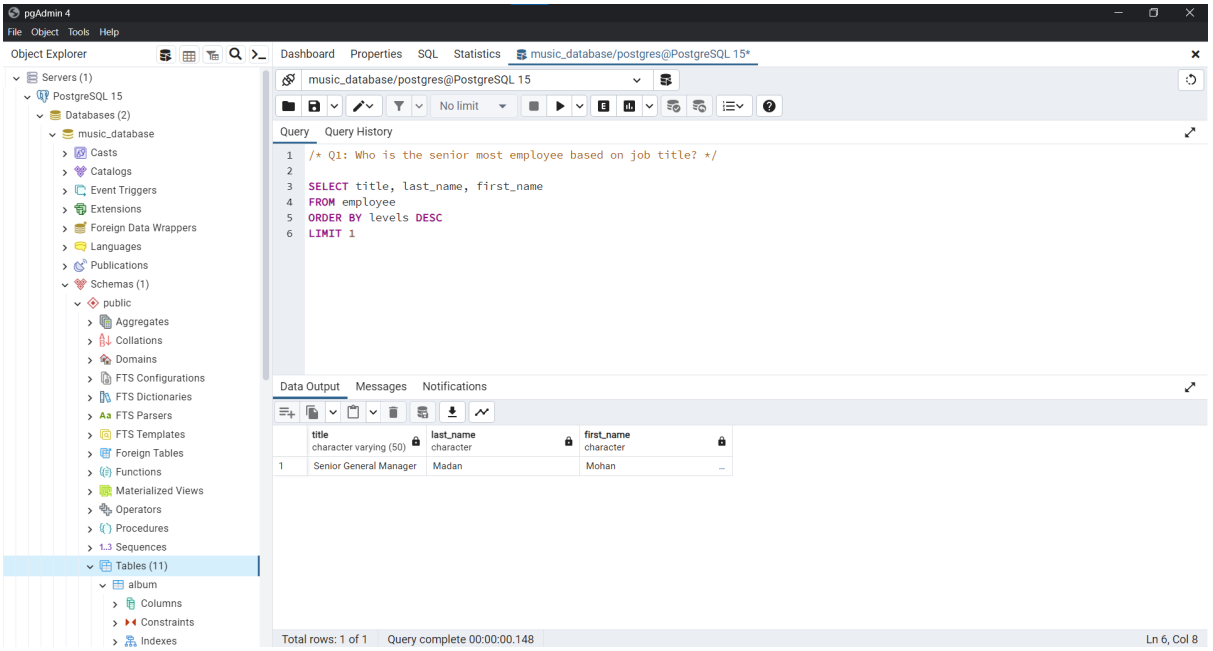
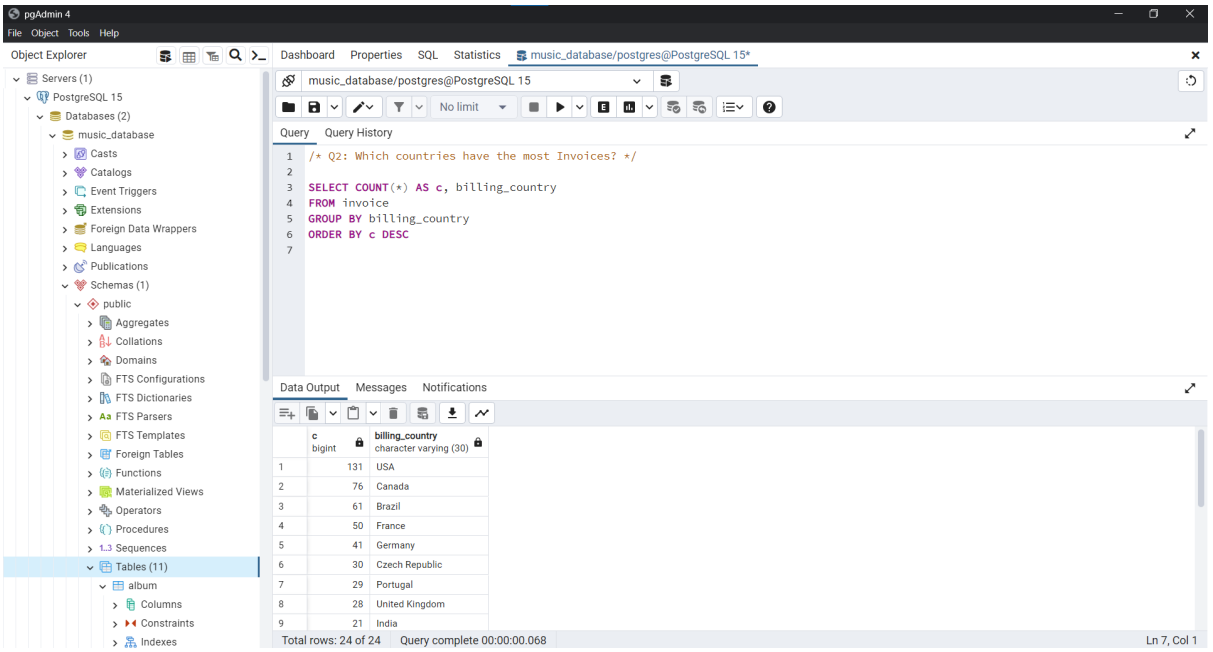
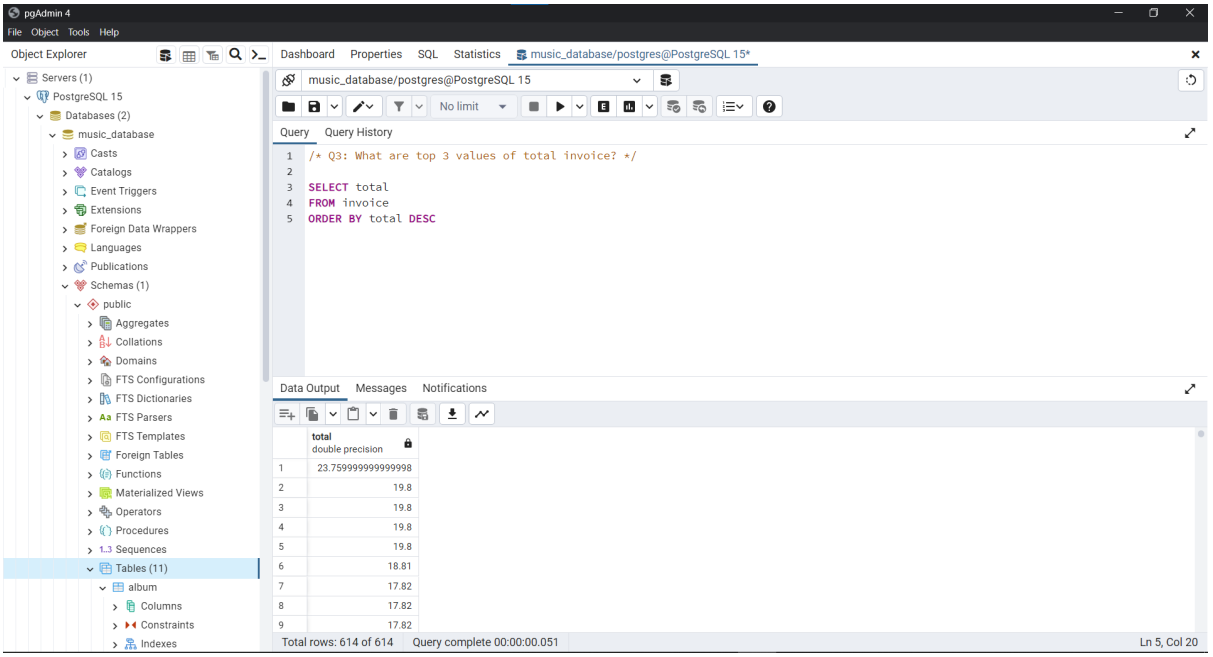
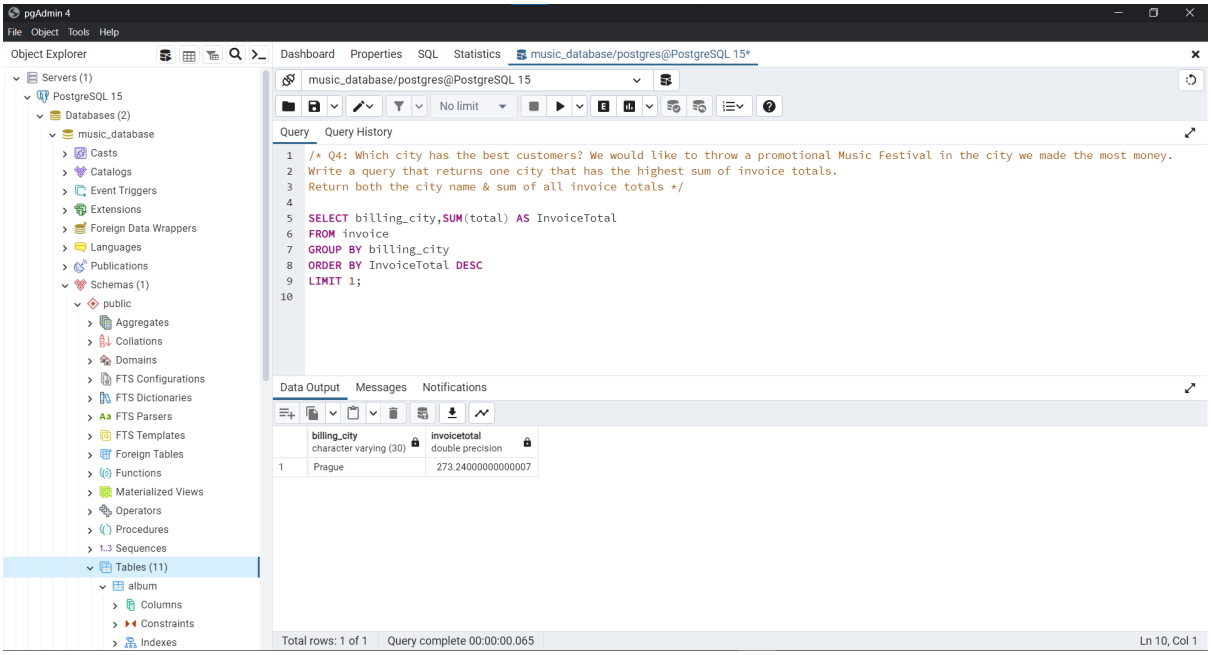
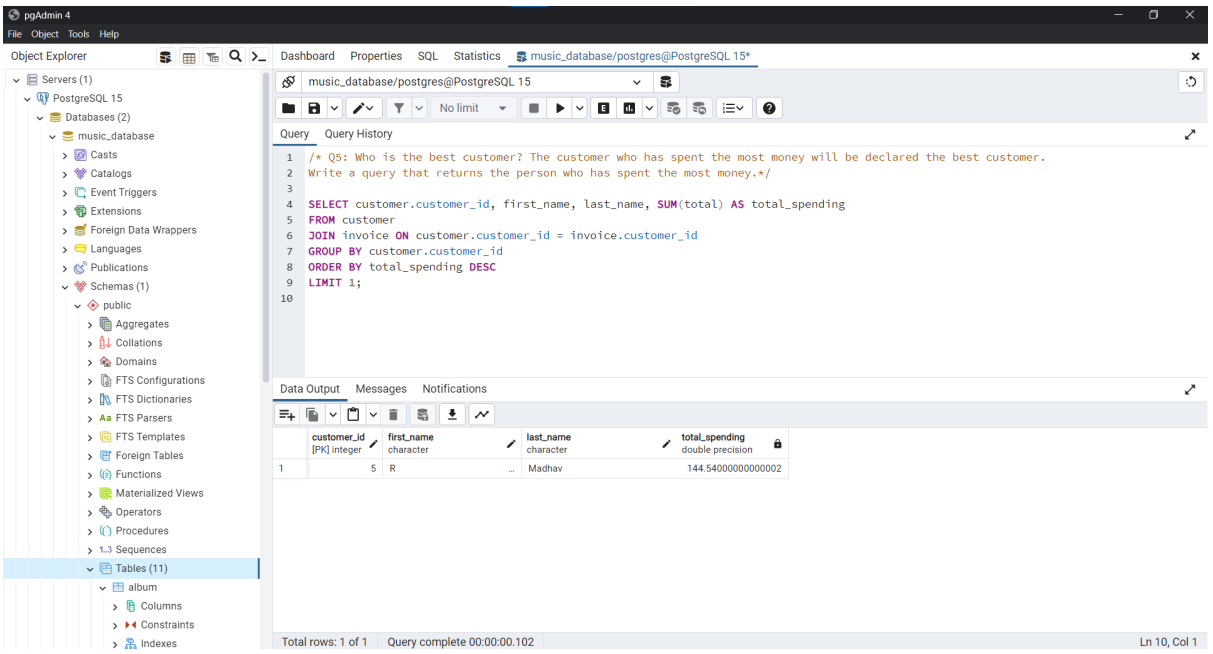
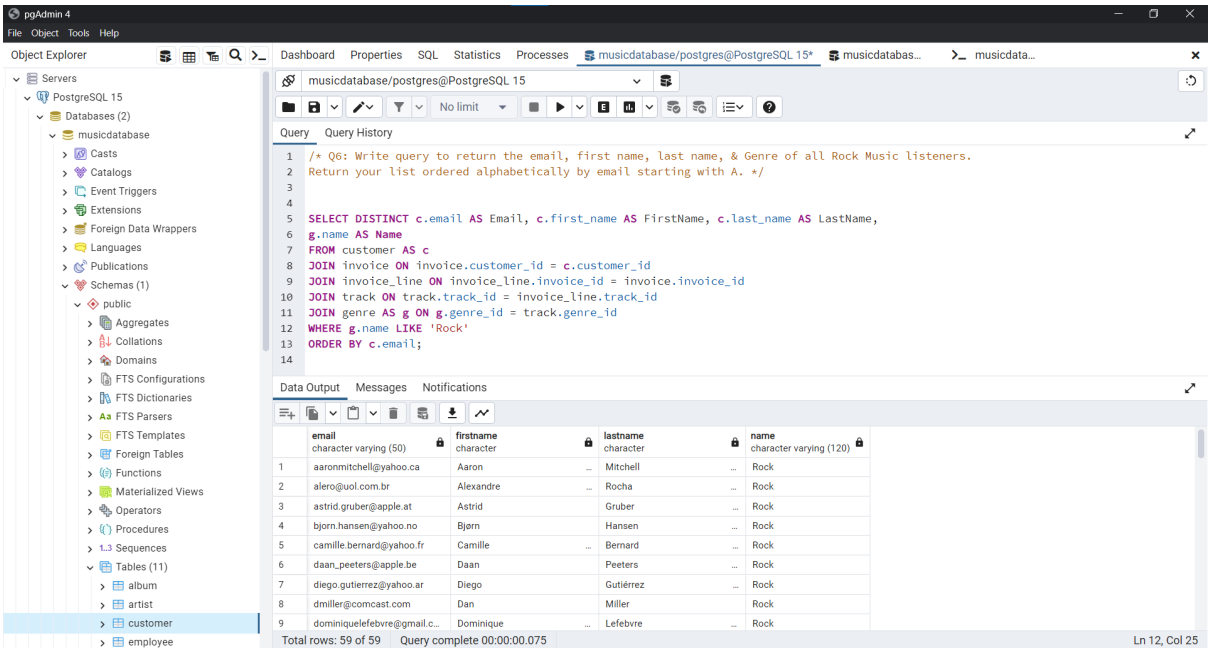

pgAdmin 4
File Object Tools Help
Object Explorer
Dashboard Properties SQL Statistics music_database/postgres@PostgreSQL 15*
music_database/postgres@PostgreSQL 15
No limit
Query Query History
1 /* Q1: Who is the senior most employee based on job title? */
2
3 SELECT title, last_name, first_name
4 FROM employee
5 ORDER BY levels DESC
6 LIMIT 1
Data Output Messages Notifications
title character varying (50)
last_name character
first_name character
1 Senior General Manager Madan Mohan
Total rows: 1 of 1
Query complete 00:00:00.148
Ln 6, Col 8

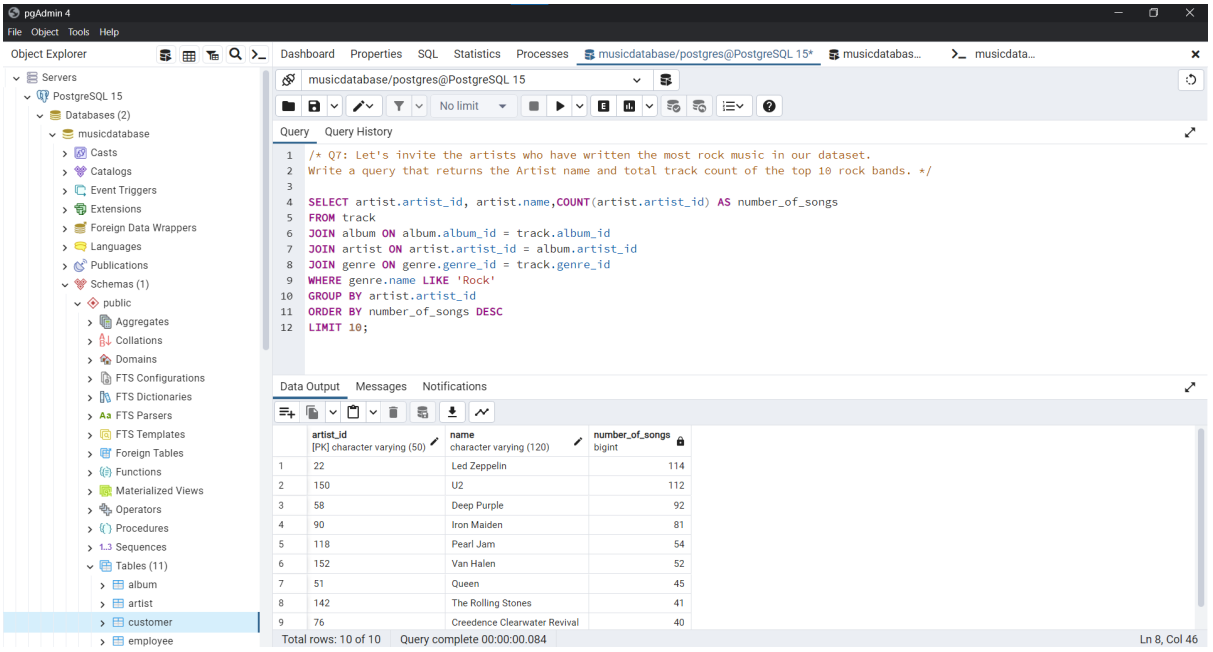

pgAdmin 4
File Object Tools Help
Object Explorer
Dashboard Properties SQL Statistics music_database/postgres@PostgreSQL 15*
music_database/postgres@PostgreSQL 15
No limit
Query Query History
1 /* Q2: Which countries have the most Invoices? */
2
3 SELECT COUNT(*) AS c, billing_country
4 FROM invoice
5 GROUP BY billing_country
6 ORDER BY c DESC
7
Data Output Messages Notifications
c bigint
billing_country character varying (30)
1 131 USA
2 76 Canada
3 61 Brazil
4 50 France
5 41 Germany
6 30 Czech Republic
7 29 Portugal
8 28 United Kingdom
9 21 India
Total rows: 24 of 24
Query complete 00:00:00.068
Ln 7, Col 1

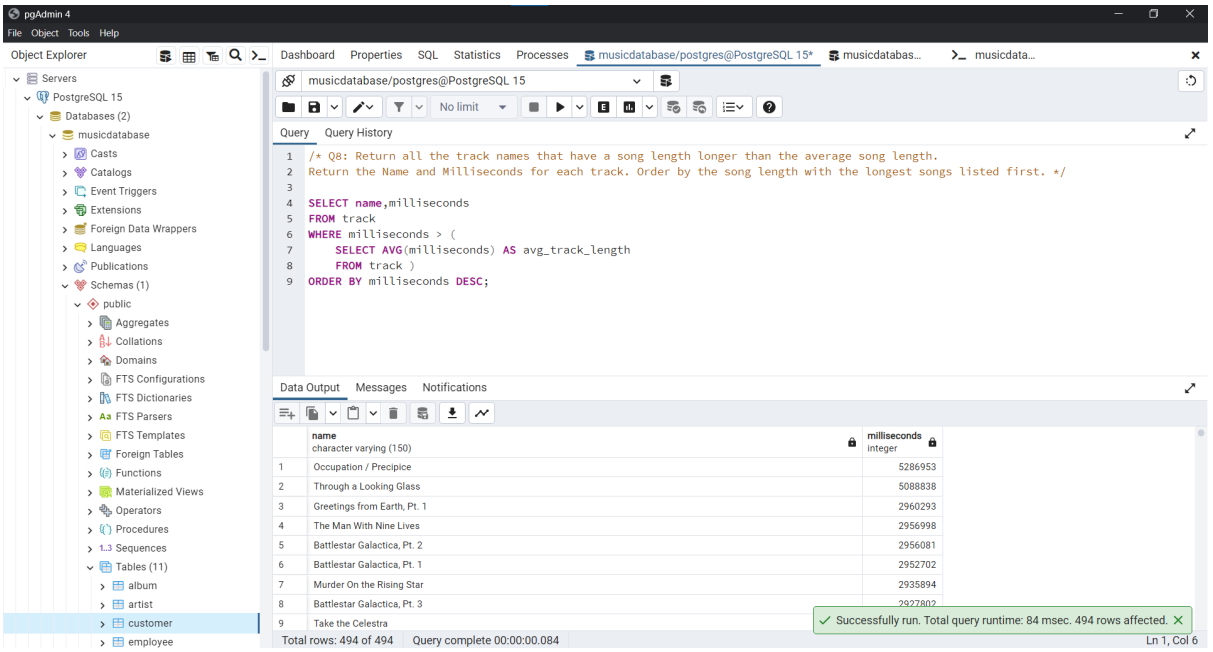

pgAdmin 4
File Object Tools Help
Object Explorer
Dashboard Properties SQL Statistics music_database/postgres@PostgreSQL 15*
music_database/postgres@PostgreSQL 15
No limit
Query Query History
/* Q3: What are top 3 values of total invoice? */
SELECT total
FROM invoice
ORDER BY total DESC
Data Output Messages Notifications
total
double precision
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
Total rows: 614 of 614
Query complete 00:00:00.051
Ln 5, Col 20

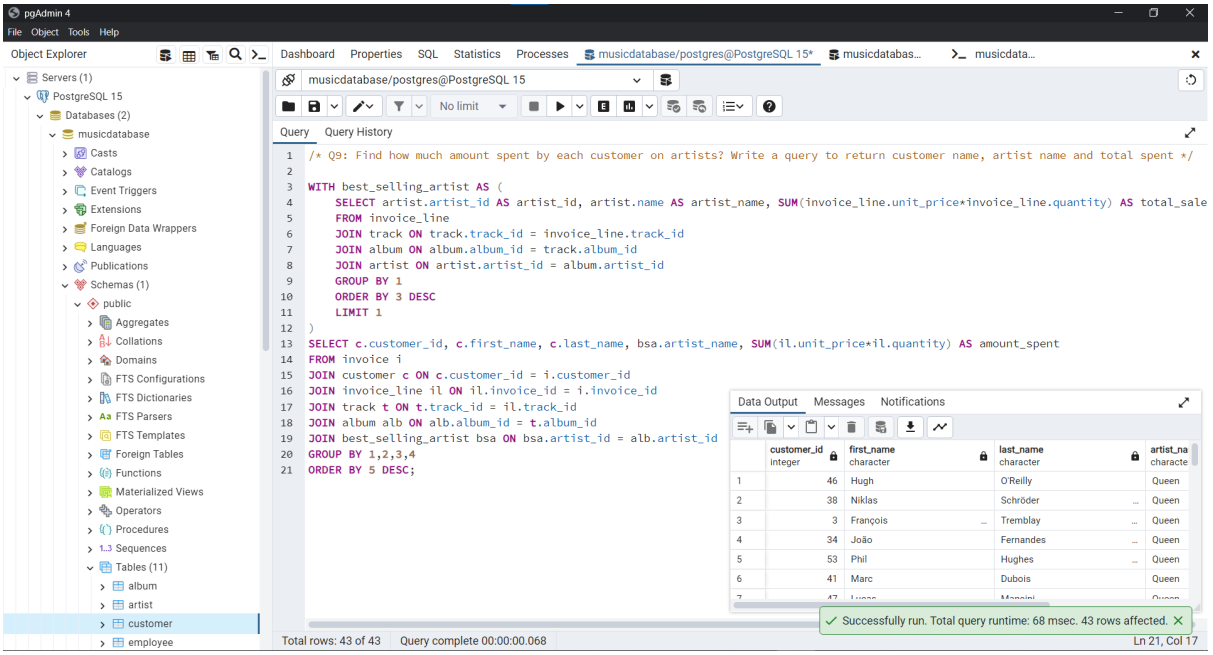

pgAdmin 4
File Object Tools Help
Object Explorer
Dashboard Properties SQL Statistics music_database/postgres@PostgreSQL 15*
music_database/postgres@PostgreSQL 15
No limit
Query Query History
/* Q4: Which city has the best customers? We would like to throw a promotional Music Festival in the city we made the most money.
Write a query that returns one city that has the highest sum of invoice totals.
Return both the city name & sum of all invoice totals */
SELECT billing_city,SUM(total) AS InvoiceTotal
FROM invoice
GROUP BY billing_city
ORDER BY InvoiceTotal DESC
LIMIT 1;
Data Output Messages Notifications
billing_city
character varying (30)
invoicetotal
double precision
1 Prague 273.24000000000007
Total rows: 1 of 1
Query complete 00:00:00.065
Ln 10, Col 1

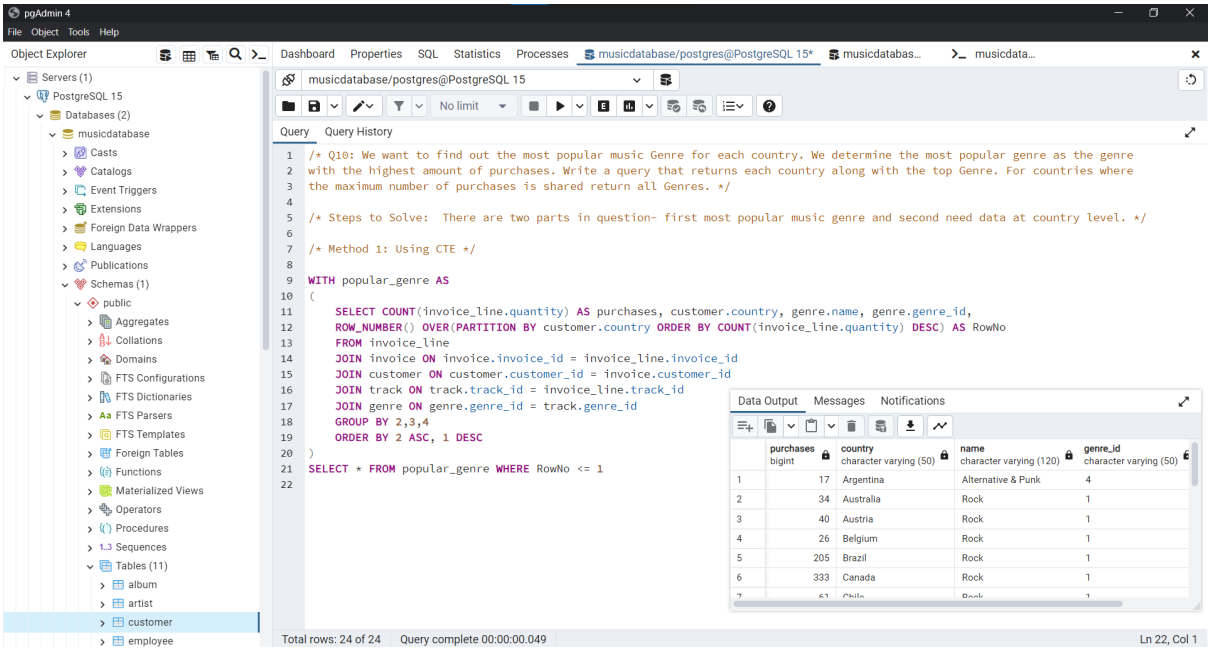

pgAdmin 4
File Object Tools Help
Object Explorer
Dashboard Properties SQL Statistics music_database/postgres@PostgreSQL 15*
music_database/postgres@PostgreSQL 15
No limit
Query Query History
/* Q5: Who is the best customer? The customer who has spent the most money will be declared the best customer.
Write a query that returns the person who has spent the most money.*/
SELECT customer.customer_id, first_name, last_name, SUM(total) AS total_spending
FROM customer
JOIN invoice ON customer.customer_id = invoice.customer_id
GROUP BY customer.customer_id
ORDER BY total_spending DESC
LIMIT 1;
Data Output Messages Notifications
customer_id [PK] integer
first_name character
last_name character
total_spending double precision
1 5 R Madhav 144.54000000000002
Total rows: 1 of 1
Query complete 00:00:00.102
Ln 10, Col 1

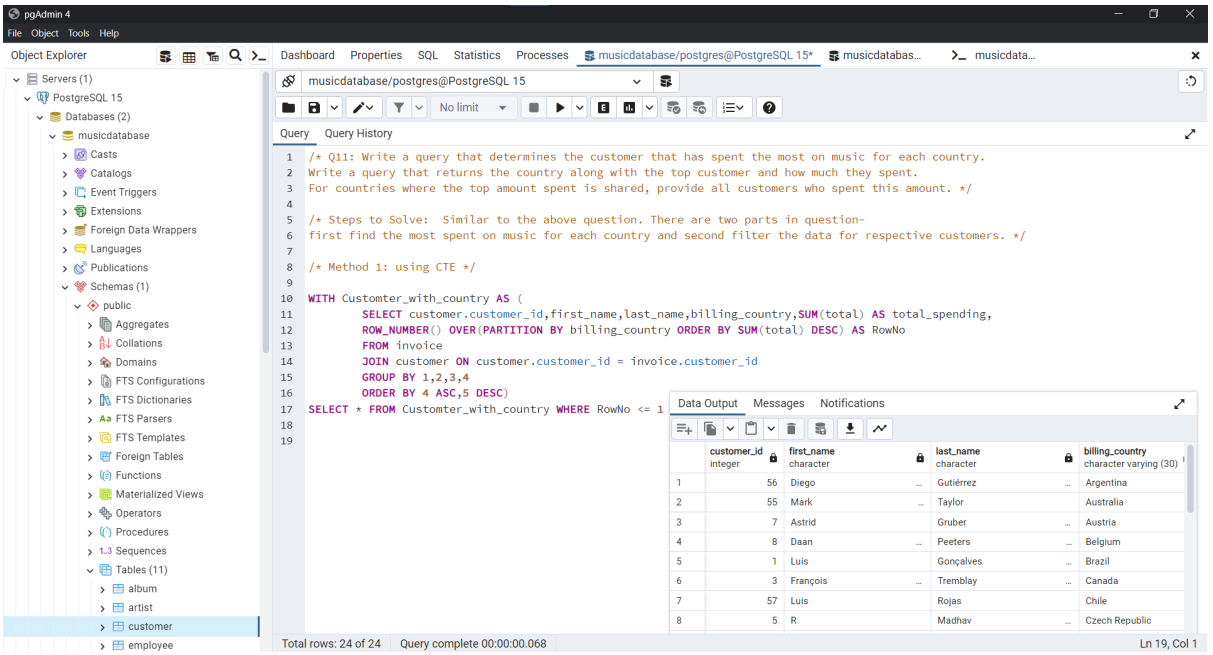

pgAdmin 4
File Object Tools Help
Object Explorer
Dashboard Properties SQL Statistics Processes musicdatabase/postgres@PostgreSQL 15*
musicdatabase/postgres@PostgreSQL 15
No limit
Query Query History
/* Q6: Write query to return the email, first name, last name, & Genre of all Rock Music listeners.
Return your list ordered alphabetically by email starting with A. */
SELECT DISTINCT c.email AS Email, c.first_name AS FirstName, c.last_name AS LastName,
g.name AS Name
FROM customer AS c
JOIN invoice ON invoice.customer_id = c.customer_id
JOIN invoice_line ON invoice_line.invoice_id = invoice.invoice_id
JOIN track ON track.track_id = invoice_line.track_id
JOIN genre AS g ON g.genre_id = track.genre_id
WHERE g.name LIKE 'Rock'
ORDER BY c.email;
Data Output Messages Notifications
email character varying (50)
firstname character
lastname character
name character varying (120)
1 aaronmitchell@yahoo.ca Aaron Mitchell Rock
2 alero@uol.com.br Alexandre Rocha Rock
3 astrid.gruber@apple.at Astrid Gruber Rock
4 bjorn.hansen@yahoo.no Bjørn Hansen Rock
5 camille.bernard@yahoo.fr Camille Bernard Rock
6 daan_peeters@apple.be Daan Peeters Rock
7 diego.gutierrez@yahoo.ar Diego Gutiérrez Rock
8 dmiller@comcast.com Dan Miller Rock
9 dominiquelefebvre@gmail.c... Dominique Lefebvre Rock
Total rows: 59 of 59
Query complete 00:00:00.075
Ln 12, Col 25


pgAdmin 4
File Object Tools Help
Object Explorer
Dashboard Properties SQL Statistics Processes musicdatabase/postgres@PostgreSQL 15* musicdatabas... musicdata...
Servers
PostgreSQL 15
Databases (2)
musicdatabase
Casts
Catalogs
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
Procedures
Sequences
Tables (11)
album
artist
customer
employee
musicdatabase/postgres@PostgreSQL 15
No limit
Query Query History
/* Q7: Let's invite the artists who have written the most rock music in our dataset.
Write a query that returns the Artist name and total track count of the top 10 rock bands. */
SELECT artist.artist_id, artist.name,COUNT(artist.artist_id) AS number_of_songs
FROM track
JOIN album ON album.album_id = track.album_id
JOIN artist ON artist.artist_id = album.artist_id
JOIN genre ON genre.genre_id = track.genre_id
WHERE genre.name LIKE 'Rock'
GROUP BY artist.artist_id
ORDER BY number_of_songs DESC
LIMIT 10;
Data Output Messages Notifications
artist_id [PK] character varying (50)
name character varying (120)
number_of_songs bigint
1 22 Led Zeppelin 114
2 150 U2 112
3 58 Deep Purple 92
4 90 Iron Maiden 81
5 118 Pearl Jam 54
6 152 Van Halen 52
7 51 Queen 45
8 142 The Rolling Stones 41
9 76 Creedence Clearwater Revival 40
Total rows: 10 of 10
Query complete 00:00:00.084
Ln 8, Col 46


pgAdmin 4
File Object Tools Help
Object Explorer
Dashboard Properties SQL Statistics Processes musicdatabase/postgres@PostgreSQL 15* musicdatabas... musicdata...
Servers
PostgreSQL 15
Databases (2)
musicdatabase
Casts
Catalogs
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
Procedures
Sequences
Tables (11)
album
artist
customer
employee
musicdatabase/postgres@PostgreSQL 15
No limit
Query Query History
/* Q8: Return all the track names that have a song length longer than the average song length.
Return the Name and Milliseconds for each track. Order by the song length with the longest songs listed first. */
SELECT name,milliseconds
FROM track
WHERE milliseconds > (
SELECT AVG(milliseconds) AS avg_track_length
FROM track )
ORDER BY milliseconds DESC;
Data Output Messages Notifications
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
9 Take the Celestra
Successfully run. Total query runtime: 84 msec. 494 rows affected.
Total rows: 494 of 494
Query complete 00:00:00.084
Ln 1, Col 6


pgAdmin 4
File Object Tools Help
Object Explorer
Servers (1)
PostgreSQL 15
Databases (2)
musicdatabase
Casts
Catalogs
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
Procedures
Sequences
Tables (11)
album
artist
customer
employee
Dashboard Properties SQL Statistics Processes musicdatabase/postgres@PostgreSQL 15* musicdatabas... musicdata...
musicdatabase/postgres@PostgreSQL 15
No limit
Query Query History
/* Q9: Find how much amount spent by each customer on artists? Write a query to return customer name, artist name and total spent */
WITH best_selling_artist AS (
SELECT artist.artist_id AS artist_id, artist.name AS artist_name, SUM(invoice_line.unit_price*invoice_line.quantity) AS total_sale
FROM invoice_line
JOIN track ON track.track_id = invoice_line.track_id
JOIN album ON album.album_id = track.album_id
JOIN artist ON artist.artist_id = album.artist_id
GROUP BY 1
ORDER BY 3 DESC
LIMIT 1
)
SELECT c.customer_id, c.first_name, c.last_name, bsa.artist_name, SUM(il.unit_price*il.quantity) AS amount_spent
FROM invoice i
JOIN customer c ON c.customer_id = i.customer_id
JOIN invoice_line il ON il.invoice_id = i.invoice_id
JOIN track t ON t.track_id = il.track_id
JOIN album alb ON alb.album_id = t.album_id
JOIN best_selling_artist bsa ON bsa.artist_id = alb.artist_id
GROUP BY 1,2,3,4
ORDER BY 5 DESC;
Data Output Messages Notifications
customer_id integer
first_name character
last_name character
artist_na characte
1 46 Hugh O'Reilly Queen
2 38 Niklas Schröder Queen
3 3 François Tremblay Queen
4 34 João Fernandes Queen
5 53 Phil Hughes Queen
6 41 Marc Dubois Queen
Successfully run. Total query runtime: 68 msec. 43 rows affected.
Total rows: 43 of 43
Query complete 00:00:00.068
Ln 21, Col 17


pgAdmin 4
File Object Tools Help
Object Explorer
Dashboard Properties SQL Statistics Processes musicdatabase/postgres@PostgreSQL 15* musicdatabas... musicdata...
musicdatabase/postgres@PostgreSQL 15
No limit
Query Query History
/* Q10: We want to find out the most popular music Genre for each country. We determine the most popular genre as the genre
with the highest amount of purchases. Write a query that returns each country along with the top Genre. For countries where
the maximum number of purchases is shared return all Genres. */
/* Steps to Solve: There are two parts in question- first most popular music genre and second need data at country level. */
/* Method 1: Using CTE */
WITH popular_genre AS
(
SELECT COUNT(invoice_line.quantity) AS purchases, customer.country, genre.name, genre.genre_id,
ROW_NUMBER() OVER(PARTITION BY customer.country ORDER BY COUNT(invoice_line.quantity) DESC) AS RowNo
FROM invoice_line
JOIN invoice ON invoice.invoice_id = invoice_line.invoice_id
JOIN customer ON customer.customer_id = invoice.customer_id
JOIN track ON track.track_id = invoice_line.track_id
JOIN genre ON genre.genre_id = track.genre_id
GROUP BY 2,3,4
ORDER BY 2 ASC, 1 DESC
)
SELECT * FROM popular_genre WHERE RowNo <= 1
Data Output Messages Notifications
purchases bigint
country character varying (50)
name character varying (120)
genre_id character varying (50)
1 17 Argentina Alternative & Punk 4
2 34 Australia Rock 1
3 40 Austria Rock 1
4 26 Belgium Rock 1
5 205 Brazil Rock 1
6 333 Canada Rock 1
Total rows: 24 of 24
Query complete 00:00:00.049
Ln 22, Col 1


pgAdmin 4
File Object Tools Help
Object Explorer
Dashboard Properties SQL Statistics Processes musicdatabase/postgres@PostgreSQL 15* musicdatabas... musicdata...
musicdatabase/postgres@PostgreSQL 15
No limit
Query Query History
/* Q11: Write a query that determines the customer that has spent the most on music for each country.
Write a query that returns the country along with the top customer and how much they spent.
For countries where the top amount spent is shared, provide all customers who spent this amount. */
/* Steps to Solve: Similar to the above question. There are two parts in question-
first find the most spent on music for each country and second filter the data for respective customers. */
/* Method 1: using CTE */
WITH Customter_with_country AS (
SELECT customer.customer_id,first_name,last_name,billing_country,SUM(total) AS total_spending,
ROW_NUMBER() OVER(PARTITION BY billing_country ORDER BY SUM(total) DESC) AS RowNo
FROM invoice
JOIN customer ON customer.customer_id = invoice.customer_id
GROUP BY 1,2,3,4
ORDER BY 4 ASC,5 DESC)
SELECT * FROM Customter_with_country WHERE RowNo <= 1
Data Output Messages Notifications
customer_id integer
first_name character
last_name character
billing_country character varying (30)
1 56 Diego Gutiérrez Argentina
2 55 Mark Taylor Australia
3 7 Astrid Gruber Austria
4 8 Daan Peeters Belgium
5 1 Luís Gonçalves Brazil
6 3 François Tremblay Canada
7 57 Luis Rojas Chile
8 5 R Madhav Czech Republic
Total rows: 24 of 24
Query complete 00:00:00.068
Ln 19, Col 1