

pgAdmin 4

File Object Tools Help

Object Explorer

- public
 - Aggregates
 - Collations
 - Domains
 - FTS Configurations
 - FTS Dictionaries
 - FTS Parsers
 - FTS Templates
 - Foreign Tables
 - Functions
 - Materialized Views
 - Operators
 - Procedures
 - Sequences
 - Tables (11)**
 - album
 - artist
 - customer
 - employee
 - genre
 - invoice
 - invoice_line
 - media_type
 - playlist
 - playlist_track
 - track
 - Trigger Functions

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

music_database/postgres@PostgreSQL 15

Query Query History Scratch Pad x

```
1 Question 1 : Who is the senior most employee based on job titles ?
2
3
4 SELECT * FROM employee
5 ORDER BY levels desc
6 LIMIT 1 ;
```

Data Output Messages Notifications

	employee_id [PK] character varying (50)	last_name character	first_name character	title character varying (50)	reports_to character varying (30)	levels character varying (10)
1	9	Madan	Mohan	Senior General Manager	[null]	L7

Total rows: 1 of 1 Query complete 00:00:00.142 Ln 6, Col 11

pgAdmin 4

File Object Tools Help

Object Explorer

- public
 - Aggregates
 - Collations
 - Domains
 - FTS Configurations
 - FTS Dictionaries
 - FTS Parsers
 - FTS Templates
 - Foreign Tables
 - Functions
 - Materialized Views
 - Operators
 - Procedures
 - Sequences
 - Tables (11)**
 - album
 - artist
 - customer
 - employee
 - genre
 - invoice
 - invoice_line
 - media_type
 - playlist
 - playlist_track
 - track
 - Trigger Functions

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

music_database/postgres@PostgreSQL 15

Query Query History Scratch Pad x

```
1 Question 2 : Which top 3 countries have the most invoices ?
2
3 SELECT COUNT(*) AS maxi_count , billing_country
4 FROM invoice
5 GROUP BY billing_country
6 ORDER BY maxi_count desc
7 LIMIT 3 ;
```

Data Output Messages Notifications

	maxi_count bigint	billing_country character varying (30)
1	131	USA
2	76	Canada
3	61	Brazil

Total rows: 3 of 3 Query complete 00:00:00.203 Ln 7, Col 11

pgAdmin 4

File Object Tools Help

Object Explorer

- Aggregates
- Collations
- Domains
- FTS Configurations
- FTS Dictionaries
- FTS Parsers
- FTS Templates
- Foreign Tables
- Functions
- Materialized Views
- Operators
- Procedures
- Sequences
- Tables (11)
 - album
 - artist
 - customer
 - employee
 - genre
 - invoice
 - invoice_line
 - media_type
 - playlist
 - playlist_track
 - track
- Trigger Functions

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

music_database/postgres@PostgreSQL 15

Query Query History

```

1 Question 3 : What are top 10 values of total invoices ?
2
3 SELECT total
4 FROM invoice
5 ORDER BY total desc
6 LIMIT 10 ;

```

Data Output Messages Notifications

	total
1	23.759999999999998
2	19.8
3	19.8
4	19.8
5	19.8
6	18.81
7	17.82
8	17.82
9	17.82
10	17.82

Total rows: 10 of 10 Query complete 00:00:00.139 Ln 6, Col 12

pgAdmin 4

File Object Tools Help

Object Explorer

- Aggregates
- Collations
- Domains
- FTS Configurations
- FTS Dictionaries
- FTS Parsers
- FTS Templates
- Foreign Tables
- Functions
- Materialized Views
- Operators
- Procedures
- Sequences
- Tables (11)
 - album
 - Columns (3)
 - Constraints
 - Indexes
 - RLS Policies
 - Rules
 - Triggers
 - artist
 - customer
 - employee
 - genre
 - invoice

Properties SQL Statistics Dependencies Dependents Processes testdb/postgre... music_database/postgres@PostgreSQL 15*

music_database/postgres@PostgreSQL 15

Query Query History

```

1 Question 4 : Which city has the best customer ? We would like to throw a promotional
2 music festival in the city, we made the most money. Write a query that return one city that
3 has the highest sum of the invoice total Returns both name & Sum of the all invoices total?
4
5 Select sum(total) As invoice_total , billing_city
6 from invoice
7 Group by billing_city
8 Order by invoice_total desc ;

```

Data Output Messages Notifications

	invoice_total	billing_city
1	273.240000000000007	Prague
2	169.29	Mountain View
3	166.32	London
4	158.4	Berlin
5	151.47	Paris
6	129.69	São Paulo
7	114.839999999999997	Dublin
8	111.869999999999999	Delhi
9	108.899999999999998	São José dos Campos
10	106.919999999999999	Brasília

Total rows: 53 of 53 Query complete 00:00:00.411 Ln 8, Col 31

pgAdmin 4

File Object Tools Help

Object Explorer

- FTS Dictionaries
- FTS Parsers
- FTS Templates
- Foreign Tables
- Functions
- Materialized Views
- Operators
- Procedures
- Sequences
- Tables (11)
 - album
 - artist
 - customer
 - employee
 - genre
 - invoice
 - invoice_line
 - media_type
 - playlist
 - playlist_track
 - track
- Trigger Functions
- Types
- Views
- Subscriptions
- postgres

Properties SQL Statistics Dependencies Dependents Processes testdb/postgre... music_database/postgres@PostgreSQL 15*

music_database/postgres@PostgreSQL 15

Query Query History

```

1 Question 5 : Who is the best customer ? The customer who has spent the most money will be declared
2 the best customer. Write a query that returns the person who has spent the most money ?
3
4 Select customer.customer_id , customer.first_name , customer.last_name , Sum(invoice.total) As total_spending
5 from customer
6 Join invoice on customer.customer_id = invoice.customer_id
7 Group by customer.customer_id
8 Order by total_spending desc
9 Limit 1 ;
10

```

Data Output Messages Notifications

	customer_id [PK] integer	first_name character	last_name character	total_spending double precision
1	5	R	Madhav	144.54000000000000

Total rows: 1 of 1 Query complete 00:00:00.165 Ln 10, Col 2

pgAdmin 4

File Object Tools Help

Object Explorer

- Foreign Data Wrappers
- Languages
- Publications
- Schemas (1)
 - public
 - Aggregates
 - Collations
 - Domains
 - FTS Configurations
 - FTS Dictionaries
 - FTS Parsers
 - FTS Templates
 - Foreign Tables
 - Functions
 - Materialized Views
 - Operators
 - Procedures
 - Sequences
 - Tables (11)
 - album
 - artist
 - customer
 - employee
 - genre
 - invoice
 - invoice_line
 - media_type

Properties SQL Statistics Dependencies Dependents Processes testdb/postgre... music_database/postgres@PostgreSQL 15*

music_database/postgres@PostgreSQL 15

Query Query History

```

1 Question 6 :Write query to return the email, first name , last name & Genre of all Rock music listeners . Return
2 your list ordered alphabetically by email starting A ?
3
4 Select distinct email,first_name , last_name
5 From customer
6 Join Invoice on customer.customer_id = invoice.customer_id
7 Join invoice_line on invoice.invoice_id = invoice_line.invoice_id
8 Where track_id in (
9 Select track_id from track
10 Join genre on track.genre_id = genre.genre_id
11 Where genre.name Like 'Rock')
12 Order by email ;

```

Data Output Messages Notifications

	email character varying (50)	first_name character	last_name character
1	aaronmitchell@yahoo.ca	Aaron	Mitchell
2	alero@uol.com.br	Alexandre	Rocha
3	astrid.gruber@apple.at	Astrid	Gruber
4	bjorn.hansen@yahoo.no	Bjorn	Hansen
5	camille.bernard@yahoo.fr	Camille	Bernard
6	daan.peeters@apple.be	Daan	Peeters
7	diego.outierrez@yahoo.ar	Diego	Gutiérrez

Total rows: 59 of 59 Query complete 00:00:00.155 Ln 12, Col 18

pgAdmin 4

File Object Tools Help

Object Explorer

- Servers (1)
 - PostgreSQL 15
 - Databases (3)
 - music_database
 - Casts
 - Catalogs (2)
 - ANSI (information_schema)
 - PostgreSQL Catalog (pg_catalog)
 - Event Triggers
 - Extensions
 - Foreign Data Wrappers
 - Languages
 - Publications
 - Schemas (1)
 - Subscriptions
 - postgres
 - testdb
 - Casts
 - Catalogs
 - Event Triggers
 - Extensions
 - Foreign Data Wrappers
 - Languages
 - Publications
 - Schemas (1)
 - public

music_database/postgres@PostgreSQL 15*

Query

```

1 Question 7 : Let's invite the artist who have written the most rock music in our dataset . Write a query that
2 returns the artist name and total track count of the top 5 records ?
3
4 Select artist.artist_id, artist.name, Count(artist.artist_id ) As numer_of_songs
5 From track
6 Join album on album.album_id = track.album_id
7 Join artist on artist.artist_id = album.artist_id
8 Join genre on genre.genre_id = track.genre_id
9 Where genre.name like 'Rock'
10 Group by artist.artist_id
11 Order by numer_of_songs
12 Limit 5 ;

```

Data Output

	artist_id [PK] character varying (50)	name character varying (120)	numer_of_songs bigint
1	157	Dread Zeppelin	1
2	200	The Posies	2
3	2	Accept	4
4	136	Terry Bozzio, Tony Levin & Steve Stevens	7
5	23	Frank Zappa & Captain Beefheart	9

Total rows: 5 of 5 Query complete 00:00:00.161 Ln 4, Col 1

pgAdmin 4

File Object Tools Help

Object Explorer

- Servers
 - PostgreSQL 15
 - Databases (3)
 - music_database
 - Casts
 - Catalogs (2)
 - ANSI (information_schema)
 - PostgreSQL Catalog (pg_catalog)
 - Event Triggers
 - Extensions
 - Foreign Data Wrappers
 - Languages
 - Publications
 - Schemas (1)
 - public
 - Aggregates
 - Collations
 - Domains
 - FTS Configurations
 - FTS Dictionaries
 - FTS Parsers
 - FTS Templates
 - Foreign Tables
 - Functions
 - Materialized Views
 - Operators

music_database/postgres@PostgreSQL 15*

Query

```

1 Question 8 : Return all the track names that have a song length longer than the average song length.
2 Return the Name and Milliseconds for each track.
3 Order by the song length with the longest songs listed first.
4
5 SELECT name,milliseconds
6 FROM track
7 WHERE milliseconds > (
8     SELECT AVG(milliseconds) AS avg_track_length
9     FROM track )
10 ORDER BY milliseconds DESC ;
11

```

Data Output

	name character varying (150)	milliseconds integer
1	Occupation / Precipice	5286953
2	Through a Looking Glass	5088838
3	Greetings from Earth, Pt. 1	2960293
4	The Man With Nine Lives	2956998
5	Battlestar Galactica, Pt. 2	2956081
6	Battlestar Galactica, Pt. 1	2952702
7	Murder On the Rising Star	2935894
8	Battlestar Galactica, Pt. 3	2927802

Total rows: 494 of 494 Query complete 00:00:00.448 Ln 11, Col 1

pgAdmin 4

File Object Tools Help

Object Explorer

- Servers
 - PostgreSQL 15
 - Databases (3)
 - music_database
 - Casts
 - Catalogs (2)
 - ANSI (information_schema)
 - PostgreSQL Catalog (pg_catalog)
 - Event Triggers
 - Extensions
 - Foreign Data Wrappers
 - Languages
 - Publications
 - Schemas (1)
 - public
 - Aggregates
 - Collations
 - Domains
 - FTS Configurations
 - FTS Dictionaries
 - FTS Parsers
 - FTS Templates
 - Foreign Tables
 - Functions
 - Materialized Views
 - Operators

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

Query Query History

```

1 Question 9 : Find how much amount spent by each customer on artist ? Write a query to return customer name
2 artist name and total spent ?
3
4 WITH best_selling_artist AS (
5   SELECT artist.artist_id AS artist_id, artist.name AS artist_name, SUM(invoice_line.unit_price*invoice_line.quantity) AS amount_spent
6   FROM invoice_line
7   JOIN track ON track.track_id = invoice_line.track_id
8   JOIN album ON album.album_id = track.album_id
9   JOIN artist ON artist.artist_id = album.artist_id
10  GROUP BY 1
11  ORDER BY 3 DESC
12  LIMIT 1
13 )
14 SELECT c.customer_id, c.first_name, c.last_name, bsa.artist_name, SUM(il.unit_price*il.quantity) AS amount_spent
15 FROM invoice i
16 JOIN customer c ON c.customer_id = i.customer_id
17 JOIN invoice_line il ON il.invoice_id = i.invoice_id
18 JOIN track t ON t.track_id = il.track_id
19 JOIN album alb ON alb.album_id = t.album_id
20 JOIN best_selling_artist bsa ON bsa.artist_id = alb.artist_id
21 GROUP BY 1,2,3,4
22 ORDER BY 5 DESC;
23

```

Data Output Messages Notifications

Total rows: 43 of 43 Query complete 00:00:00.414 Ln 23, Col 1

pgAdmin 4

File Object Tools Help

Object Explorer

- Servers
 - PostgreSQL 15
 - Databases (3)
 - music_database
 - Casts
 - Catalogs (2)
 - ANSI (information_schema)
 - PostgreSQL Catalog (pg_catalog)
 - Event Triggers
 - Extensions
 - Foreign Data Wrappers
 - Languages
 - Publications
 - Schemas (1)
 - public
 - Aggregates
 - Collations
 - Domains
 - FTS Configurations
 - FTS Dictionaries
 - FTS Parsers
 - FTS Templates
 - Foreign Tables
 - Functions
 - Materialized Views
 - Operators

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

Query Query History

```

1 Question 9 : Find how much amount spent by each customer on artist ? Write a query to return customer name
2 artist name and total spent ?
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23

```

Data Output Messages Notifications

	customer_id integer	first_name character	last_name character	artist_name character varying (120)	amount_spent double precision
9	20	Dan	Miller	Queen	3.96
10	5	R	Madhav	Queen	3.96
11	23	John	Gordon	Queen	2.9699999999999998
12	54	Steve	Murray	Queen	2.9699999999999998
13	31	Martha	Silk	Queen	2.9699999999999998
14	16	Frank	Harris	Queen	1.98
15	17	Jack	Smith	Queen	1.98
16	24	Frank	Ralston	Queen	1.98
17	30	Edward	Francis	Queen	1.98
18	35	Madalena	Sampaio	Queen	1.98
19	36	Hannah	Schneider	Queen	1.98
20	11	Alexandre	Rocha	Queen	1.98
21	8	Daan	Peeters	Queen	1.98
22	42	Wyatt	Girard	Queen	1.98

Total rows: 43 of 43 Query complete 00:00:00.414 Ln 23, Col 1

pgAdmin 4

File Object Tools Help

Object Explorer

- Servers
 - PostgreSQL 15
 - Databases (3)
 - music_database
 - Cast
 - Catalogs (2)
 - ANSI (information_schema)
 - PostgreSQL Catalog (pg_catalog)
 - Event Triggers
 - Extensions
 - Foreign Data Wrappers
 - Languages
 - Publications
 - Schemas (1)
 - public
 - Aggregates
 - Collations
 - Domains
 - FTS Configurations
 - FTS Dictionaries
 - FTS Parsers
 - FTS Templates
 - Foreign Tables
 - Functions
 - Materialized Views
 - Operators

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

music_database/postgres@PostgreSQL 15

Query Query History

```

1 Question 10: We want to find out the most popular music Genre for each country. We determine the most popular genre
2 with the highest amount of purchases. Write a query that returns each country along with the top Genre. For countries
3 the maximum number of purchases is shared return all Genres.
4 WITH popular_genre AS
5 (
6     SELECT COUNT(invoice_line.quantity) AS purchases, customer.country, genre.name, genre.genre_id,
7     ROW_NUMBER() OVER(PARTITION BY customer.country ORDER BY COUNT(invoice_line.quantity) DESC) AS RowNo
8     FROM invoice_line
9     JOIN invoice ON invoice.invoice_id = invoice_line.invoice_id
10    JOIN customer ON customer.customer_id = invoice.customer_id
11    JOIN track ON track.track_id = invoice_line.track_id
12    JOIN genre ON genre.genre_id = track.genre_id
13    GROUP BY 2,3,4
14    ORDER BY 2 ASC, 1 DESC
15 )
16 SELECT * FROM popular_genre WHERE RowNo <= 1
17

```

Data Output Messages Notifications

	purchases bigint	country character varying (50)	name character varying (120)	genre_id character varying (50)	rowno bigint
1	17	Argentina	Alternative & Punk	4	1
2	34	Australia	Rock	1	1
3	40	Austria	Rock	1	1
4	26	Belgium	Rock	1	1

Total rows: 24 of 24 Query complete 00:00:00.223 Ln 14, Col 24

pgAdmin 4

File Object Tools Help

Object Explorer

- Servers
 - PostgreSQL 15
 - Databases (3)
 - music_database
 - Cast
 - Catalogs (2)
 - ANSI (information_schema)
 - PostgreSQL Catalog (pg_catalog)
 - Event Triggers
 - Extensions
 - Foreign Data Wrappers
 - Languages
 - Publications
 - Schemas (1)
 - public
 - Aggregates
 - Collations
 - Domains
 - FTS Configurations
 - FTS Dictionaries
 - FTS Parsers
 - FTS Templates
 - Foreign Tables
 - Functions
 - Materialized Views
 - Operators

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

music_database/postgres@PostgreSQL 15

Query Query History

```

1 Question 11 : Write a query that determines the customer that has spent the most on music for each country. Write a
2 query that returns the country along with the top customer and how much they spent. For countries where the top
3 amount spent is shared, provide all customers who spent this amount.
4
5 WITH Customer_with_country AS (
6     SELECT customer.customer_id, first_name, last_name, billing_country, SUM(total) AS total_spending,
7     ROW_NUMBER() OVER(PARTITION BY billing_country ORDER BY SUM(total) DESC) AS RowNo
8     FROM invoice
9     JOIN customer ON customer.customer_id = invoice.customer_id
10    GROUP BY 1,2,3,4
11    ORDER BY 4 ASC,5 DESC)
12 SELECT * FROM Customer_with_country WHERE RowNo <= 1
13
14

```

Data Output Messages Notifications

	customer_id integer	first_name character	last_name character	billing_country character varying (30)	total_spending double precision	rowno bigint
1	56	Diego	Gutiérrez	Argentina	39.6	1
2	55	Mark	Taylor	Australia	81.18	1
3	7	Astrid	Gruber	Austria	69.3	1
4	8	Daan	Peeters	Belgium	60.38999999999999	1
5	1	Luis	Gonçalves	Brazil	108.89999999999998	1
6	3	François	Tremblay	Canada	99.99	1

Total rows: 24 of 24 Query complete 00:00:00.257 Ln 14, Col 1