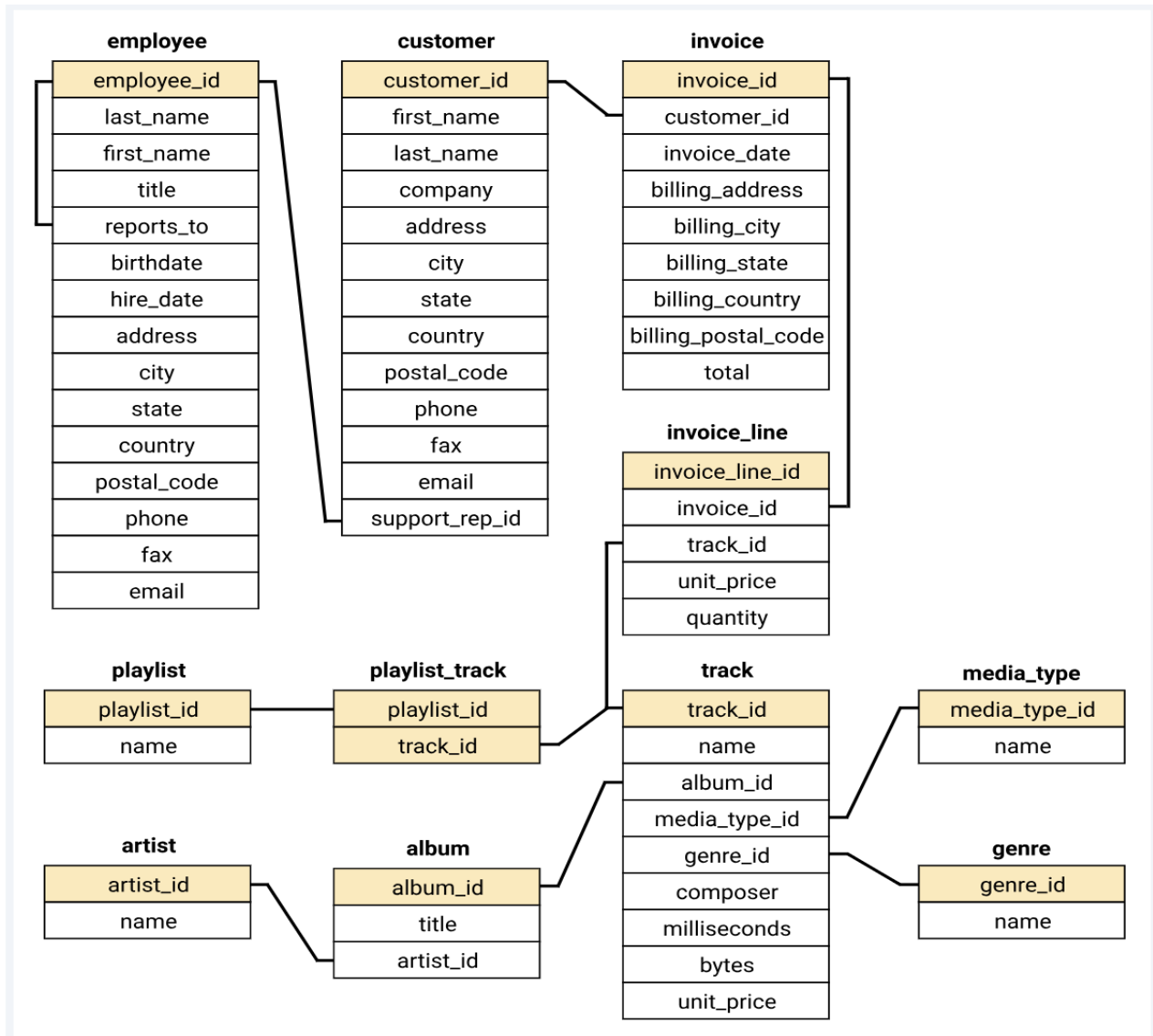


MySQL Project - Music Store Analysis



Database and Tools

- MySQL
- Schema- Music Store Database



Case study Questions

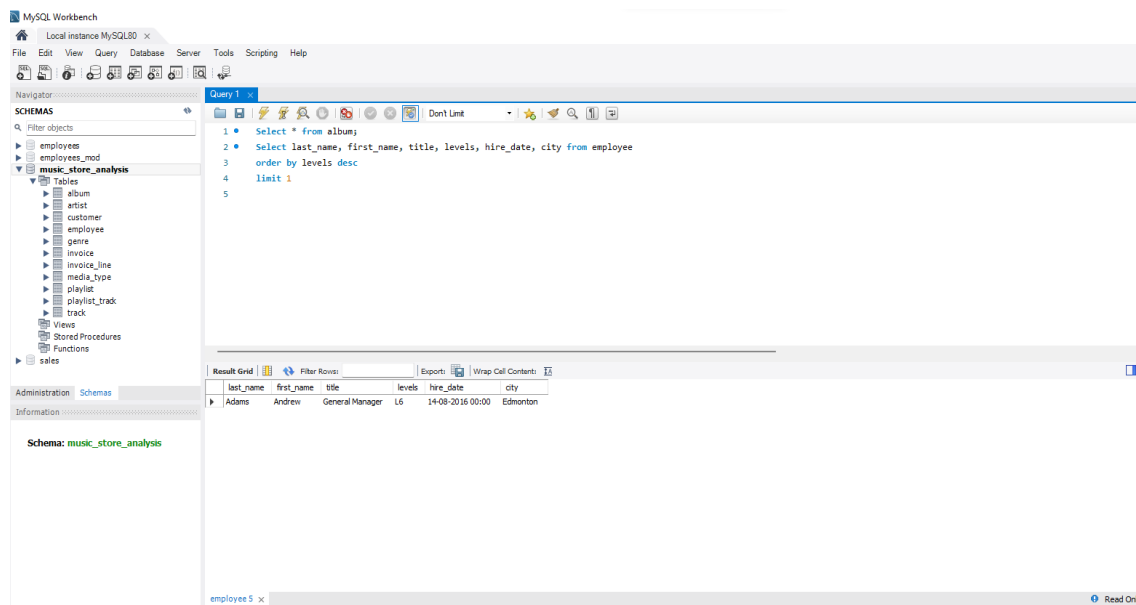
Q1: Who is the senior most employee based on job title?

Ans- SELECT title, last_name, first_name

FROM employee

ORDER BY levels DESC

LIMIT 1;



Q2: Which countries have the most Invoices?

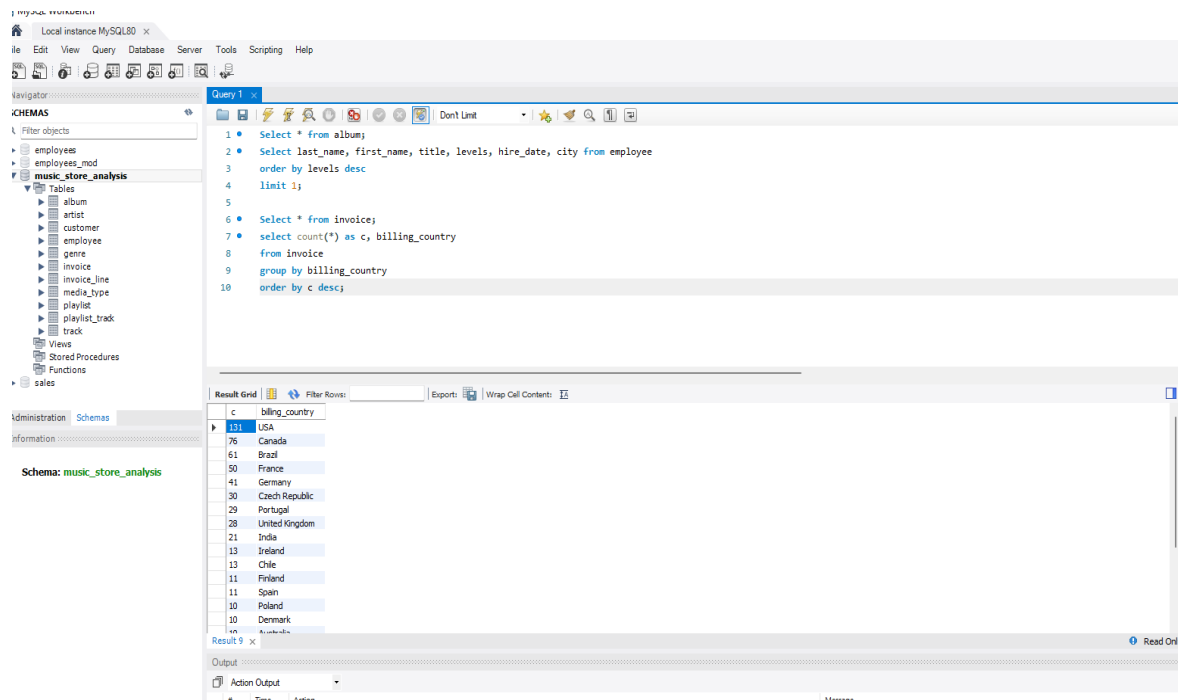
Ans-

SELECT COUNT(*) AS c, billing_country

FROM invoice

GROUP BY billing_country

ORDER BY c DESC



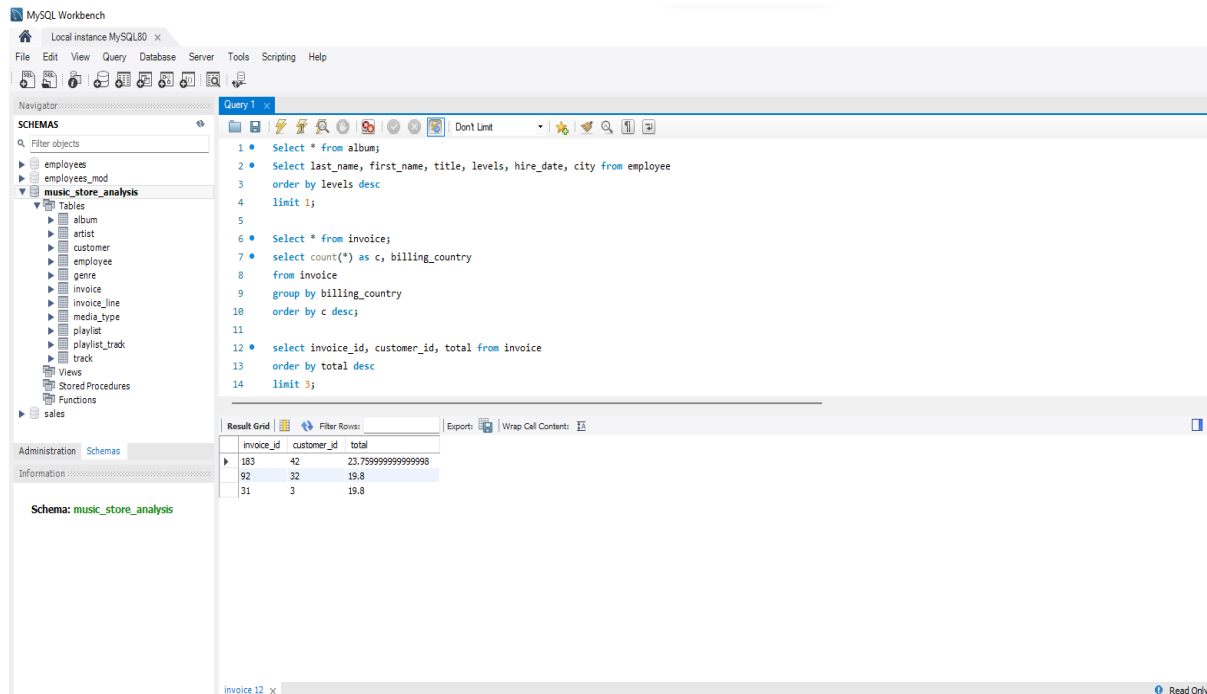
Q3: What are top 3 values of total invoice?

Ans-

SELECT total

FROM invoice

ORDER BY total DESC



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

Ans-

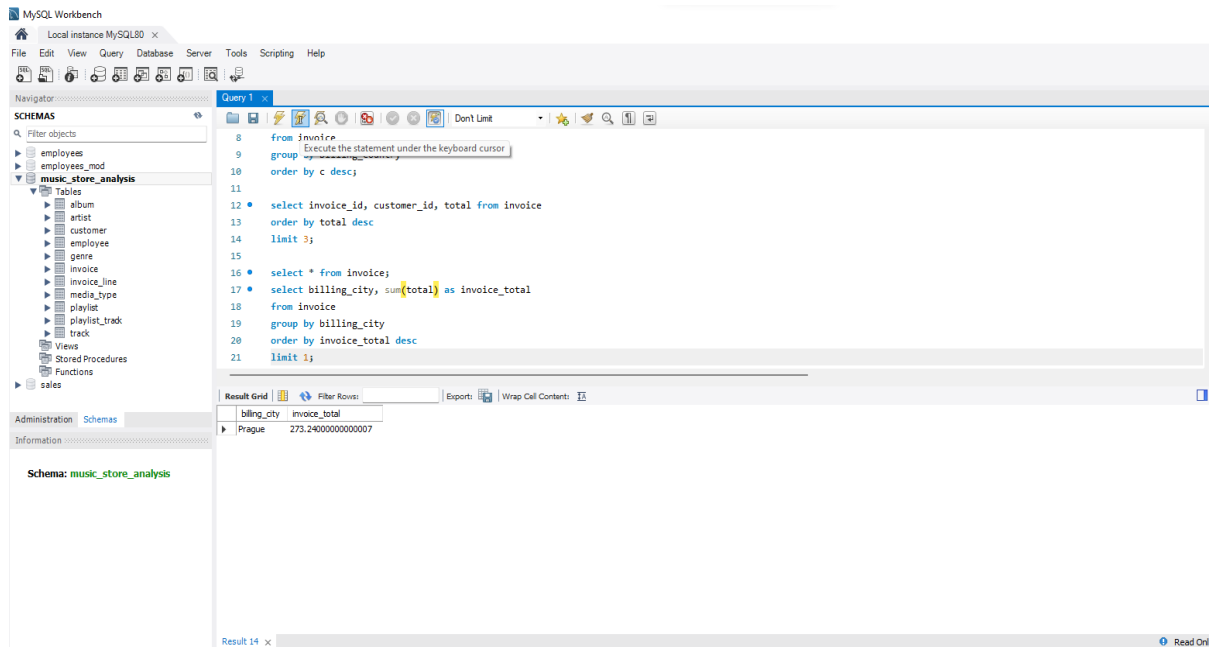
SELECT billing_city,SUM(total) AS InvoiceTotal

FROM invoice

GROUP BY billing_city

ORDER BY InvoiceTotal DESC

LIMIT 1;



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.

Ans-

```

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

```

```

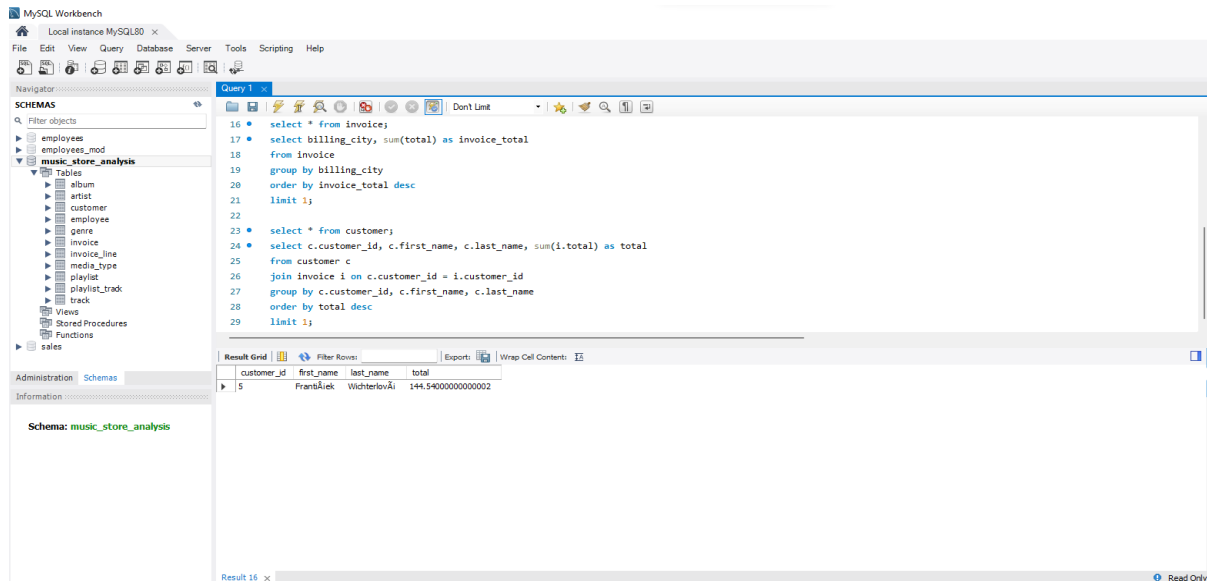
ORDER BY total_spending DESC

```

```

LIMIT 1

```



Q6: 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.

Ans-

SELECT DISTINCT email,first_name, last_name

FROM customer

JOIN invoice ON customer.customer_id = invoice.customer_id

JOIN invoiceline ON invoice.invoice_id = invoiceline.invoice_id

WHERE track_id IN(

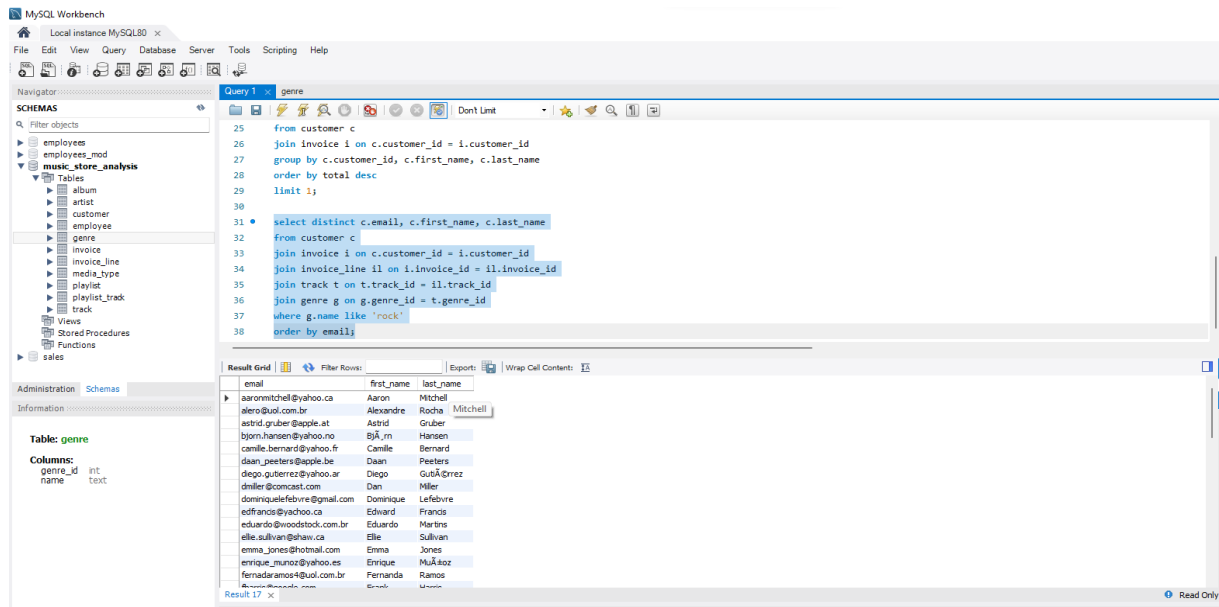
SELECT track_id FROM track

JOIN genre ON track.genre_id = genre.genre_id

WHERE genre.name LIKE 'Rock'

)

ORDER BY email;



Q7: 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.

Ans-

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;

The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' pane displays a tree view of the database structure, including tables like 'album', 'artist', 'customer', 'employee', 'genre', 'invoice', 'invoice_line', 'media_type', 'playlist', 'playlist_track', 'track', 'views', 'stored_procedures', 'functions', and 'sales'. The 'genre' table is selected. The main query editor shows a SQL query that joins 'invoice', 'invoice_line', 'track', and 'genre' tables, filters for 'rock' genre, and counts the number of songs per artist. The query is as follows:

```

33 join invoice i on c.customer_id = i.customer_id
34 join invoice_line il on i.invoice_id = il.invoice_id
35 join track t on t.track_id = il.track_id
36 join genre g on g.genre_id = t.genre_id
37 where g.name like 'rock'
38 order by email;
39
40 select a.artist_id, a.name, count(a.artist_id) as No_Of_Songs
41 from track t
42 join album al on al.album_id = t.album_id
43 join artist a on a.artist_id = t.artist_id
44 join genre g on g.genre_id = t.genre_id
45 where g.name like 'rock'
46 group by a.artist_id, a.name
47 order by No_Of_Songs desc
48 limit 10;

```

The 'Result Grid' at the bottom shows the results of the query, ordered by the number of songs in descending order:

artist_id	name	No_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

Q8: 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.

Ans-

SELECT name,milliseconds

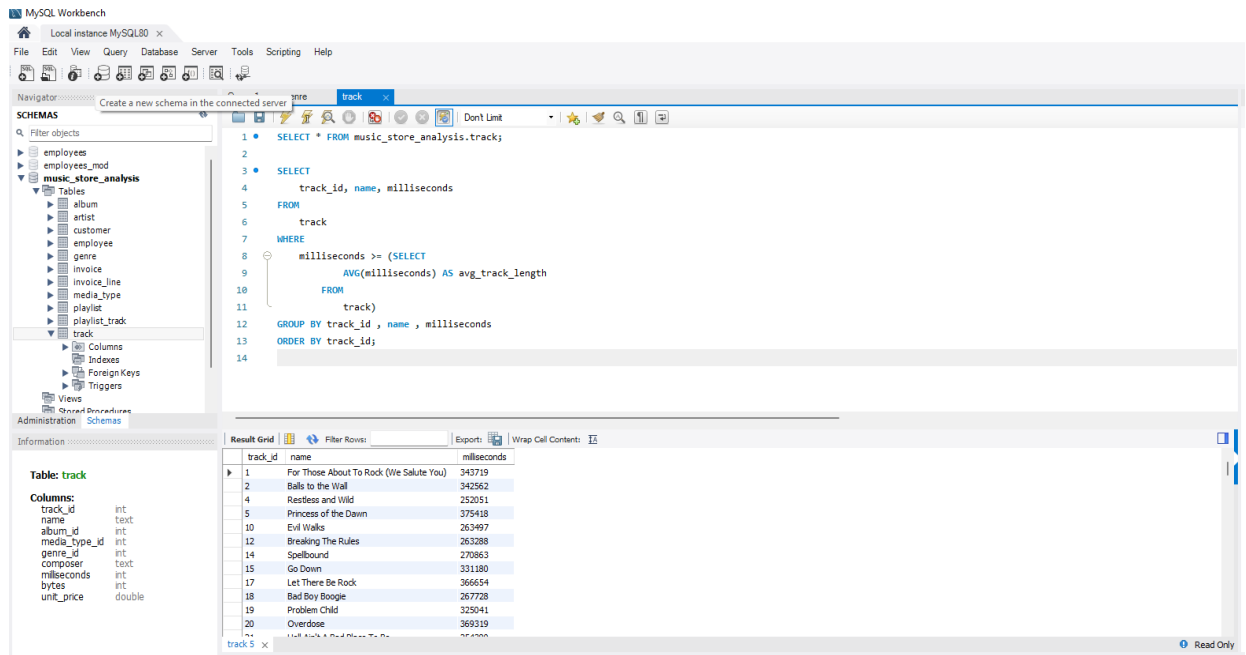
FROM track

WHERE milliseconds > (

SELECT AVG(milliseconds) AS avg_track_length

FROM track)

ORDER BY milliseconds DESC;



Q9: Find how much amount spent by each customer on artists? Write a query to return customer name, artist name and total spent

Ans-

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;
```

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with 'main_store_analysis' selected. The main editor window contains a SQL query (Query 1) that joins several tables to calculate the total amount spent by customer, grouped by customer ID, first name, last name, and artist name, and ordered by the total amount spent in descending order. The query is as follows:

```
50 with best_selling_artist as (SELECT
51   a.artist_id,
52   a.name AS artist_name,
53   SUM(il.unit_price * quantity)
54 FROM
55   invoice_line il
56   JOIN
57   track t ON t.track_id = il.track_id
58   JOIN
59   album al ON al.album_id = t.album_id
60   JOIN
61   artist a ON a.artist_id = al.artist_id
62 GROUP BY a.artist_id, a.name
63 ORDER BY 5 desc
64 LIMIT 1)
65 select c.customer_id, c.first_name, c.last_name, bsa.artist_name, sum(il.unit_price*il.quantity) as amt_spt
66 from invoice i
67 join customer c on c.customer_id = i.customer_id
68 join invoice_line il on il.invoice_id = i.invoice_id
69 join track t on t.track_id = il.track_id
70 join album al on al.album_id = t.album_id
71 join best_selling_artist bsa on bsa.artist_id = al.artist_id
72 group by 1,2,3,4
73 order by 5 desc;
74
```

The 'Result Grid' at the bottom shows the results of the query, with 21 rows returned. The columns are customer_id, first_name, last_name, artist_name, and amt_spt. The results are ordered by the total amount spent in descending order.

customer_id	first_name	last_name	artist_name	amt_spt
54	Steve	Murphy	AC/DC	10.99
53	Phil	Hughes	AC/DC	10.99
21	Kathy	Chase	AC/DC	10.99
49	Stam	Law	W.A.S.P.	9.9
1	LuLu	Gonzales	AC/DC	7.920000000000001
24	Frank	Rakoton	AC/DC	7.920000000000001
31	Martha	Sik	AC/DC	3.95
16	Frank	Harris	AC/DC	2.9699999999999998

The 'Output' pane at the bottom shows the execution details of the query, including the time taken (0.016 sec / 0.000 sec) and the number of rows returned (42 rows).

Q10: 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.

Ans- 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

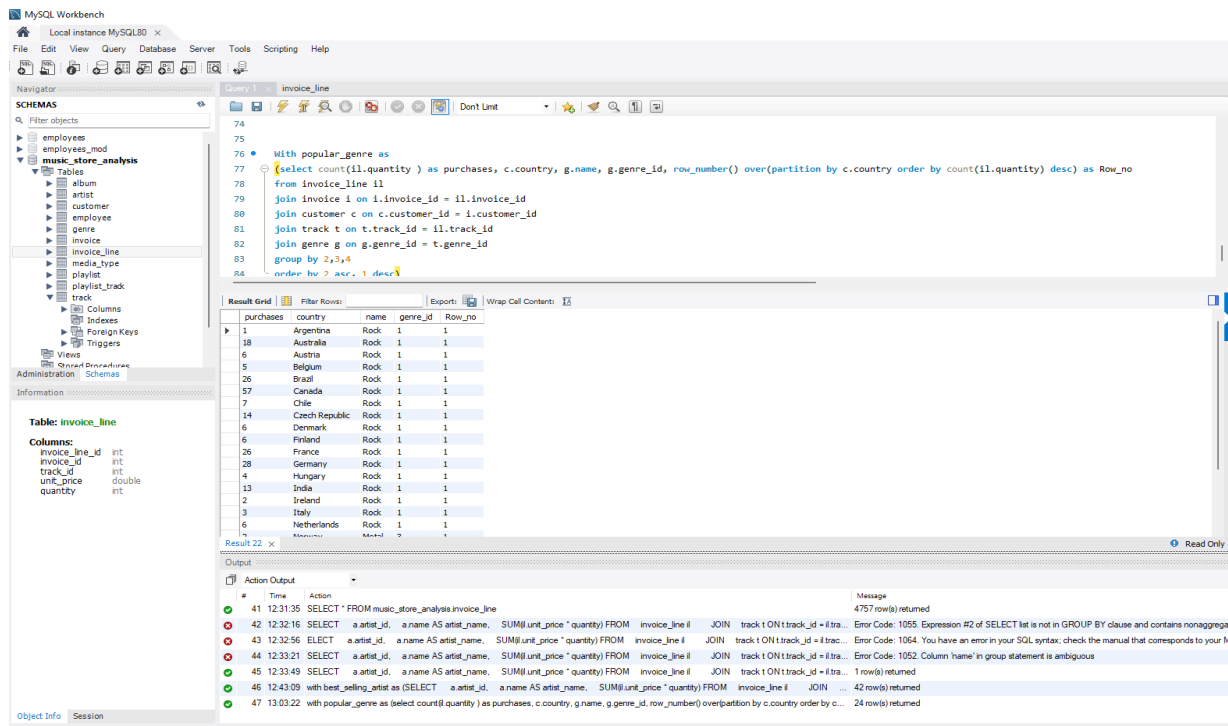
JOIN genre ON genre.genre_id = track.genre_id

GROUP BY 2,3,4

ORDER BY 2 ASC, 1 DESC

)

SELECT * FROM popular_genre WHERE RowNo <= 1



Q11: 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.

Ans-

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

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

- Filter objects
- employees
- employees_mod
- music_store_analysis
 - Tables
 - album
 - artist
 - customer
 - employee
 - genre
 - invoice
 - invoice_line
 - media_type
 - playlist_track
 - track
 - Columns
 - Indexes
 - Foreign Keys
 - Triggers
- Views
- Stored Procedures

Administration Schemas

Information

Table: invoice_line

Columns:

- invoice_line_id int
- invoice_id int
- track_id int
- unit_price double
- quantity int

Query 1: invoice_line

```
86
87
88
89 with money_spent as (select row_number() over(partition by c.country) as row_no, c.customer_id, concat(c.first_name,c.last_name) as name, sum(i.total), c.country
90 from customer c
91 join Invoice i on c.customer_id = i.customer_id
92 group by c.country,3,3
93 order by 1)
94 select * from money_spent where row_no <= 1;
```

Result Grid

row_no	customer_id	name	sum(i.total)	country
1	56	DiegoGustÃ³rrez	39.6	Argentina
1	55	MarkTaylor	81.18	Australia
1	7	AstridGruber	69.3	Austria
1	8	DaanPeeters	60.389999999999999	Belgium
1	12	RobertoAlmeida	82.17	Brazil
1	59	RohabhMishra	71.28	India
1	40	DominiqueLefebvre	72.27	France
1	48	JohannesVan der Berg	65.34	Netherlands
1	44	TerttuSavolainen	79.2	Finland
1	30	EdwardFrancis	91.08	Canada
1	57	LuizRojas	97.020000000000001	Chile
1	9	Karalieshen	37.619999999999999	Denmark
1	38	NiklasSchÃ¼tler	75.259999999999999	Germany
1	47	LucasManoni	50.49	Italy
1	46	HughO'Reilly	114.839999999999997	Ireland
1	6	HelenakollÃ¡s	128.7	Czech Rep...
1	45	LadislavKovÃ¡cs	78.21	Hungary
1	4	BjÃrntsen	72.270000000000001	Norway
1	49	StanisÅlawWiÅki	76.229999999999999	Poland
1	35	MadalenaSampaio	82.17	Portugal
1	50	EnriqueMÃ¼Ã±oz	98.01	Spain
1	51	JoakimJohansson	75.24	Sweden
1	54	SteveMurray	79.2	United King...
1	20	DanMiller	95.039999999999999	USA

Result 40

Output

Action Output

Time Action Message

- 73 15:30:39 select row_number() over(partition by c.country), c.customer_id, concat(c.first_name,c.last_name) as name, (i.total), c.country from customer c join... Error Code: 1055. Expression #4 of SELECT list is not in GROUP BY clause and contains nonaggrega...
- 74 15:30:54 select row_number() over(partition by c.country), c.customer_id, concat(c.first_name,c.last_name) as name, (i.total), c.country from customer c join... 433 row(s) returned
- 75 15:33:08 with money_spent as (select row_number() over(partition by c.country) as row_no, c.customer_id, concat(c.first_name,c.last_name) as name, sum... 24 row(s) returned

Object Info Session

