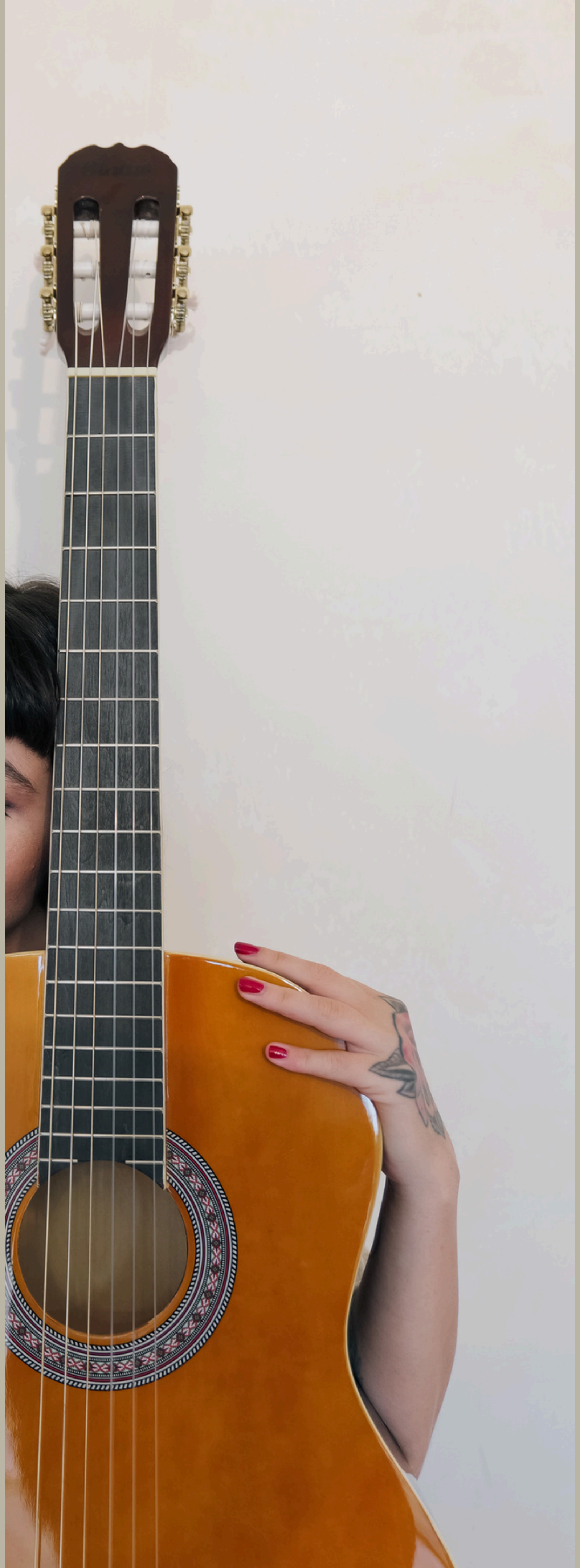
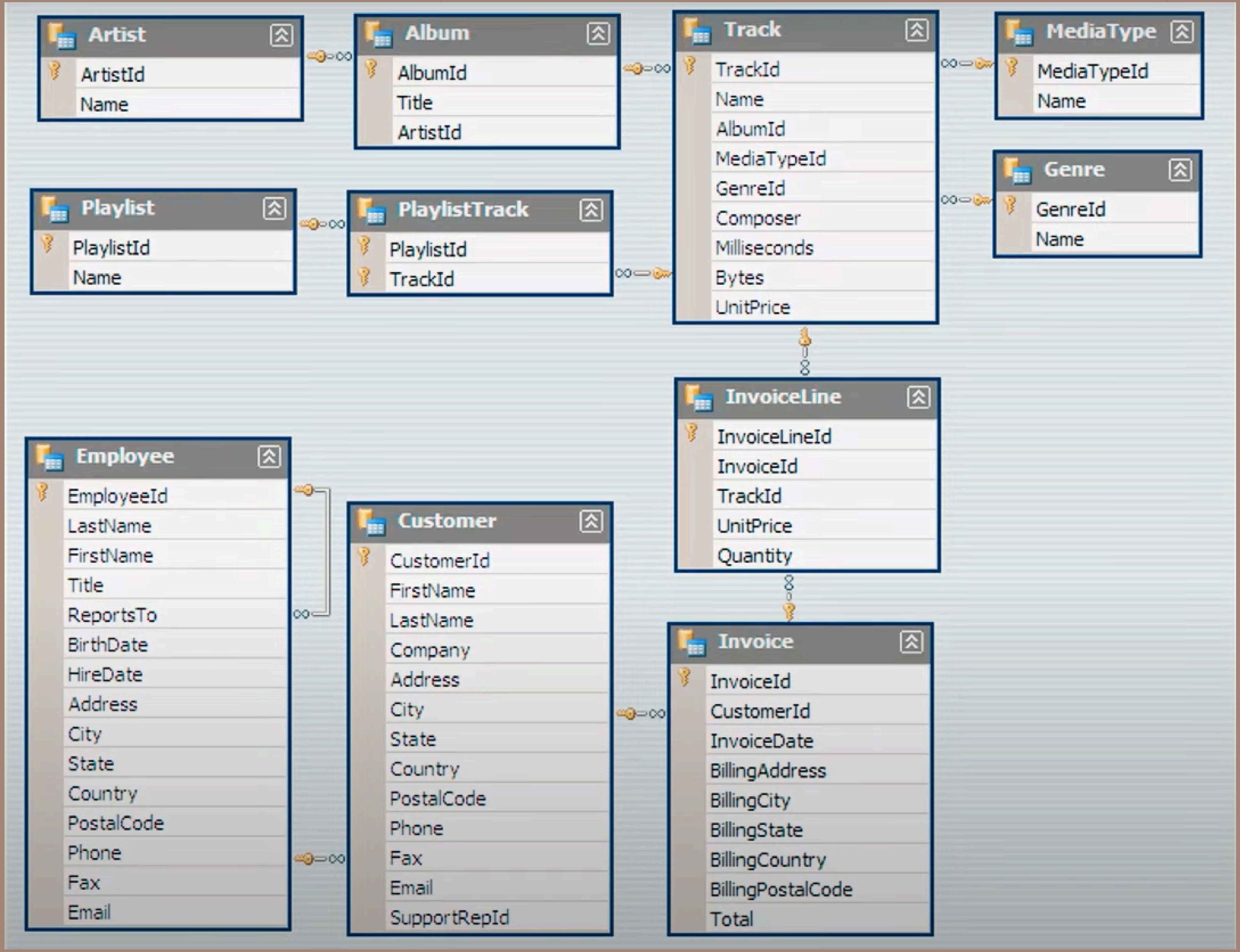


SQL PROJECT ON MUSIC STORE DATASET

INTRODUCTION

This project involves analyzing a music store database, which contains various tables representing different aspects of the store's operations. The database is structured to capture essential information about the music store's inventory, customer interactions, and sales activities. The primary objectives of this project are to analyze the Sales Performance ,Customer Insights and Operational Efficiency. This project aims to leverage the structured data within the music store's database to uncover valuable insights, drive better business decisions, and ultimately improve the store's overall performance and customer satisfaction.

SCHEMA



Artist
- ArtistId
- Name

Album
- AlbumId
- Title
- ArtistId

Track
- TrackId
- Name
- AlbumId
- MediaTypeId
- GenreId
- Composer
- Milliseconds
- Bytes
- UnitPrice

MediaType
- MediaTypeId
- Name

Genre
- GenreId
- Name

Playlist
- PlaylistId
- Name

PlaylistTrack
- PlaylistId
- TrackId

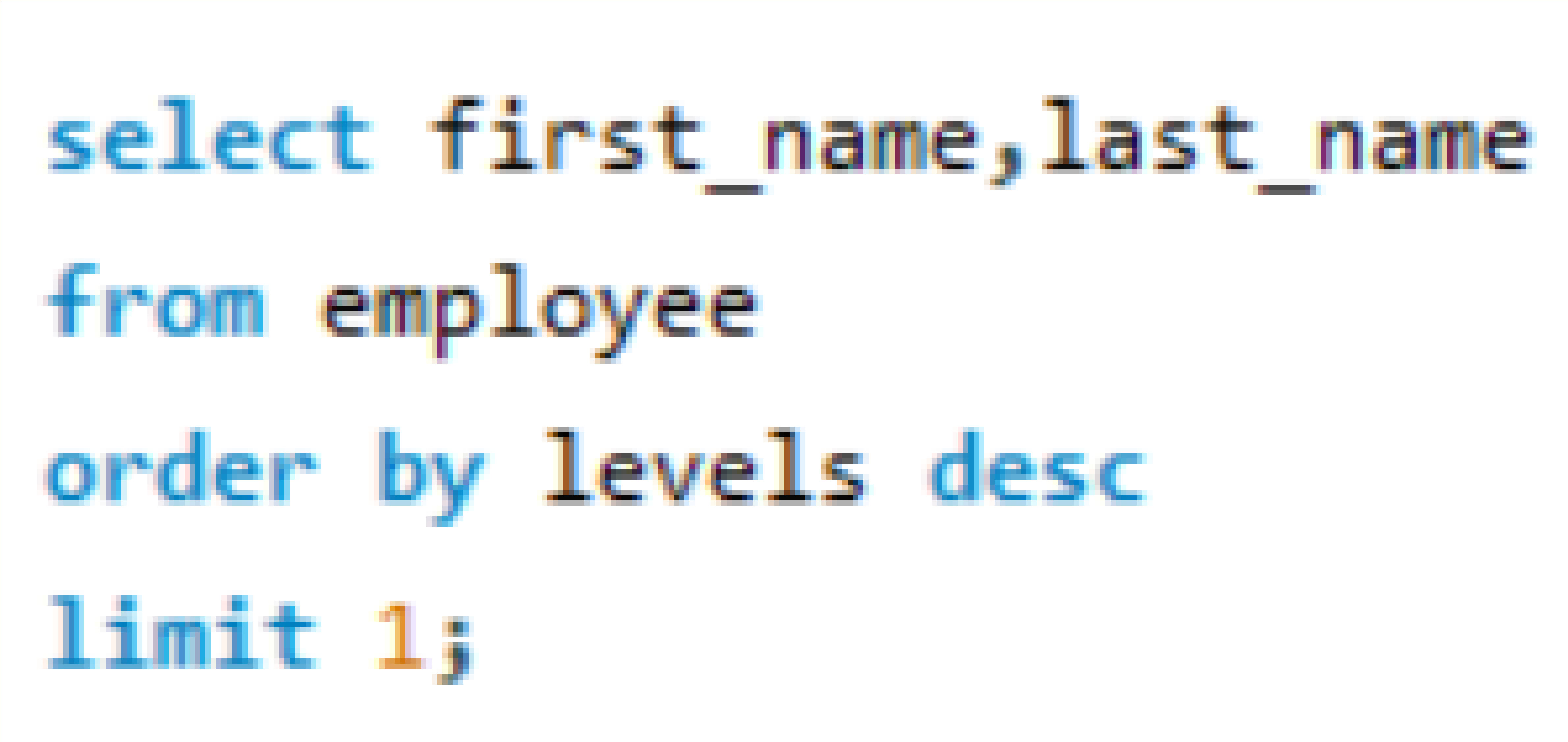
InvoiceLine
- InvoiceLineId
- InvoiceId
- TrackId
- UnitPrice
- Quantity

Employee
- EmployeeId
- LastName
- FirstName
- Title
- ReportsTo
- BirthDate
- HireDate
- Address
- City
- State
- Country
- PostalCode
- Phone
- Fax
- Email

Customer
- CustomerId
- FirstName
- LastName
- Company
- Address
- City
- State
- Country
- PostalCode
- Phone
- Fax
- Email
- SupportRepId

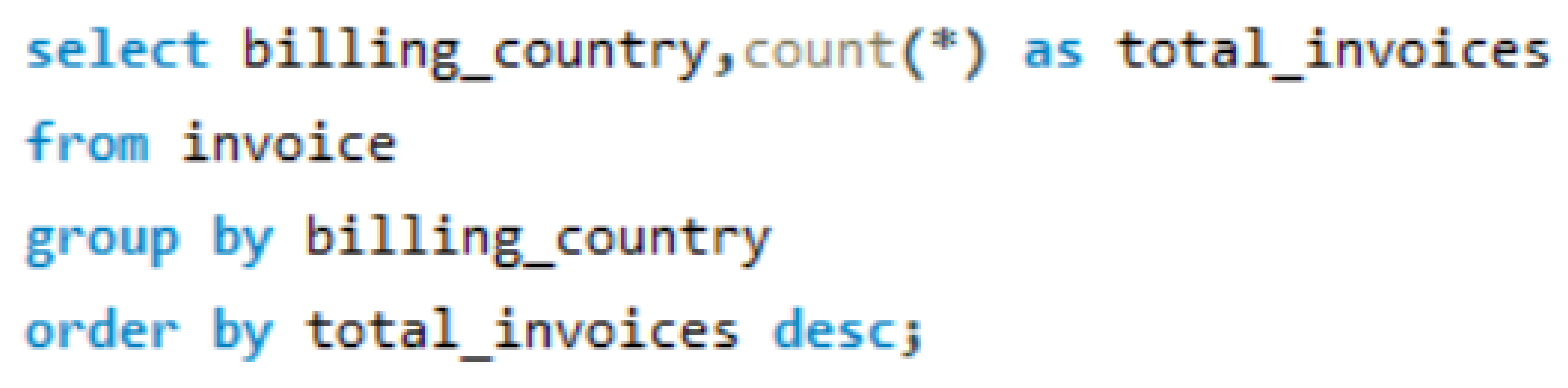
Invoice
- InvoiceId
- CustomerId
- InvoiceDate
- BillingAddress
- BillingCity
- BillingState
- BillingCountry
- BillingPostalCode
- Total

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



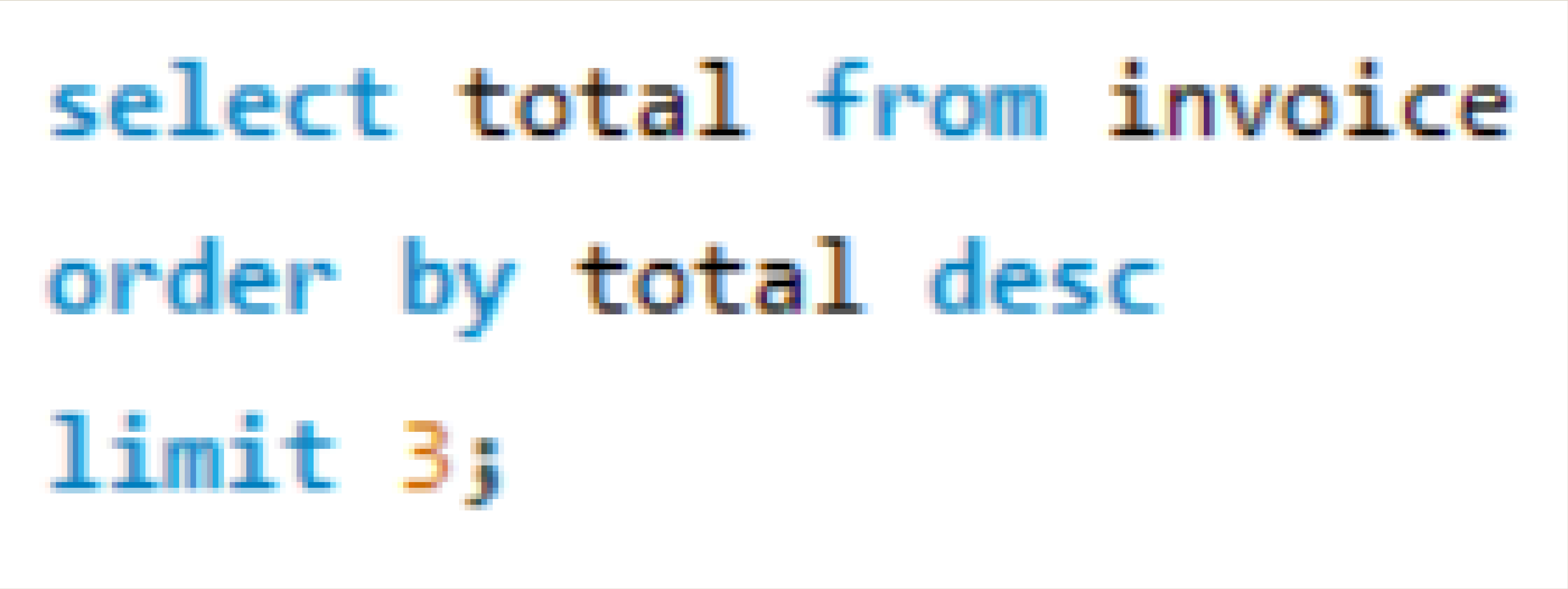
```
select first_name,last_name
from employee
order by levels desc
limit 1;
```

2.Which countries have the most Invoices?



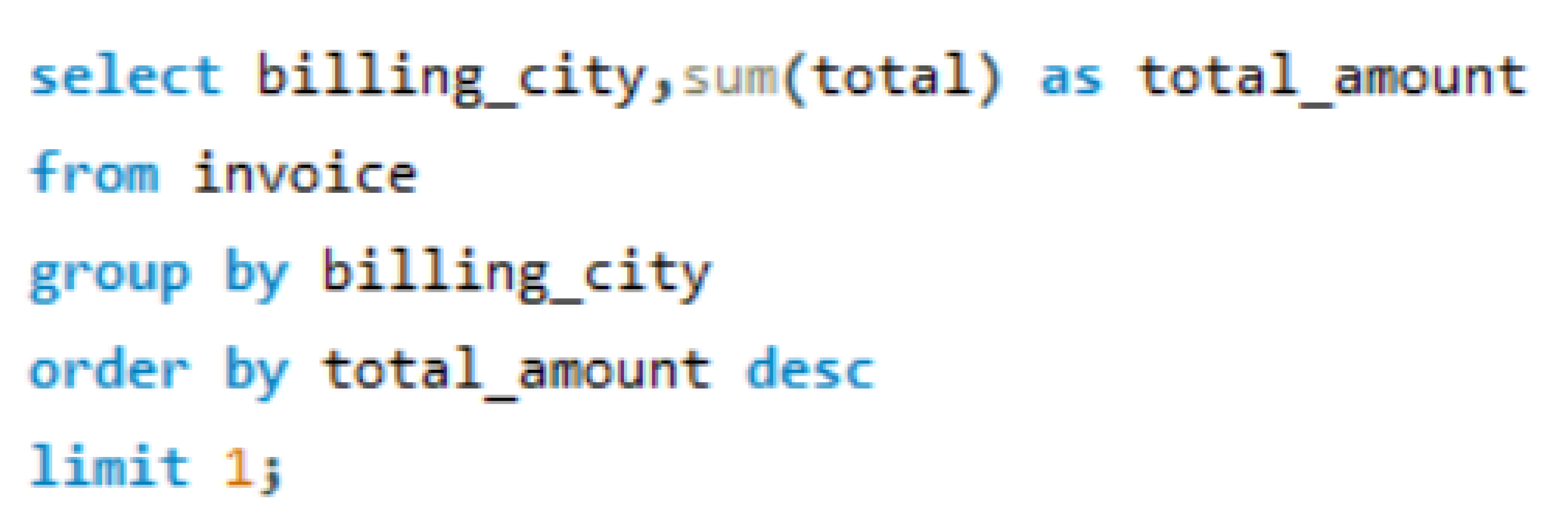
```
select billing_country,count(*) as total_invoices
from invoice
group by billing_country
order by total_invoices desc;
```

3.What are top 3 values of total invoice



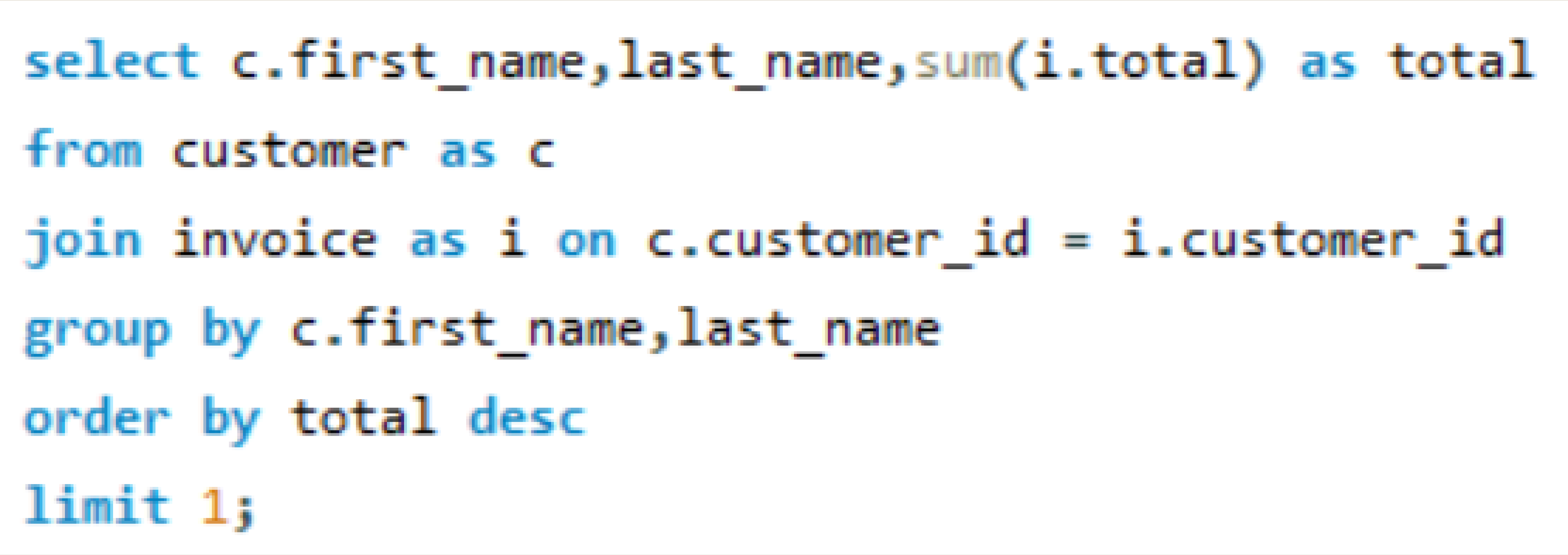
```
select total from invoice
order by total desc
limit 3;
```

4. 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.



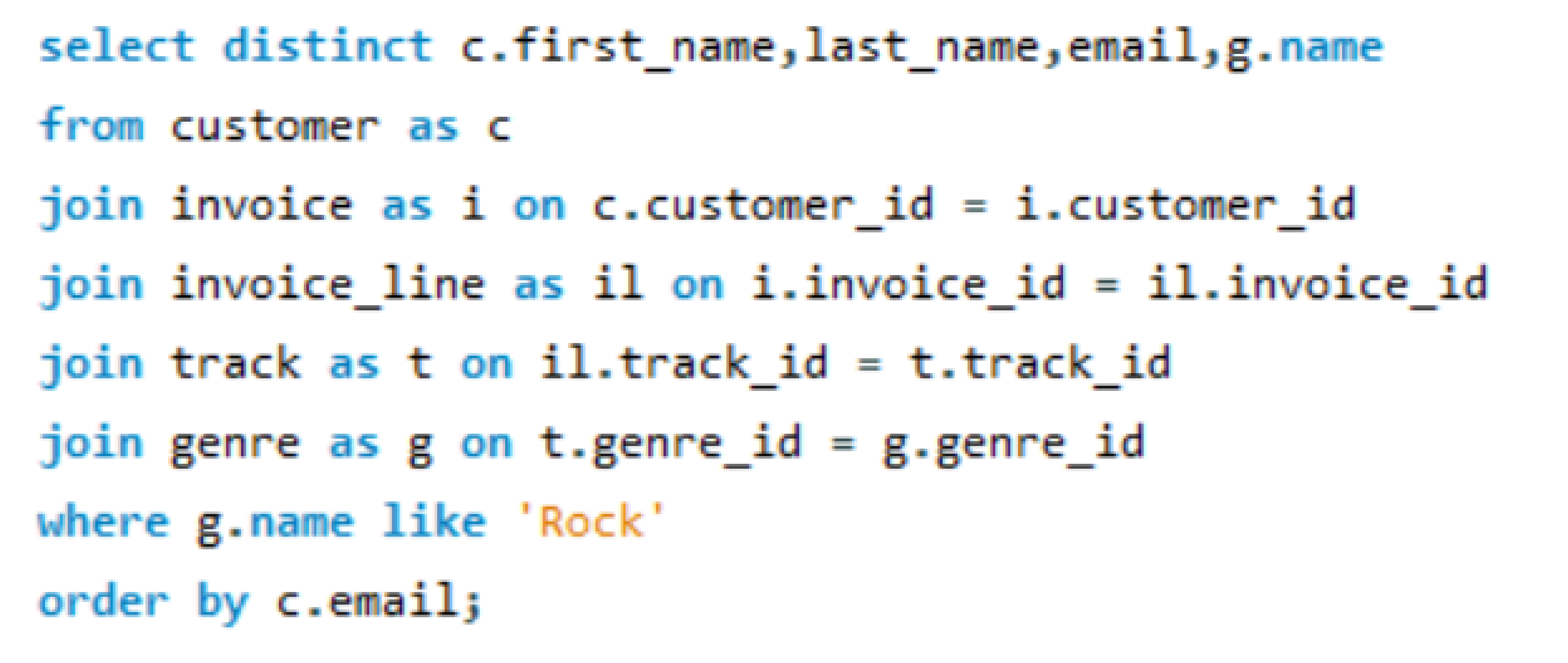
```
select billing_city,sum(total) as total_amount
from invoice
group by billing_city
order by total_amount desc
limit 1;
```

5.Write a query that returns the person who has spent the most money.



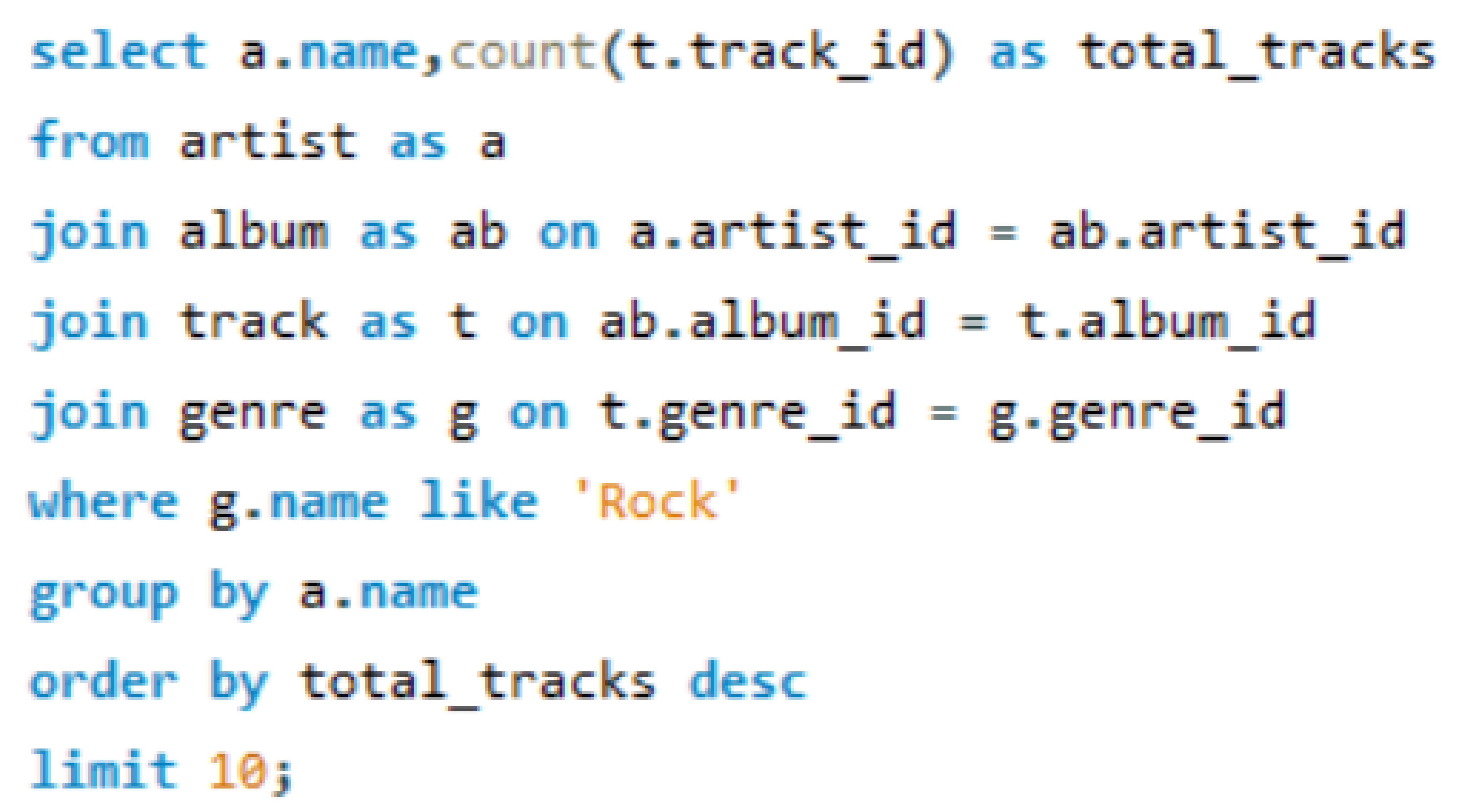
```
select c.first_name,last_name,sum(i.total) as total
from customer as c
join invoice as i on c.customer_id = i.customer_id
group by c.first_name,last_name
order by total desc
limit 1;
```

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.



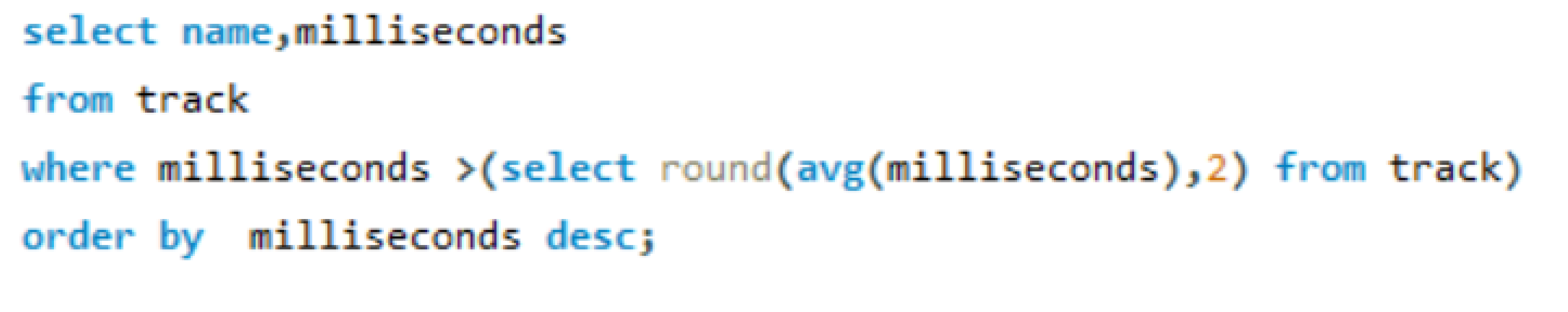
```sql
select distinct c.first_name,last_name,email,g.name
from customer as c
join invoice as i on c.customer_id = i.customer_id
join invoice_line as il on i.invoice_id = il.invoice_id
join track as t on il.track_id = t.track_id
join genre as g on t.genre_id = g.genre_id
where g.name like 'Rock'
order by c.email;
```

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.



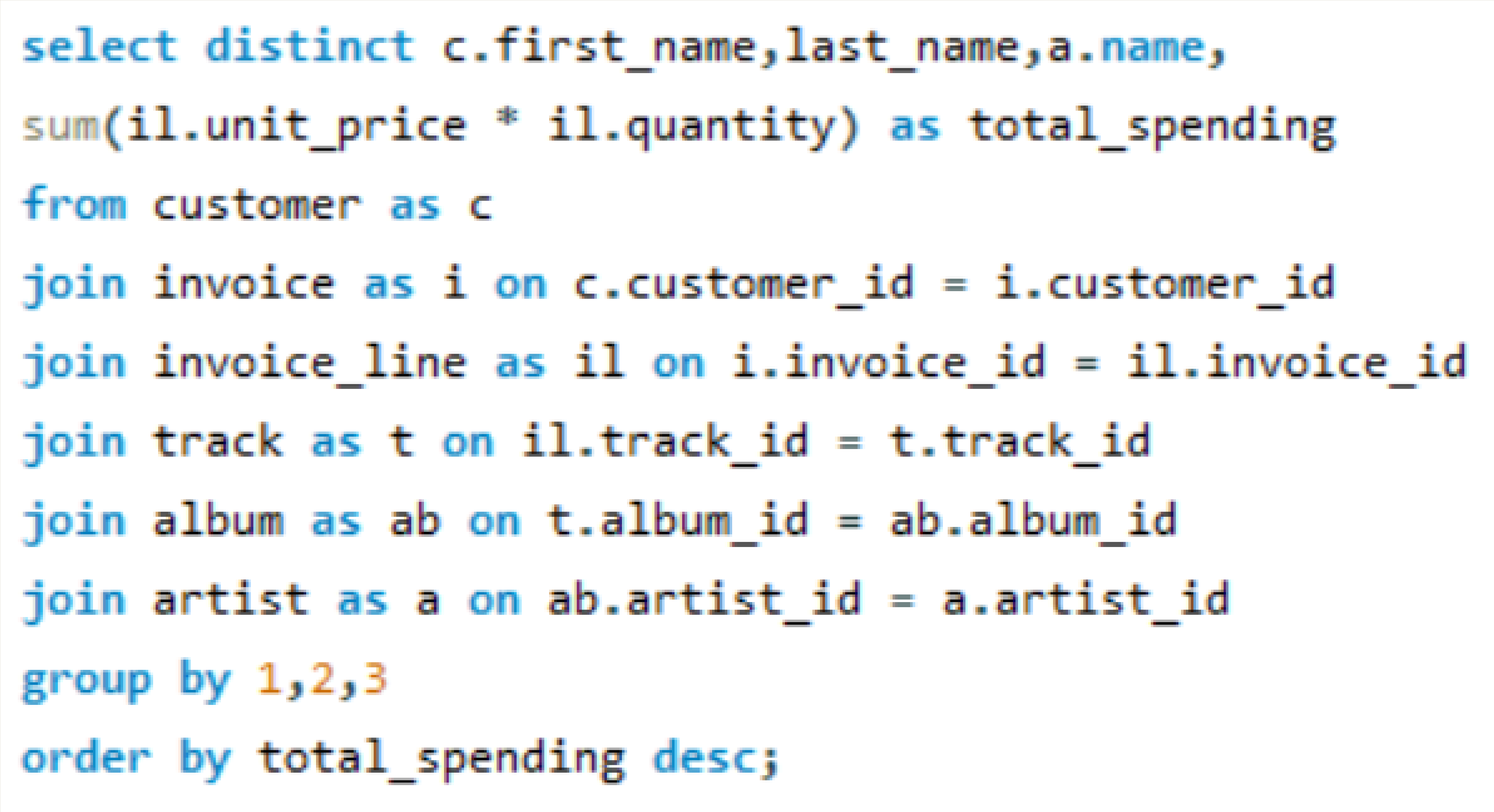
```
select a.name,count(t.track_id) as total_tracks
from artist as a
join album as ab on a.artist_id = ab.artist_id
join track as t on ab.album_id = t.album_id
join genre as g on t.genre_id = g.genre_id
where g.name like 'Rock'
group by a.name
order by total_tracks desc
limit 10;
```

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.



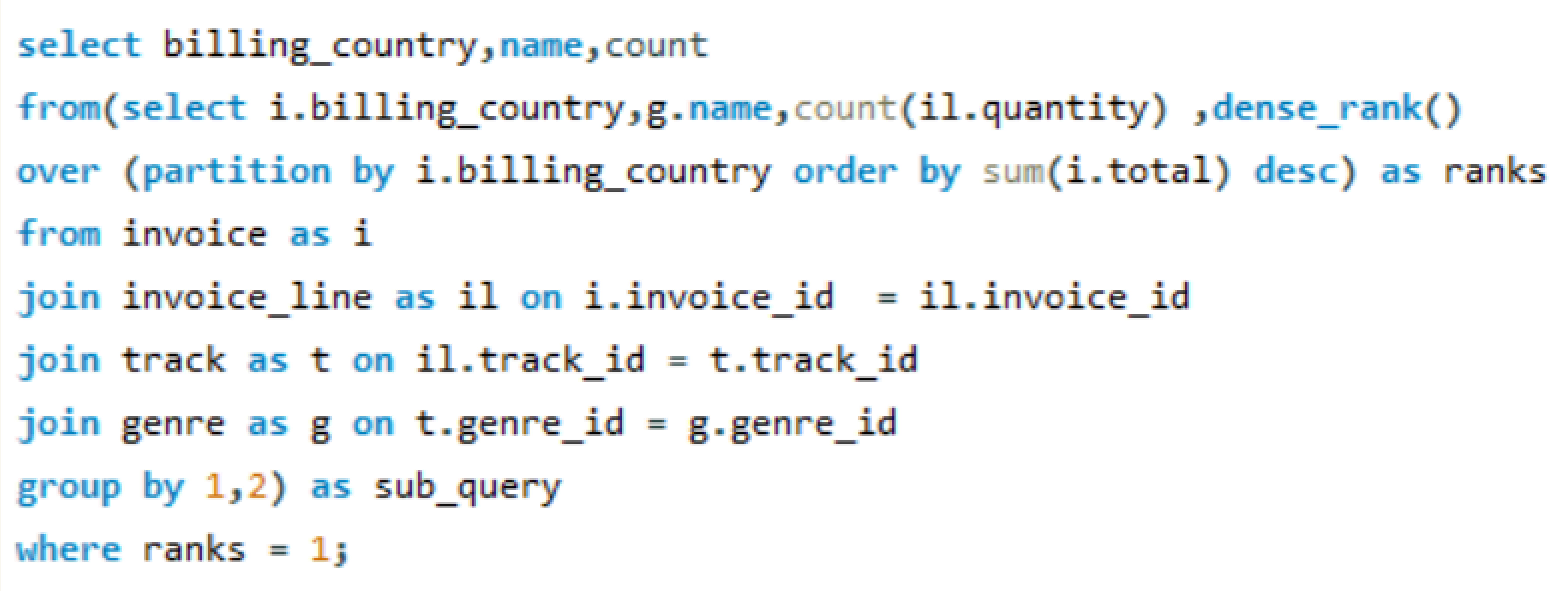
```
select name,milliseconds
from track
where milliseconds >(select round(avg(milliseconds),2) from track)
order by  milliseconds desc;
```

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



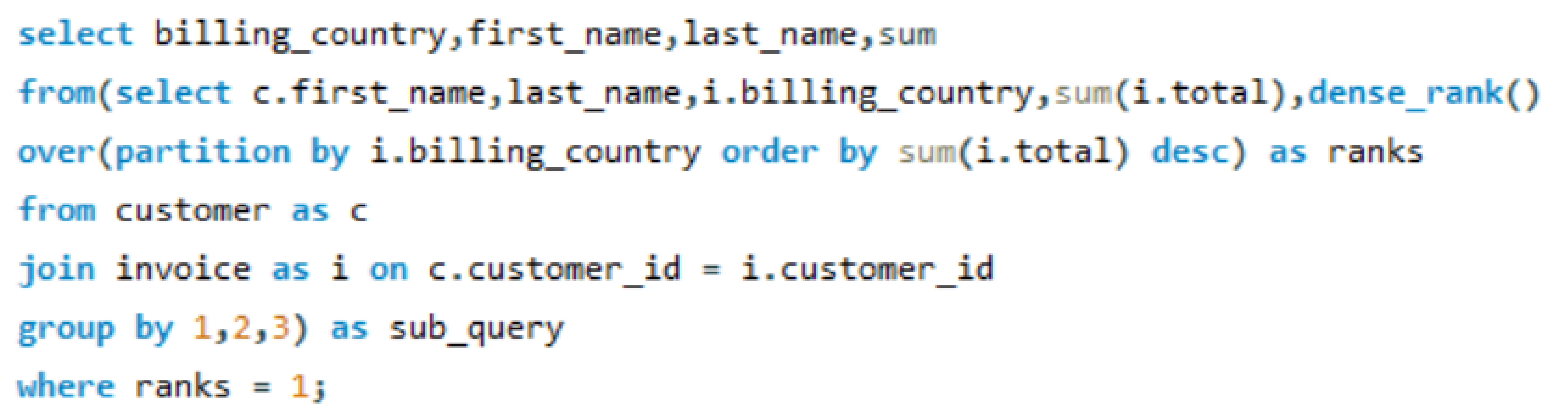
```
select distinct c.first_name,last_name,a.name,
sum(il.unit_price * il.quantity) as total_spending
from customer as c
join invoice as i on c.customer_id = i.customer_id
join invoice_line as il on i.invoice_id = il.invoice_id
join track as t on il.track_id = t.track_id
join album as ab on t.album_id = ab.album_id
join artist as a on ab.artist_id = a.artist_id
group by 1,2,3
order by total_spending desc;
```

10.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 billing_country,name,count
from(select i.billing_country,g.name,count(il.quantity) ,dense_rank()
over (partition by i.billing_country order by sum(i.total) desc) as ranks
from invoice as i
join invoice_line as il on i.invoice_id  = il.invoice_id
join track as t on il.track_id = t.track_id
join genre as g on t.genre_id = g.genre_id
group by 1,2) as sub_query
where ranks = 1;
```

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.



```
select billing_country,first_name,last_name,sum
from(select c.first_name,last_name,i.billing_country,sum(i.total),dense_rank()
over(partition by i.billing_country order by sum(i.total) desc) as ranks
from customer as c
join invoice as i on c.customer_id = i.customer_id
group by 1,2,3) as sub_query
where ranks = 1;
```