



SQL PROJECT- MUSIC STORE DATA ANALYSIS

1. Who is the senior most employee based on job title?

pgAdmin 4

File Object Tools Help

Object Explorer

- 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

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

Music_database/postgres@PostgreSQL 15

Query Query History

```

1 1. Who is the senior most employee based on job title?
2
3 select * from employee
4 order by levels desc
5 limit 1;

```

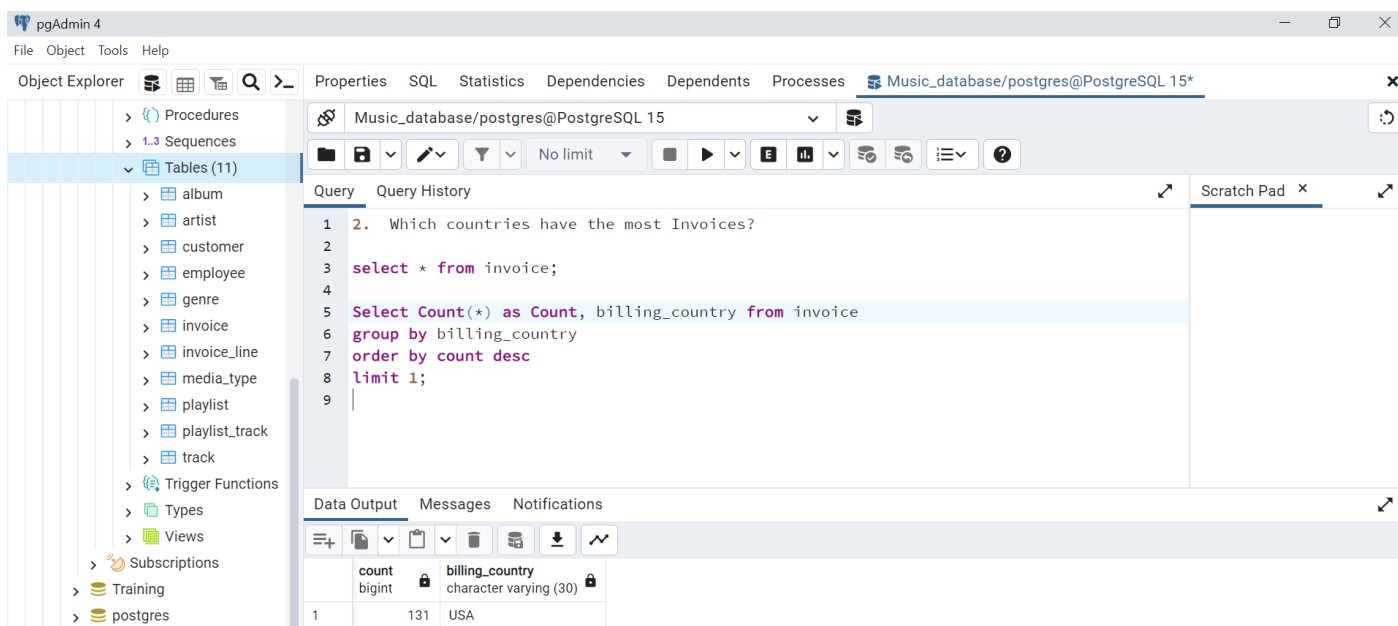
Scratch Pad

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 varyir
1	9	Madan	Mohan	Senior General Manager	[null]	L7

Subscriptions

2. Which countries have the most Invoices?



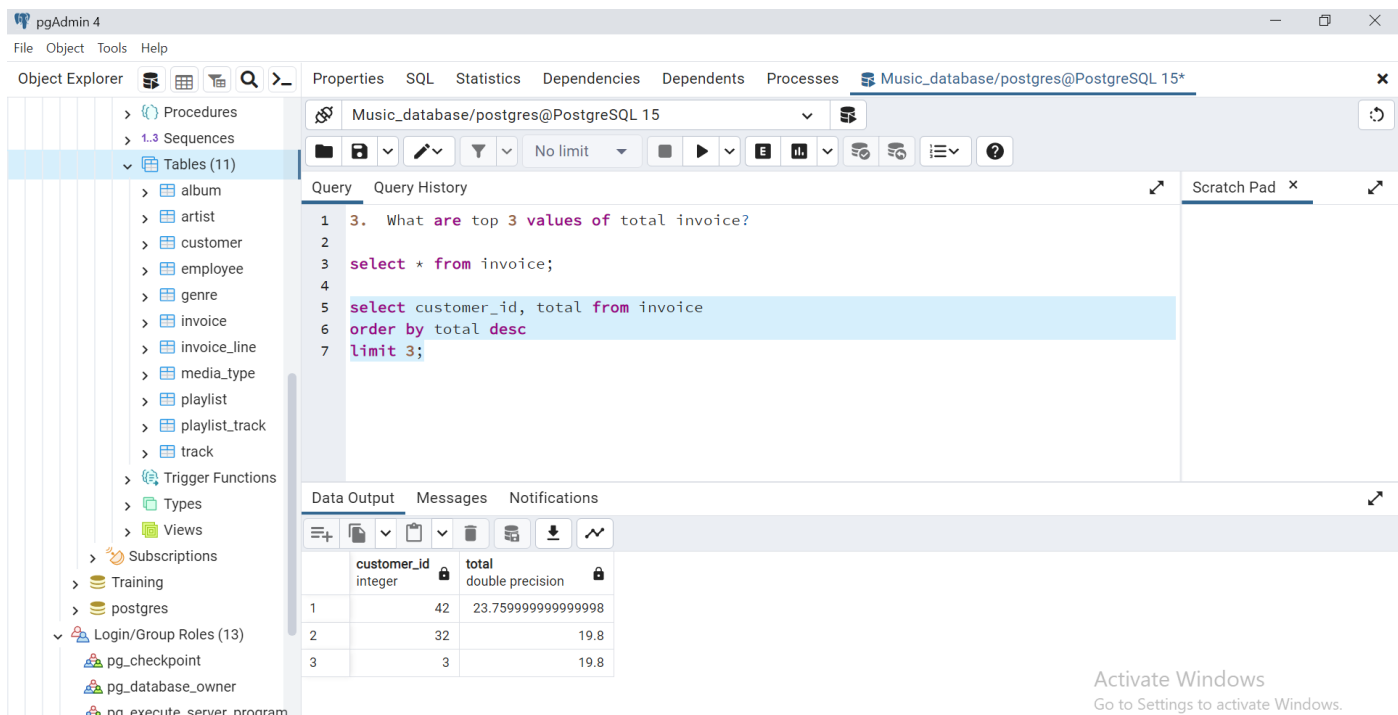
The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database structure, including tables like album, artist, customer, employee, genre, invoice, invoice_line, media_type, playlist, playlist_track, track, trigger_functions, types, views, subscriptions, training, and postgres. The main pane shows the SQL query editor with the following query:

```
1 2. Which countries have the most Invoices?
2
3 select * from invoice;
4
5 Select Count(*) as Count, billing_country from invoice
6 group by billing_country
7 order by count desc
8 limit 1;
9
```

The Data Output tab shows the results of the query:

	count bigint	billing_country character varying (30)
1	131	USA

3. What are top 3 values of total invoice?



The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database structure, including tables like album, artist, customer, employee, genre, invoice, invoice_line, media_type, playlist, playlist_track, track, trigger_functions, types, views, subscriptions, training, postgres, login/group roles, pg_checkpoint, pg_database_owner, and pg_execute_server_program. The main pane shows the SQL query editor with the following query:

```
1 3. What are top 3 values of total invoice?
2
3 select * from invoice;
4
5 select customer_id, total from invoice
6 order by total desc
7 limit 3;
```

The Data Output tab shows the results of the query:

	customer_id integer	total double precision
1	42	23.759999999999998
2	32	19.8
3	3	19.8

4. 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.

pgAdmin 4

File Object Tools Help

Object Explorer

- Procedures
- Sequences
- Tables (11)
 - album
 - artist
 - customer
 - employee
 - genre
 - invoice
 - invoice_line
 - media_type
 - playlist
 - playlist_track
 - track
- Trigger Functions
- Types
- Views
- Subscriptions
- Training
- postgres
- Login/Group Roles (13)
 - pg_checkpoint
 - pg_database_owner
 - pg_execute_server_program
 - pg_monitor

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

Music_database/postgres@PostgreSQL 15

Query Query History

```

1 4. Which city has the best customers? We would like to throw a promotional Music
2 Festival in the city we made the most money. Write a query that returns one city that
3 has the highest sum of invoice totals. Return both the city name & sum of all
4 invoice totals.
5
6 Select * from invoice;
7
8 Select billing_city, sum(total) as Sum_invoice from invoice
9 group by billing_city
10 order by Sum_invoice desc;

```

Data Output Messages Notifications

	billing_city	sum_invoice
	character varying (30)	double precision
1	Prague	273.240000000000007
2	Mountain View	169.29
3	London	166.32
4	Berlin	158.4
5	Paris	151.47
6	São Paulo	129.69
7	Dublin	114.830000000000007

Total rows: 53 of 53 Query complete 00:00:00.081 Ln 7, Col 1

Activate Windows
Go to Settings to activate Windows.

5. 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.

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

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

Music_database/postgres@PostgreSQL 15

Query Query History

```

1 5. Who is the best customer? The customer who has spent the most money will be declared the best customer.
2 Write a query that returns the person who has spent the most money.
3 select * from customer;
4 select * from invoice;
5 select customer.customer_id, concat(customer.first_name, ' ', customer.last_name), sum(invoice.total) as total
6 from customer
7 join invoice on customer.customer_id = invoice.customer_id
8 group by customer.customer_id
9 order by total desc
10 limit 1;

```

Data Output Messages Notifications

	customer_id	concat	total
	[PK] integer	text	double precision
1	5	R Madhav	144.540000000000002

6. 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.

File Object Tools Help

Object Explorer

- 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
- Training
- postgres
- Login/Group Roles (13)

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

Music_database/postgres@PostgreSQL 15

Query Query History

```

1 6. Write query to return the email, first name, last name, & Genre of all Rock Music listeners. Return your
2 list ordered alphabetically by email starting with A.
3
4 Select Distinct email, Concat(first_name,' ',last_name) as customer_name from customer
5 join invoice on customer.customer_id = invoice.customer_id
6 join invoice_line on invoice.invoice_id = invoice_line.invoice_id
7 where track_id
8 In (Select track.track_id from track join genre on track.genre_id = genre.genre_id
9 where genre.name = 'Rock'
10 ) order by email;

```

Data Output Messages Notifications

	email character varying (50)	customer_name text	
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	Björn Hansen	...
5	camille.bernard@yahoo.fr	Camille Bernard	...
6	daan.peeters@apple.be	Daan Peeters	...

Total rows: 59 of 59 Query complete 00:00:00.150 Ln 3, Col 1

Activate Windows
Go to Settings to activate Windows.

7. 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.

File Object Tools Help

Object Explorer

- 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
- Training
- postgres
- Login/Group Roles (13)

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

Music_database/postgres@PostgreSQL 15

Query Query History

```

1 7. Lets invite the artists who have written the most rock music in our dataset. Write a query that returns the
2 Artist name and total track count of the top 10 rock bands.
3 select * from track
4 Select artist.artist_id, artist.name, count(artist.artist_id) as Total_track from artist
5 join album on artist.artist_id = album.artist_id
6 join track on album.album_id = track.album_id
7 Join genre on track.genre_id = genre.genre_id
8 where genre.name like 'Rock'
9 group by artist.artist_id
10 order by Total_track Desc
11 limit 10;
12

```

Data Output Messages Notifications

	artist_id [PK] character varying (50)	name character varying (120)	number_of_songs bigint
1	22	Lea 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

Total rows: 10 of 10 Query complete 00:00:00.112 Ln 13, Col 1

Activate Windows
Go to Settings to activate Windows.

8. 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.

File Object Tools Help

Object Explorer Properties SQL Statistics Dependencies Dependents Processes Music_database/postgres@PostgreSQL 15*

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 Training postgres

Music_database/postgres@PostgreSQL 15

Query Query History

```

1 8. Return all the track names that have a song length longer than the average song length. Return the Name and
2 Milliseconds for each track. Order by the song length with the longest songs listed first.
3 Select * from track;
4 Select name, milliseconds from track
5 where milliseconds > (select avg(milliseconds) from track)
6 order by milliseconds desc;
7

```

Data Output Messages Notifications

	name	milliseconds
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

Total rows: 494 of 494 Query complete 00:00:00.090 Ln 3, Col 1

9. Find how much amount spent by each customer on artists? Write a query to return customer name, artist name and total spent.

File Object Tools Help

Object Explorer Properties SQL Statistics Dependencies Dependents Processes Music_database/postgres@PostgreSQL 15*

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 Types Views Subscriptions Training postgres Login/Group Roles Tablespace

Music_database/postgres@PostgreSQL 15

Query Query History

```

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

```

Data Output Messages Notifications

	customer_id	first_name	last_name	artist_name	amount_spent
1	46	Hugh	O'Reilly	Queen	27.719999999999985
2	6	Helena	Holy	Red Hot Chili Peppers	19.799999999999997
3	38	Niklas	Schröder	Queen	18.81

Total rows: 119 of 119 Query complete 00:00:00.170 Ln 24, Col 17

10. 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.

File Object Tools Help

Object Explorer SQL Statistics Dependencies Dependents Processes Music_database/postgres@PostgreSQL 15*

Music_database/postgres@PostgreSQL 15

Query Query History

10. 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.

```

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)	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
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 00:00:00.085 Ln 18, Col 1

Activate Windows
Go to Settings to activate Windows.

11. 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

File Object Tools Help

Object Explorer SQL Statistics Dependencies Dependents Processes Music_database/postgres@PostgreSQL 15*

Music_database/postgres@PostgreSQL 15

Query Query History

11. 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 Customer_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 Customer_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	60.389999999999999	1
5	1	Luis	Gonçalves	Brazil	108.899999999999998	1
6	3	François	Tremblay	Canada	99.99	1
7	57	Luis	Rojas	Chile	97.020000000000001	1
8	5	R	Madhav	Czech Republic	144.540000000000002	1
9	9	Kara	Nielsen	Denmark	37.619999999999999	1
10	44	Terhi	Hämäläinen	Finland	79.2	1
11	42	Wyatt	Girard	France	99.99	1
12	37	Fynn	Zimmermann	Germany	94.050000000000001	1
13	45	Ladislav	Kovács	Hungary	78.21	1

Total rows: 24 of 24 Query complete 00:00:00.083 Ln 4, Col 1

Activate Windows
Go to Settings to activate Windows.