

SQL QUERY

```
1 SELECT COUNT(name) AS Number_of_Tracks FROM tracks
```

 Execute Clear Beautify Minify Reload**RESULT****Number_of_Tracks**

3503

SQL QUERY

```
1 SELECT AVG(total) AS Invoice_Average FROM invoices
```

 Execute Clear Beautify Minify Reload**RESULT****Invoice_Average**

5.651941747572825

SQL QUERY

```
1 SELECT Country, COUNT(CustomerId) AS Number_of_Customers FROM customers
2 GROUP BY Country
3 HAVING COUNT(CustomerId) > 2
4 ORDER BY COUNT(CustomerId) DESC
```

▶ Execute

✍ Clear

✍ Beautify

✂ Minify

↻ Reload

RESULT

Country	Number_of_Customers
USA	13
Canada	8
Brazil	5

SQL QUERY

```
1 SELECT BillingCountry, SUM(Total) AS Regional_Total FROM invoices
2 ORDER BY Regional_Total DESC
3
4
5 The above code should work and with the logic applied to other databases, does
6 I'm going to take the loss on this one because I'm clueless as to what is going on
7 under the hood.
```

▶ Execute

✍ Clear

✍ Beautify

✂ Minify

↻ Reload

RESULT

BillingCountry	Regional_Total
India	2328.600000000004

SQL QUERY

```
1 SELECT Name, MAX(Milliseconds) AS Duration FROM tracks
```

▶ Execute

✂ Clear

✍ Beautify

✖ Minify

↻ Reload

RESULT

Name	Duration
Occupation / Precipice	5286953

main.sql

```
1 .open chinook.db
2 .tables
3 /*
4  Display the name of the album along with the name of it's shortest
5  track. The actual duration should NOT be displayed.
6 */
7 SELECT title, tracks.Name FROM albums
8 JOIN tracks ON albums.AlbumId = tracks.AlbumId
9 GROUP BY tracks.AlbumId
10 ORDER BY MIN(tracks.Milliseconds)
11 LIMIT 5
```

Interpret

17s on 13:01:28, 02/09

Body Count
House of Pain
Os Cães Ladram Mas A Caravana Não Pára
As Canções de Eu Tu Eles
SQLite version 3.39.4 2022-09-29 15:55:41
Enter ".help" for usage hints.

Interpret

1m on 13:01:47, 02/09

-- Loading resources from main.sql
albums employees invoices playlists
artists genres media_types tracks
customers invoice_items playlist_track
O Samba Poconé|É Uma Partida De Futebol
Body Count|Now Sports
House of Pain|Commercial 1
Os Cães Ladram Mas A Caravana Não Pára|Bossa
As Canções de Eu Tu Eles|Casinha Feliz
SQLite version 3.39.4 2022-09-29 15:55:41
Enter ".help" for usage hints.

Interpret

1m on 13:03:14, 02/09

-- Loading resources from main.sql
albums employees invoices playlists
artists genres media_types tracks
customers invoice_items playlist_track
O Samba Poconé|É Uma Partida De Futebol|1071
Body Count|Now Sports|4884
House of Pain|Commercial 1|7941
Os Cães Ladram Mas A Caravana Não Pára|Bossa|29048
As Canções de Eu Tu Eles|Casinha Feliz|32287
SQLite version 3.39.4 2022-09-29 15:55:41
Enter ".help" for usage hints.

AI SQL

Ln 7, Col 26 • Spaces: 2 History

```
main.sql x +
main.sql
1 .open chinook.db
2 .tables
3 /*
4  Display the top 10 tracks.name and sum of invoice_items.UnitPrice
  (as Total_Earned). Sort the results by those tracks that earned the
  most in descending order.
5  The following is not yielding a correct answer but it is what I think
  the code should look like, much like 5.2.1.
6  */
7 SELECT tracks.name, SUM(invoice_items.UnitPrice)/invoice_items.Quantity
  AS Total_Earned FROM tracks
8 INNER JOIN invoice_items ON tracks.TrackId = invoice_items.TrackId
9 ORDER BY Total_Earned DESC
10 LIMIT 10

Console x Shell x +
Interpret 2m on 11:34:50, 02/14
Enter ".help" for usage hints.
>

Interpret 18s on 11:36:57, 02/14
-- Loading resources from main.sql
albums      employees  invoices    playlists
artists     genres      media_types tracks
customers   invoice_items playlist_track
Balls to the Wall|2328.599999999996
SQLite version 3.39.4 2022-09-29 15:55:41
Enter ".help" for usage hints.
>

Interpret 2m on 11:37:16, 02/14
-- Loading resources from main.sql
albums      employees  invoices    playlists
artists     genres      media_types tracks
customers   invoice_items playlist_track
Balls to the Wall|2328.599999999996
SQLite version 3.39.4 2022-09-29 15:55:41
Enter ".help" for usage hints.
>

Interpret 1m on 11:39:37, 02/14
-- Loading resources from main.sql
albums      employees  invoices    playlists
artists     genres      media_types tracks
customers   invoice_items playlist_track
Balls to the Wall|2328.599999999996
SQLite version 3.39.4 2022-09-29 15:55:41
Enter ".help" for usage hints.
>
```

```
main.sql x +
main.sql
1 .open chinook.db
2 .tables
3 /*
4  Display the average number of tracks per album. You may have to use a
  subquery within FROM, to count the tracks (grouped by album) before
  attempting to average the results.
5  */
6 SELECT AVG(tracks.AlbumId) as average_tracks FROM tracks
7 GROUP BY tracks.AlbumId;

Console x Shell x +
Interpret 14s on 11:46:19
314.0
315.0
316.0
317.0
318.0
319.0
320.0
321.0
322.0
323.0
324.0
325.0
326.0
327.0
328.0
329.0
330.0
331.0
332.0
333.0
334.0
335.0
336.0
337.0
338.0
339.0
340.0
341.0
342.0
343.0
344.0
345.0
346.0
347.0
SQLite version 3.39.4 2022-09-29 15:55:41
Enter ".help" for usage hints.
>
```