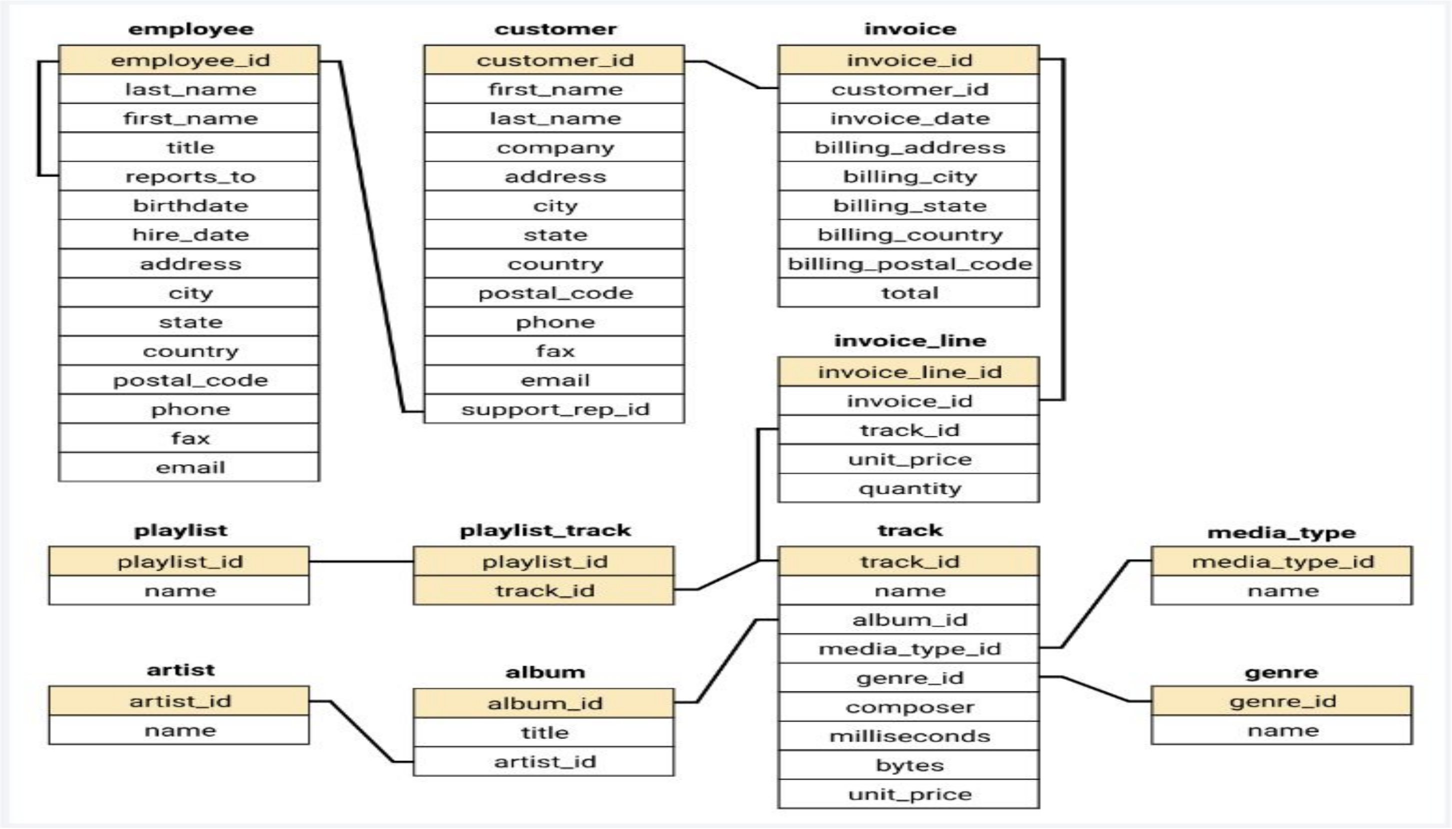
SQL PROJECT

MUSIC STORE DATA ANALYSIS

Entity Relationship Diagram:



employee
employee_id
last_name
first_name
title
reports_to
birthdate
hire_date
address
city
state
country
postal_code
phone
fax
email

customer
customer_id
first_name
last_name
company
address
city
state
country
postal_code
phone
fax
email
support_rep_id

invoice
invoice_id
customer_id
invoice_date
billing_address
billing_city
billing_state
billing_country
billing_postal_code
total

invoice_line
invoice_line_id
invoice_id
track_id
unit_price
quantity

playlist
playlist_id
name

playlist_track
playlist_id
track_id

track
track_id
name
album_id
media_type_id
genre_id
composer
milliseconds
bytes
unit_price

media_type
media_type_id
name

artist
artist_id
name

album
album_id
title
artist_id

genre
genre_id
name

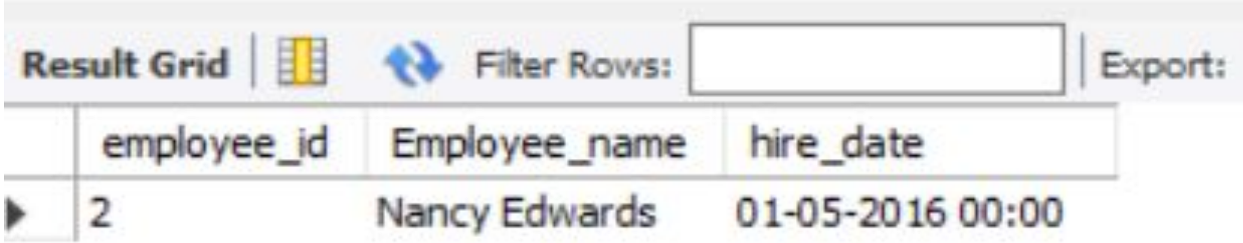
Description:

- Dataset The dataset used in this analysis can be found in the data folder. It consists of below CSV files like employee , customer , track , artist , invoice , invoice_line as soon as .
- Getting Started To run the analysis using SQL, you will need to import the tables into a database management system (DBMS) such as MySQL, PostgreSQL or SQLite…etc.
- The analysis consists of several SQL queries, each exploring different aspects of the data. ..
- Exploratory Data Analysis is an initial exploration of the dataset to understand its structure and contents.

- ## Who is the senior most employee based on job title?

```
Select  employee_id , CONCAT(first_name," ", last_name) as Employee_name , hire_date
from  employe
order by  str_to_date(hire_date, '%d-%m-%Y %H:%i')
LIMIT 1;
```

Output :

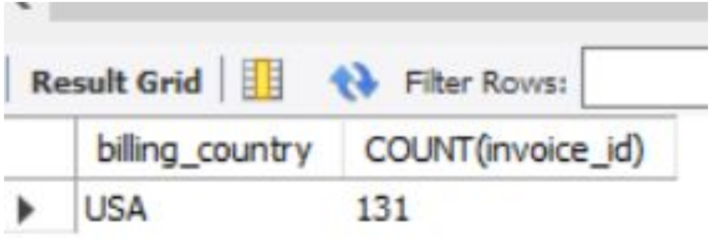


employee_id	Employee_name	hire_date
2	Nancy Edwards	01-05-2016 00:00

Which countries have the most Invoices?

```
select billing_country , COUNT(invoice_id)   from invoice
GROUP BY billing_country
ORDER BY COUNT(invoice_id) desc
 limit 1;
```

Output:

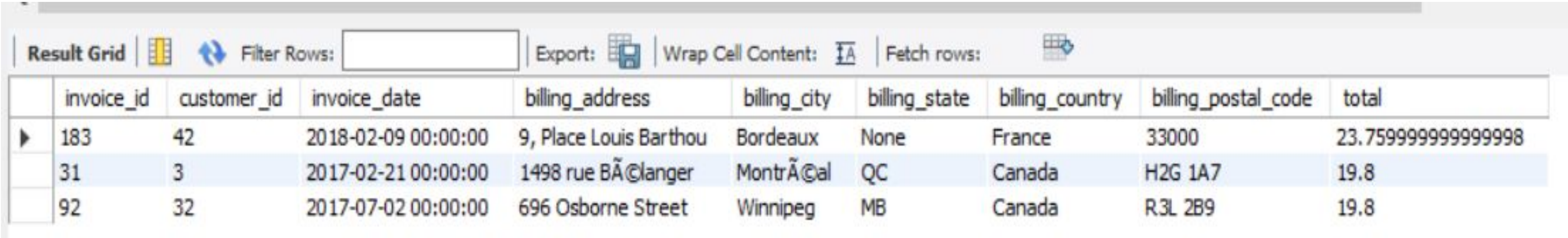


billing_country	COUNT(invoice_id)
USA	131

- **What are the top 3 values of total invoice?**

```
select  *  from invoice
order  by  total  desc  limit  3;
```

Output:

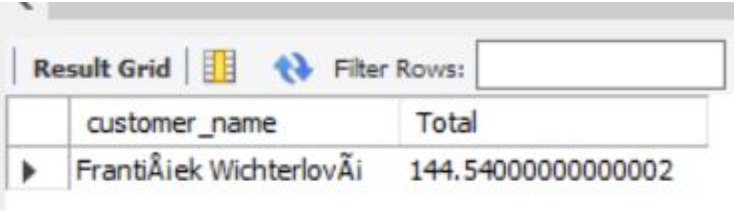


invoice_id	customer_id	invoice_date	billing_address	billing_city	billing_state	billing_country	billing_postal_code	total
183	42	2018-02-09 00:00:00	9, Place Louis Barthou	Bordeaux	None	France	33000	23.759999999999998
31	3	2017-02-21 00:00:00	1498 rue BÃ©langer	MontrÃ©al	QC	Canada	H2G 1A7	19.8
92	32	2017-07-02 00:00:00	696 Osborne Street	Winnipeg	MB	Canada	R3L 2B9	19.8

- **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 concat (c.first_name," ", c.last_name)  as  customer_name  , sum(invoice.total)  as Total
from   invoice
Left  join  customer  c
on  invoice . customer_id  =  c . customer_id
group  by  invoice . customer_id
order  by  Total  desc  limit  1;
```

Output:

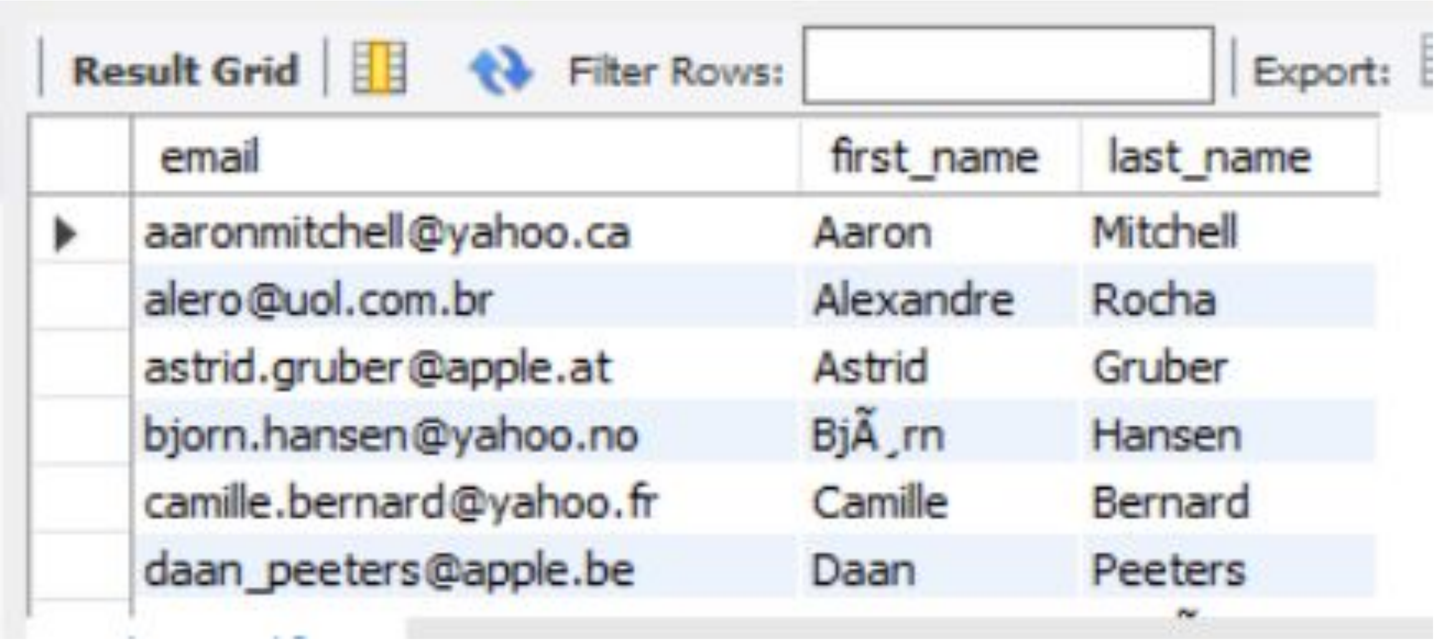


	customer_name	Total
▶	FrantiÅiek WichterlovÃi	144.5400000000002

- **Write a 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   c.email ,  c.first_name ,  c.last_name  from   customer  c
where   c.customer_id  IN  (
Select   i.customer_id
from   invoice  i
where  i.invoice_id IN (
select   il.invoice_id  from   invoice_line  il
where  il.track_id IN (
select  t.track_id  from  track t
where  t.genre_id = (
select  genre_id  from   genre
where  name = 'Rock'))))
Order  by  c.email  asc;
```

output:

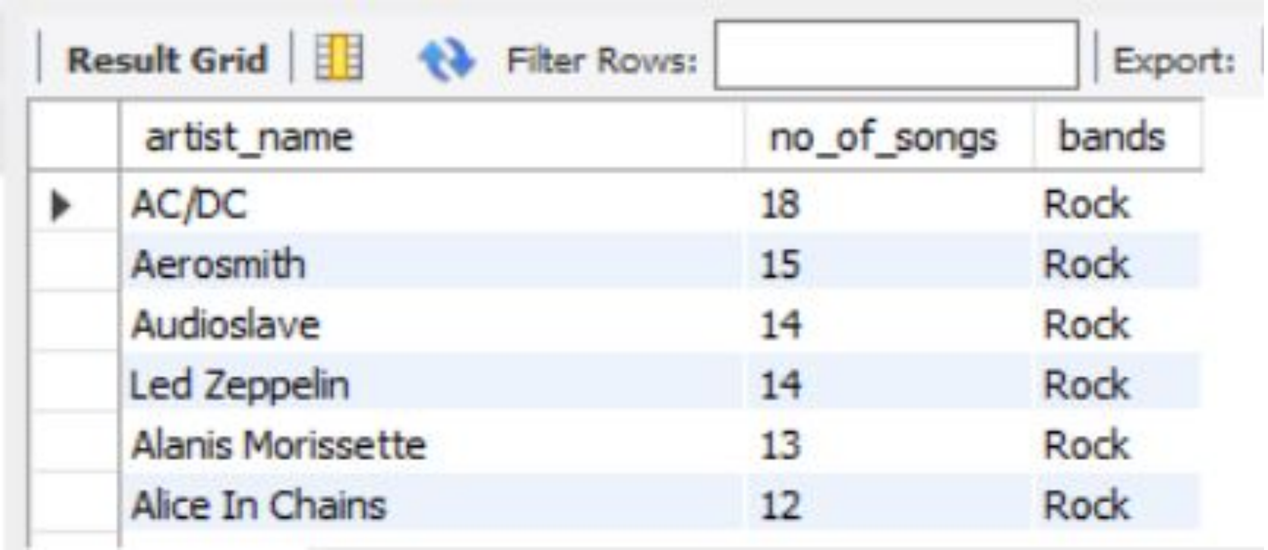


email	first_name	last_name
aaronmitchell@yahoo.ca	Aaron	Mitchell
alero@uol.com.br	Alexandre	Rocha
astrid.gruber@apple.at	Astrid	Gruber
bjorn.hansen@yahoo.no	BjÃ¸rn	Hansen
camille.bernard@yahoo.fr	Camille	Bernard
daan_peeters@apple.be	Daan	Peeters

- **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  t1.name  as  artist_name ,  count(*) as no_of_songs ,  t4.name as  bands
from  artist  as t1
join  album11  as t2
on t1.artist_id = t2.artist_id
join  track as  t3
on t2.album_id = t3.album_id
join genre  as t4
on t3.genre_id = t4.genre_id
where  t4.name = "Rock"
group  by t1.artist_id
order by no_of_songs  desc  limit 10;
```

output:

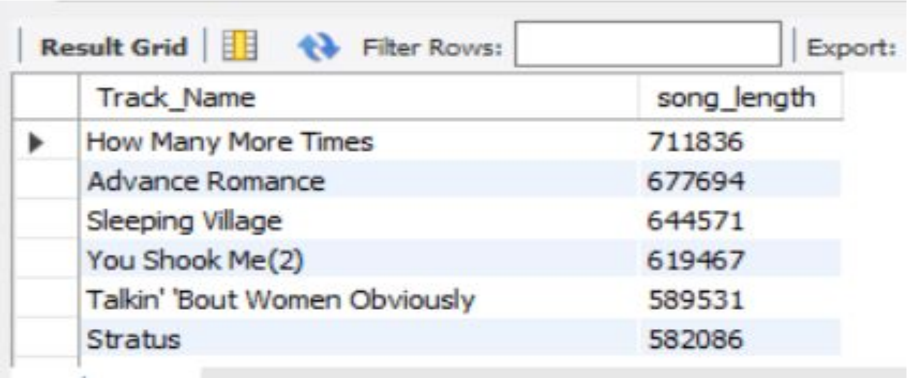


artist_name	no_of_songs	bands
AC/DC	18	Rock
Aerosmith	15	Rock
Audioslave	14	Rock
Led Zeppelin	14	Rock
Alanis Morissette	13	Rock
Alice In Chains	12	Rock

- **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 as Track_Name , milliseconds  as song_length  from  track
where milliseconds > (select avg(milliseconds) as avg_time  from  track)
order  by  song_length desc;
```

output:

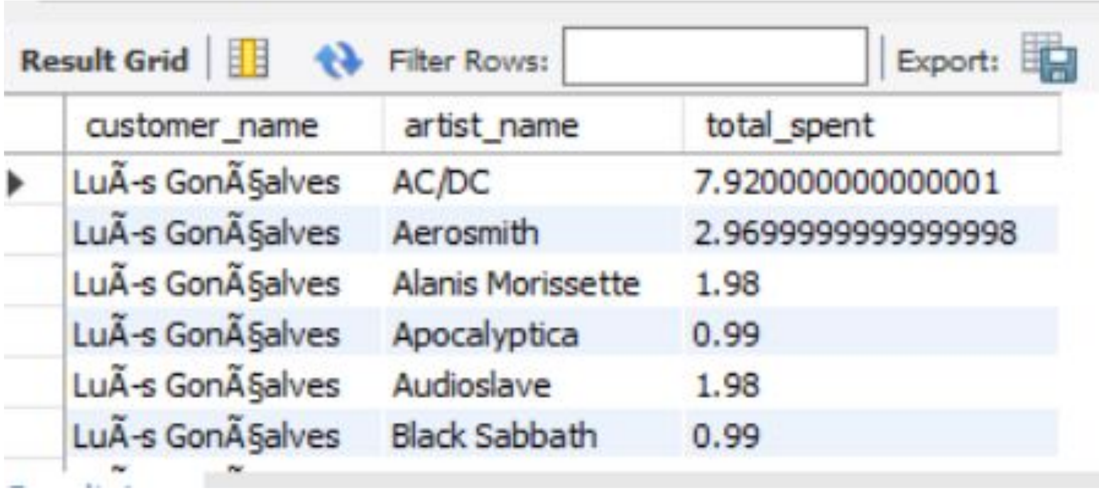


Track_Name	song_length
How Many More Times	711836
Advance Romance	677694
Sleeping Village	644571
You Shook Me(2)	619467
Talkin' 'Bout Women Obviously	589531
Stratus	582086

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

```
SELECT (
        SELECT  concat (c.first_name," ",c.last_name)
        FROM  customer c
       WHERE  c.customer_id  =  i.customer_id
         )
       AS  customer_name (
       SELECT   a.name   FROM   artist  as  a
       WHERE   a.artist_id   =   al.artist_id
       )  AS  artist_name,
       SUM(il.unit_price * il.quantity)  AS   total_spent
       FROM   invoice_line    il
       JOIN invoice i  ON  il.invoice_id  =  i.invoice_id
       JOIN  track t   ON   il.track_id  =  t.track_id
       JOIN  album11 al ON t.album_id  =  al.album_id
      GROUP BY  i.customer_id ,  al.artist_id;
```

output:

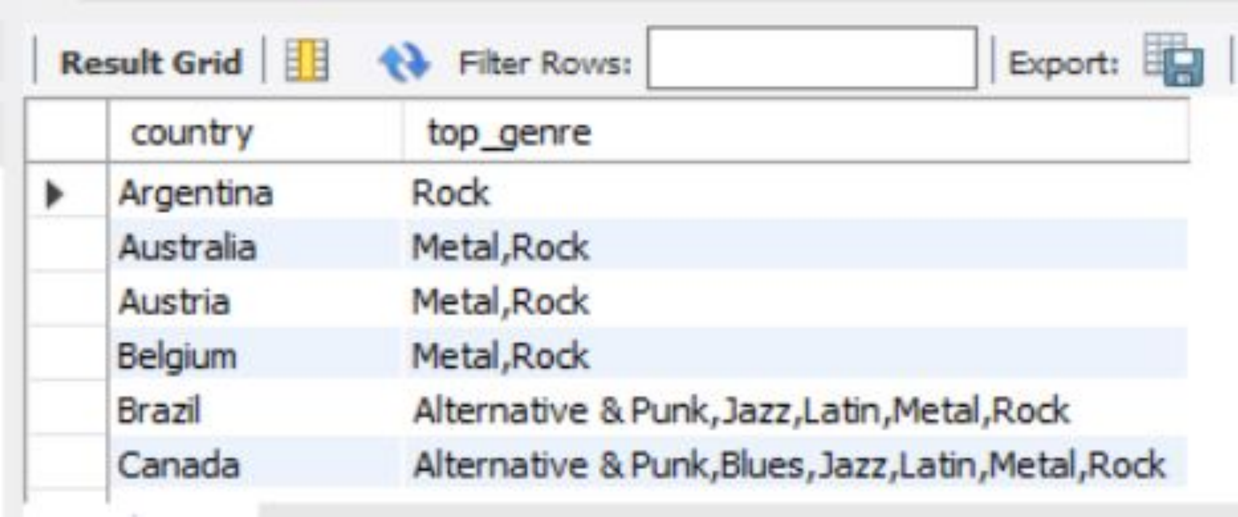


customer_name	artist_name	total_spent
LuÃ-s GonÃ§alves	AC/DC	7.920000000000001
LuÃ-s GonÃ§alves	Aerosmith	2.9699999999999998
LuÃ-s GonÃ§alves	Alanis Morissette	1.98
LuÃ-s GonÃ§alves	Apocalyptica	0.99
LuÃ-s GonÃ§alves	Audioslave	1.98
LuÃ-s GonÃ§alves	Black Sabbath	0.99

- **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 ?**

```
SELECT
        i.billing_country   AS  country,
                        CASE
                            WHEN COUNT(*)  >  1  THEN  GROUP_CONCAT(DISTINCT g.name ORDER BY g.name)
                            ELSE  g.name
                        END  AS  top_genre
FROM   invoice i
JOIN  invoice_line il  ON  i.invoice_id  =  il.invoice_id
JOIN  track  t  ON  il.track_id  =  t.track_id
jOIN  genre  g  ON  t.genre_id  =  g.genre_id
GROUP BY  i.billing_country
HAVING  COUNT(DISTINCT g.name)  >  1  OR  COUNT(DISTINCT g.name)  =  1
ORDER BY  i.billing_country;
```

output:

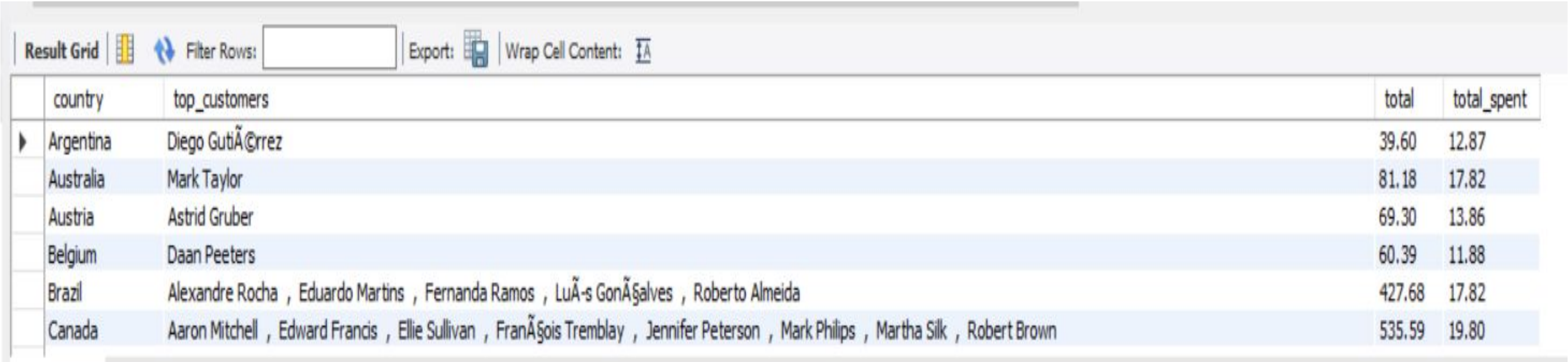


country	top_genre
Argentina	Rock
Australia	Metal,Rock
Austria	Metal,Rock
Belgium	Metal,Rock
Brazil	Alternative & Punk,Jazz,Latin,Metal,Rock
Canada	Alternative & Punk,Blues,Jazz,Latin,Metal,Rock

- **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 ?**

```
SELECT
          i.billing_country   AS   country,
           CASE
                      WHEN   count (DISTINCT  c.customer_id)  >   1
                      THEN  GROUP_CONCAT
                      (DISTINCT  c.first_name  , ' ' , c.last_name  ORDER BY  c.first_name  ,  c.last_name     SEPARATOR ' ,   '  )
                      ELSE   CONCAT (c.first_name , ' ' ,  c.last_name)
           END AS top_customers,
           round(SUM(il.unit_price) , 2 )  as  total ,
           round(max(total),2)  AS  total_spent
FROM    customer    c
JOIN     invoice  i  ON  c.customer_id  =  i.customer_id
JOIN   invoice_line  il  ON  i.invoice_id  =  il.invoice_id
GROUP BY  i.billing_country
HAVING  COUNT(DISTINCT c.customer_id)  >  1  or  COUNT(DISTINCT   c.customer_id)  =  1
ORDER BY  i.billing_country;
```

output:



country	top_customers	total	total_spent
Argentina	Diego GutiÃ©rrez	39.60	12.87
Australia	Mark Taylor	81.18	17.82
Austria	Astrid Gruber	69.30	13.86
Belgium	Daan Peeters	60.39	11.88
Brazil	Alexandre Rocha , Eduardo Martins , Fernanda Ramos , LuÃ-s GonÃ§alves , Roberto Almeida	427.68	17.82
Canada	Aaron Mitchell , Edward Francis , Ellie Sullivan , FranÃ§ois Tremblay , Jennifer Peterson , Mark Philips , Martha Silk , Robert Brown	535.59	19.80

conclusion:

Through the analysis of the music store dataset using SQL, we were able to gain insights into the store's customers, products, and sales. These insights can be used to make informed business decisions, such as which products to stock, which marketing strategies to use, and which customer segments to target.

THE END