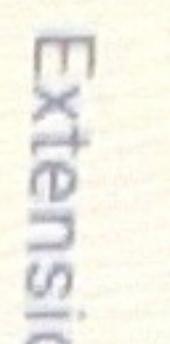
greencycles



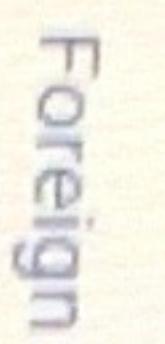
Catalogs



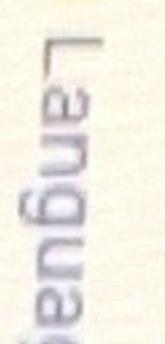
Event Triggers



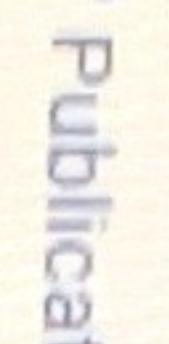
Extensions



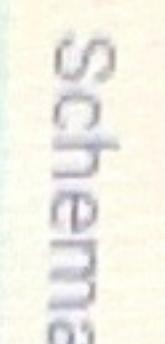
Foreign Data Wrappers



Languages



Publications



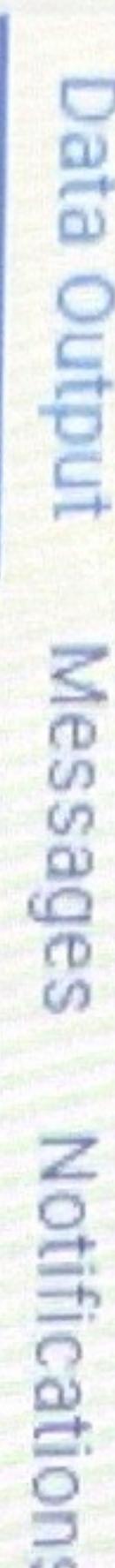
Schemas (1)



public



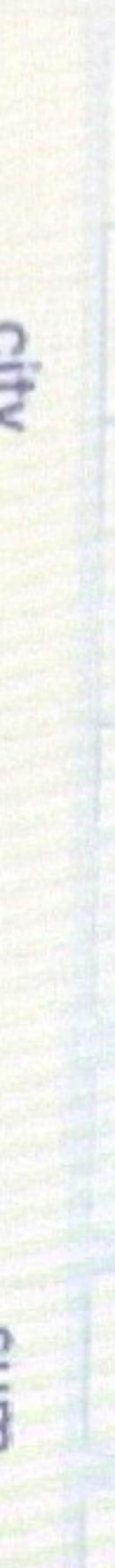
Data Output Messages Notifications



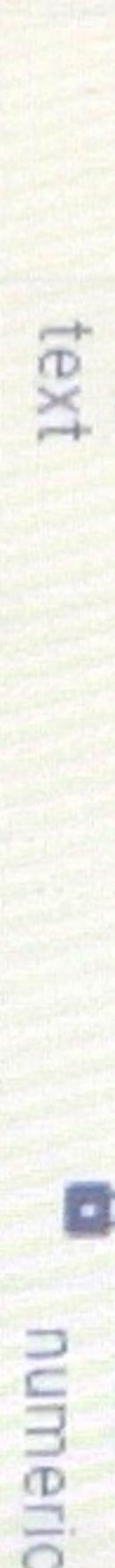
Data Output Messages Notifications



Data Output Messages Notifications



Data Output Messages Notifications



Data Output Messages Notifications



Data Output Messages Notifications



Data Output Messages Notifications



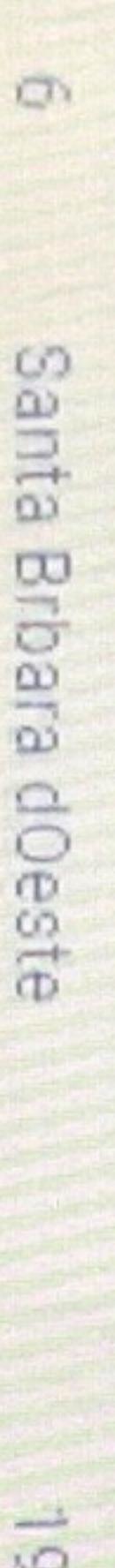
Data Output Messages Notifications



Data Output Messages Notifications



Data Output Messages Notifications



Data Output Messages Notifications



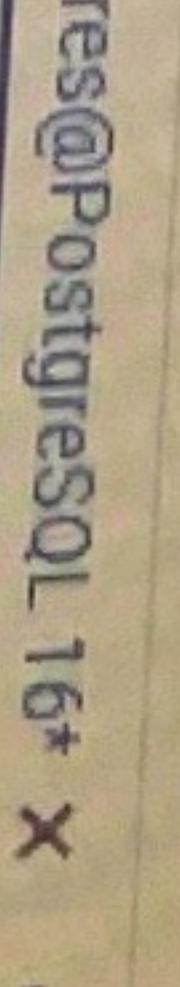
Data Output Messages Notifications



greencycles/postgres@PostgreSQL 16



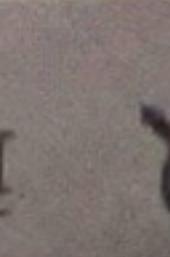
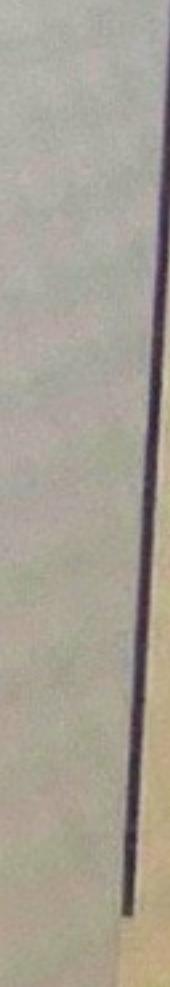
greencycles/postgres@PostgreSQL 16*



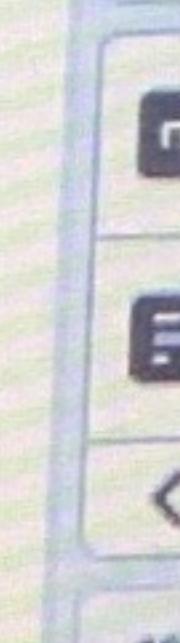
demo/postgres@...



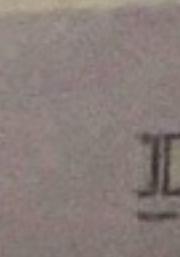
demo/postgres@...



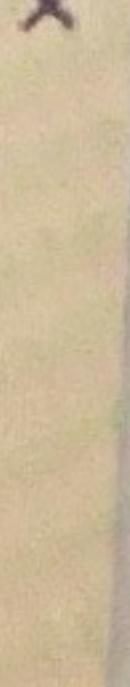
Query History



Scratch Pad



Scratch Pad



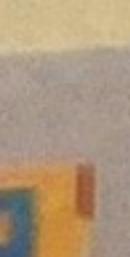
Scratch Pad



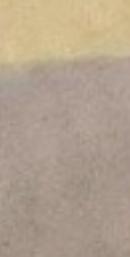
Scratch Pad



Scratch Pad



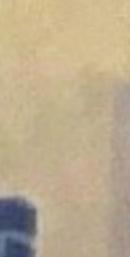
Scratch Pad



Scratch Pad



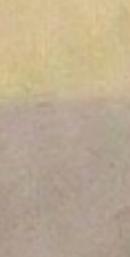
Scratch Pad



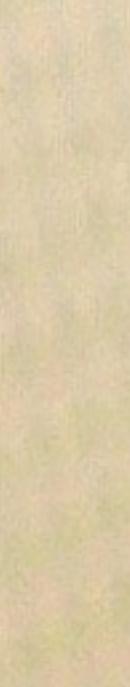
Scratch Pad



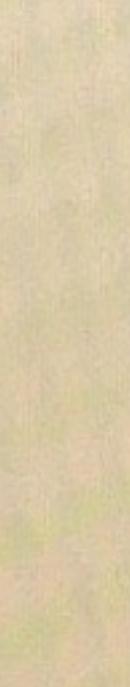
Scratch Pad



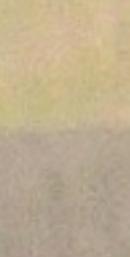
Scratch Pad



Scratch Pad



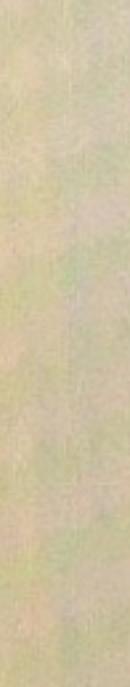
Scratch Pad



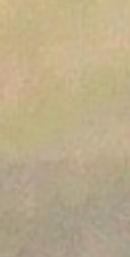
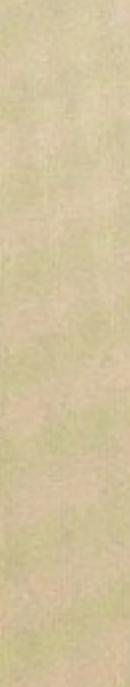
Scratch Pad



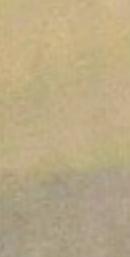
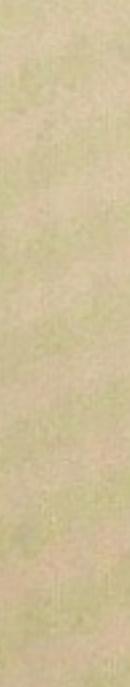
Scratch Pad



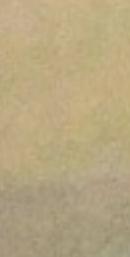
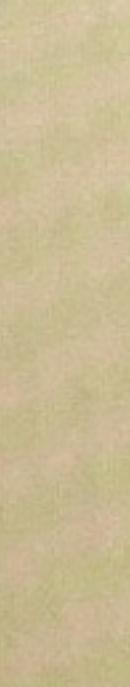
Scratch Pad



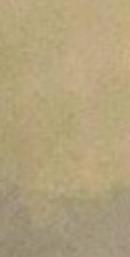
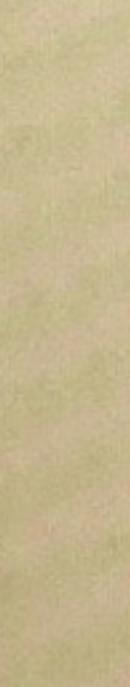
Scratch Pad



Scratch Pad



Scratch Pad



Scratch Pad



Scratch Pad



/*Question 7:
 Level: Moderate, Topic: JOIN & GROUP BY
 Task: Create the overview of the sales to determine the from which city
 (we are interested in the city in which the customer lives, not where the store is)
 Question: What city is that and how much is the amount?
 Answer: Cape Coral with a total amount of 221.55*/

```
SELECT city, SUM(amount) FROM payment p LEFT JOIN customer c
ON p.customer_id=c.customer_id LEFT JOIN address a
ON a.address_id=c.address_id
LEFT JOIN city ci ON ci.city_id=a.city_id GROUP BY city ORDER BY 2 DESC
```

```
1
2
3
4
5
6
7
8
9
10
11
```

```
SELECT city, SUM(amount) FROM payment p LEFT JOIN customer c
ON p.customer_id=c.customer_id LEFT JOIN address a
ON a.address_id=c.address_id
LEFT JOIN city ci ON ci.city_id=a.city_id GROUP BY city ORDER BY 2 DESC
```

```
1
2
3
4
5
6
7
8
9
10
11
```

```
SELECT city, SUM(amount) FROM payment p LEFT JOIN customer c
ON p.customer_id=c.customer_id LEFT JOIN address a
ON a.address_id=c.address_id
LEFT JOIN city ci ON ci.city_id=a.city_id GROUP BY city ORDER BY 2 DESC
```

```
1
2
3
4
5
6
7
8
9
10
11
```

```
SELECT city, SUM(amount) FROM payment p LEFT JOIN customer c
ON p.customer_id=c.customer_id LEFT JOIN address a
ON a.address_id=c.address_id
LEFT JOIN city ci ON ci.city_id=a.city_id GROUP BY city ORDER BY 2 DESC
```

```
1
2
3
4
5
6
7
8
9
10
11
```

```
SELECT city, SUM(amount) FROM payment p LEFT JOIN customer c
ON p.customer_id=c.customer_id LEFT JOIN address a
ON a.address_id=c.address_id
LEFT JOIN city ci ON ci.city_id=a.city_id GROUP BY city ORDER BY 2 DESC
```

```
1
2
3
4
5
6
7
8
9
10
11
```

```
SELECT city, SUM(amount) FROM payment p LEFT JOIN customer c
ON p.customer_id=c.customer_id LEFT JOIN address a
ON a.address_id=c.address_id
LEFT JOIN city ci ON ci.city_id=a.city_id GROUP BY city ORDER BY 2 DESC
```

```
1
2
3
4
5
6
7
8
9
10
11
```

```
SELECT city, SUM(amount) FROM payment p LEFT JOIN customer c
ON p.customer_id=c.customer_id LEFT JOIN address a
ON a.address_id=c.address_id
LEFT JOIN city ci ON ci.city_id=a.city_id GROUP BY city ORDER BY 2 DESC
```

```
1
2
3
4
5
6
7
8
9
10
11
```

```
SELECT city, SUM(amount) FROM payment p LEFT JOIN customer c
ON p.customer_id=c.customer_id LEFT JOIN address a
ON a.address_id=c.address_id
LEFT JOIN city ci ON ci.city_id=a.city_id GROUP BY city ORDER BY 2 DESC
```

```
1
2
3
4
5
6
7
8
9
10
11
```

```
SELECT city, SUM(amount) FROM payment p LEFT JOIN customer c
ON p.customer_id=c.customer_id LEFT JOIN address a
ON a.address_id=c.address_id
LEFT JOIN city ci ON ci.city_id=a.city_id GROUP BY city ORDER BY 2 DESC
```

```
1
2
3
4
5
6
7
8
9
10
11
```

```
SELECT city, SUM(amount) FROM payment p LEFT JOIN customer c
ON p.customer_id=c.customer_id LEFT JOIN address a
ON a.address_id=c.address_id
LEFT JOIN city ci ON ci.city_id=a.city_id GROUP BY city ORDER BY 2 DESC
```

```
1
2
3
4
5
6
7
8
9
10
11
```

```
SELECT city, SUM(amount) FROM payment p LEFT JOIN customer c
ON p.customer_id=c.customer_id LEFT JOIN address a
ON a.address_id=c.address_id
LEFT JOIN city ci ON ci.city_id=a.city_id GROUP BY city ORDER BY 2 DESC
```

```
1
2
3
4
5
6
7
8
9
10
11
```

```
SELECT city, SUM(amount) FROM payment p LEFT JOIN customer c
ON p.customer_id=c.customer_id LEFT JOIN address a
ON a.address_id=c.address_id
LEFT JOIN city ci ON ci.city_id=a.city_id GROUP BY city ORDER BY 2 DESC
```

```
1
2
3
4
5
6
7
8
9
10
11
```

```
SELECT city, SUM(amount) FROM payment p LEFT JOIN customer c
ON p.customer_id=c.customer_id LEFT JOIN address a
ON a.address_id=c.address_id
LEFT JOIN city ci ON ci.city_id=a.city_id GROUP BY city ORDER BY 2 DESC
```

```
1
2
3
4
5
6
7
8
9
10
11
```

```
SELECT city, SUM(amount) FROM payment p LEFT JOIN customer c
ON p.customer_id=c.customer_id LEFT JOIN address a
ON a.address_id=c.address_id
LEFT JOIN city ci ON ci.city_id=a.city_id GROUP BY city ORDER BY 2 DESC
```

```
1
2
3
4
5
6
7
8
9
10
11
```

```
SELECT city, SUM(amount) FROM payment p LEFT JOIN customer c
ON p.customer_id=c.customer_id LEFT JOIN address a
ON a.address_id=c.address_id
LEFT JOIN city ci ON ci.city_id=a.city_id GROUP BY city ORDER BY 2 DESC
```

```
1
2
3
4
5
6
7
8
9
10
11
```

```
SELECT city, SUM(amount) FROM payment p LEFT JOIN customer c
ON p.customer_id=c.customer_id LEFT JOIN address a
ON a.address_id=c.address_id
LEFT JOIN city ci ON ci.city_id=a.city_id GROUP BY city ORDER BY 2 DESC
```

```
1
2
3
4
5
6
7
8
9
10
11
```

```
SELECT city, SUM(amount) FROM payment p LEFT JOIN customer c
ON p.customer_id=c.customer_id LEFT JOIN address a
ON a.address_id=c.address_id
LEFT JOIN city ci ON ci.city_id=a.city_id GROUP BY city ORDER BY 2 DESC
```

```
1
2
3
4
5
6
7
8
9
10
11
```

```
SELECT city, SUM(amount) FROM payment p LEFT JOIN customer c
ON p.customer_id=c.customer_id LEFT JOIN address a
ON a.address_id=c.address_id
LEFT JOIN city ci ON ci.city_id=a.city_id GROUP BY city ORDER BY 2 DESC
```

```
1
2
3
4
5
6
7
8
9
10
11
```

```
SELECT city, SUM(amount) FROM payment p LEFT JOIN customer c
ON p.customer_id=c.customer_id LEFT JOIN address a
ON a.address_id=c.address_id
LEFT JOIN city ci ON ci.city_id=a.city_id GROUP BY city ORDER BY 2 DESC
```

```
1
2
3
4
5
6
7
8
9
10
11
```

```
SELECT city, SUM(amount) FROM payment p LEFT JOIN customer c
ON p.customer_id=c.customer_id LEFT JOIN address a
ON a.address_id=c.address_id
LEFT JOIN city ci ON ci.city_id=a.city_id GROUP BY city ORDER BY 2 DESC
```

```
1
2
3
4
5
6
7
8
9
10
11
```

```
SELECT city, SUM(amount) FROM payment p LEFT JOIN customer c
ON p.customer_id=c.customer_id LEFT JOIN address a
ON a.address_id=c.address_id
LEFT JOIN city ci ON ci.city_id=a.city_id GROUP BY city ORDER BY 2 DESC
```

```
1
2
3
4
5
6
7
8
9
10
11
```

```
SELECT city, SUM(amount) FROM payment p LEFT JOIN customer c
ON p.customer_id=c.customer_id LEFT JOIN address a
ON a.address_id=c.address_id
LEFT JOIN city ci ON ci.city_id=a.city_id GROUP BY city ORDER BY 2 DESC
```

```
1
2
3
4
5
6
7
8
9
10
11
```

Object Explorer File Object Tools Help

Servers (1) PostgreSQL 16 Databases (4) Demo demo greencycles/postgres@PostgreSQL 16* greencycles/postgres@PostgreSQL 16*

Dependencies Dependents Processes greencycles/postgres@PostgreSQL 16* demo/postgres@... greencycles/postgres@PostgreSQL 16*

Query Query History Scratch Pad

```

1 /*Question 8:
2 Level: Moderate to difficult, Topic: JOIN & GROUP BY
3 Task: Create an overview of the revenue (sum of amount) grouped by a column in the f
4 Question: Which country, city has the least sales?
5 Answer: United States, Tallahassee with a total amount of 50.85.*/
6
7 SELECT country || ', ' || city,
8 SUM(amount) FROM payment p LEFT JOIN customer c ON p.customer_id=c.customer_id
9 LEFT JOIN address a ON a.address_id=c.address_id LEFT JOIN city ci
10 ON ci.city_id=a.city_id LEFT JOIN country co ON co.country_id=ci.country_id
11 GROUP BY country || ', ' || city ORDER BY 2 ASC

```

Data Output Messages Notifications

	?column?	sum
		numeric
1	United States, Tallahassee	50.85
2	China, Fuzhou	50.86
3	Poland, Bydgoszcz	52.88
4	Sudan, al-Qadarif	57.81
5	Mozambique, Tete	58.82
6	Switzerland, Basel	58.86
7	China, Enshi	59.86
8	India, Hubli-Dharwad	62.85

Total rows: 507 of 507 Query complete 00:00:00 145

1010 Col 16

Object Tools Help

Servers (1)

- PostgreSQL 16
- Databases (4)
 - Demo
 - demo
 - greencycles
 - Casts
 - Catalogs
 - Event Triggers
 - Extensions
 - Foreign Data Wrappers
 - Languages
 - Publications
 - Schemas (1)
 - public

Dependencies X Dependents X Processes X

greencycles/postgres@PostgreSQL 16 X demo/postgres@... X greencycles/postgres@PostgreSQL 16* X

Query History

Query

```

1 /*Question 9:
2 Level: Difficult, Topic: Uncorrelated subquery
3 Task: Create a list with the average of the sales amount each staff_id has per customer
4 Question: Which staff_id makes on average more revenue per customer?
5 Answer: staff_id 2 with an average revenue of 56.64 per customer...*/
6
7 SELECT staff_id, ROUND(AVG(total),2) as avg_amount
8 FROM (SELECT SUM(amount) as total, customer_id, staff_id
9 FROM payment GROUP BY customer_id, staff_id) a
10 GROUP BY staff_id
  
```

Scratch Pad X

Data Output Messages Notifications

	staff_id	avg_amount
1	smallint	numeric
2	2	56.64
1	1	55.91

40°F

40°F


9:53 AM 3/17/2024

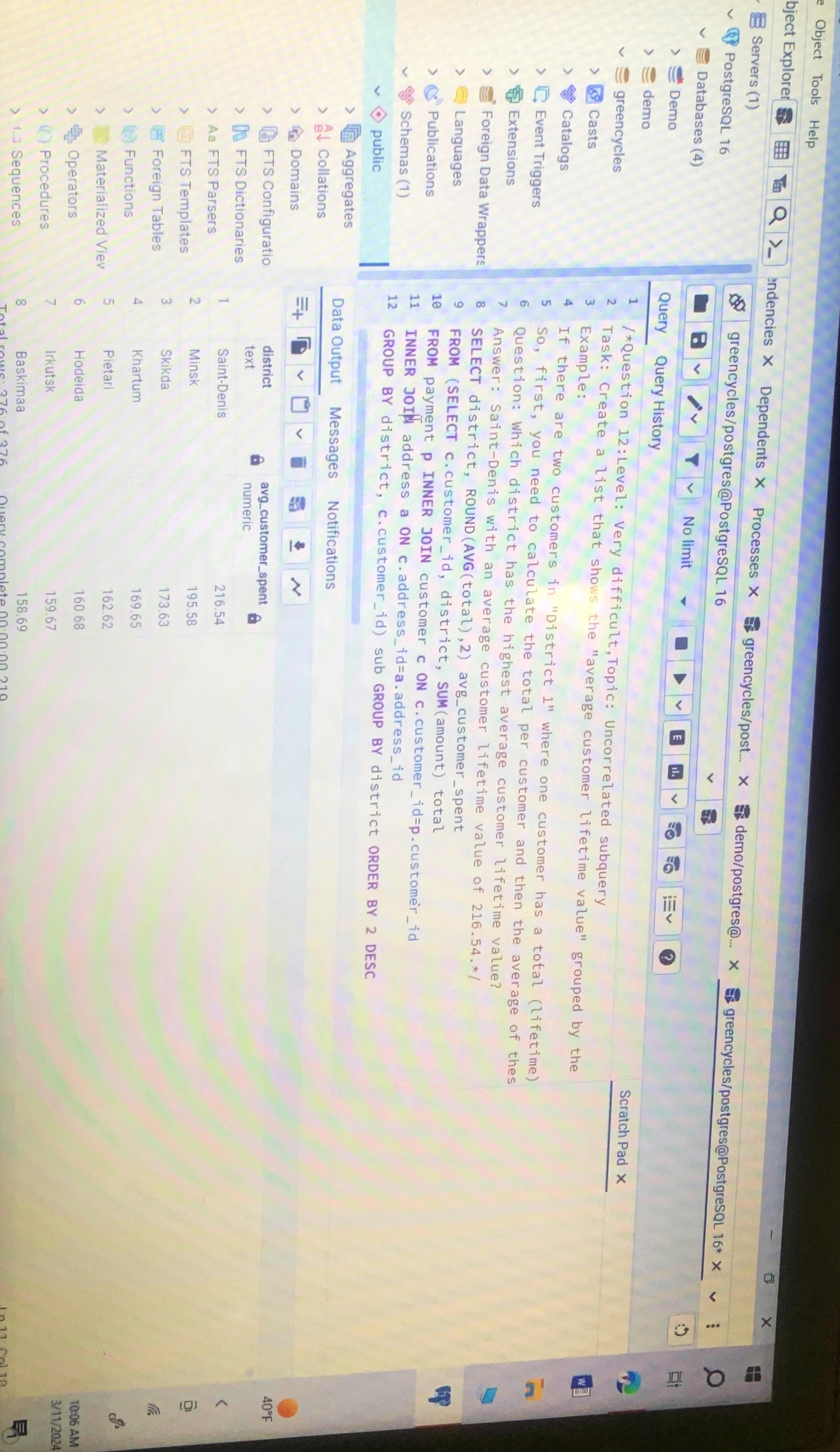
The screenshot shows the pgAdmin 4 interface with the following details:

- File**, **Object**, **Tools**, **Help** menu at the top.
- Object Explorer** sidebar on the left listing databases, schemas, and objects like 'public' and 'demo'.
- Databases** section expanded, showing 'greencycles' and 'demo'.
- Query History** tab active, displaying a numbered list of previous queries:

 - /*Level: Difficult to very difficult
 - Topic: EXTRACT + Uncorrelated subquery
 - Task: Create a query that shows average daily revenue of all Sundays.
 - Question: What is the daily average revenue of all Sundays?
 - Answer: 1410.65*/
 - SELECT AVG(total)
 - FROM (SELECT SUM(amount) as total, DATE(payment_date), EXTRACT(dow from payment_date))
 - FROM payment WHERE EXTRACT(dow from payment_date)=0
 - GROUP BY DATE(payment_date), weekday) daily

- Scratch Pad** tab is also visible.
- Bottom status bar shows 'Total rows: 1 of 1' and 'Query complete 00:00:00.00 123'.





Object Explorer Servers (1) Postgres (16) Databases (4)

greencycles/postgres@PostgreSQL 16* greencycles/postgres@... demo/postgres@... greencycles/postgres@PostgreSQL 16*

Dependencies X Dependents X Processes X greencycles/postgres@... X demo/postgres@... X greencycles/postgres@PostgreSQL 16*

greencycles/postgres@PostgreSQL 16

No limit ▾ □ ▾ ▾ ▾ ▾ ▾ ▾ ▾ ▾ ▾ ▾ ▾ ▾ ▾ ▾ ▾ ▾ ▾

Query History

2 Task: Create a list that shows the "average customer lifetime value" grouped by the different districts.

3 Example: If there are two customers in "District 1" where one customer has a total (lifetime) spent of \$1000 and the second customer has a total spent of \$2000 then the "average customer lifetime spent" in this district is \$1500.

4 So, first, you need to calculate the total per customer and then the average of these totals per district.

5 Question: Which district has the highest average customer lifetime value?

6 Answer: Saint-Denis with an average customer lifetime value of 216.54.* /

7

```

8 SELECT title, amount, name, payment_id, (SELECT SUM(amount) FROM payment p LEFT JOIN rental r
9 ON r.rental_id=p.rental_id LEFT JOIN inventory i ON i.inventory_id=r.inventory_id LEFT JOIN film_category fc ON fc.film_id=f.film_id LEFT JOIN category c1
10 ON c1.category_id=fc.category_id WHERE c1.name=c.name) FROM payment p LEFT JOIN rental r ON r.rental_id=p.rental_id
11 LEFT JOIN inventory i ON i.inventory_id=r.inventory_id LEFT JOIN film f ON f.film_id=i.film_id LEFT JOIN category c ON c.category_id=fc.category_id ORDER BY name
12
13

```

8 REAR TRADING 3.99 Action 18998 4375.85

9 Total rows: 1000 of 16049 Query complete 00:02:47 101

Data Output Messages Notifications

title amount name paymentId sum

	text	numeric (5,2)	text	integer	numeric
1	SUSPECTS QUILLS	4.99	Action	30011	4375.85
2	DEVIL DESIRE	4.99	Action	31822	4375.85
3	CASUALTIES ENCINO	8.99	Action	18149	4375.85
4	DREAM PICKUP	2.99	Action	18152	4375.85
5	REAR TRADING	0.99	Action	30545	4375.85
6	SHRUNK DIVINE	2.99	Action	19027	4375.85
7	PATRIOT ROMAN	3.99	Action	19002	4375.85
8	REAR TRADING	3.99	Action	18998	4375.85

Wind a. < > ⏪ ⏩

10:16 AM 3/11/2022