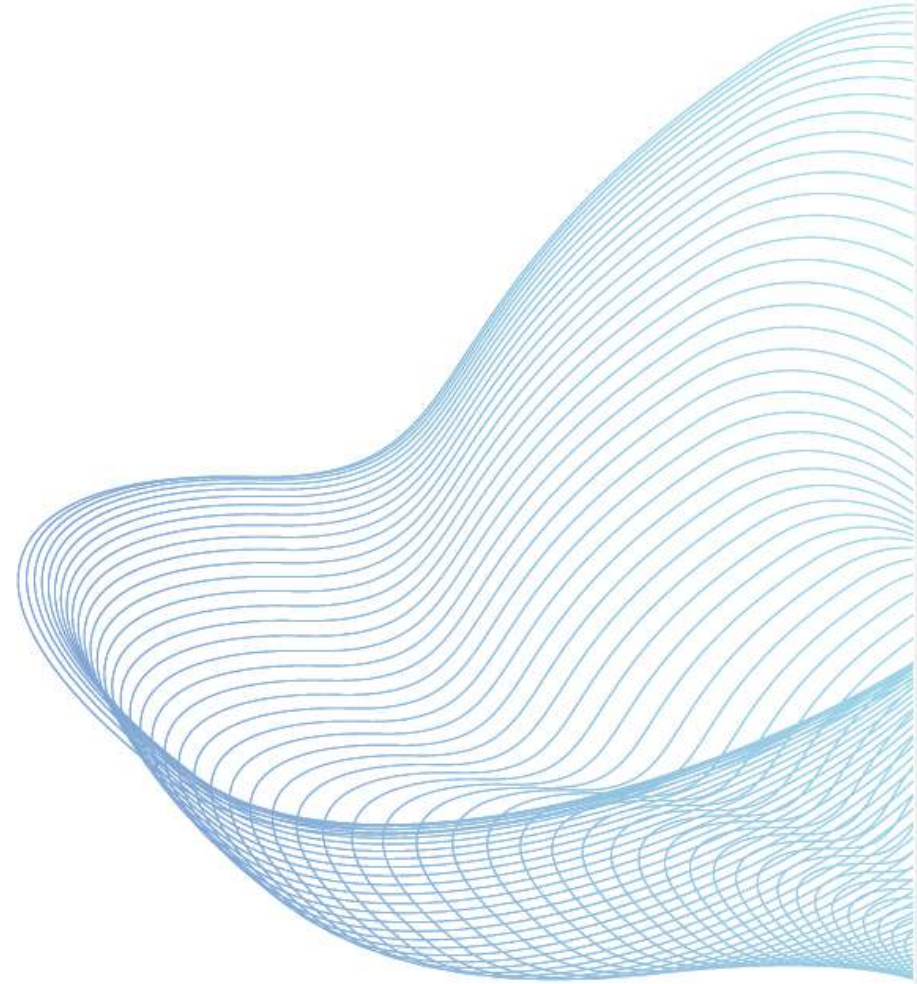
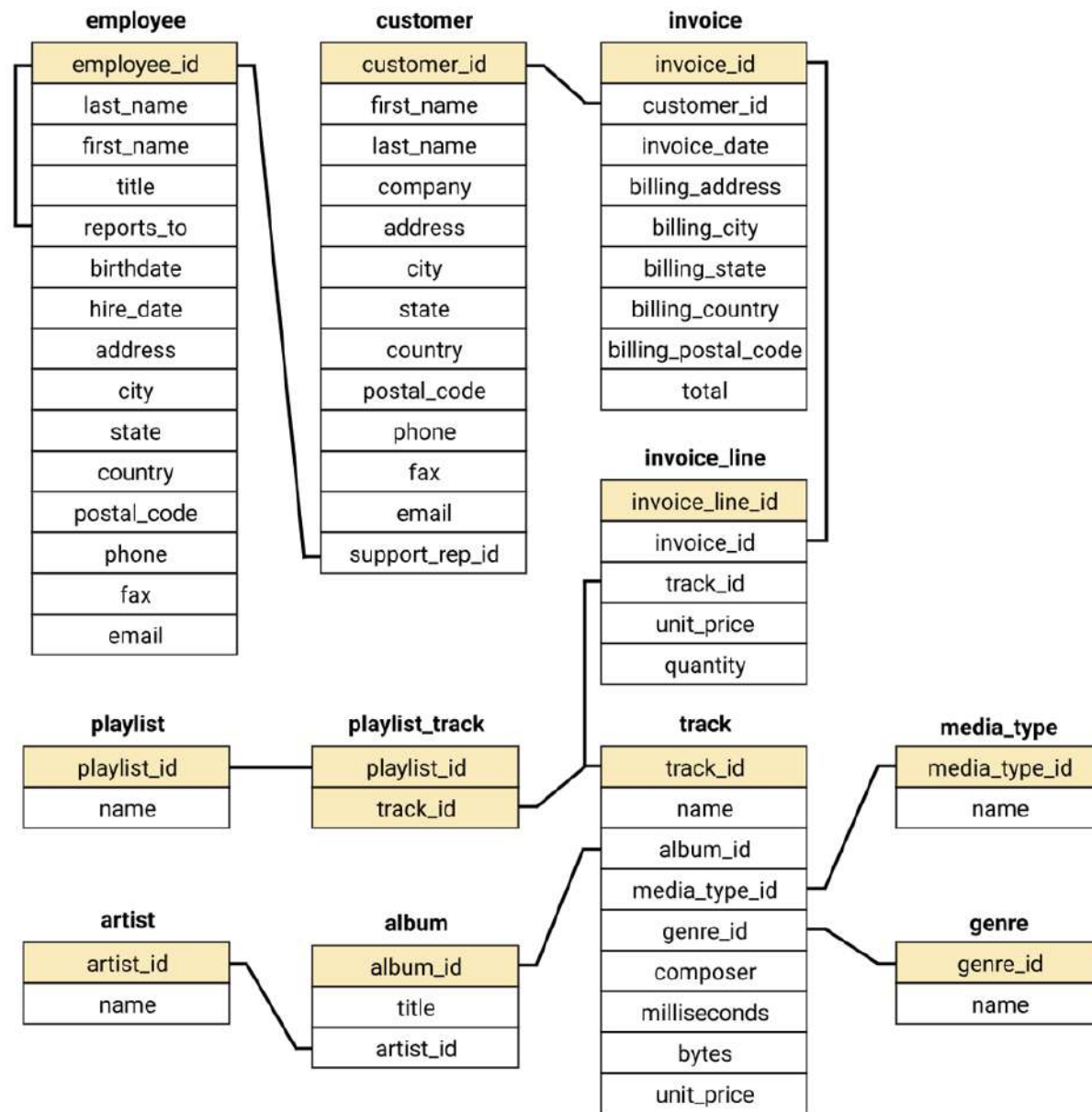


# **MUSIC STORE DATA ANALYSIS PROJECT**





pgAdmin 4

File Object Tools Help

Object Explorer

- Schemas (1)
  - public
    - Aggregates
    - Collations
    - Domains
    - FTS Configurations
    - FTS Dictionaries
    - FTS Parsers
    - FTS Templates
    - Foreign Tables
    - Functions
    - Materialized Views
    - Operators
    - Procedures
    - Sequences
    - Tables (12)
      - album
      - album2
      - artist
      - customer
      - employee
      - genre
      - invoice
      - invoice\_line
      - media\_type
      - playlist
      - playlist\_track
      - track
      - Trigger Functions
      - Types
      - Views

music\_database/postgres@localhost

Query

```
--1. Who is the senior most employee based on job title?
select * from employee
order by levels desc
limit 1;
```

Data Output

	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)	bl tin
1	9	Madan	Mohan	Senior General Manager	[null]	L7	19

Total rows: 1 of 1 Query complete 00:00:00.142

Ln 4, Col 9

16°C Fog 02:30 PM 11-01-2024

pgAdmin 4

File Object Tools Help

Object Explorer

- Scemas (1)
  - public
    - Aggregates
    - Collations
    - Domains
    - FTS Configurations
    - FTS Dictionaries
    - FTS Parsers
    - FTS Templates
    - Foreign Tables
    - Functions
    - Materialized Views
    - Operators
    - Procedures
    - Sequences
    - Tables (12)
      - album
      - album2
      - artist
      - customer
      - employee
      - genre
      - invoice
      - Invoice\_line
      - media\_type
      - playlist
      - playlist\_track
      - track
    - Trigger Functions
    - Types
    - Views

music\_database/postgres@localhost

Query

```
--2. Which countries have the most Invoices?  
select billing_country,count(total) from invoice  
group by 1  
order by 2 desc
```

Data Output

	billing_country character varying (30)	count bigint
1	USA	131
2	Canada	76
3	Brazil	61
4	France	50
5	Germany	41
6	Czech Republic	30
7	Portugal	29
8	United Kingdom	28
9	India	21

Total rows: 24 of 24 Query complete 00:00:00.127

Ln 5, Col 1

02:35 PM 11-01-2024



pgAdmin 4

FileObjectToolsHelp

Object Explorer

Scchemas (1)

public

Aggregates

Collations

Domains

FTS Configurations

FTS Dictionaries

FTS Parsers

FTS Templates

Foreign Tables

Functions

Materialized Views

Operators

Procedures

Sequences

Tables (12)

album

album2

artist

customer

employee

genre

invoice

Invoice\_line

media\_type

playlist

playlist\_track

track

Trigger Functions

Types

Views

music\_database/postgres@localhost

Query

1

2

3

4

5

6

7

8

9

--3. What are top 3 values of total invoice?  
select \* from invoice  
order by total desc  
limit 3;

Data Output

Messages

Graph Visualiser

Notifications

Invoice\_Id

customer\_Id

Invoice\_date

billing\_address

billing\_city

billing\_state

billing\_country

billing\_cha

[PK] integer

integer

timestamp without time zone

character varying (120)

character varying (30)

character varying (30)

character varying (30)

character

1

2

3

183

92

31

42

32

3

2018-02-09 00:00:00

2017-07-02 00:00:00

2017-02-21 00:00:00

9, Place Louis Barthou

696 Osborne Street

1498 rue Bélanger

Bordeaux

Winnipeg

Montréal

None

MB

QC

France

Canada

Canada

3300

R3L 2

H2G

Total rows: 3 of 3

Query complete 00:00:00.273

Ln 9, Col 1

pgAdmin 4

File Object Tools Help

Object Explorer

- Scemas (1)
  - public
    - Aggregates
    - Collations
    - Domains
    - FTS Configurations
    - FTS Dictionaries
    - FTS Parsers
    - FTS Templates
    - Foreign Tables
    - Functions
    - Materialized Views
    - Operators
    - Procedures
    - Sequences
    - Tables (12)
      - album
      - album2
      - artist
      - customer
      - employee
      - genre
      - invoice
      - Invoice\_line
      - media\_type
      - playlist
      - playlist\_track
      - track
    - Trigger Functions
    - Types
    - Views

music\_database/postgres@localhost

Query

```
--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  
select billing_city,sum(total) from invoice  
group by 1  
order by 2 desc  
limit 1;
```

Data Output

	billing_city character varying (30)	sum double precision
1	Prague	273.24000000000007

Total rows: 1 of 1 Query complete 00:00:00.329

Ln 9, Col 1

16°C Fog 02:42 PM 11-01-2024

pgAdmin 4

File Object Tools Help

Object Explorer

- Scemas (1)
  - public
    - Aggregates
    - Collations
    - Domains
    - FTS Configurations
    - FTS Dictionaries
    - FTS Parsers
    - FTS Templates
    - Foreign Tables
    - Functions
    - Materialized Views
    - Operators
    - Procedures
    - Sequences
    - Tables (12)
      - album
      - album2
      - artist
      - customer
      - employee
      - genre
      - invoice
      - Invoice\_line
      - media\_type
      - playlist
      - playlist\_track
      - track
    - Trigger Functions
    - Types
    - Views

music\_database/postgres@localhost

Query

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

Data Output

	customer_name		total
	text		double precision
1	R	Madhav	144.54000000000002

Total rows: 1 of 1 Query complete 00:00:00.342 Ln 11, Col 1

16°C Fog 02:53 PM 11-01-2024

pgAdmin 4

File Object Tools Help

Object Explorer

- Schemas (1)
  - public
    - Aggregates
    - Collations
    - Domains
    - FTS Configurations
    - FTS Dictionaries
    - FTS Parsers
    - FTS Templates
    - Foreign Tables
    - Functions
    - Materialized Views
    - Operators
    - Procedures
    - Sequences
    - Tables (12)
      - album
      - album2
      - artist
      - customer
      - employee
      - genre
      - invoice
      - Invoice\_line
      - media\_type
      - playlist
      - playlist\_track
      - track
    - Trigger Functions
    - Types
    - Views

music\_database/postgres@localhost

Query

```
--Question Set 2 - Moderate
--1. 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 * from customer,genre
select distinct c.email,c.first_name,c.last_name,g.name from customer as c
join invoice as i
on c.customer_id=i.customer_id
join invoice_line as al
on i.invoice_id=al.invoice_id
join track as t
on al.track_id=t.track_id
join genre as g
on t.genre_id=g.genre_id
where g.name='Rock'
order by c.email ;
```

Data Output

	email character varying (50)	first_name character	last_name 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	Bjorn	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	driller@comcast.com	Don	Miller	Rock

Total rows: 59 of 59 Query complete 00:00:00.315

Ln 1, Col 28

16°C Fog 03:33 PM 11-01-2024



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 (12)
  - album
  - album2
  - artist
  - customer
  - employee
  - genre
  - invoice
  - invoice\_line
  - media\_type
  - playlist
  - playlist\_track
  - track
- Trigger Functions
- Types
- Views
- Subscriptions
- postgres

Dashboard Properties SQL Statistics Dependencies Dependents Processes music\_database/postgres@localhost

music\_database/postgres@localhost

No limit

Query Query History

```
1 --2. Let's invite the artists who have written the most rock music in our dataset.
2 --Write a query that returns the Artist name and total track count of the top 10 rock bands
3 select a.artist_id,a.name,count(a.name) as number_of_songs from artist as a
4 join album as al
5 on a.artist_id=al.artist_id
6 join track as t
7 on al.album_id=t.album_id
8 join genre as g
9 on t.genre_id=g.genre_id
10 where g.name='Rock'
11 group by a.artist_id, a.name
12 order by count(g.name) desc
13 limit 10;
14
15
```

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

Total rows: 10 of 10 Query complete 00:00:00.109

Ln 2, Col 3

16°C Cloudy 10:40 AM 12-01-2024

pgAdmin 4

FileObjectToolsHelp

Object Explorer

Aggregates

Collations

Domains

FTS Configurations

FTS Dictionaries

FTS Parsers

FTS Templates

Foreign Tables

Functions

Materialized Views

Operators

Procedures

Sequences

Tables (12)

album

album2

artist

customer

employee

genre

invoice

invoice\_line

media\_type

playlist

playlist\_track

track

Trigger Functions

Types

Views

Subscriptions

postgres

DashboardPropertiesSQLStatisticsDependenciesDependentsProcessesmusic\_database/postgres@localhost

music\_database/postgres@localhost

QueryQuery History

1--3. 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

4select name,milliseconds from track

5where milliseconds>(select avg(milliseconds) from track)

6order by milliseconds desc;

Data OutputMessagesNotifications

	name	milliseconds
	character varying (150)	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. 2	2927802

Total rows: 494 of 494Query complete 00:00:00.188

Ln 4, Col 1

pgAdmin 4

FileObjectToolsHelp

Object Explorer

Aggregates

Collations

Domains

FTS Configurations

FTS Dictionaries

FTS Parsers

FTS Templates

Foreign Tables

Functions

Materialized Views

Operators

Procedures

Sequences

Tables (12)

album

album2

artist

customer

employee

genre

invoice

invoice\_line

media\_type

playlist

playlist\_track

track

Trigger Functions

Types

Views

Subscriptions

postgres

DashboardPropertiesSQLStatisticsDependenciesDependentsProcessesmusic\_database/postgres@localhost\*

music\_database/postgres@localhost

QueryQuery History

--1. Find how much amount spent by each customer on artists?  
--Write a query to return customer name, artist name and total spent  
select c.first\_name,c.last\_name,a.name as artist\_name,sum(i.unit\_price \* i.quantity) as Total\_amount from customer as c  
join invoice as inv  
on c.customer\_id=inv.customer\_id  
join invoice\_line as i  
on inv.invoice\_id=i.invoice\_id  
join track as t  
on i.track\_id=t.track\_id  
join album as al  
on t.album\_id=al.album\_id  
join artist as a  
on al.artist\_id = a.artist\_id  
group by c.first\_name,c.last\_name,a.name  
order by Total\_amount desc ;

Data OutputMessagesNotifications

	first_name character	last_name character	artist_name character varying (120)	total_amount double precision
1	Hugh	O'Reilly	Queen	27.719999999999985
2	Wyatt	Girard	Frank Sinatra	23.759999999999999
3	Aaron	Mitchell	James Brown	19.799999999999997
4	François	Tremblay	The Who	19.799999999999997
5	Robert	Brown	Creedence Clearwater Revival	19.799999999999997
6	Helena	Holý	Red Hot Chili Peppers	19.799999999999997
7	R	Madhav	Kiss	19.799999999999997
8	Heather	Leacock	House Of Pain	18.81

Total rows: 1000 of 2189Query complete 00:00:00.174

Type here to search

11:20 AM  
12-01-2024



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 (12)
  - album
  - album2
  - artist
  - customer
  - employee
  - genre
  - invoice
  - invoice\_line
  - media\_type
  - playlist
  - playlist\_track
  - track
- Trigger Functions
- Types
- Views
- Subscriptions

music\_database/postgres@localhost

Dashboard Properties SQL Statistics Dependencies Dependents Processes music\_database/postgres@localhost

Query Query History

```
1 ---2. We want to find out the most popular music Genre for each country.
2 --We determine the most popular genre as the genre with the highest amount of purchases.
3 --Write a query that returns each country along with the top Genre.
4 --For countries where the maximum number of purchases is shared return all Genres
5 WITH popular_genre AS
6 (
7     SELECT COUNT(invoice_line.quantity) AS purchases, customer.country, genre.name, genre.genre_id,
8     ROW_NUMBER() OVER(PARTITION BY customer.country ORDER BY COUNT(invoice_line.quantity) DESC) AS RowNo
9     FROM invoice_line
10    JOIN invoice ON invoice.invoice_id = invoice_line.invoice_id
11    JOIN customer ON customer.customer_id = invoice.customer_id
12    JOIN track ON track.track_id = invoice_line.track_id
13    JOIN genre ON genre.genre_id = track.genre_id
14    GROUP BY 2,3,4
15    ORDER BY 2 ASC, 1 DESC
16 )
17 SELECT * FROM popular_genre WHERE RowNo <= 1
18
```

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

Total rows: 24 of 24 Query complete 00:00:00.135 Ln 2, Col 89

16°C Cloudy 01:27 PM 12-01-2024



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 (12)
  - album
  - album2
  - artist
  - customer
  - employee
  - genre
  - invoice
  - invoice\_line
  - media\_type
  - playlist
  - playlist\_track
  - track
- Trigger Functions
- Types
- Views
- Subscriptions

music\_database/postgres@localhost

Query

```
--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 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

	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	Luís	Gonçalves	Brazil	108.89999999999998	1
6	3	François	Tremblay	Canada	99.99	1

Total rows: 24 of 24 Query complete 00:00:00.108

Ln 12, Col 1