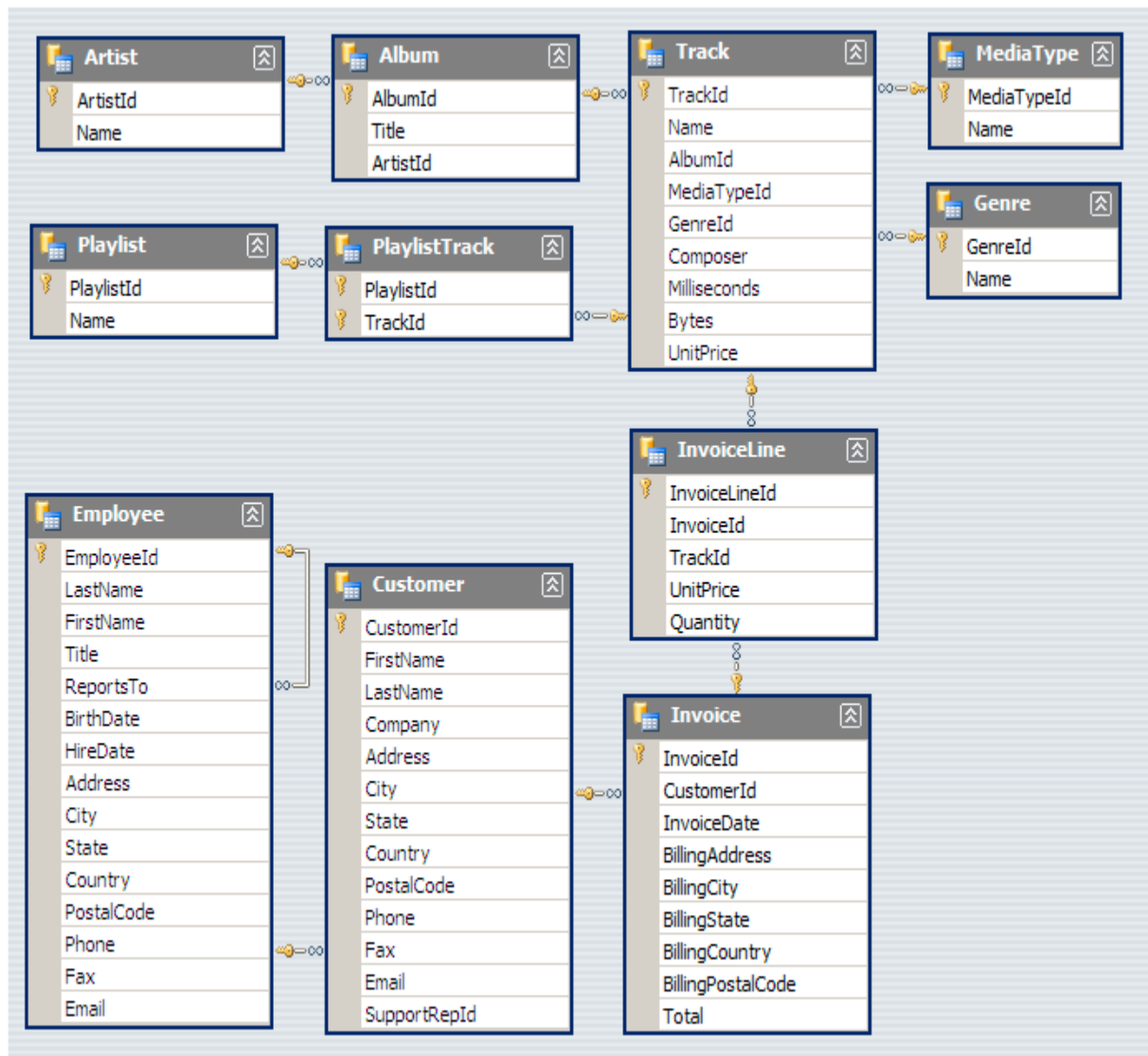


Project Name: Music Store Analysis

Database link:

<https://drive.google.com/file/d/18ZytGB8Nlr1NSIVgC4qyT9IfQoV48NI/view?usp=sharing>

Schema Diagram:



Question Set 1 - Easy

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

employee_id	last_name	first_name	title	reports_to	levels	birthdate	hire_date	address	city
1	Adams	Andrew	General Manager	9	L6	18-02-1962 00:00	14-08-2016 00:00	11120 Jasper Ave NW	Edmonton

state	country	postal_code	phone	fax	email
AB	Canada	T5K 2N1	+1 (780) 428-9482	+1 (780) 428-3457	andrew@chinookcorp.com

Q2: Which countries have the most Invoices?

SELECT COUNT(*) AS c, billing_country

FROM invoice

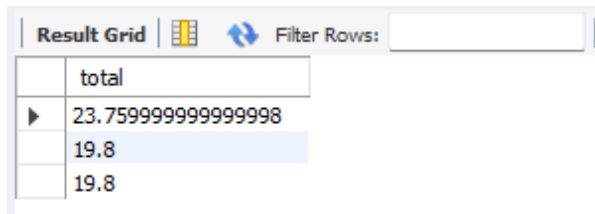
GROUP BY billing_country

ORDER BY c **DESC**

count(*)	billing_country
131	USA
76	Canada
50	France
10	Poland
21	India
41	Germany
29	Portugal
11	Finland
61	Brazil
10	Denmark
28	United Kingdom
9	Norway
10	Australia
10	Hungary
13	Ireland
10	Sweden
11	Spain
13	Chile
9	Italy
30	Czech Republic
10	Netherlands
7	Belgium
9	Austria
5	Argentina

Q3: What are top 3 values of total invoice? */

```
SELECT total
FROM invoice
ORDER BY total DESC
```



The screenshot shows a 'Result Grid' with a 'Filter Rows' input field. The grid contains three rows of data. The first row has a value of 23.759999999999998. The second row has a value of 19.8 and is highlighted. The third row has a value of 19.8.

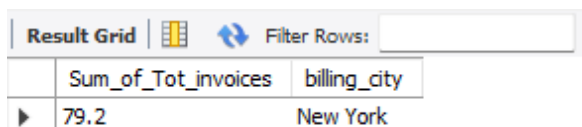
	total
▶	23.759999999999998
	19.8
	19.8

Q4: 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

```
SELECT billing_city,SUM(total) AS InvoiceTotal
FROM invoice
GROUP BY billing_city
ORDER BY InvoiceTotal DESC
LIMIT 1;
```



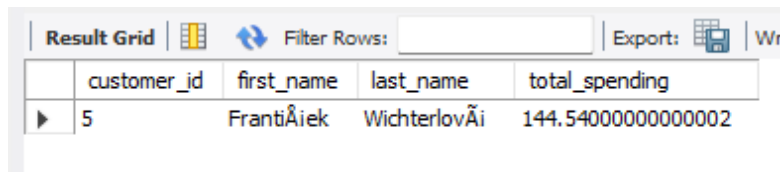
The screenshot shows a 'Result Grid' with a 'Filter Rows' input field. The grid contains one row of data. The first column is labeled 'Sum_of_Tot_invoices' and the second column is labeled 'billing_city'. The values are 79.2 and New York respectively.

	Sum_of_Tot_invoices	billing_city
▶	79.2	New York

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;
```



The screenshot shows a database interface with a 'Result Grid' tab. It displays a single row of data for customer_id 5, with first_name 'František', last_name 'Wichterlov', and total_spending 144.54000000000002. The interface includes a 'Filter Rows' search bar and an 'Export' button.

	customer_id	first_name	last_name	total_spending
▶	5	František	Wichterlov	144.54000000000002

Note : IF YOU GOT THE ERROR LIKE THIS --.>

Error Code: 1055. Expression #2 of SELECT list is not in GROUP BY clause and contains nonaggregated column 'sql_project.customer.first_name' which is not functionally dependent on columns in GROUP BY clause; this is incompatible with sql_mode=only_full_group_by
0.000 sec

SOLUTION : (Run this command and then rerun your group by query)

```
set sql_mode = (SELECT REPLACE(@@sql_mode, 'ONLY_FULL_GROUP_BY',''));
```

Question Set 2 - Moderate

Q1: 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 DISTINCT email AS Email, first_name AS FirstName, last_name AS LastName,  
genre.name AS Name
```

```
FROM customer
```

```
JOIN invoice ON invoice.customer_id = customer.customer_id
```

```
JOIN invoiceline ON invoiceline.invoice_id = invoice.invoice_id
```

```
JOIN track ON track.track_id = invoiceline.track_id
```

```
JOIN genre ON genre.genre_id = track.genre_id
```

```
WHERE genre.name LIKE 'Rock'
```

```
ORDER BY email;
```

Result Grid			Filter Rows:
	genre_id	name	
▶	1	Rock	
	2	Jazz	
	3	Metal	
	4	Alternative & Punk	
	5	Rock And Roll	
	6	Blues	
	7	Latin	
	8	Reggae	
	9	Pop	
	10	Soundtrack	
	11	Bossa Nova	
	12	Easy Listening	
	13	Heavy Metal	
	14	R&B/Soul	
	15	Electronica/Dance	
	16	World	
	17	Hip Hop/Rap	
	18	Science Fiction	
	19	TV Shows	
	20	Sci Fi & Fantasy	
	21	Drama	
	22	Comedy	
	23	Alternative	
	24	Classical	
	25		

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;
```

	artist_id	name	number_of_songs
▶	1	AC/DC	18
	3	Aerosmith	15
	8	Audioslave	14
	22	Led Zeppelin	14
	4	Alanis Morissette	13
	5	Alice In Chains	12
	23	Frank Zappa & Captain Beefheart	9
	2	Accept	4

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;
```

Result Grid	Filter Rows:
name	milliseconds
Book Of Thel	494393
You Oughta Know (Alternate)	491885
Terra	482429
Snoopy's search-Red baron	456071
Sozinho (Hitmakers Classic Mix)	436636
Master Of Puppets	436453
Stone Crazy	433397
Snowblind	420022
Computadores Fazem Arte	404323
Jerusalem	402390
Dazed and Confused	401920
The Winner Loses	392254
Love, Hate, Love	387134
Construção / Deus Lhe Pague	383059
Black Sabbath	382066
Livin' On The Edge	381231
I Can't Quit You Baby(2)	380551
Princess of the Dawn	375418
Harvester Of Sorrow	374543
Whole Lotta Love	373394
Momma's Gotta Die Tonight	371539
Vai Passar	369763
Wherever I May Roam	369345
Overdose	369319

Result Grid	Filter Rows:
name	milliseconds
Harvester Of Sorrow	374543
Whole Lotta Love	373394
Momma's Gotta Die Tonight	371539
Vai Passar	369763
Wherever I May Roam	369345
Overdose	369319
N.I.B.	368770
O Boto (Bã 'to)	366837
Let There Be Rock	366654
The Beginning... At Last	365662
Carolina Hard-Core Ecstasy	359680
World Of Trouble	359157
Trupets Of Jericho	359131
Amazing	356519
Under The Sun/Every Day Comes and ...	350458
Welcome Home (Sanitarium)	350197
There Goes The Neighborhood	350171
Sea Of Sorrow	349831
Confusion	344163
For Those About To Rock (We Salute Y...	343719
Shadow on the Sun	343457
Balls to the Wall	342562
Machine Men	341655
I am the Highway	334942

Question Set 3 - Advance

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  
)  
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;
```


Result Grid					
		Filter Rows:		Export:	Wrap Cell Content:
	customer_id	first_name	last_name	artist_name	amount_spent
▶	54	Steve	Murray	AC/DC	17.82
	53	Phil	Hughes	AC/DC	10.89
	21	Kathy	Chase	AC/DC	10.89
	49	Stanisław	Wójcik	AC/DC	9.9
	1	Luís	Gonçalves	AC/DC	7.920000000000001
	24	Frank	Ralston	AC/DC	7.920000000000001
	31	Martha	Silk	AC/DC	3.96
	16	Frank	Harris	AC/DC	2.9699999999999998
	42	Wyatt	Girard	AC/DC	2.9699999999999998
	6	Helena	Holmér	AC/DC	2.9699999999999998
	38	Niklas	Schröder	AC/DC	2.9699999999999998
	35	Madalena	Sampaio	AC/DC	2.9699999999999998
	44	Terhi	Härmä	AC/DC	2.9699999999999998
	9	Kara	Nielsen	AC/DC	1.98
	34	João	Fernandes	AC/DC	1.98
	57	Luis	Rojas	AC/DC	1.98
	27	Patrick	Gray	AC/DC	1.98
	20	Dan	Miller	AC/DC	1.98
	30	Edward	Francis	AC/DC	1.98
	5	František	Wichterlov	AC/DC	1.98
	47	Lucas	Mancini	AC/DC	0.99
	43	Isabelle	Mercier	AC/DC	0.99
	19	Tim	Goyer	AC/DC	0.99
	39	Camille	Bernard	AC/DC	0.99

Result Grid					
		Filter Rows:		Export:	Wrap Cell Content: fA
	customer_id	first_name	last_name	artist_name	amount_spent
	30	Edward	Francis	AC/DC	1.98
	5	František	Wichterlov	AC/DC	1.98
	47	Lucas	Mancini	AC/DC	0.99
	43	Isabelle	Mercier	AC/DC	0.99
	19	Tim	Goyer	AC/DC	0.99
	39	Camille	Bernard	AC/DC	0.99
	8	Daan	Peeters	AC/DC	0.99
	15	Jennifer	Peterson	AC/DC	0.99
	58	Manoj	Pareek	AC/DC	0.99
	46	Hugh	O'Reilly	AC/DC	0.99
	32	Aaron	Mitchell	AC/DC	0.99
	45	Ladislav	Kovács	AC/DC	0.99
	29	Robert	Brown	AC/DC	0.99
	26	Richard	Cunningham	AC/DC	0.99
	17	Jack	Smith	AC/DC	0.99
	14	Mark	Philips	AC/DC	0.99
	2	Leonie	Köhler	AC/DC	0.99
	56	Diego	Gutiérrez	AC/DC	0.99
	48	Johannes	Van der Berg	AC/DC	0.99
	13	Fernanda	Ramos	AC/DC	0.99
	55	Mark	Taylor	AC/DC	0.99
	7	Astrid	Gruber	AC/DC	0.99
	59	Rishabh	Mishra	AC/DC	0.99
	10	Eduardo	Martins	AC/DC	0.99

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 where

the maximum number of purchases is shared return all Genres.

Using CTE Method:

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

Result Grid		Filter Rows:			Export:
	purchases	country	name	genre_id	RowNo
▶	1	Argentina	Rock	1	1
	18	Australia	Rock	1	1
	6	Austria	Rock	1	1
	5	Belgium	Rock	1	1
	26	Brazil	Rock	1	1
	57	Canada	Rock	1	1
	7	Chile	Rock	1	1
	14	Czech Republic	Rock	1	1
	6	Denmark	Rock	1	1
	6	Finland	Rock	1	1
	26	France	Rock	1	1
	28	Germany	Rock	1	1
	4	Hungary	Rock	1	1
	13	India	Rock	1	1