[Buzzing Tone Musics APP]

SQL Data Analysis project

Kshitij bante [kshitijbante@gmail.com]

2-12-2025

This project features a collection of SQL queries crafted to extract distinct insights from a simulated Buzzing tones Musics database.

❖ Purpose

The Buzzing tones Musics SQL Project uses SQL queries to explore and understand a typical music store's data. The queries aim to answer important questions about Buzztone Musics' operations, customer habits, and sales, offering useful insights that can help improve the business and increase profits.

❖ Data

The data is stored in several tables within the database. Based on the queries, some of the tables include employee, invoice, customer, invoice_line, track, genre, artist, and album.

❖ Analysis Approach

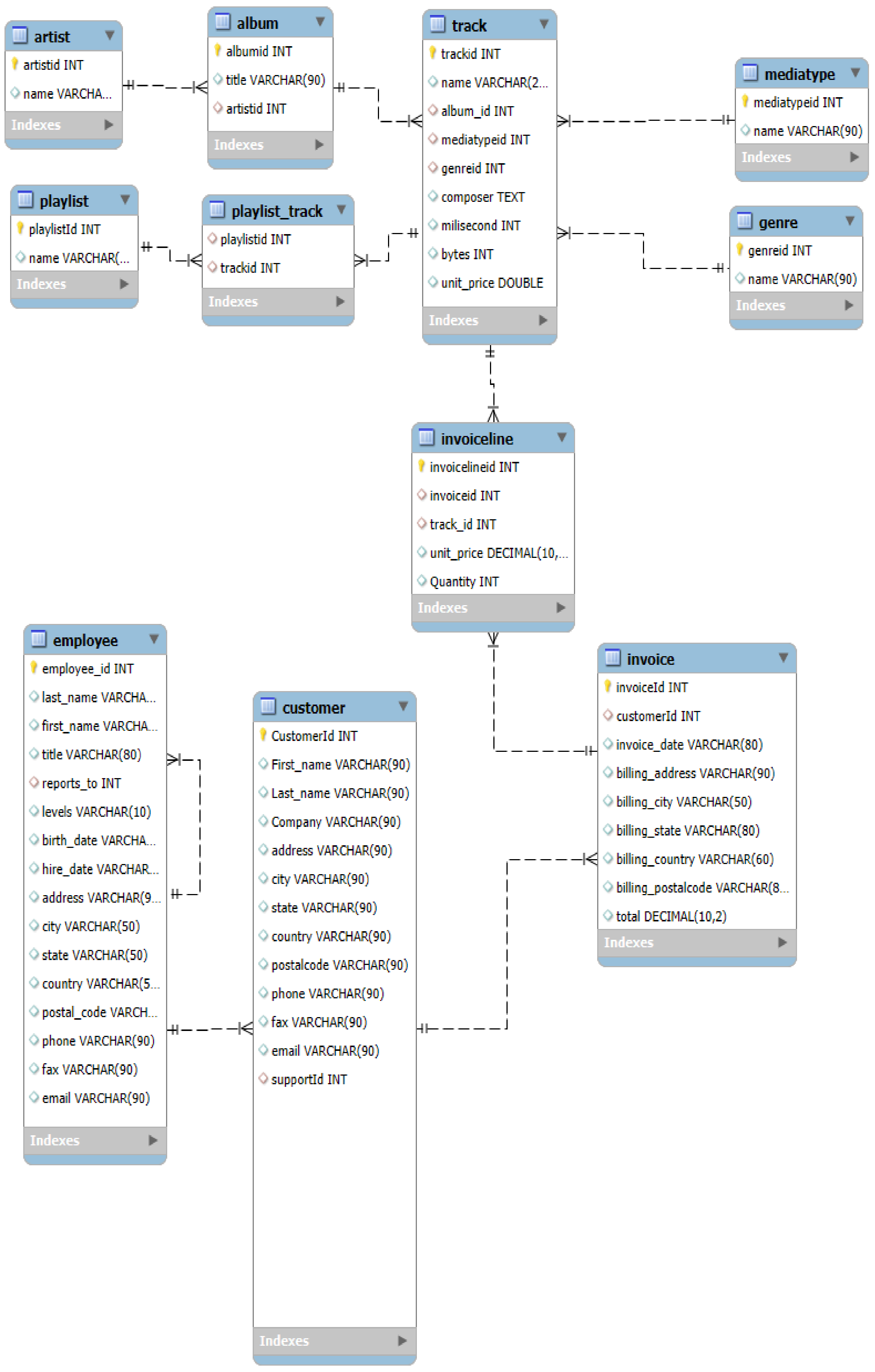
The project is segmented into three tiers of complexity: **Easy, Moderate, and Advanced.**

- **Easy Level:** Features basic queries that focus on direct data retrieval, such as pinpointing top customers or employees.
- **Moderate Level:** Encompasses intermediate queries that dig deeper, using complex JOIN operations, GROUP BY clauses, and aggregate functions like SUM and COUNT to derive more nuanced insights.
- **Advanced Level:** Showcases the power of advanced SQL techniques. It prominently uses Common Table Expressions (CTEs) and window functions like ROW_NUMBER to answer complex queries.

❖ SQL Constructs Used

The project showcases a wide range of SQL constructs to address various querying needs:

- **Data Retrieval:** SELECT, DISTINCT, and FROM.
- **Filtering:** WHERE, IN, and LIMIT.
- **Aggregation:** SUM, COUNT, AVG.
- **Sorting:** ORDER BY.
- **Joining Tables:** JOIN.
- **Grouping Data:** GROUP BY.
- **Window Functions:** ROW_NUMBER.
- **Subqueries and Derived Tables:** WITH (for CTEs).



ER-Diagram of BuzzingTone

- **Easy Level Questions:**

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.
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.

- **Moderate Level Questions:**

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.

2. 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.
3. Returnallthetracknamesthathaveasonglengthlongerthantheaveragesonglength. Return the Name
and Milliseconds for each track. Order by the song length with the longest songs listed first.

- **Advance Level Questions:**

1. Find how much amount spent by each customer on artists? Write a query to return customer name,
artist name and total spent.
2. 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.
3. Writeaquerythatdeterminesthecustomerthathasspentthemostonmusicforeach 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.

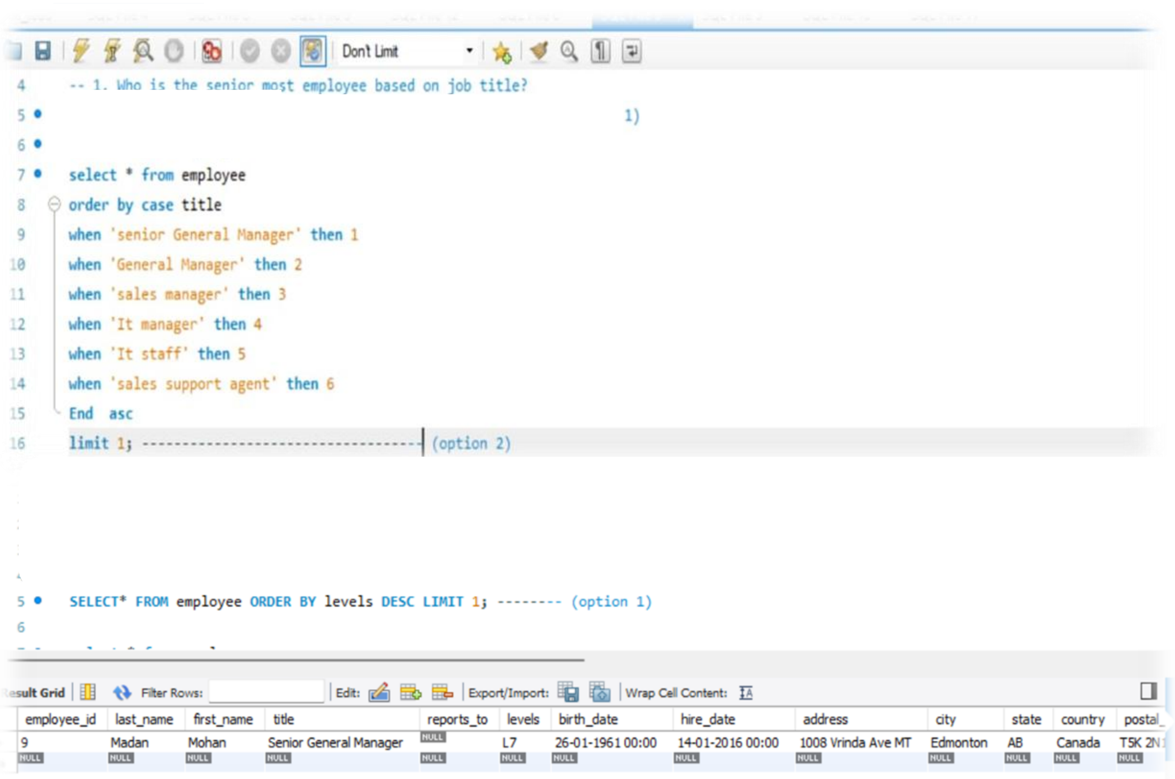
1. Who is the senior most employee based on job title?

Method:1

```
SELECT * FROM employee ORDER BY levels
DESC LIMIT 1;
```

Method:2

```
select * from employee
order by case title
when 'senior General Manager' then 1
when 'General Manager' then 2
when 'sales manager' then 3
when 'It manager' then 4
when 'It staff' then 5
when 'sales support agent' then 6
End  asc
limit 1;
```



```
-- 1. Who is the senior most employee based on job title?
                                                          1)


select * from employee
order by case title
when 'senior General Manager' then 1
when 'General Manager' then 2
when 'sales manager' then 3
when 'It manager' then 4
when 'It staff' then 5
when 'sales support agent' then 6
End  asc
limit 1; ----------------------------------| (option 2)
```

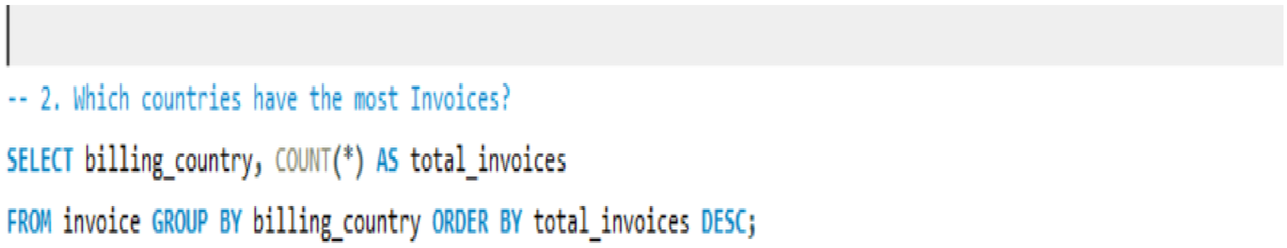
```
SELECT* FROM employee ORDER BY levels DESC LIMIT 1; -------- (option 1)
```

employee_id	last_name	first_name	title	reports_to	levels	birth_date	hire_date	address	city	state	country	postal_
9	Madan	Mohan	Senior General Manager	NULL	L7	26-01-1961 00:00	14-01-2016 00:00	1008 Vrinda Ave MT	Edmonton	AB	Canada	T5K 2N
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

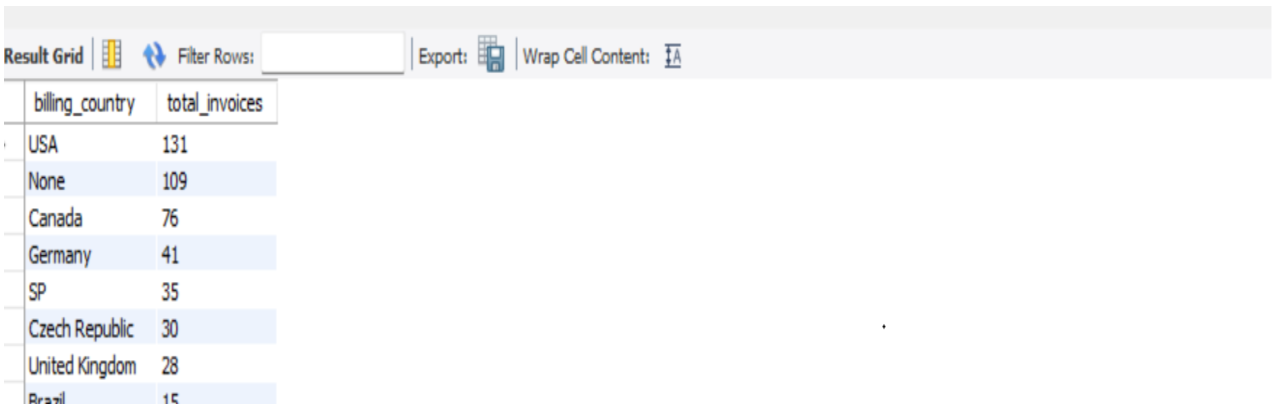
2. Which countries have the most Invoices?

SELECT billing_country, COUNT(*) AS total_invoices

FROM invoice GROUP BY billing_country ORDER BY total_invoices DESC;



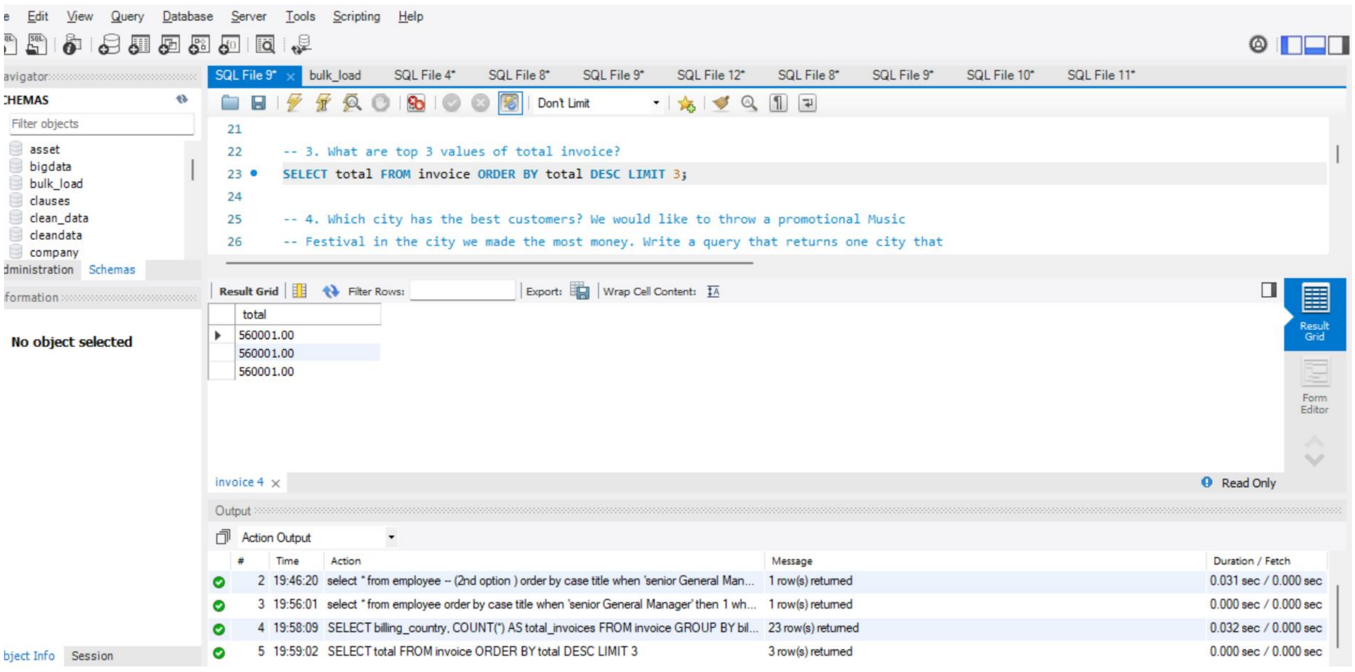
```
-- 2. Which countries have the most Invoices?
SELECT billing_country, COUNT(*) AS total_invoices
FROM invoice GROUP BY billing_country ORDER BY total_invoices DESC;
```



billing_country	total_invoices
USA	131
None	109
Canada	76
Germany	41
SP	35
Czech Republic	30
United Kingdom	28
Brazil	15

3. What are top 3 values of total invoice?

SELECT total FROM invoice ORDER BY total DESC LIMIT 3;


Edit View Query Database Server Tools Scripting Help
SQL File 9* bulk_load SQL File 4* SQL File 8* SQL File 9* SQL File 12* SQL File 8* SQL File 9* SQL File 10* SQL File 11*
Don't Limit
asset
bigdata
bulk_load
clauses
clean_data
cleandata
company
No object selected
Result Grid Filter Rows: Export: Wrap Cell Content:
invoice 4
Read Only
Output
Action Output

```
-- 3. What are top 3 values of total invoice?
SELECT total FROM invoice ORDER BY total DESC LIMIT 3;

-- 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
```

total
560001.00
560001.00
560001.00

#	Time	Action	Message	Duration / Fetch
2	19:46:20	select * from employee -- (2nd option) order by case title when 'senior General Man...	1 row(s) returned	0.031 sec / 0.000 sec
3	19:56:01	select * from employee order by case title when 'senior General Manager' then 1 wh...	1 row(s) returned	0.000 sec / 0.000 sec
4	19:58:09	SELECT billing_country, COUNT(*) AS total_invoices FROM invoice GROUP BY bil...	23 row(s) returned	0.032 sec / 0.000 sec
5	19:59:02	SELECT total FROM invoice ORDER BY total DESC LIMIT 3	3 row(s) returned	0.000 sec / 0.000 sec

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.

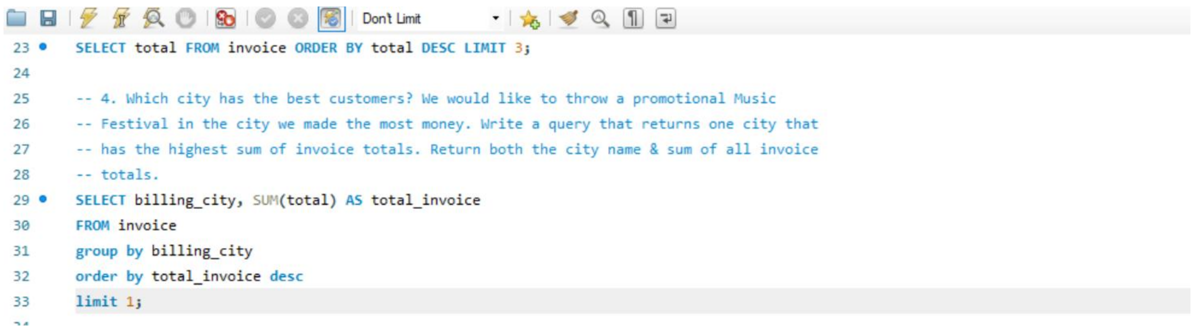
SELECT billing_city, SUM(total) AS total_invoice

FROM invoice

group by billing_city

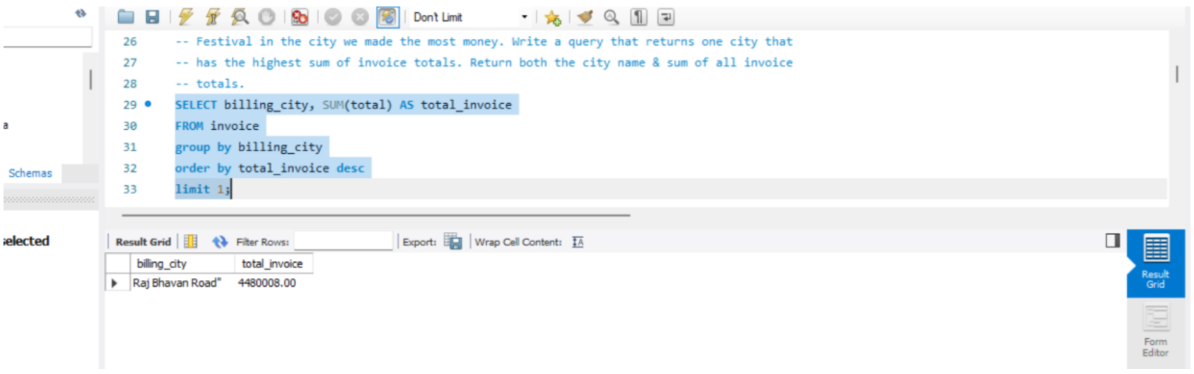
order by total_invoice desc

limit 1;



```
SELECT total FROM invoice ORDER BY total DESC LIMIT 3;

-- 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.
SELECT billing_city, SUM(total) AS total_invoice
FROM invoice
group by billing_city
order by total_invoice desc
limit 1;
```



```
-- 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 total_invoice
FROM invoice
group by billing_city
order by total_invoice desc
limit 1;
```

billing_city	total_invoice
Raj Bhavan Road"	4480008.00

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.

SELECT c.customerid, CONCAT(c.first_name," ",c.last_name) AS full_name, SUM(i.total) AS total_spend

FROM customer AS c

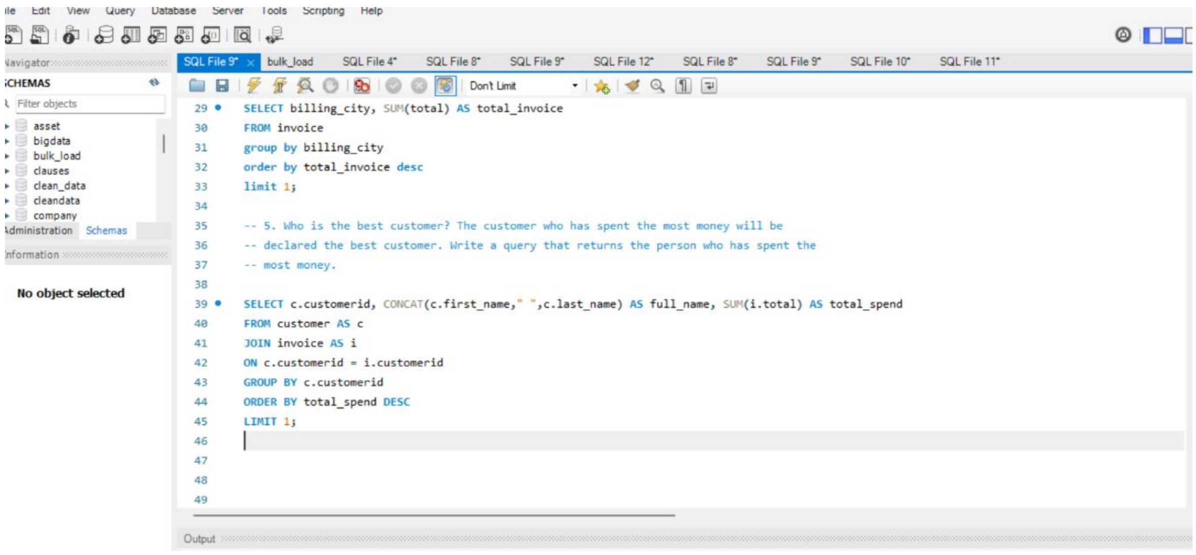
JOIN invoice AS i

ON c.customerid = i.customerid

GROUP BY c.customerid

ORDER BY total_spend DESC

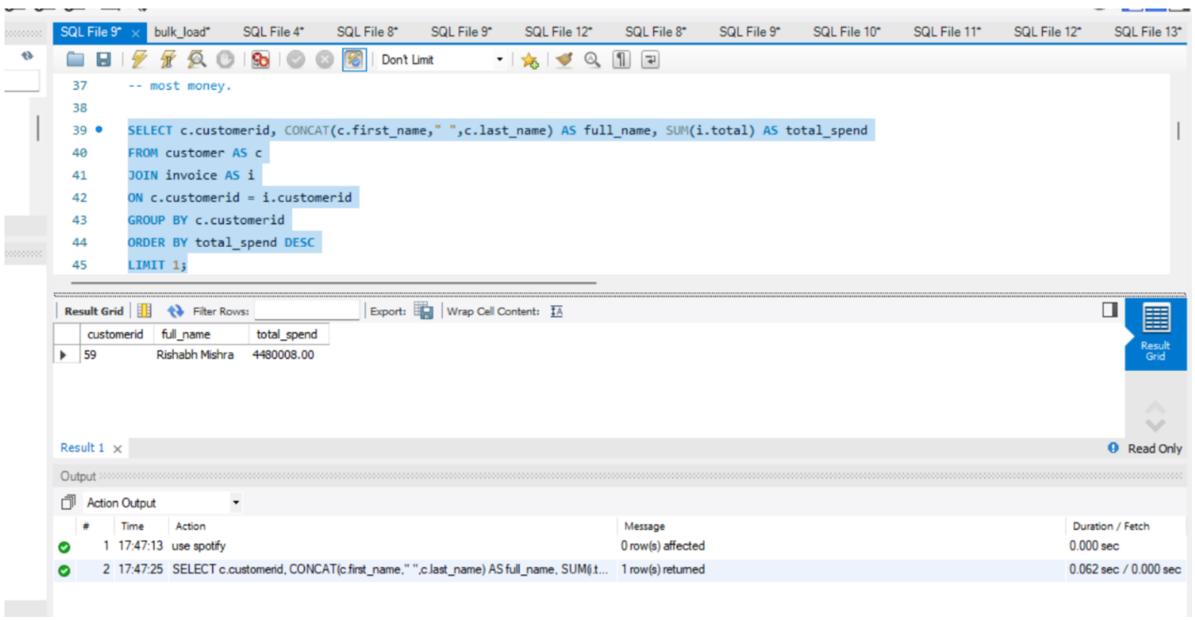
LIMIT 1;



```
SELECT billing_city, SUM(total) AS total_invoice
FROM invoice
group by billing_city
order by total_invoice desc
limit 1;

-- 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.

SELECT c.customerid, CONCAT(c.first_name," ",c.last_name) AS full_name, SUM(i.total) AS total_spend
FROM customer AS c
JOIN invoice AS i
ON c.customerid = i.customerid
GROUP BY c.customerid
ORDER BY total_spend DESC
LIMIT 1;
```



```
-- most money.

SELECT c.customerid, CONCAT(c.first_name," ",c.last_name) AS full_name, SUM(i.total) AS total_spend
FROM customer AS c
JOIN invoice AS i
ON c.customerid = i.customerid
GROUP BY c.customerid
ORDER BY total_spend DESC
LIMIT 1;
```

```
customerid   full_name        total_spend
59           Rishabh Mishra   4480008.00
```

```
#   Time       Action                                                                    Message               Duration / Fetch
1   17:47:13   use spotify                                                               0 row(s) affected     0.000 sec
2   17:47:25   SELECT c.customerid, CONCAT(c.first_name," ",c.last_name) AS full_name, SUM(i.t...   1 row(s) returned     0.062 sec / 0.000 sec
```

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

SELECT DISTINCT c.first_name, c.last_name, c.email, g.`name`

FROM customer AS c

JOIN invoice AS i ON c.customerId = i.customerid

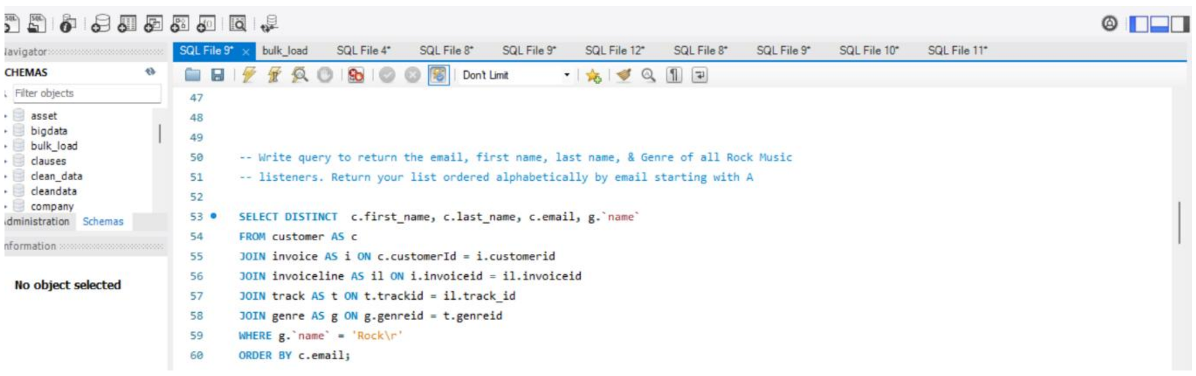
JOIN invoiceline AS il ON i.invoiceid = il.invoiceid

JOIN track AS t ON t.trackid = il.track_id

JOIN genre AS g ON g.genreid = t.genreid

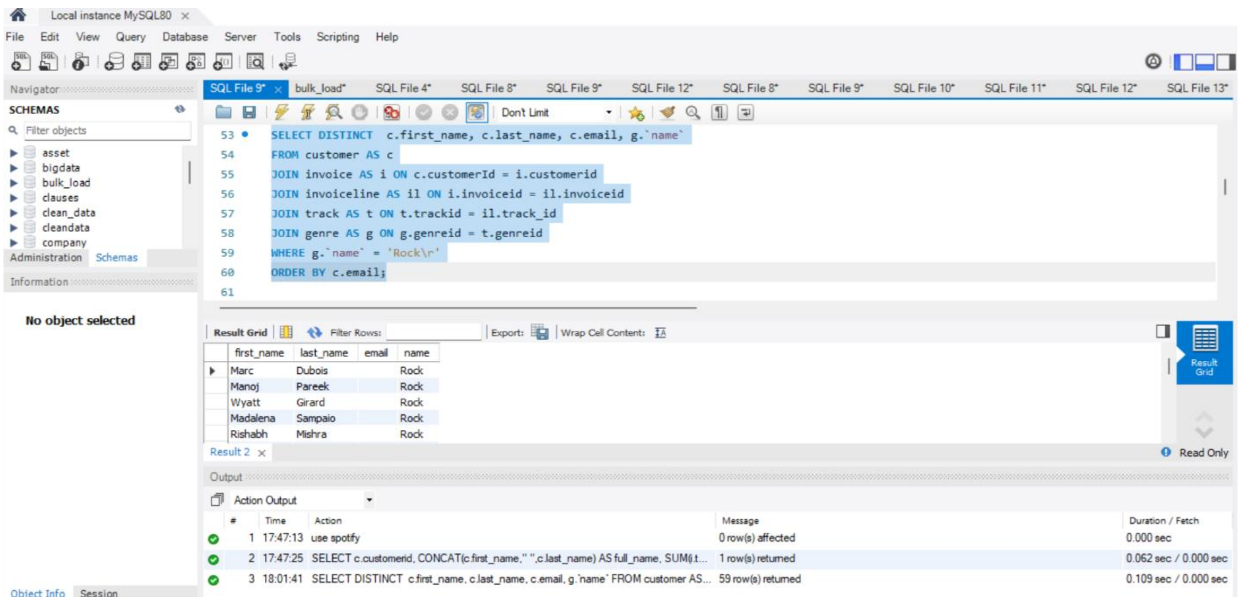
WHERE g.`name` = 'Rock\r'

ORDER BY c.email;



```
-- 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  c.first_name, c.last_name, c.email, g.`name`
FROM customer AS c
JOIN invoice AS i ON c.customerId = i.customerid
JOIN invoiceline AS il ON i.invoiceid = il.invoiceid
JOIN track AS t ON t.trackid = il.track_id
JOIN genre AS g ON g.genreid = t.genreid
WHERE g.`name` = 'Rock\r'
ORDER BY c.email;
```



```
SELECT DISTINCT  c.first_name, c.last_name, c.email, g.`name`
FROM customer AS c
JOIN invoice AS i ON c.customerId = i.customerid
JOIN invoiceline AS il ON i.invoiceid = il.invoiceid
JOIN track AS t ON t.trackid = il.track_id
JOIN genre AS g ON g.genreid = t.genreid
WHERE g.`name` = 'Rock\r'
ORDER BY c.email;
```

first_name	last_name	email	name
Marc	Dubois		Rock
Manoj	Pareek		Rock
Wyatt	Girard		Rock
Madalena	Sampaio		Rock
Rishabh	Mishra		Rock

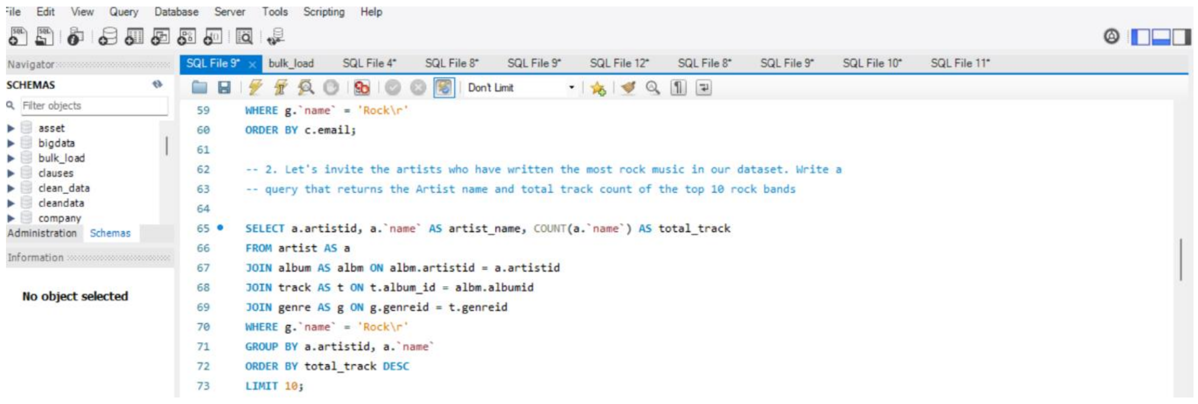
Output

#	Time	Action	Message	Duration / Fetch
1	17:47:13	use spotify	0 row(s) affected	0.000 sec
2	17:47:25	SELECT c.customerid, CONCAT(c.first_name," ",c.last_name) AS full_name, SUM(t...	1 row(s) returned	0.062 sec / 0.000 sec
3	18:01:41	SELECT DISTINCT c.first_name, c.last_name, c.email, g.`name` FROM customer AS...	59 row(s) returned	0.109 sec / 0.000 sec

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

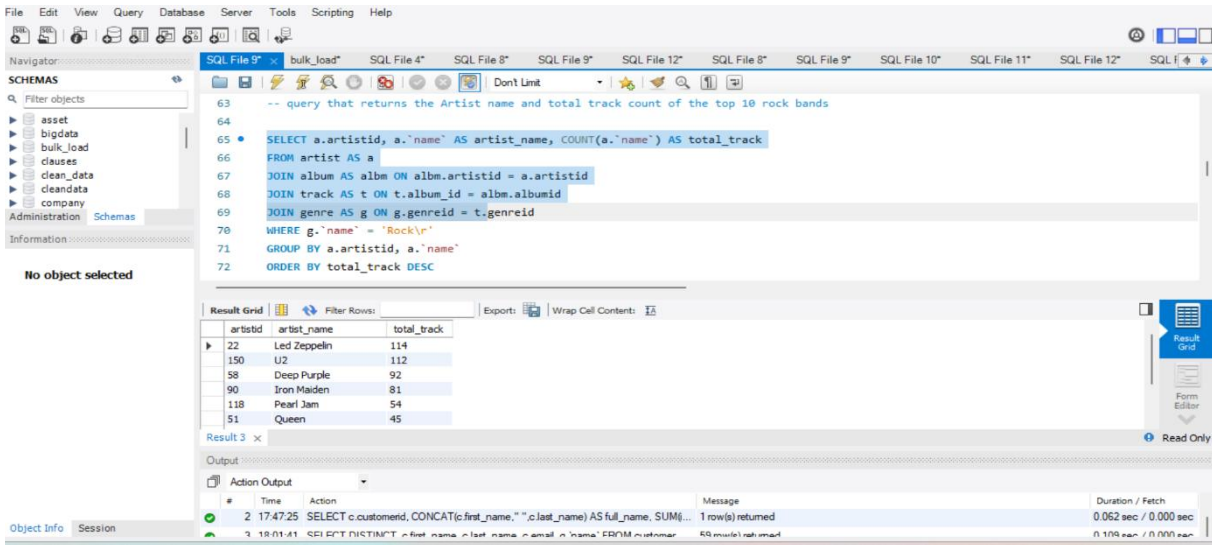
```
SELECT a.artistid, a.`name` AS artist_name, COUNT(a.`name`) AS total_track
FROM artist AS a
JOIN album AS albm ON albm.artistid = a.artistid
JOIN track AS t ON t.album_id = albm.albumid
JOIN genre AS g ON g.genreid = t.genreid
WHERE g.`name` = 'Rock\r'
GROUP BY a.artistid, a.`name`
ORDER BY total_track DESC
LIMIT 10;
```



```
WHERE g.`name` = 'Rock\r'
ORDER BY c.email;

-- 2. 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 a.artistid, a.`name` AS artist_name, COUNT(a.`name`) AS total_track
FROM artist AS a
JOIN album AS albm ON albm.artistid = a.artistid
JOIN track AS t ON t.album_id = albm.albumid
JOIN genre AS g ON g.genreid = t.genreid
WHERE g.`name` = 'Rock\r'
GROUP BY a.artistid, a.`name`
ORDER BY total_track DESC
LIMIT 10;
```



```
-- query that returns the Artist name and total track count of the top 10 rock bands

SELECT a.artistid, a.`name` AS artist_name, COUNT(a.`name`) AS total_track
FROM artist AS a
JOIN album AS albm ON albm.artistid = a.artistid
JOIN track AS t ON t.album_id = albm.albumid
JOIN genre AS g ON g.genreid = t.genreid
WHERE g.`name` = 'Rock\r'
GROUP BY a.artistid, a.`name`
ORDER BY total_track DESC
```

artistid	artist_name	total_track
22	Led Zeppelin	114
150	U2	112
58	Deep Purple	92
90	Iron Maiden	81
118	Pearl Jam	54
51	Queen	45

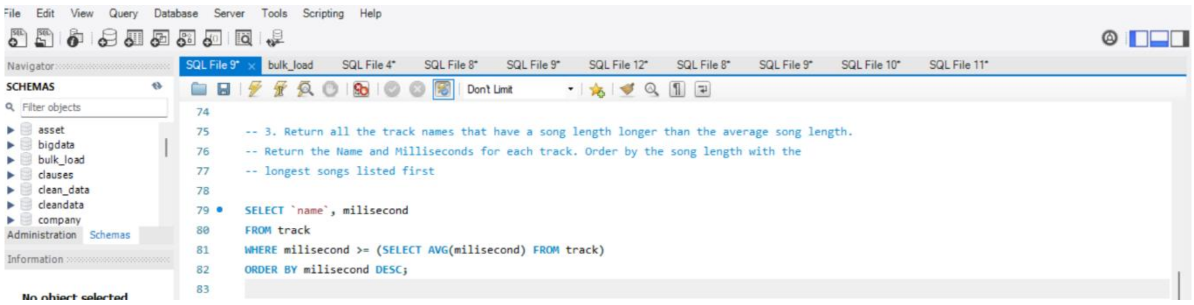
Result 3

Output

#	Time	Action	Message	Duration / Fetch
2	17:47:25	SELECT c.customerid, CONCAT(c.first_name," ",c.last_name) AS full_name, SUM(...	1 row(s) returned	0.062 sec / 0.000 sec
3	18:01:41	SELECT DISTINCT c.first_name, c.last_name, c.email, g.`name` FROM customer ...	59 row(s) returned	0.109 sec / 0.000 sec

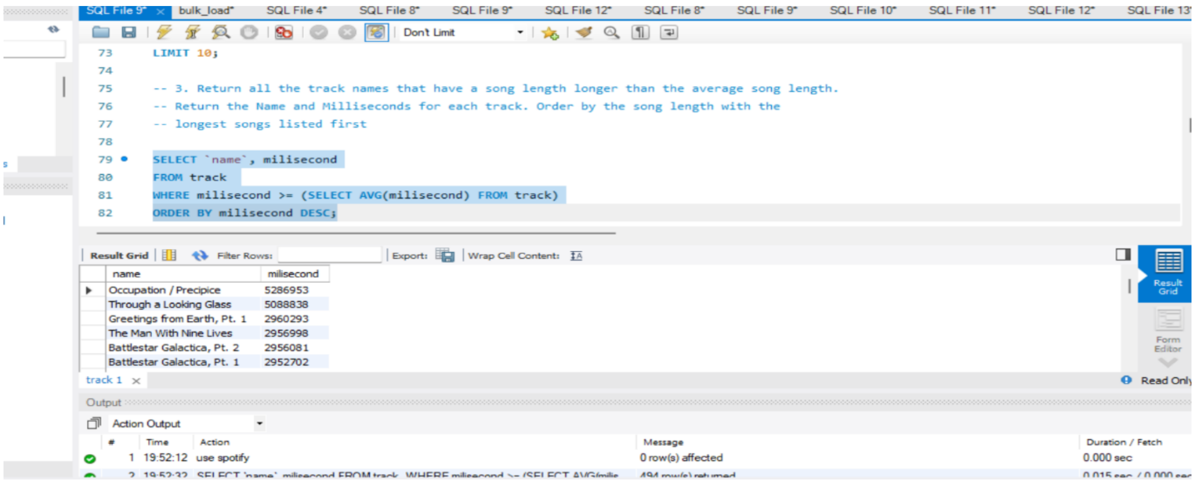
8. Return all the track names that have a song length longer than the average song length. Return the Name and Milliseconds for each track. Order by the song length with the longest songs listed first

```
SELECT `name`, milisecond
FROM track
WHERE milisecond >= (SELECT AVG(milisecond) FROM track)
ORDER BY milisecond DESC;
```



```
-- 3. 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`, milisecond
FROM track
WHERE milisecond >= (SELECT AVG(milisecond) FROM track)
ORDER BY milisecond DESC;
```



```
LIMIT 10;

-- 3. 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`, milisecond
FROM track
WHERE milisecond >= (SELECT AVG(milisecond) FROM track)
ORDER BY milisecond DESC;
```

name	milisecond
Occupation / Precipice	5286953
Through a Looking Glass	5088838
Greetings from Earth, Pt. 1	2960293
The Man With Nine Lives	2956998
Battlestar Galactica, Pt. 2	2956081
Battlestar Galactica, Pt. 1	2952702

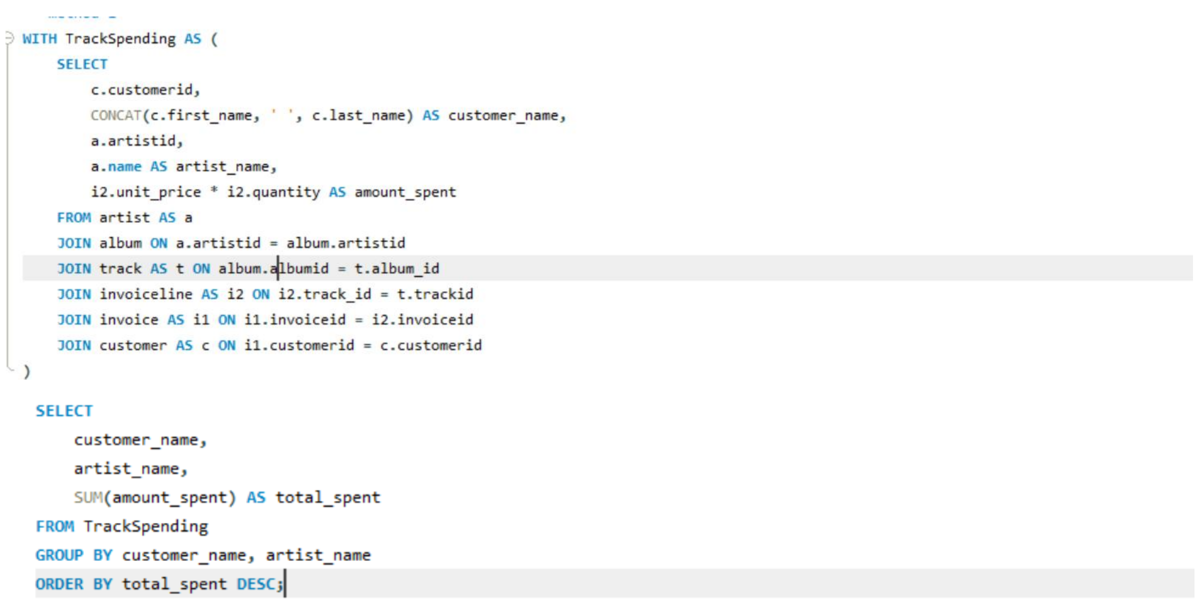
#	Time	Action	Message	Duration / Fetch
1	19:52:12	use spotify	0 row(s) affected	0.000 sec
2	19:52:32	SELECT `name`, milisecond FROM track WHERE milisecond >= (SELECT AVG(milis	494 row(s) returned	0.015 sec / 0.000 sec

9) Find how much amount spent by each customer on artists? Write a query to return

```
-- customer name, artist name and total spent
-- CTE to calculate spending at the track level and attribute it to artists
-- method 1
WITH TrackSpending AS (
    SELECT
        c.customerid,
        CONCAT(c.first_name, ' ', c.last_name) AS customer_name,
        a.artistid,
        a.name AS artist_name,
        i2.unit_price * i2.quantity AS amount_spent
    FROM artist AS a
    JOIN album ON a.artistid = album.artistid
    JOIN track AS t ON album.albumid = t.album_id
    JOIN invoiceline AS i2 ON i2.track_id = t.trackid
    JOIN invoice AS i1 ON i1.invoiceid = i2.invoiceid
    JOIN customer AS c ON i1.customerid = c.customerid)
SELECT
    customer_name,
    artist_name,
    SUM(amount_spent) AS total_spent
FROM TrackSpending
GROUP BY customer_name, artist_name
ORDER BY total_spent DESC;
```

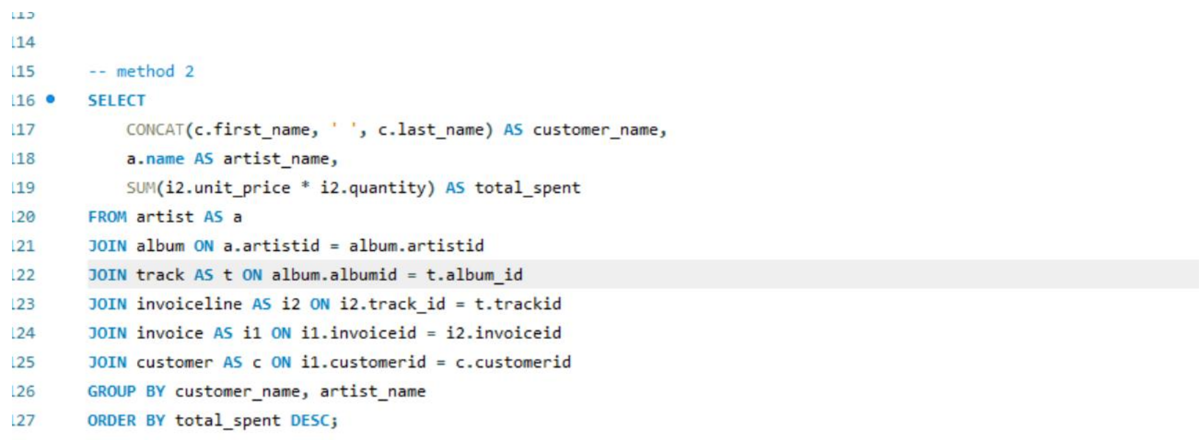
method 2

```
SELECT
  CONCAT(c.first_name, ' ', c.last_name) AS customer_name,
    a.name AS artist_name,
    SUM(i2.unit_price * i2.quantity) AS total_spent
FROM artist AS a
JOIN album ON a.artistid = album.artistid
JOIN track AS t ON album.albumid = t.album_id
JOIN invoiceline AS i2 ON i2.track_id = t.trackid
JOIN invoice AS i1 ON i1.invoiceid = i2.invoiceid
JOIN customer AS c ON i1.customerid = c.customerid
GROUP BY customer_name, artist_name
ORDER BY total_spent DESC;
```

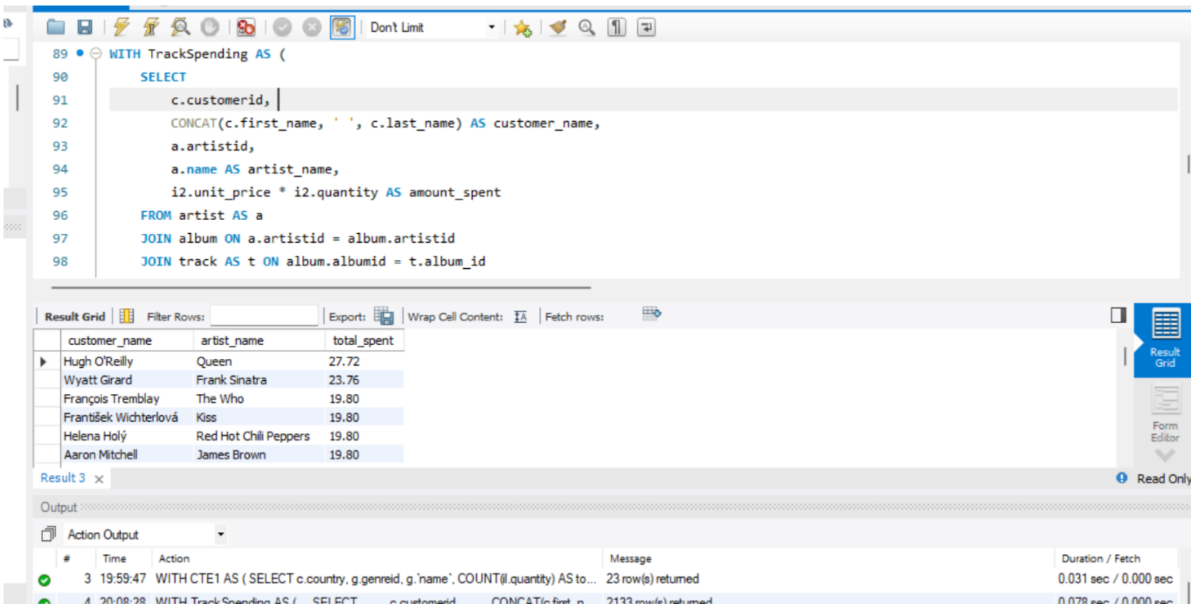


```
WITH TrackSpending AS (
    SELECT
        c.customerid,
        CONCAT(c.first_name, ' ', c.last_name) AS customer_name,
        a.artistid,
        a.name AS artist_name,
        i2.unit_price * i2.quantity AS amount_spent
    FROM artist AS a
    JOIN album ON a.artistid = album.artistid
    JOIN track AS t ON album.albumid = t.album_id
    JOIN invoiceline AS i2 ON i2.track_id = t.trackid
    JOIN invoice AS i1 ON i1.invoiceid = i2.invoiceid
    JOIN customer AS c ON i1.customerid = c.customerid
)

SELECT
    customer_name,
    artist_name,
    SUM(amount_spent) AS total_spent
FROM TrackSpending
GROUP BY customer_name, artist_name
ORDER BY total_spent DESC;
```



```
-- method 2
SELECT
    CONCAT(c.first_name, ' ', c.last_name) AS customer_name,
    a.name AS artist_name,
    SUM(i2.unit_price * i2.quantity) AS total_spent
FROM artist AS a
JOIN album ON a.artistid = album.artistid
JOIN track AS t ON album.albumid = t.album_id
JOIN invoiceline AS i2 ON i2.track_id = t.trackid
JOIN invoice AS i1 ON i1.invoiceid = i2.invoiceid
JOIN customer AS c ON i1.customerid = c.customerid
GROUP BY customer_name, artist_name
ORDER BY total_spent DESC;
```



```sql
WITH TrackSpending AS (
    SELECT
        c.customerid,
        CONCAT(c.first_name, ' ', c.last_name) AS customer_name,
        a.artistid,
        a.name AS artist_name,
        i2.unit_price * i2.quantity AS amount_spent
    FROM artist AS a
    JOIN album ON a.artistid = album.artistid
    JOIN track AS t ON album.albumid = t.album_id
```

customer_name	artist_name	total_spent
Hugh O'Reilly	Queen	27.72
Wyatt Girard	Frank Sinatra	23.76
François Tremblay	The Who	19.80
František Wichterlová	Kiss	19.80
Helena Holý	Red Hot Chili Peppers	19.80
Aaron Mitchell	James Brown	19.80

Result 3

Read Only

Output

Action Output

#	Time	Action	Message	Duration / Fetch
3	19:59:47	WITH CTE1 AS (SELECT c.country, g.genreid, g.`name`, COUNT(il.quantity) AS to...	23 row(s) returned	0.031 sec / 0.000 sec
4	20:08:28	WITH TrackSpending AS (SELECT c.customerid, CONCAT(c.first_n...	2133 row(s) returned	0.078 sec / 0.000 sec

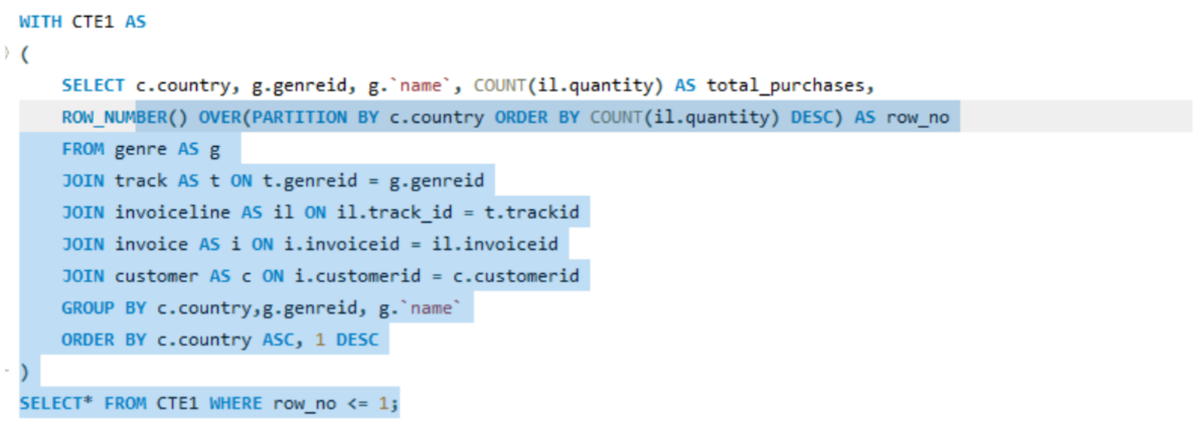
10. We want to find out the most popular music Genre for each country. We determine the

-- most popular genre as the genre with the highest amount of purchases. Write a query

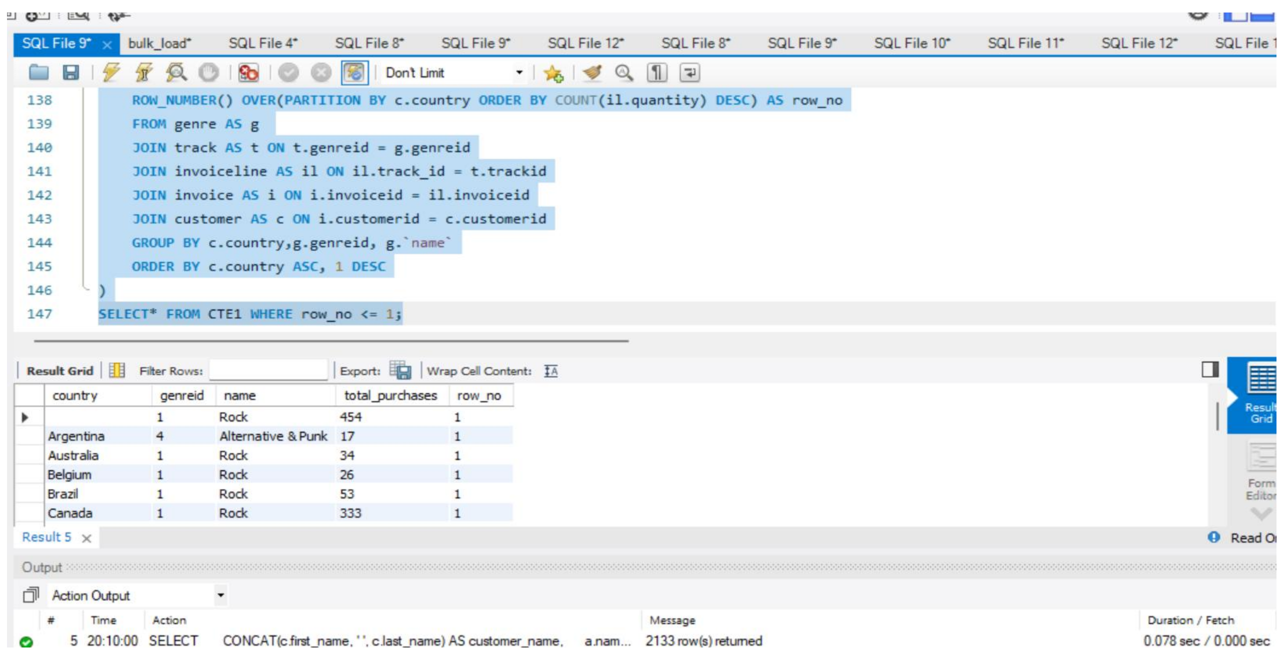
-- that returns each country along with the top Genre. For countries where the maximum

-- number of purchases is shared return all Genres

```
WITH CTE1 AS
(
	SELECT c.country, g.genreid, g.`name`, COUNT(il.quantity) AS total_purchases,
	ROW_NUMBER() OVER(PARTITION BY c.country ORDER BY
COUNT(il.quantity) DESC) AS row_no
	FROM genre AS g
	JOIN track AS t ON t.genreid = g.genreid
	JOIN invoiceline AS il ON il.track_id = t.trackid
	JOIN invoice AS i ON i.invoiceid = il.invoiceid
	JOIN customer AS c ON i.customerid = c.customerid
	GROUP BY c.country,g.genreid, g.`name`
```



```
WITH CTE1 AS
(
    SELECT c.country, g.genreid, g.`name`, COUNT(il.quantity) AS total_purchases,
    ROW_NUMBER() OVER(PARTITION BY c.country ORDER BY COUNT(il.quantity) DESC) AS row_no
    FROM genre AS g
    JOIN track AS t ON t.genreid = g.genreid
    JOIN invoiceline AS il ON il.track_id = t.trackid
    JOIN invoice AS i ON i.invoiceid = il.invoiceid
    JOIN customer AS c ON i.customerid = c.customerid
    GROUP BY c.country,g.genreid, g.`name`
    ORDER BY c.country ASC, 1 DESC
)
SELECT* FROM CTE1 WHERE row_no <= 1;
```



```
    ROW_NUMBER() OVER(PARTITION BY c.country ORDER BY COUNT(il.quantity) DESC) AS row_no
    FROM genre AS g
    JOIN track AS t ON t.genreid = g.genreid
    JOIN invoiceline AS il ON il.track_id = t.trackid
    JOIN invoice AS i ON i.invoiceid = il.invoiceid
    JOIN customer AS c ON i.customerid = c.customerid
    GROUP BY c.country,g.genreid, g.`name`
    ORDER BY c.country ASC, 1 DESC
)
SELECT* FROM CTE1 WHERE row_no <= 1;
```

country	genreid	name	total_purchases	row_no
	1	Rock	454	1
Argentina	4	Alternative & Punk	17	1
Australia	1	Rock	34	1
Belgium	1	Rock	26	1
Brazil	1	Rock	53	1
Canada	1	Rock	333	1

```
#  Time      Action                                                     Message               Duration / Fetch
5  20:10:00  SELECT  CONCAT(c.first_name, ' ', c.last_name) AS customer_name,  a.nam...  2133 row(s) returned  0.078 sec / 0.000 sec
```

11.Write a query that determines the customer that has spent the most on music for each

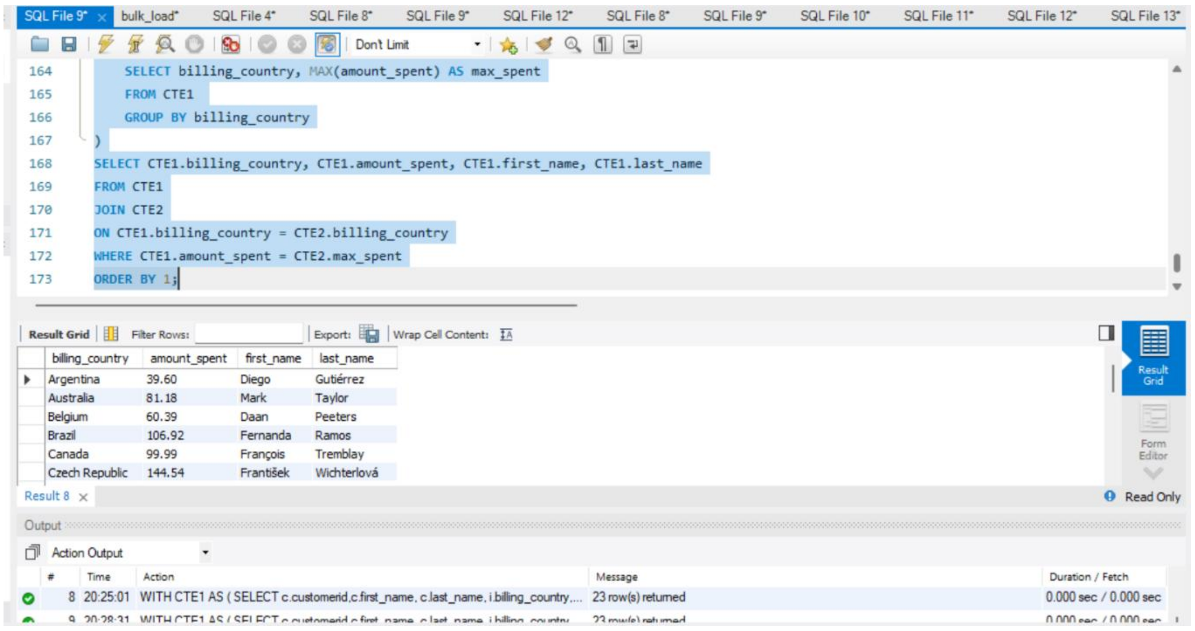
-- country. Write a query that returns the country along with the top customer and how

-- much they spent. For countries where the top amount spent is shared, provide all

-- customers who spent this amount

```
	WITH CTE1 AS
(
	SELECT c.customerid,c.first_name, c.last_name, i.billing_country, SUM(i.total) AS amount_spent
	FROM customer AS c
	JOIN invoice AS i ON c.customerid = i.customerid
	GROUP BY c.customerid,c.first_name, c.last_name, i.billing_country
	ORDER BY c.customerid,  amount_spent  DESC
),
CTE2 AS
(
	SELECT billing_country, MAX(amount_spent) AS max_spent
	FROM CTE1
	GROUP BY billing_country
)
SELECT CTE1.billing_country, CTE1.amount_spent, CTE1.first_name, CTE1.last_name
FROM CTE1
JOIN CTE2
ON CTE1.billing_country = CTE2.billing_country
WHERE CTE1.amount_spent = CTE2.max_spent
ORDER BY 1;
```

```sql
    WITH CTE1 AS
(
    SELECT c.customerid,c.first_name, c.last_name, i.billing_country, SUM(i.total) AS amount_spent
    FROM customer AS c
    JOIN invoice AS i ON c.customerid = i.customerid
    GROUP BY c.customerid,c.first_name, c.last_name, i.billing_country
    ORDER BY c.customerid,  amount_spent  DESC
),
CTE2 AS
(
    SELECT billing_country, MAX(amount_spent) AS max_spent
    FROM CTE1
    GROUP BY billing_country
)
```



SQL File 9* | bulk_load* | SQL File 4* | SQL File 8* | SQL File 9* | SQL File 12* | SQL File 8* | SQL File 9* | SQL File 10* | SQL File 11* | SQL File 12* | SQL File 13*

Don't Limit

```sql
    SELECT billing_country, MAX(amount_spent) AS max_spent
    FROM CTE1
    GROUP BY billing_country
)
SELECT CTE1.billing_country, CTE1.amount_spent, CTE1.first_name, CTE1.last_name
FROM CTE1
JOIN CTE2
ON CTE1.billing_country = CTE2.billing_country
WHERE CTE1.amount_spent = CTE2.max_spent
ORDER BY 1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

billing_country	amount_spent	first_name	last_name
Argentina	39.60	Diego	Gutiérrez
Australia	81.18	Mark	Taylor
Belgium	60.39	Daan	Peeters
Brazil	106.92	Fernanda	Ramos
Canada	99.99	François	Tremblay
Czech Republic	144.54	František	Wichterlová

Result 8 | Read Only

Output

Action Output

#	Time	Action	Message	Duration / Fetch
8	20:25:01	WITH CTE1 AS (SELECT c.customerid,c.first_name, c.last_name, i.billing_country,...	23 row(s) returned	0.000 sec / 0.000 sec
9	20:28:31	WITH CTE1 AS (SELECT c.customerid,c.first_name, c.last_name, i.billing_country	23 row(s) returned	0.000 sec / 0.000 sec