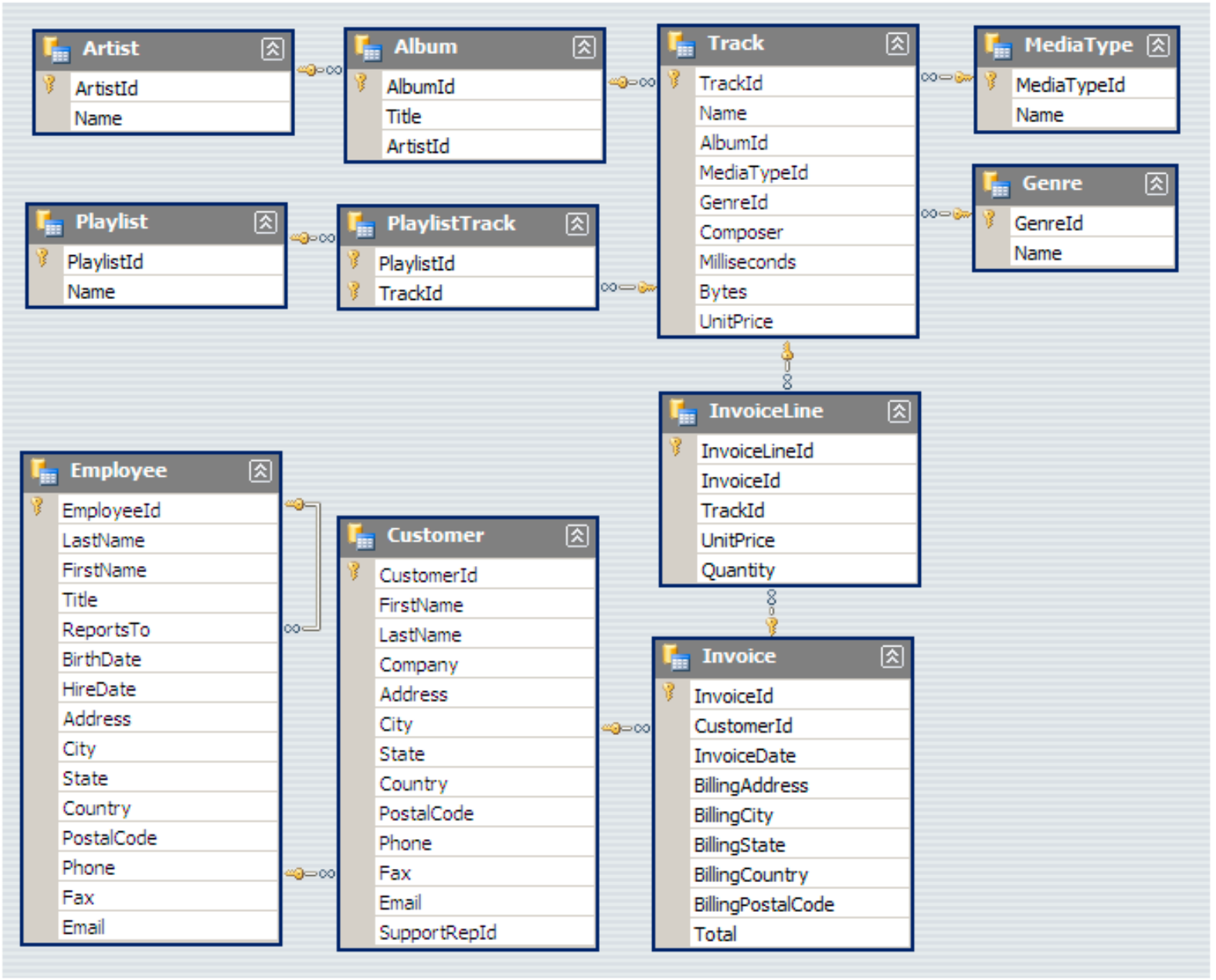
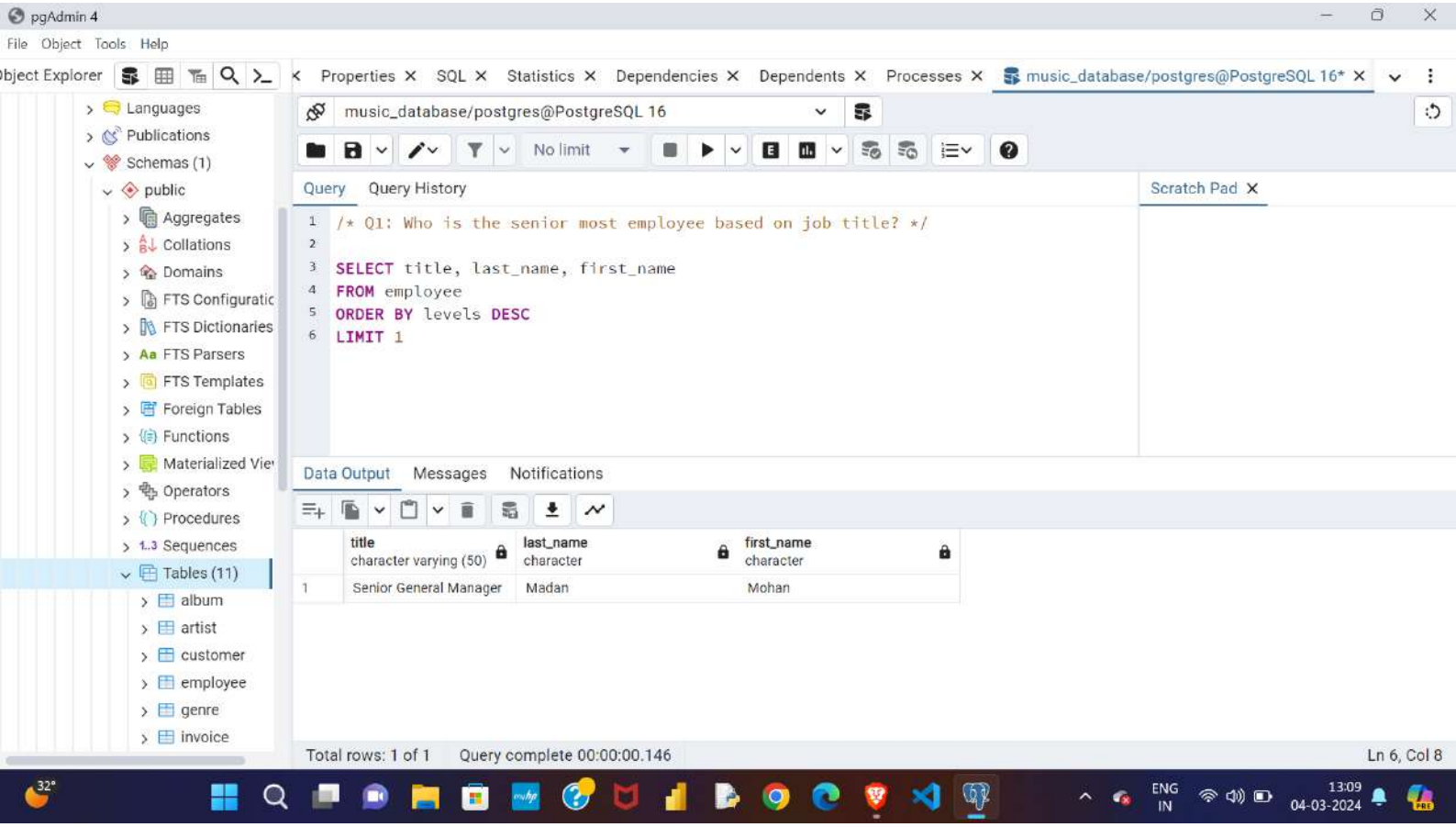
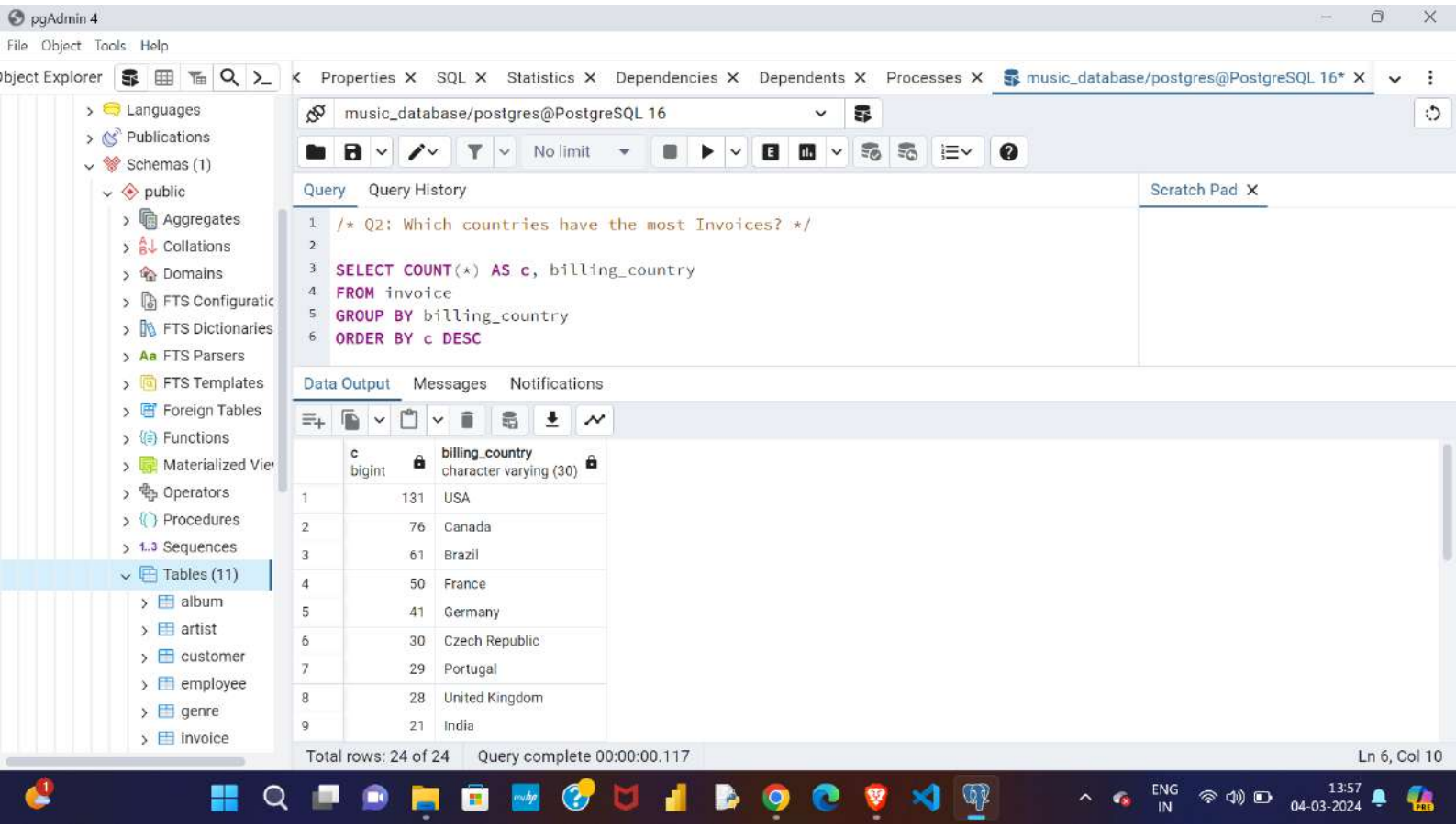
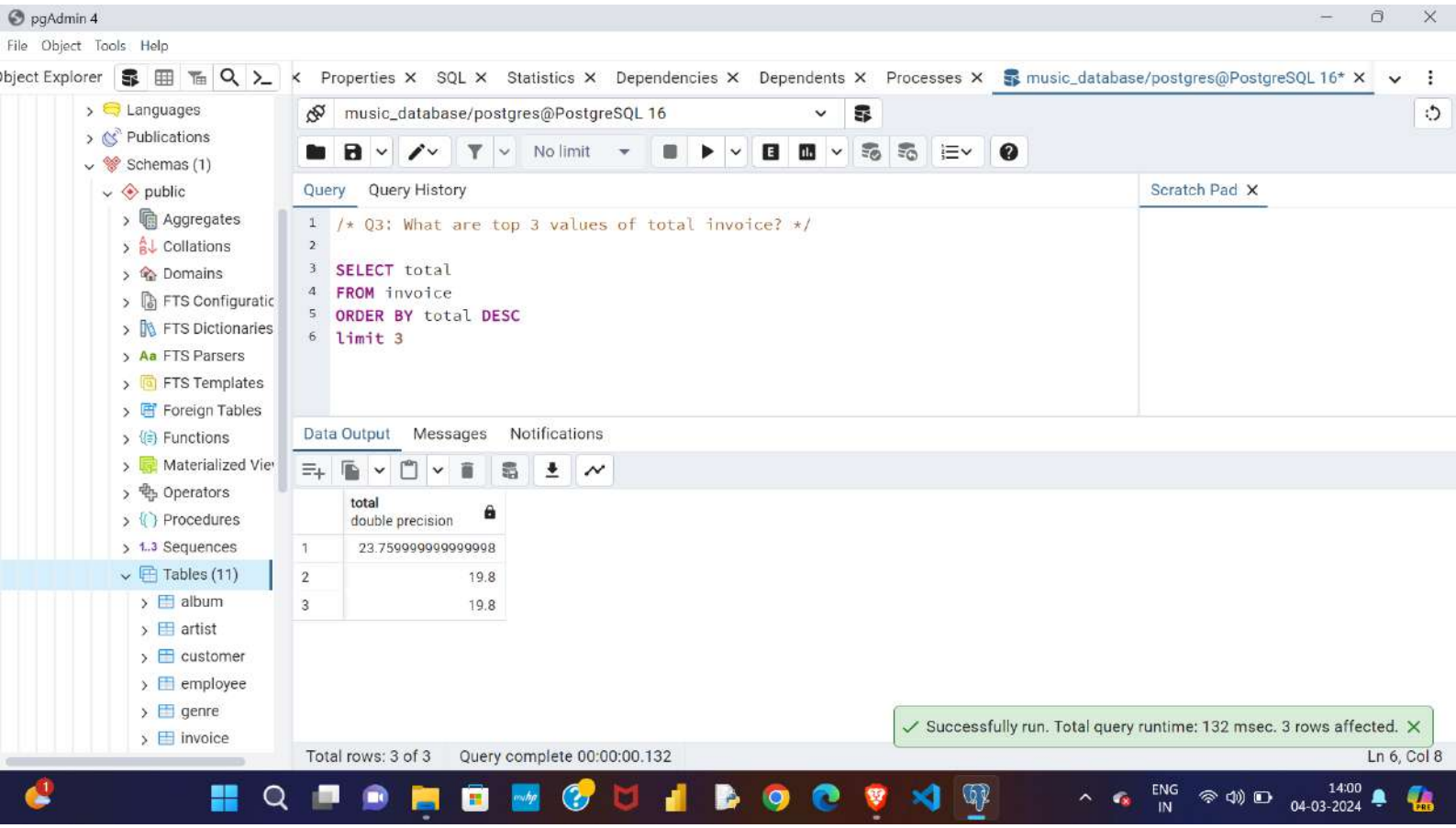
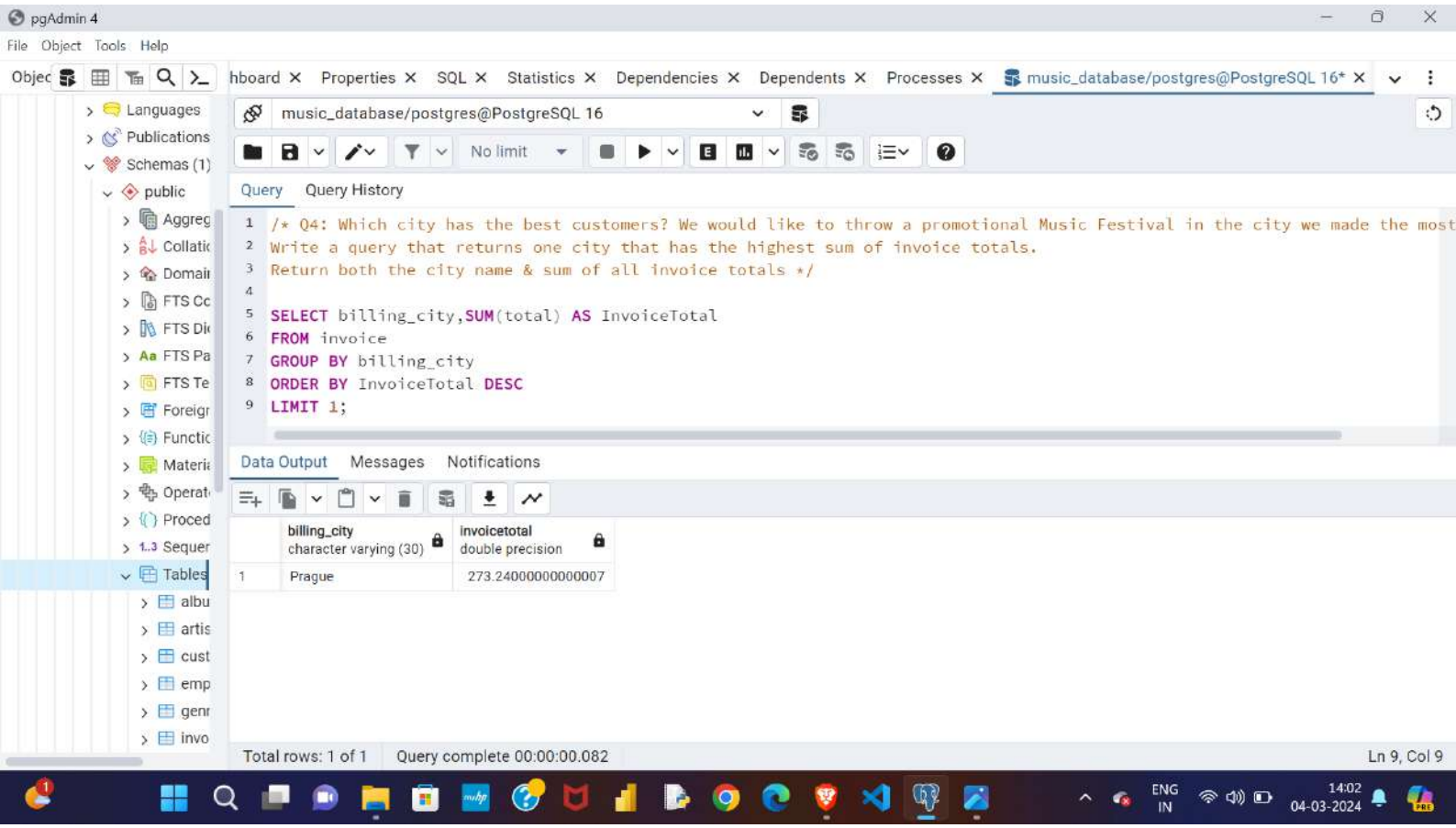
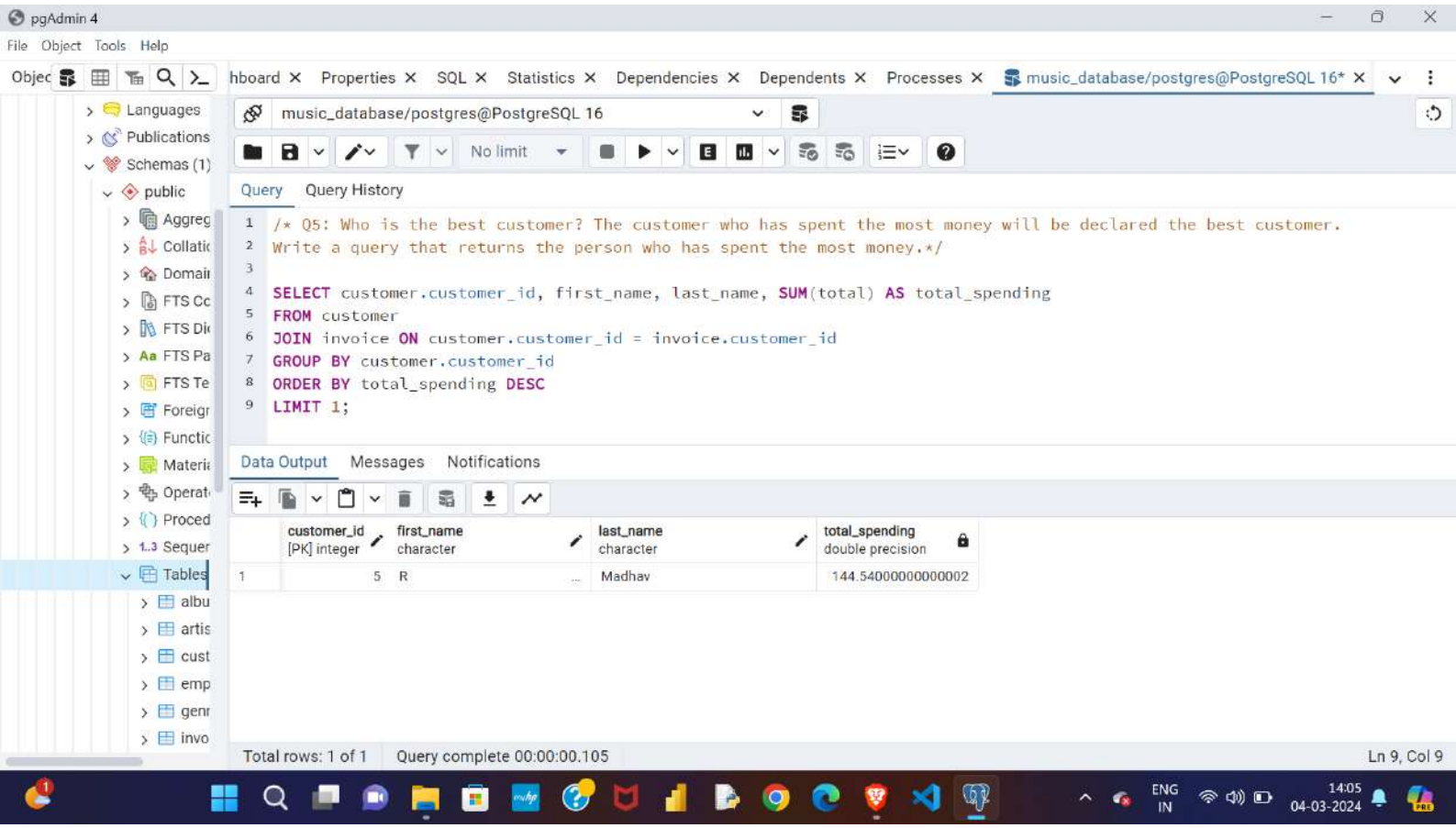

Artist
ArtistId
Name
Album
AlbumId
Title
ArtistId
Track
TrackId
Name
AlbumId
MediaTypeId
GenreId
Composer
Milliseconds
Bytes
UnitPrice
MediaType
MediaTypeId
Name
Genre
GenreId
Name
Playlist
PlaylistId
Name
PlaylistTrack
PlaylistId
TrackId
InvoiceLine
InvoiceLineId
InvoiceId
TrackId
UnitPrice
Quantity
Employee
EmployeeId
LastName
FirstName
Title
ReportsTo
BirthDate
HireDate
Address
City
State
Country
PostalCode
Phone
Fax
Email
Customer
CustomerId
FirstName
LastName
Company
Address
City
State
Country
PostalCode
Phone
Fax
Email
SupportRepId
Invoice
InvoiceId
CustomerId
InvoiceDate
BillingAddress
BillingCity
BillingState
BillingCountry
BillingPostalCode
Total

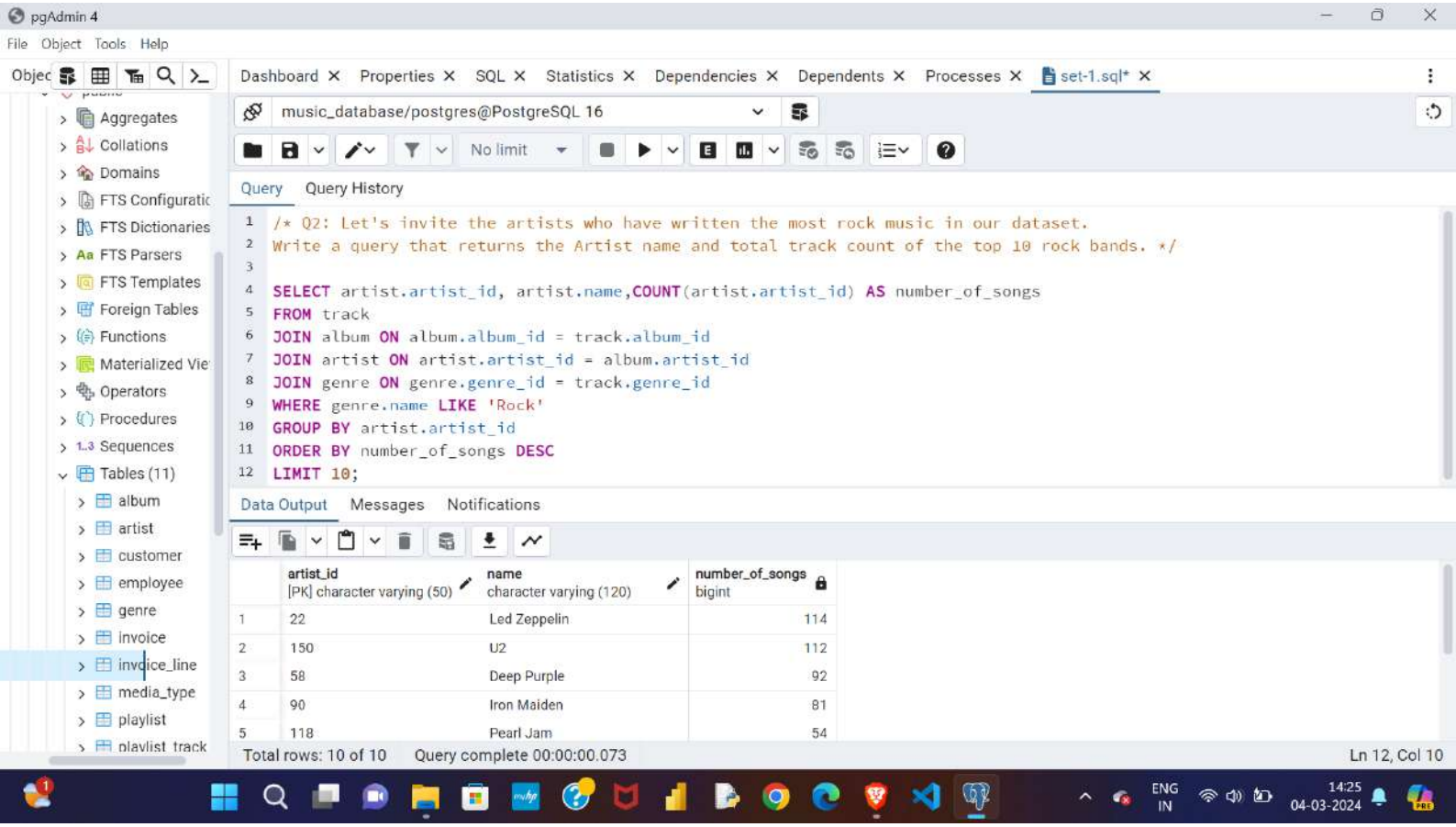

pgAdmin 4
File Object Tools Help
Object Explorer
Properties
SQL
Statistics
Dependencies
Dependents
Processes
music_database/postgres@PostgreSQL 16*
music_database/postgres@PostgreSQL 16
No limit
Languages
Publications
Schemas (1)
public
Aggregates
Collations
Domains
FTS Configuratic
FTS Dictionaries
FTS Parsers
FTS Templates
Foreign Tables
Functions
Materialized Vie
Operators
Procedures
Sequences
Tables (11)
album
artist
customer
employee
genre
invoice
Query
Query History
Scratch Pad
/* Q1: Who is the senior most employee based on job title? */
SELECT title, last_name, first_name
FROM employee
ORDER BY levels DESC
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 00:00:00.146
Ln 6, Col 8
32°
ENG
IN
13:09
04-03-2024

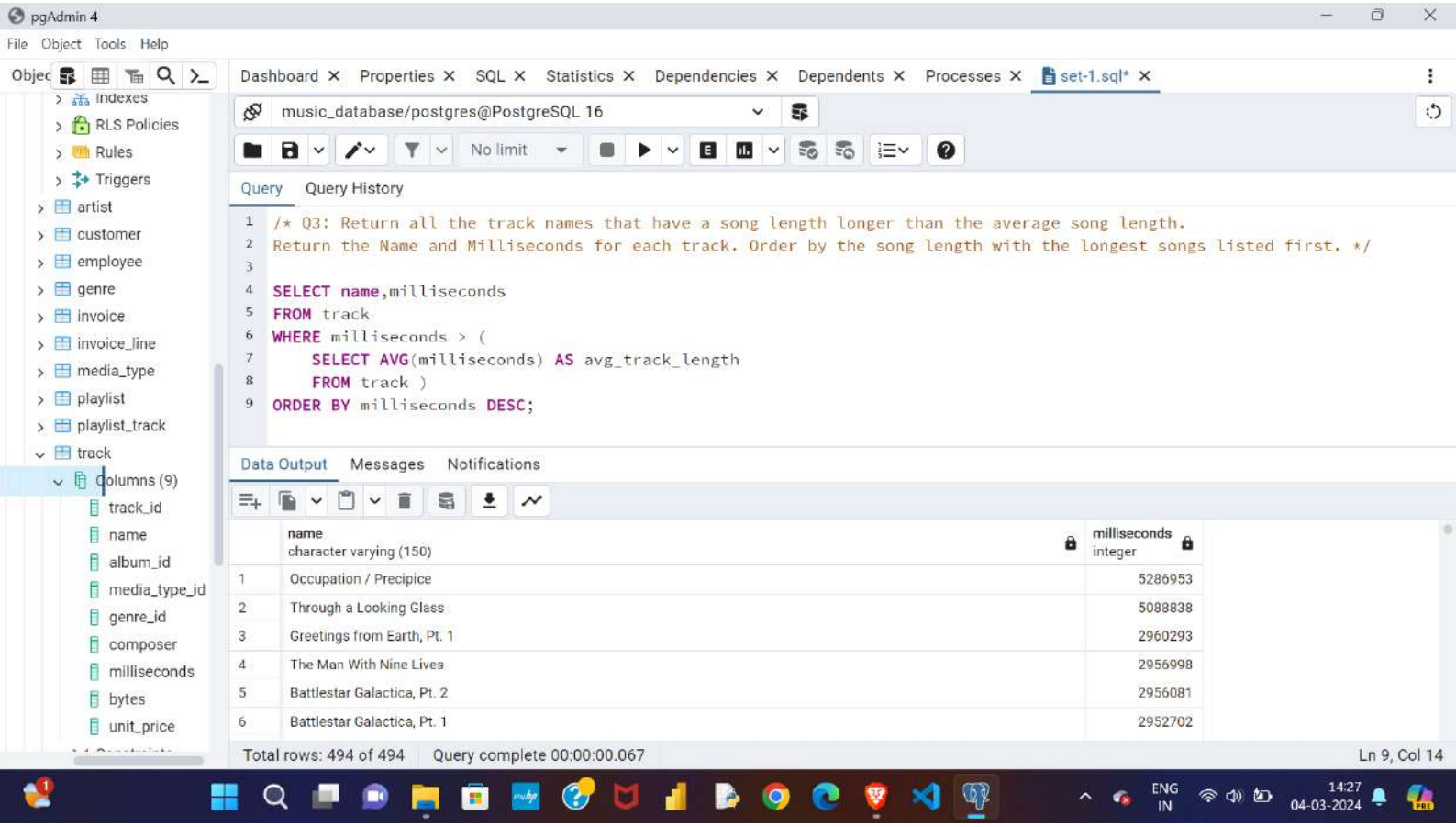

pgAdmin 4
File Object Tools Help
Object Explorer
Properties
SQL
Statistics
Dependencies
Dependents
Processes
music_database/postgres@PostgreSQL 16*
music_database/postgres@PostgreSQL 16
No limit
Languages
Publications
Schemas (1)
public
Aggregates
Collations
Domains
FTS Configuratic
FTS Dictionaries
FTS Parsers
FTS Templates
Foreign Tables
Functions
Materialized Vie
Operators
Procedures
Sequences
Tables (11)
album
artist
customer
employee
genre
invoice
Query
Query History
Scratch Pad
/* Q2: Which countries have the most Invoices? */
SELECT COUNT(*) AS c, billing_country
FROM invoice
GROUP BY billing_country
ORDER BY c DESC
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 00:00:00.117
Ln 6, Col 10
ENG
IN
13:57
04-03-2024

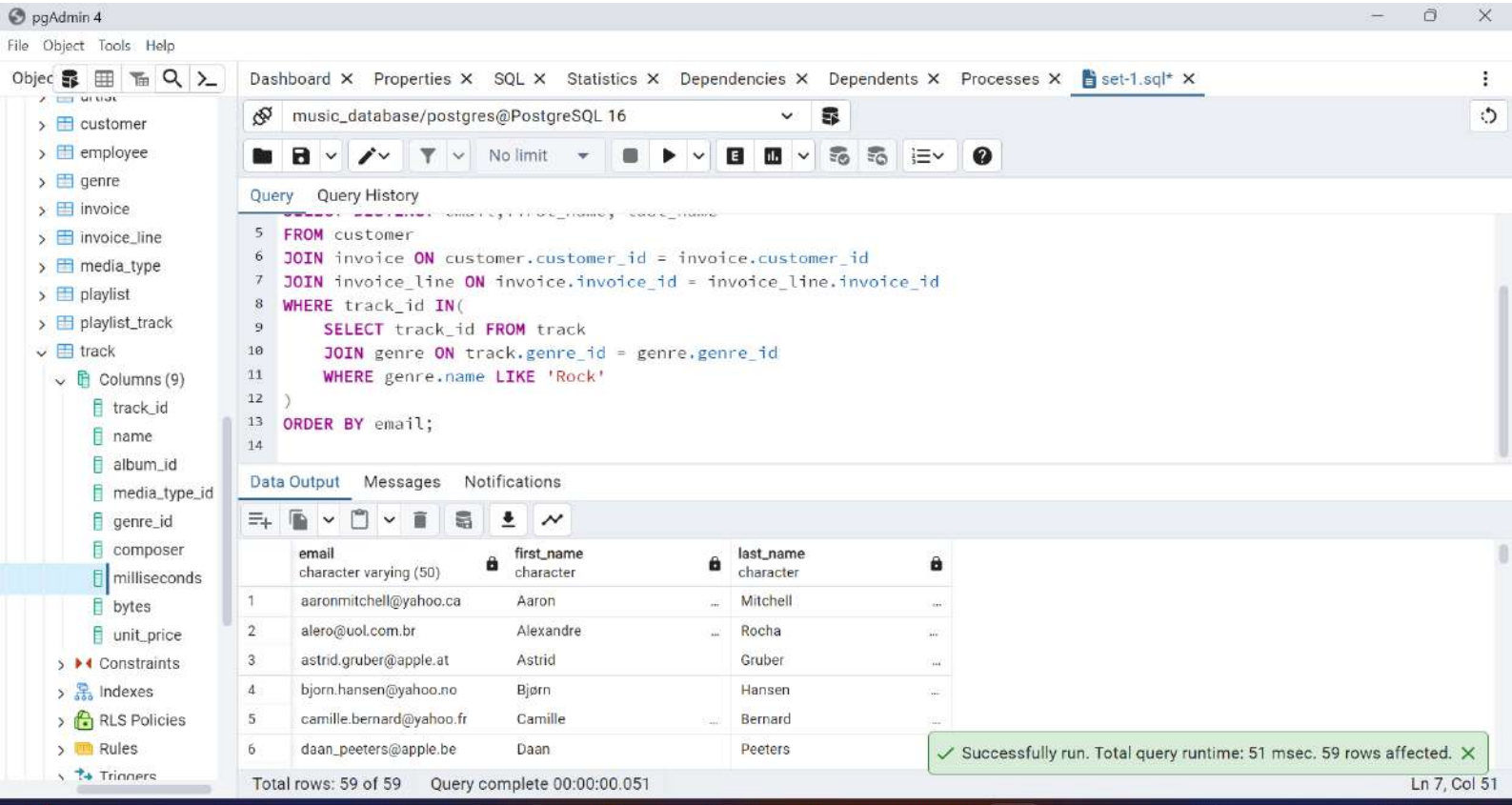

pgAdmin 4
File Object Tools Help
Object Explorer
Properties
SQL
Statistics
Dependencies
Dependents
Processes
music_database/postgres@PostgreSQL 16*
music_database/postgres@PostgreSQL 16
No limit
Languages
Publications
Schemas (1)
public
Aggregates
Collations
Domains
FTS Configuratic
FTS Dictionaries
FTS Parsers
FTS Templates
Foreign Tables
Functions
Materialized Vie
Operators
Procedures
Sequences
Tables (11)
album
artist
customer
employee
genre
invoice
Query
Query History
Scratch Pad
/* Q3: What are top 3 values of total invoice? */
SELECT total
FROM invoice
ORDER BY total DESC
limit 3
Data Output
Messages
Notifications
total
double precision
1 23.759999999999998
2 19.8
3 19.8
Successfully run. Total query runtime: 132 msec. 3 rows affected.
Total rows: 3 of 3
Query complete 00:00:00.132
Ln 6, Col 8
ENG
IN
14:00
04-03-2024

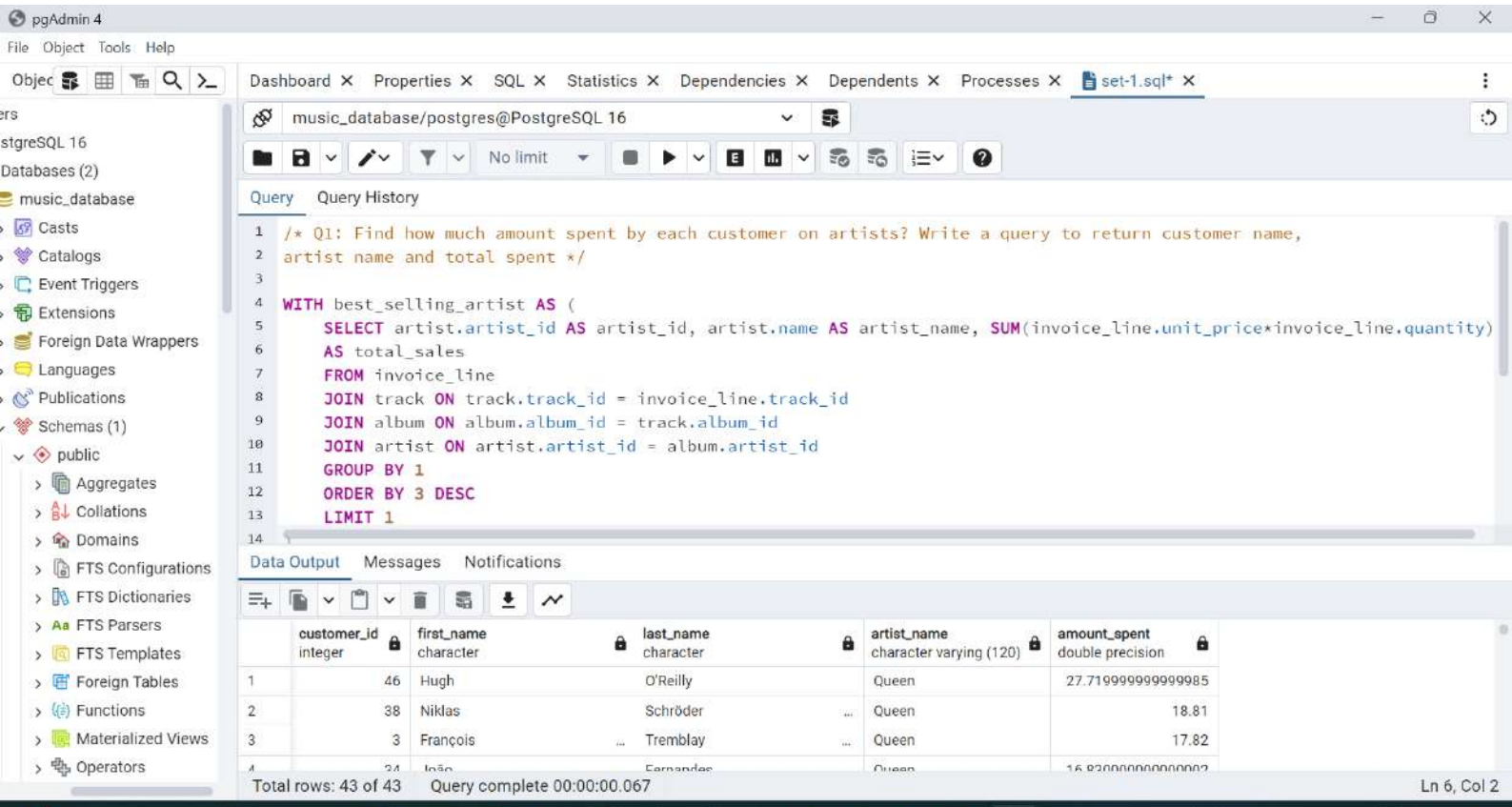

pgAdmin 4
File Object Tools Help
Properties
SQL
Statistics
Dependencies
Dependents
Processes
music_database/postgres@PostgreSQL 16*
music_database/postgres@PostgreSQL 16
No limit
Query
Query History
/* Q4: Which city has the best customers? We would like to throw a promotional Music Festival in the city we made the most
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 00:00:00.082
Ln 9, Col 9
ENG
IN
14:02
04-03-2024

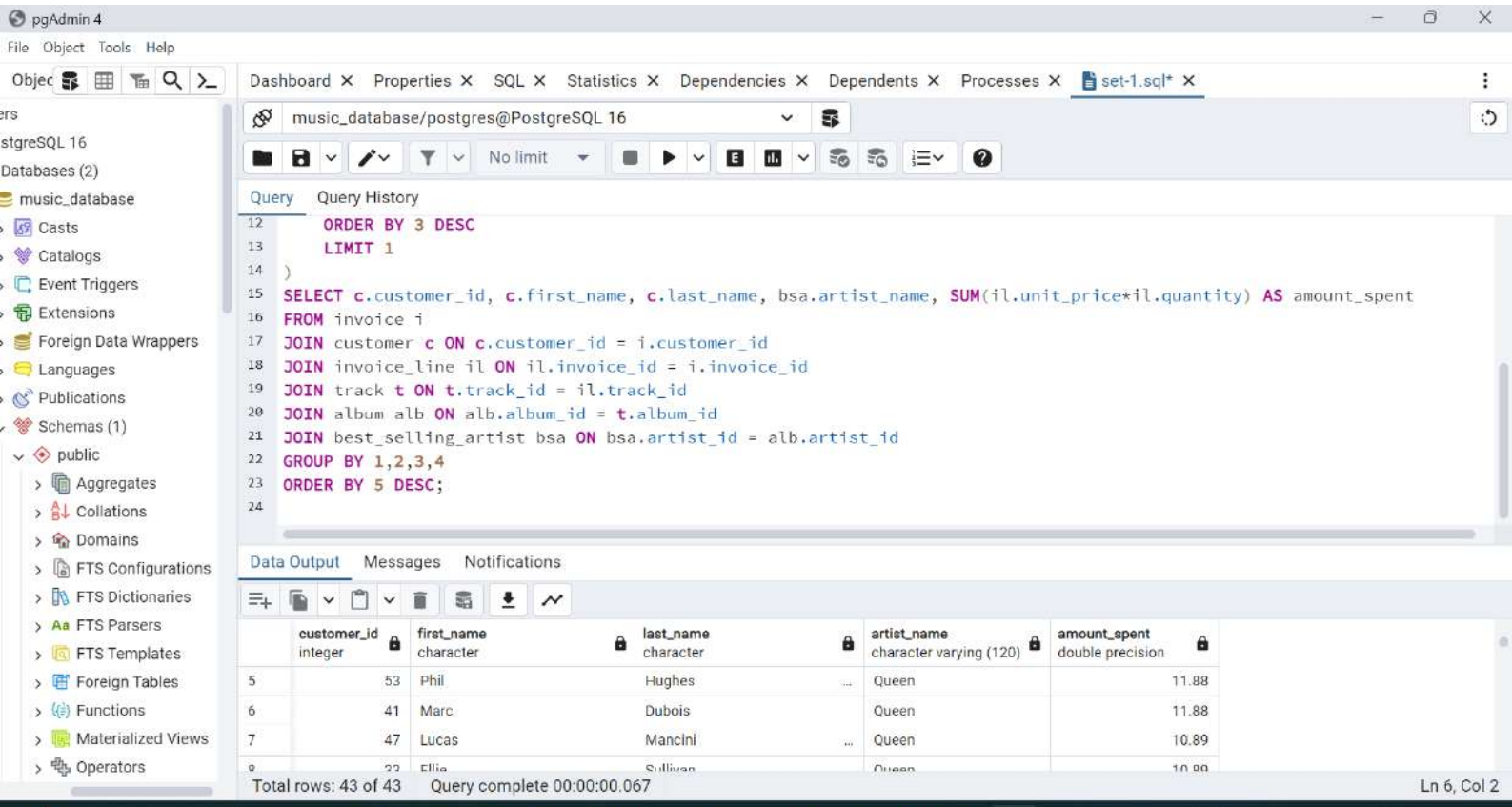

pgAdmin 4
File Object Tools Help
Properties
SQL
Statistics
Dependencies
Dependents
Processes
music_database/postgres@PostgreSQL 16*
music_database/postgres@PostgreSQL 16
No limit
Query
Query History
/* Q5: Who is the best customer? The customer who has spent the most money will be declared the best customer.
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 00:00:00.105
Ln 9, Col 9
ENG IN
14:05
04-03-2024

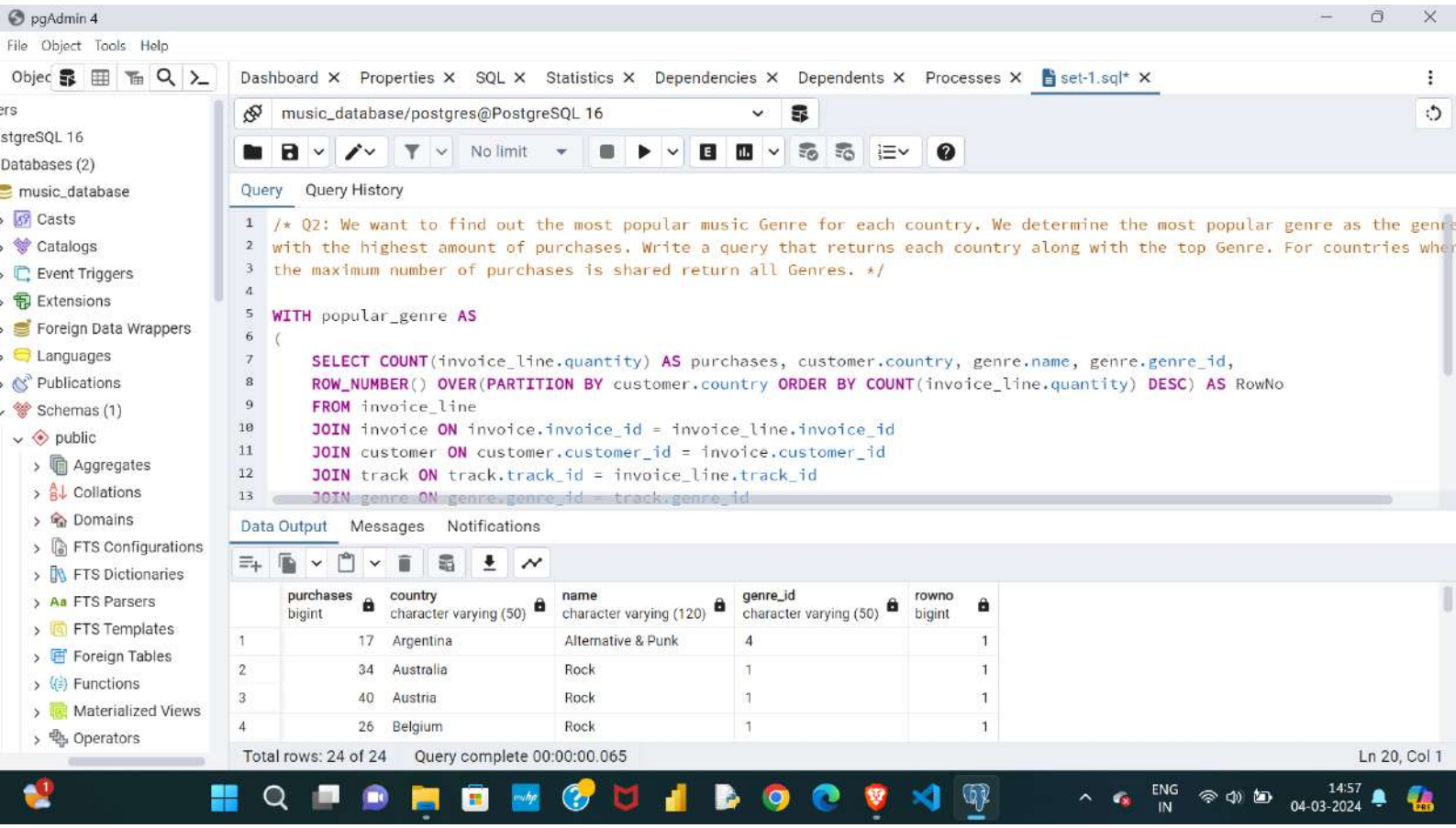

pgAdmin 4
File Object Tools Help
Dashboard
Properties
SQL
Statistics
Dependencies
Dependents
Processes
set-1.sql*
music_database/postgres@PostgreSQL 16
No limit
Query
Query History
/* Q2: Let's invite the artists who have written the most rock music in our dataset.
Write a query that returns the Artist 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
Total rows: 10 of 10
Query complete 00:00:00.073
Ln 12, Col 10
Aggregates
Collations
Domains
FTS Configuratic
FTS Dictionaries
FTS Parsers
FTS Templates
Foreign Tables
Functions
Materialized Vie
Operators
Procedures
Sequences
Tables (11)
album
artist
customer
employee
genre
invoice
invoice_line
media_type
playlist
playlist_track
ENG IN
14:25
04-03-2024

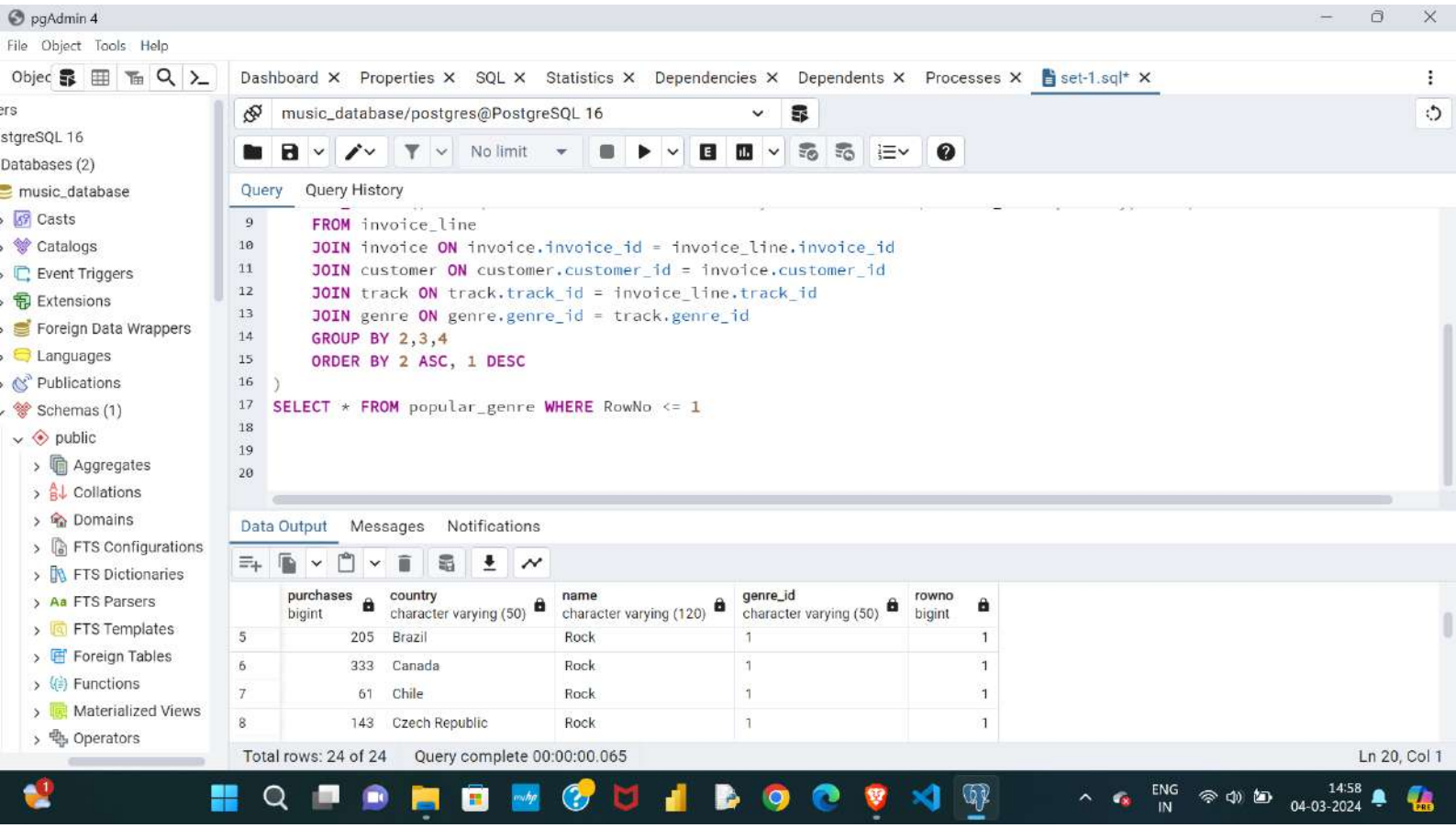

pgAdmin 4
File Object Tools Help
Dashboard
Properties
SQL
Statistics
Dependencies
Dependents
Processes
set-1.sql*
music_database/postgres@PostgreSQL 16
No limit
Query
Query History
/* Q3: Return all the track names that have a song length longer than the average song length.
Return the Name and Milliseconds 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
Total rows: 494 of 494
Query complete 00:00:00.067
Ln 9, Col 14
Indexes
RLS Policies
Rules
Triggers
artist
customer
employee
genre
invoice
invoice_line
media_type
playlist
playlist_track
track
Columns (9)
track_id
name
album_id
media_type_id
genre_id
composer
milliseconds
bytes
unit_price
ENG
IN
14:27
04-03-2024

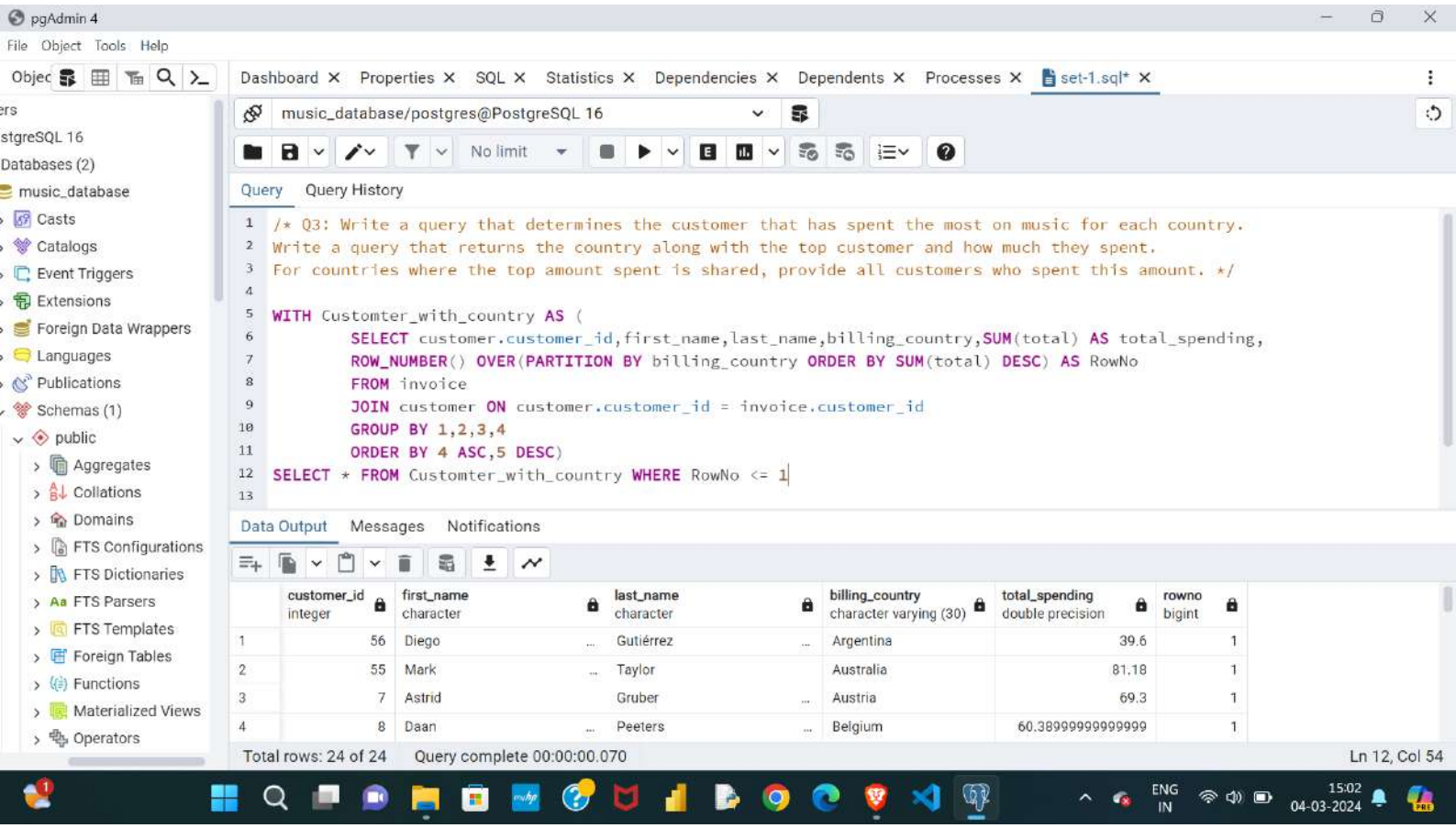

pgAdmin 4
File Object Tools Help
Dashboard
Properties
SQL
Statistics
Dependencies
Dependents
Processes
set-1.sql*
music_database/postgres@PostgreSQL 16
No limit
Query
Query History
customer
employee
genre
invoice
invoice_line
media_type
playlist
playlist_track
track
Columns (9)
track_id
name
album_id
media_type_id
genre_id
composer
milliseconds
bytes
unit_price
Constraints
Indexes
RLS Policies
Rules
Triggers
FROM customer
JOIN invoice ON customer.customer_id = invoice.customer_id
JOIN invoice_line ON invoice.invoice_id = invoice_line.invoice_id
WHERE track_id IN(
SELECT track_id FROM track
JOIN genre ON track.genre_id = genre.genre_id
WHERE genre.name LIKE 'Rock'
)
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 Bjørn Hansen
5 camille.bernard@yahoo.fr Camille Bernard
6 daan_peeters@apple.be Daan Peeters
Successfully run. Total query runtime: 51 msec. 59 rows affected.
Total rows: 59 of 59
Query complete 00:00:00.051
Ln 7, Col 51
ENG
IN
14:34
04-03-2024

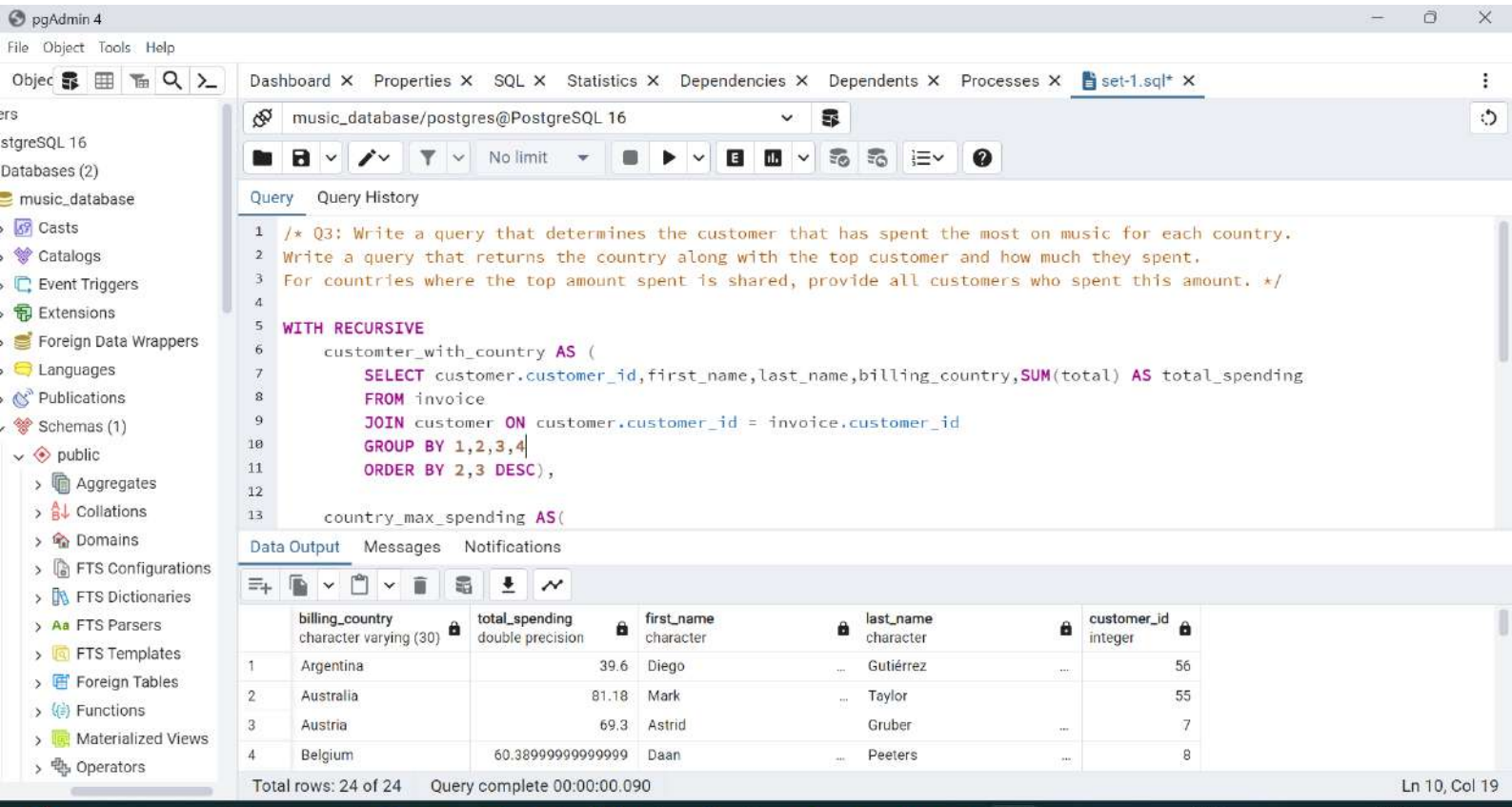

pgAdmin 4
File Object Tools Help
Dashboard Properties SQL Statistics Dependencies Dependents Processes set-1.sql*
music_database/postgres@PostgreSQL 16
No limit
Query Query History
/* Q1: Find how much amount spent by each customer on artists? Write a query to return customer name,
artist name and total 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_sales
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
Data Output Messages Notifications
customer_id integer
first_name character
last_name character
artist_name character varying (120)
amount_spent double precision
1 46 Hugh O'Reilly Queen 27.719999999999985
2 38 Niklas Schröder Queen 18.81
3 3 François Tremblay Queen 17.82
Total rows: 43 of 43
Query complete 00:00:00.067
Ln 6, Col 2
Databases (2)
music_database
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
ENG
IN
14:55
04-03-2024

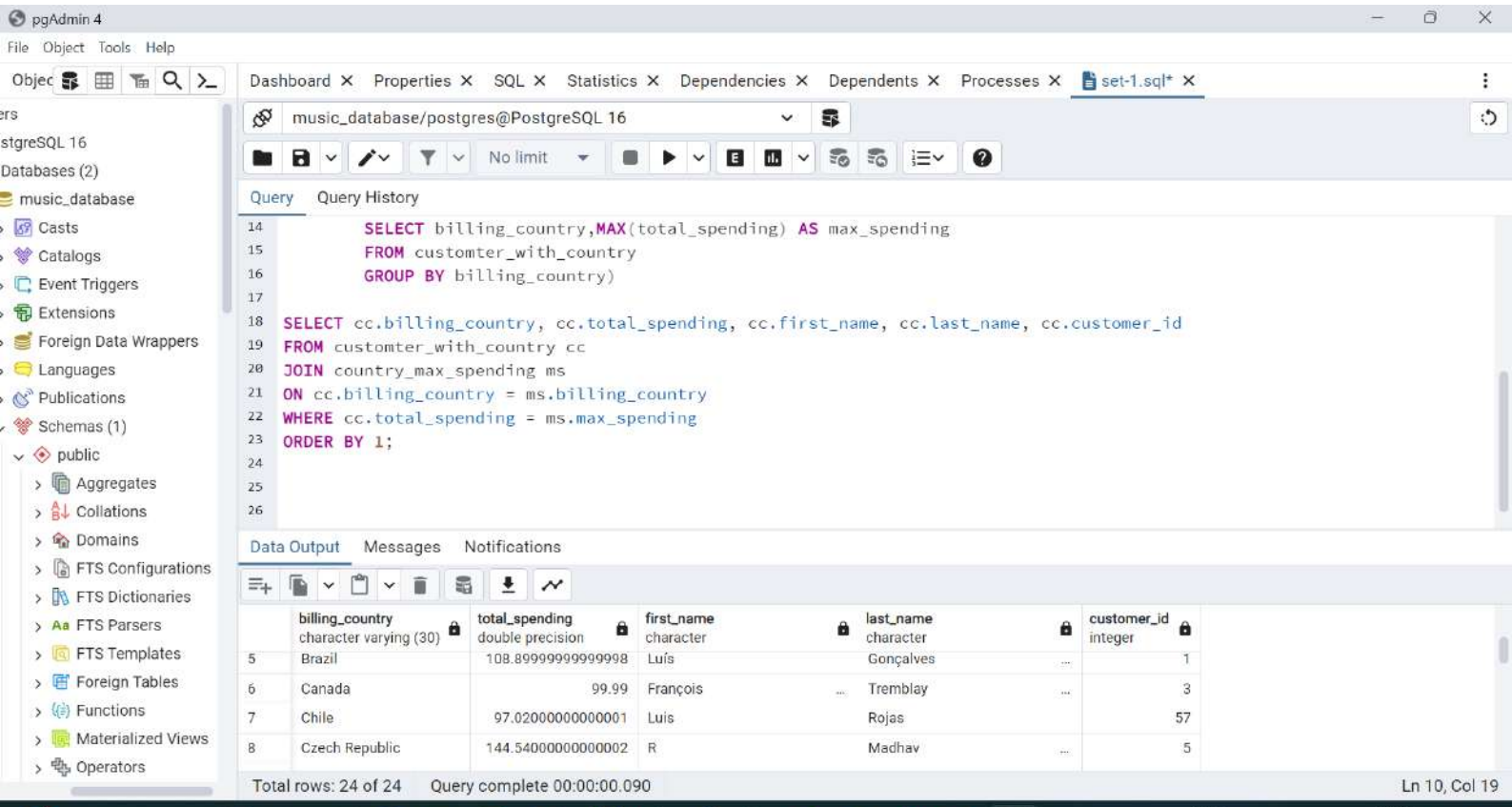

pgAdmin 4
File Object Tools Help
Dashboard
Properties
SQL
Statistics
Dependencies
Dependents
Processes
set-1.sql*
music_database/postgres@PostgreSQL 16
No limit
Query
Query History
ORDER BY 3 DESC
LIMIT 1
)
SELECT c.customer_id, c.first_name, c.last_name, bsa.artist_name, SUM(il.unit_price*il.quantity) 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_name character varying (120)
amount_spent double precision
5 53 Phil Hughes Queen 11.88
6 41 Marc Dubois Queen 11.88
7 47 Lucas Mancini Queen 10.89
Total rows: 43 of 43
Query complete 00:00:00.067
Ln 6, Col 2
Databases (2)
music_database
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
ENG
IN
14:55
04-03-2024


pgAdmin 4
File Object Tools Help
Dashboard
Properties
SQL
Statistics
Dependencies
Dependents
Processes
set-1.sql*
music_database/postgres@PostgreSQL 16
No limit
Query
Query History
/* Q2: 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 whe
the maximum number of purchases is shared return all Genres. */
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
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 00:00:00.065
Ln 20, Col 1
music_database
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
ENG
IN
14:57
04-03-2024


pgAdmin 4
File Object Tools Help
Dashboard
Properties
SQL
Statistics
Dependencies
Dependents
Processes
set-1.sql*
music_database/postgres@PostgreSQL 16
No limit
Query
Query History
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)
rowno bigint
5 205 Brazil Rock 1 1
6 333 Canada Rock 1 1
7 61 Chile Rock 1 1
8 143 Czech Republic Rock 1 1
Total rows: 24 of 24
Query complete 00:00:00.065
Ln 20, Col 1
music_database
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
ENG
IN
14:58
04-03-2024


pgAdmin 4
File Object Tools Help
Dashboard
Properties
SQL
Statistics
Dependencies
Dependents
Processes
set-1.sql*
music_database/postgres@PostgreSQL 16
No limit
Query
Query History
/* Q3: 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. */
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)
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
Total rows: 24 of 24
Query complete 00:00:00.070
Ln 12, Col 54
Schemas (1)
public
Aggregates
Collations
Domains
FTS Configurations
FTS Dictionaries
FTS Parsers
FTS Templates
Foreign Tables
Functions
Materialized Views
Operators
Foreign Data Wrappers
Languages
Publications
Event Triggers
Extensions
Catalogs
Casts
music_database
Databases (2)
PostgreSQL 16
ENG
IN
15:02
04-03-2024


pgAdmin 4
File Object Tools Help
Dashboard
Properties
SQL
Statistics
Dependencies
Dependents
Processes
set-1.sql*
music_database/postgres@PostgreSQL 16
No limit
Query
Query History
/* Q3: 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. */
WITH RECURSIVE
customter_with_country AS (
SELECT customer.customer_id,first_name,last_name,billing_country,SUM(total) AS total_spending
FROM invoice
JOIN customer ON customer.customer_id = invoice.customer_id
GROUP BY 1,2,3,4
ORDER BY 2,3 DESC),
country_max_spending AS(
Data Output
Messages
Notifications
billing_country character varying (30)
total_spending double precision
first_name character
last_name character
customer_id integer
1 Argentina 39.6 Diego Gutiérrez 56
2 Australia 81.18 Mark Taylor 55
3 Austria 69.3 Astrid Gruber 7
4 Belgium 60.38999999999999 Daan Peeters 8
Total rows: 24 of 24
Query complete 00:00:00.090
Ln 10, Col 19
Databases (2)
music_database
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
ENG
IN
15:02
04-03-2024


pgAdmin 4
File Object Tools Help
Dashboard
Properties
SQL
Statistics
Dependencies
Dependents
Processes
set-1.sql*
music_database/postgres@PostgreSQL 16
No limit
Query
Query History
SELECT billing_country,MAX(total_spending) AS max_spending
FROM customter_with_country
GROUP BY billing_country)
SELECT cc.billing_country, cc.total_spending, cc.first_name, cc.last_name, cc.customer_id
FROM customter_with_country cc
JOIN country_max_spending ms
ON cc.billing_country = ms.billing_country
WHERE cc.total_spending = ms.max_spending
ORDER BY 1;
Data Output
Messages
Notifications
billing_country character varying (30)
total_spending double precision
first_name character
last_name character
customer_id integer
5 Brazil 108.89999999999998 Luís Gonçalves 1
6 Canada 99.99 François Tremblay 3
7 Chile 97.02000000000001 Luis Rojas 57
8 Czech Republic 144.54000000000002 R Madhav 5
Total rows: 24 of 24
Query complete 00:00:00.090
Ln 10, Col 19
Databases (2)
music_database
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
ENG IN
15:03
04-03-2024