



MUSIC STORE ANALYSIS




Using SQL



Project Description

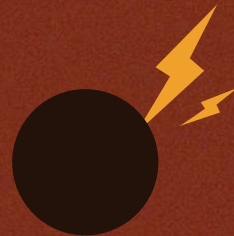


This project aims to analyze sales data from a music store to identify key trends and insights. The focus areas include determining the most popular artists, songs, genres, and countries for music purchases. By leveraging SQL queries to interrogate the database, the project seeks to provide actionable insights that can help in strategic decision-making and improving business performance.



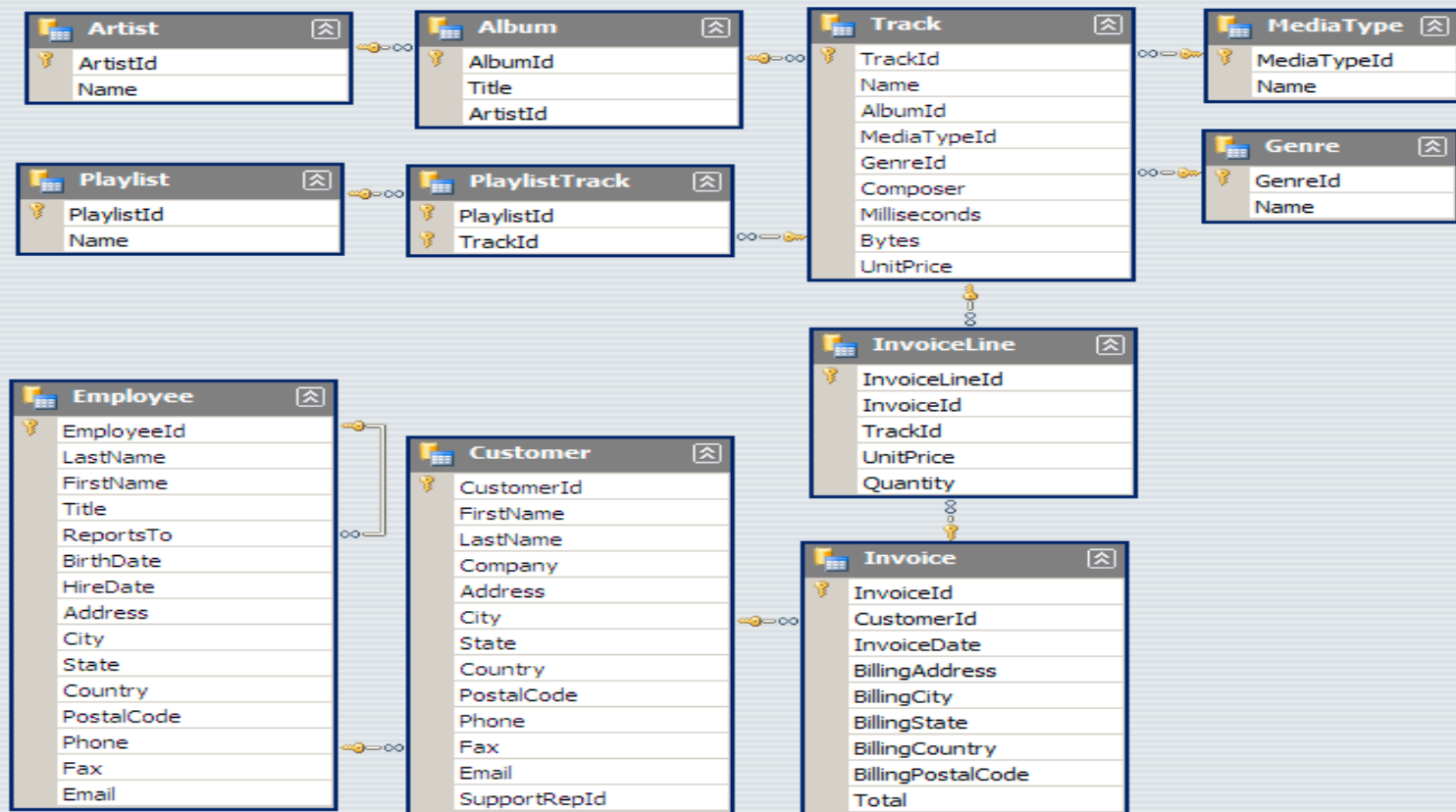


Queries















1. Who is the senior most employee based on job title?
2. Which countries have the most Invoices?
3. What are top 3 values of total invoice?
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.
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.
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.
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.
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.
9. List top 10 most popular artists?
10. List top 5 most popular songs?
11. What are the most popular countries for music purchase?

Schema





List Of Tables

- ▼  Tables (11)
 - >  album
 - >  artist
 - >  customer
 - >  employee
 - >  genre
 - >  invoice
 - >  invoice_line
 - >  media_type
 - >  playlist
 - >  playlist_track
 - >  track



Who is the senior most employee based on job title?

music_database/postgres@PostgreSQL 16

Query Query History Scratch Pad x

```
1 Q1. Who is the senior most employee based on job title
2
3 select * from employee
4 order by levels desc
5 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)	birthdate timestamp without time zone
1	9	Madan	Mohan	Senior General Manager	[null]	L7	1961-01-26 00:00:00

Which countries have the most Invoices?

Query Query History

```
1  v  Q2. Which country have most invoices?
2
3  select count(*) as s , billing_country
4  from invoice
5  group by billing_country
6  order by s desc;
7
8  |
```

Data Output Messages Notifications

	s 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 00:00:00.090



What are top 3 values of total invoice?



music_database/postgres@PostgreSQL 16

Query Query History

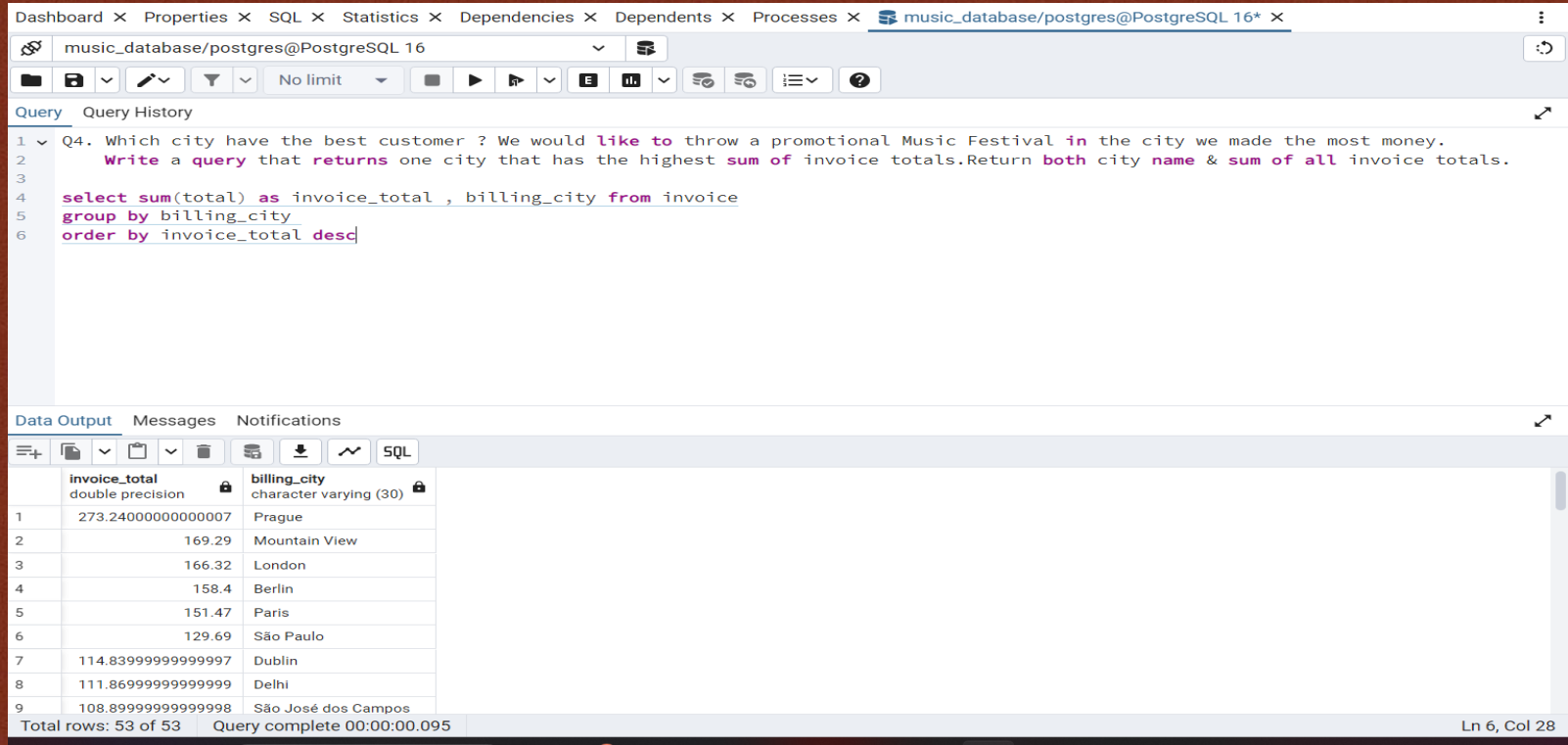
```
1 Q3. What are the top 3 values of total invoices ?
2
3 select total from invoice
4 order by total desc limit 3
```

Data Output Messages Notifications

	total double precision	
1	23.759999999999998	
2	19.8	
3	19.8	



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.



The screenshot shows a PostgreSQL database interface with a query editor and a results table. The query editor contains a SQL query to find the city with the highest sum of invoice totals. The results table displays the top 9 cities based on the sum of invoice totals.

Query:

```
Q4. Which city have the best customer ? 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 city name & sum of all invoice totals.  
  
select sum(total) as invoice_total , billing_city from invoice  
group by billing_city  
order by invoice_total desc
```

Data Output:

	invoice_total double precision	billing_city character varying (30)
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

Total rows: 53 of 53 Query complete 00:00:00.095 Ln 6, Col 28

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.

Dashboard × Properties × SQL × Statistics × Dependencies × Dependents × Processes × music_database/postgres@PostgreSQL 16* ×

music_database/postgres@PostgreSQL 16

Query Query History

Q5. Who is the best customer ? The customer who has spent the most money will be declared as best customer.
Write a query that returns the person who has spent the most money.

```
select customer.customer_id , customer.first_name , customer.last_name , sum(invoice.total) as total
from customer join invoice on customer.customer_id = invoice.customer_id
group by customer.customer_id
order by total desc limit 1
```

Data Output Messages Notifications

	customer_id [PK] integer	first_name character	last_name character	total double precision
1	5	R	Madhav	144.54000000000002

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'.

music_database/postgres@PostgreSQL 16

Query Query History

```
1 Q6. Write a query to return the email , first_name , last_name & Genre of all Rock
2 Return your list ordered alphabetically by email starting with A.
3
4
5 select DISTINCT email , first_name , last_name 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 = 'Rock'
12 )
13 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
7	diego.gutierrez@yahoo.ar	Diego	Gutiérrez
8	dmiller@comcast.com	Dan	Miller
9	dominiquelefebvre@gmail.c...	Dominique	Lefebvre

Total rows: 59 of 59 Query complete 00:00:00.078

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.

Dashboard X Properties X SQL X Statistics X Dependencies X Dependents X Processes X

music_database/postgres@PostgreSQL 16

Query Query History

```
1 -- Let's invite the artists who have written the most rock music in our dataset
2 -- Write a query that return the artist name and total track count of the top 1
3
4 select artist.artist_id,artist.name , COUNT(artist.artist_id) as number_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 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 00:00:00.085

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.

Dashboard × Properties × SQL × Statistics × Dependencies × Dependents × Processes × music_database/postgres@PostgreSQL 16* ×

music_database/postgres@PostgreSQL 16

No limit

Query History

```
1 -- 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 .Order by the song length with the longest songs listed first
3
4 select name, milliseconds
5 from track where milliseconds >
6 (select avg(milliseconds) as average_track_length from track)
7 order by milliseconds desc;
8
9
10
11
12
13
```

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	2927802
9	Take the Celestra	2927677

Total rows: 494 of 494 Query complete 00:00:00.081

List top 10 most popular artists?

music_database/postgres@PostgreSQL 16

Query Query History

```
1  -- List top 10 most popular artists?
2
3  SELECT COUNT(invoice_line.quantity) AS purchases, artist.name AS artist_name
4  FROM invoice_line
5  JOIN track ON track.track_id = invoice_line.track_id
6  JOIN album ON album.album_id = track.album_id
7  JOIN artist ON artist.artist_id = album.artist_id
8  GROUP BY artist.name
9  ORDER BY purchases DESC
10 LIMIT 10;
```

Data Output Messages Notifications

	purchases bigint	artist_name character varying (120)
1	192	Queen
2	187	Jimi Hendrix
3	130	Nirvana
4	130	Red Hot Chili Peppers
5	129	Pearl Jam
6	124	Guns N' Roses
7	124	AC/DC
8	121	Foo Fighters
9	117	The Rolling Stones

Total rows: 10 of 10 Query complete 00:00:00.197

List top 5 most popular songs?

Dashboard X Properties X SQL X Statistics X Dependencies X Dependents X Processes X

music_database/postgres@PostgreSQL 16

No limit

Query Query History

```
1  --List top 5 the most popular song?
2
3
4  SELECT COUNT(invoice_line.quantity) AS purchases, track.name AS song_name
5  FROM invoice_line
6  JOIN track ON track.track_id = invoice_line.track_id
7  GROUP BY track.name
8  ORDER BY purchases DESC
9  LIMIT 5;
```

Data Output Messages Notifications

	purchases bigint	song_name character varying (150)
1	33	War Pigs
2	14	Changes
3	14	Are You Experienced?
4	14	Highway Chile
5	13	Hey Joe

Total rows: 5 of 5 Query complete 00:00:00.083

What are the most popular countries for music purchase?

Dashboard × Properties × SQL × Statistics × Dependencies × Dependents × Processes × music_d

music_database/postgres@PostgreSQL 16

Query Query History

```
1 --What are the most popular countries for music purchases?
2
3 SELECT customer.country AS country, ROUND(SUM(invoice.total)) AS total_sales
4 FROM invoice
5 JOIN customer ON invoice.customer_id = customer.customer_id
6 GROUP BY customer.country
7 ORDER BY total_sales DESC
8 LIMIT 10;
```

Data Output Messages Notifications

	country character varying (50)	total_sales double precision
1	USA	1040
2	Canada	536
3	Brazil	428
4	France	389
5	Germany	335
6	Czech Republic	273
7	United Kingdom	246
8	Portugal	185
9	India	183

Total rows: 10 of 10 Query complete 00:00:00.087



Insights of the Project

1. Senior Most Employee: Madan Mohan (Senior General Manager)
2. Top Country for Invoices: USA (131 invoices)
3. Top Invoice Total: \$23.76 (Invoice 183, France)
4. Best Customer City: Prague (\$273.24 total sales)
5. Best Customer: František Wichterlová (\$144.54 total spend)
6. Top Rock Artist: Led Zeppelin (114 tracks)
7. Longest Track: "Occupation / Precipice" (5286953 ms)
8. Most Popular Artist: Queen (192 purchases)
9. Most Popular Song: "War Pigs" (33 purchases)
10. Most Popular Countries for Purchases: USA (1051 purchases), Canada (541 purchases)
11. Rock Music Listeners: Targeted list includes customers' emails, first and last names for marketing purposes.

Thank
You

