**ACTIVIDAD 2**

**Tipo actividad: Taller de consultas SQL**

**Taller de consultas SQL**

Utilizando el editor de consultas (queries) que se puede ver en la figura 7 orientar a los estudiantes para que, escriban consultas SQL que permitan responder las siguientes preguntas:

1. Ejecutar y revisar el resultado de las siguientes consultas:

SELECT \* FROM Invoice;

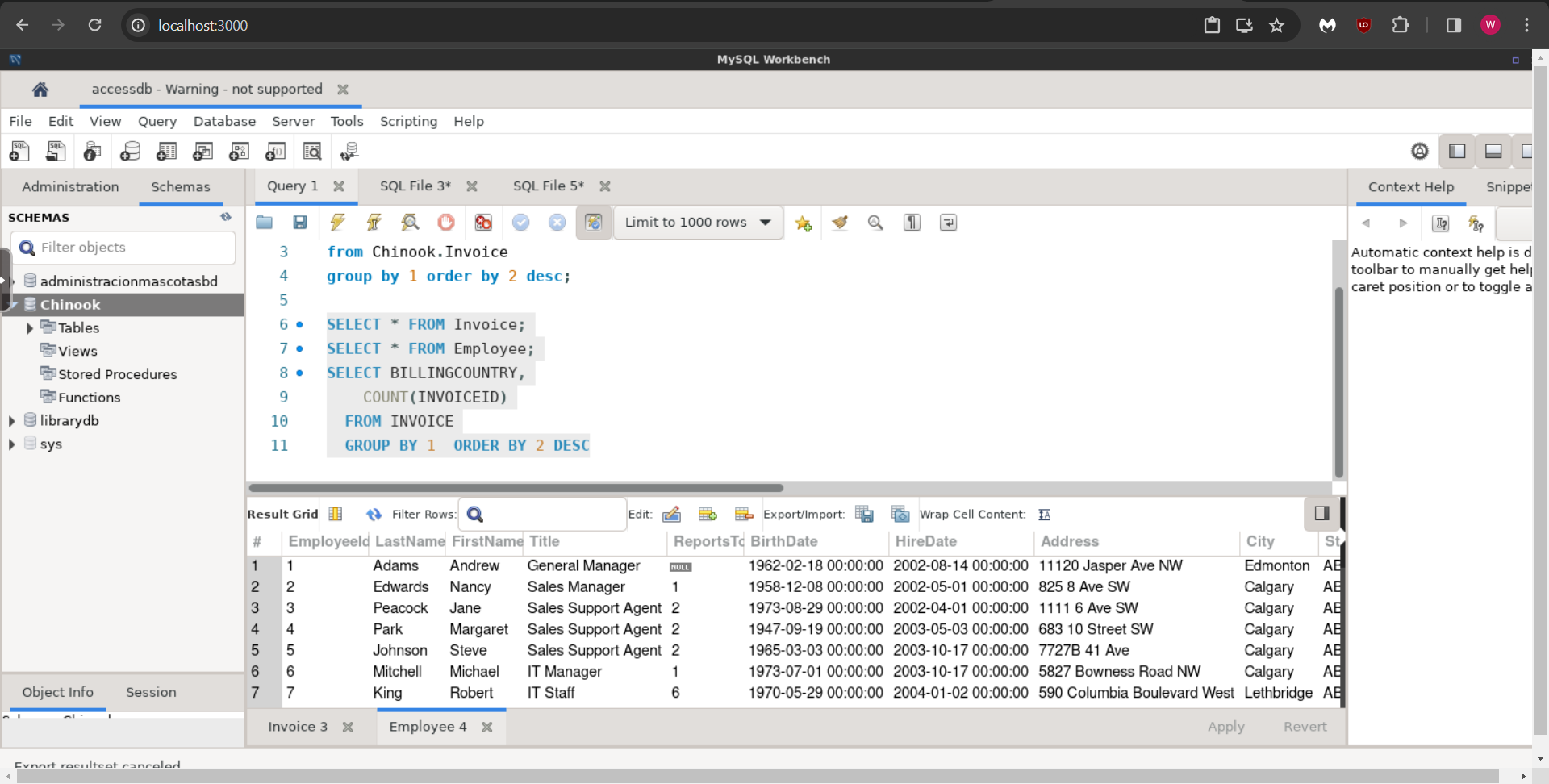
SELECT \* FROM Employee;

SELECT BILLINGCOUNTRY,

COUNT(INVOICEID)

FROM INVOICE

GROUP BY 1 ORDER BY 2 DESC



1. ¿Qué país tiene más facturas?

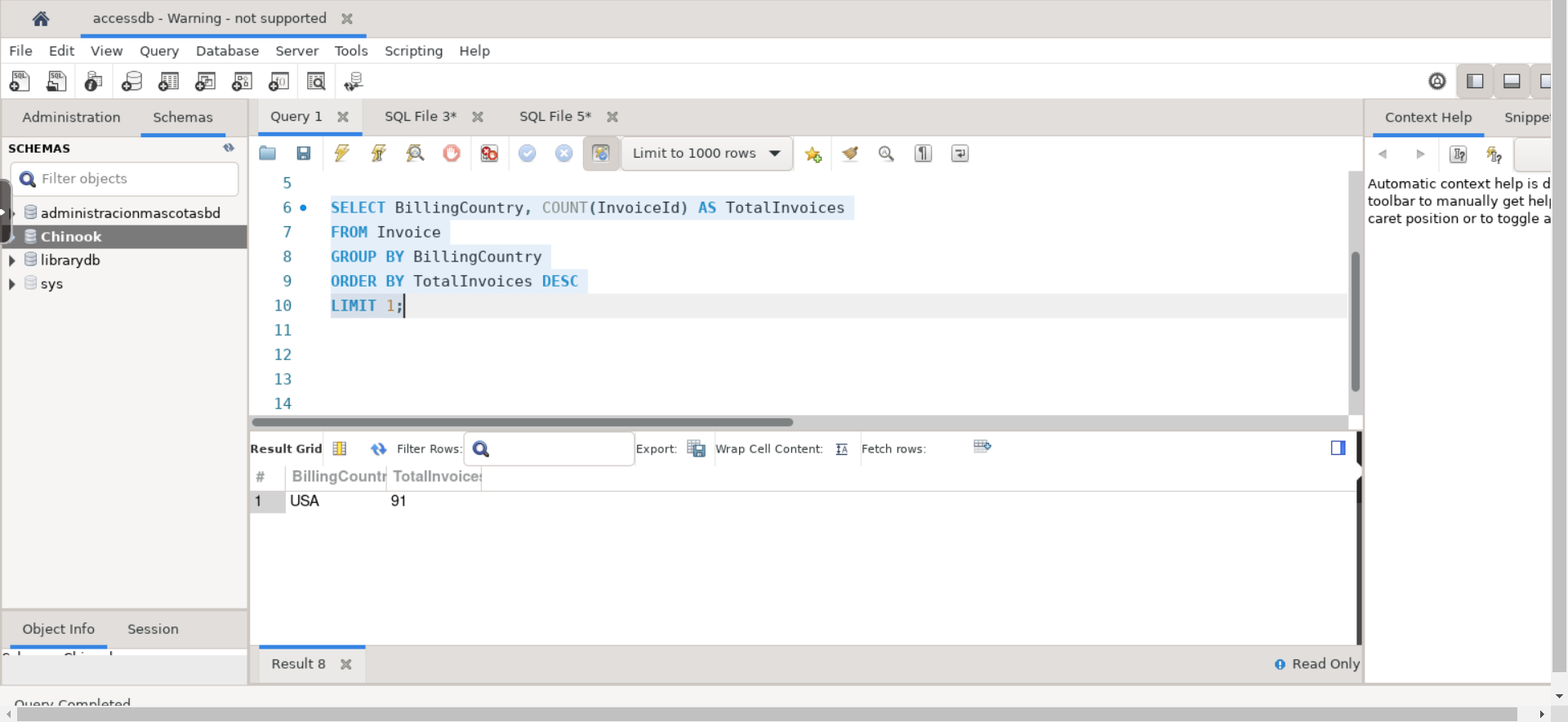
SELECT BillingCountry, COUNT(InvoiceId) AS TotalInvoices

FROM Invoice

GROUP BY BillingCountry

ORDER BY TotalInvoices DESC

LIMIT 1;



USA con 91 facturas

1. se quiere promocionar un nuevo festival musical, para ello es necesario saber en qué ciudad hay más facturas (invoices) . determinar en qué ciudad la suma de facturas es la mayor.

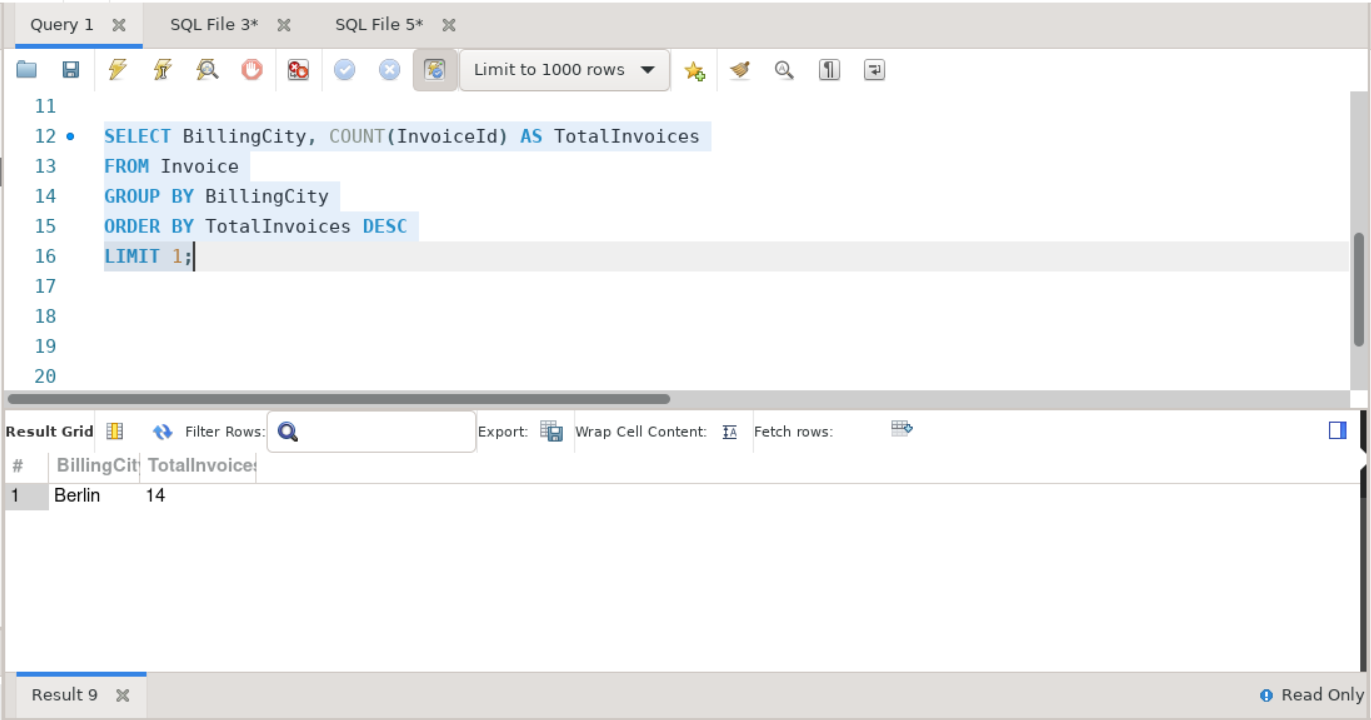
SELECT BillingCity, COUNT(InvoiceId) AS TotalInvoices

FROM Invoice

GROUP BY BillingCity

ORDER BY TotalInvoices DESC

LIMIT 1;



Berlín es la ciudad con más facturas 14

1. Determinar quién es el mejor cliente (el que más ha gastado en el sistema)

SELECT c.FirstName, c.LastName, SUM(i.Total) AS TotalSpent

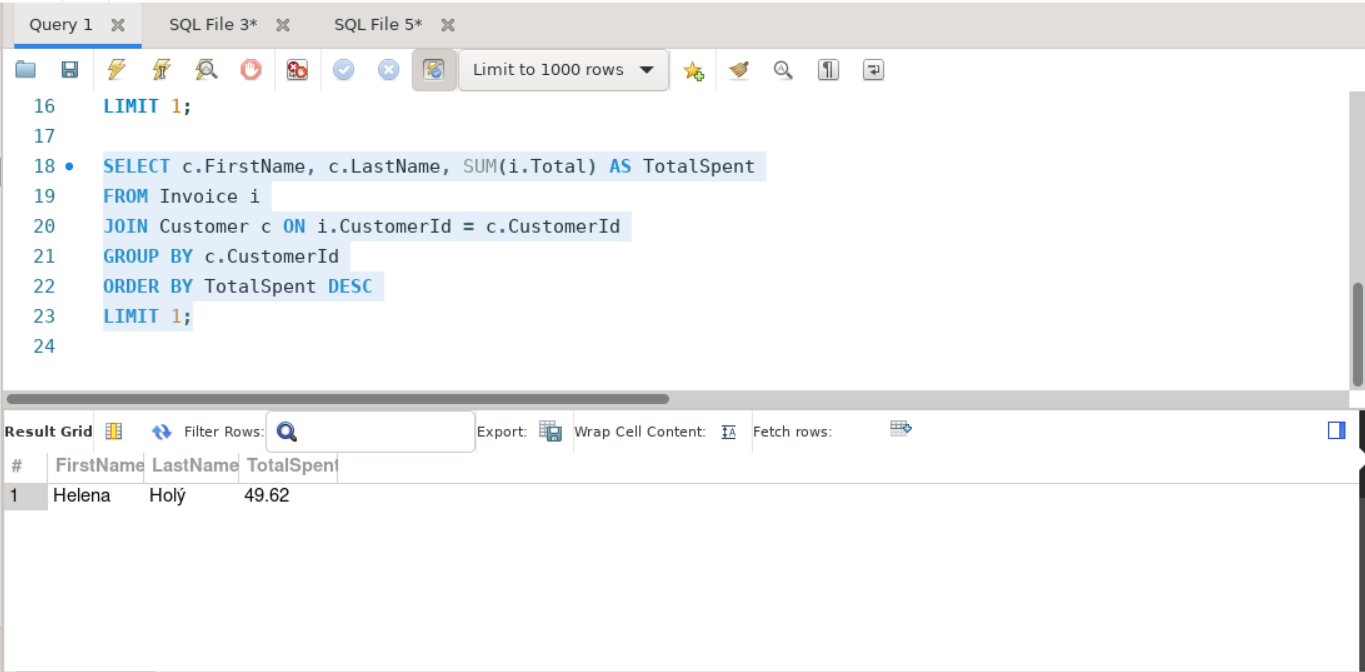
FROM Invoice i

JOIN Customer c ON i.CustomerId = c.CustomerId

GROUP BY c.CustomerId

ORDER BY TotalSpent DESC

LIMIT 1;



El cliente que más ha gastado es Helena Holý 48.62

1. Obtener una tabla con el correo, nombre y apellido de todos las personas que escuchan Rock. Retornar la lista por orden alfabético

SELECT c.Email, c.FirstName, c.LastName

FROM Customer c

JOIN Invoice i ON c.CustomerId = i.CustomerId

JOIN InvoiceLine il ON i.InvoiceId = il.InvoiceId

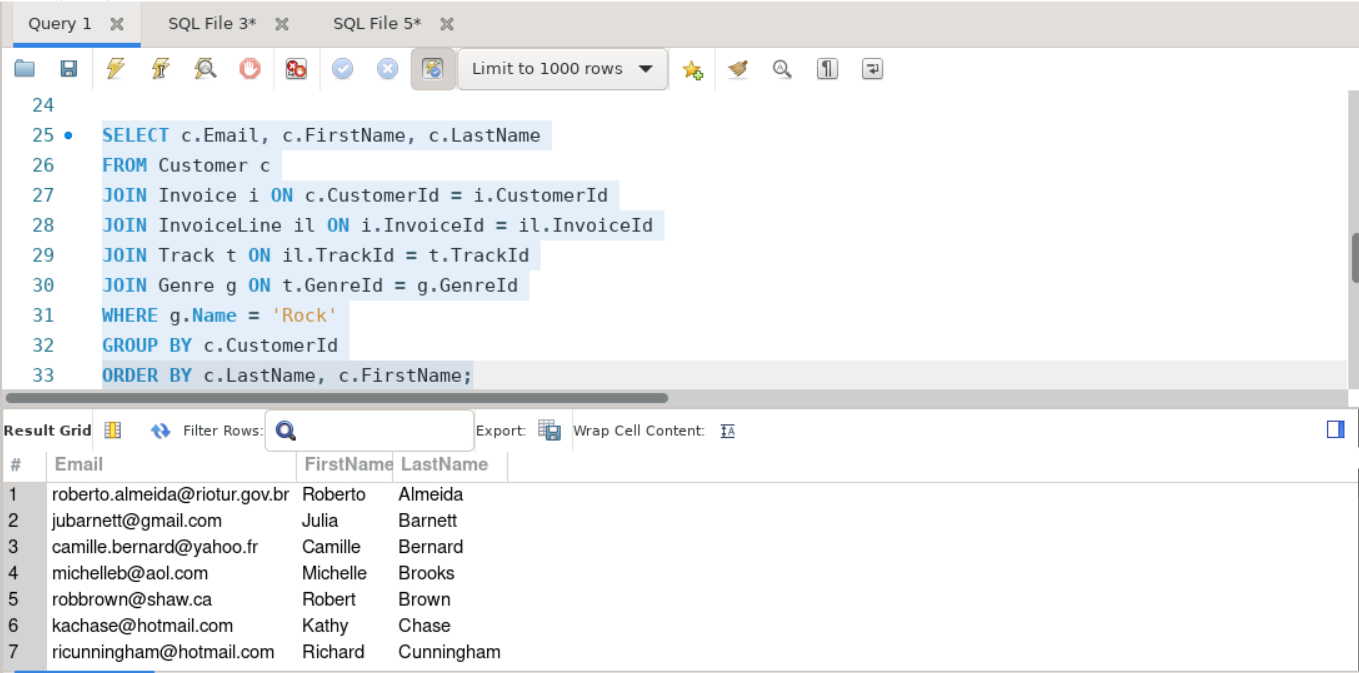
JOIN Track t ON il.TrackId = t.TrackId

JOIN Genre g ON t.GenreId = g.GenreId

WHERE g.Name = 'Rock'

GROUP BY c.CustomerId

ORDER BY c.LastName, c.FirstName;



1. Sacar una lista con todos los artistas que generan música rock

SELECT DISTINCT ar.Name AS ArtistName

FROM Artist ar

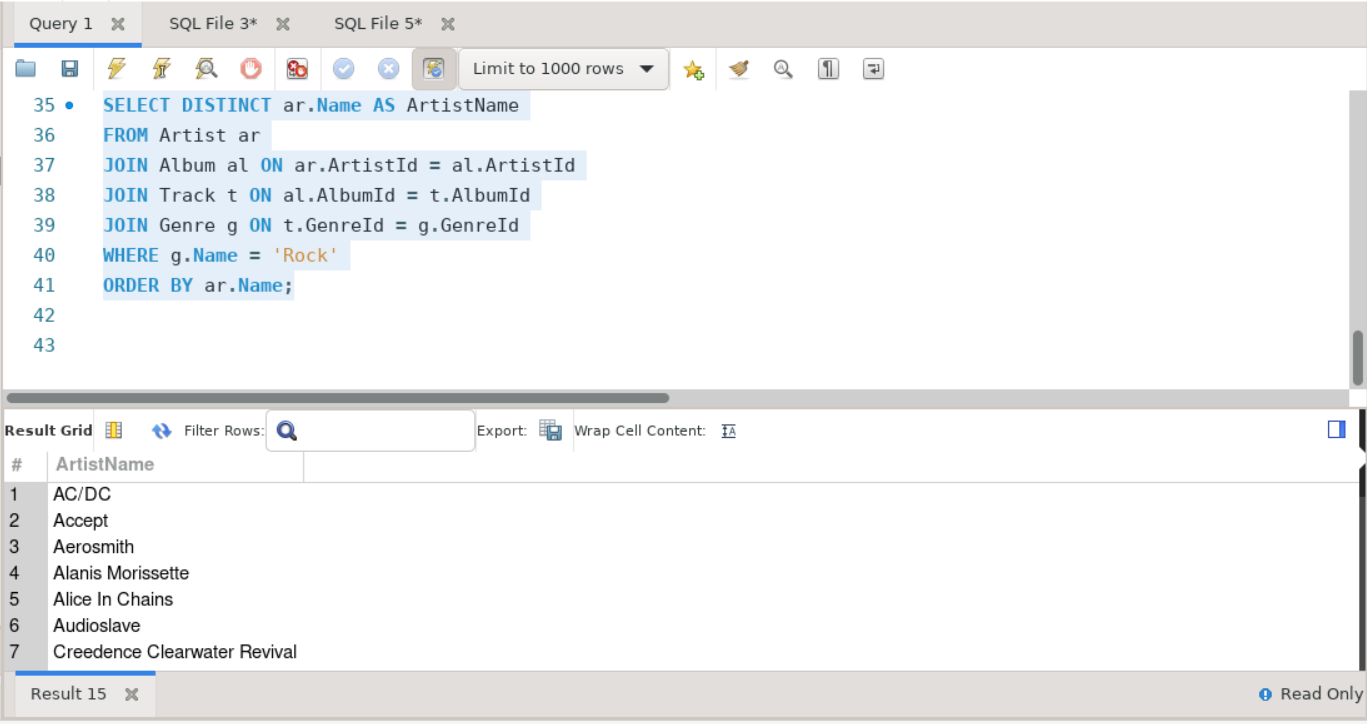
JOIN Album al ON ar.ArtistId = al.ArtistId

JOIN Track t ON al.AlbumId = t.AlbumId

JOIN Genre g ON t.GenreId = g.GenreId

WHERE g.Name = 'Rock'

ORDER BY ar.Name;



1. Encontrar cual es el artista que más ha ganado de acuerdo al campo invoiceLines.

SELECT ar.Name AS ArtistName, SUM(il.UnitPrice \* il.Quantity) AS TotalEarned

FROM Artist ar

JOIN Album al ON ar.ArtistId = al.ArtistId

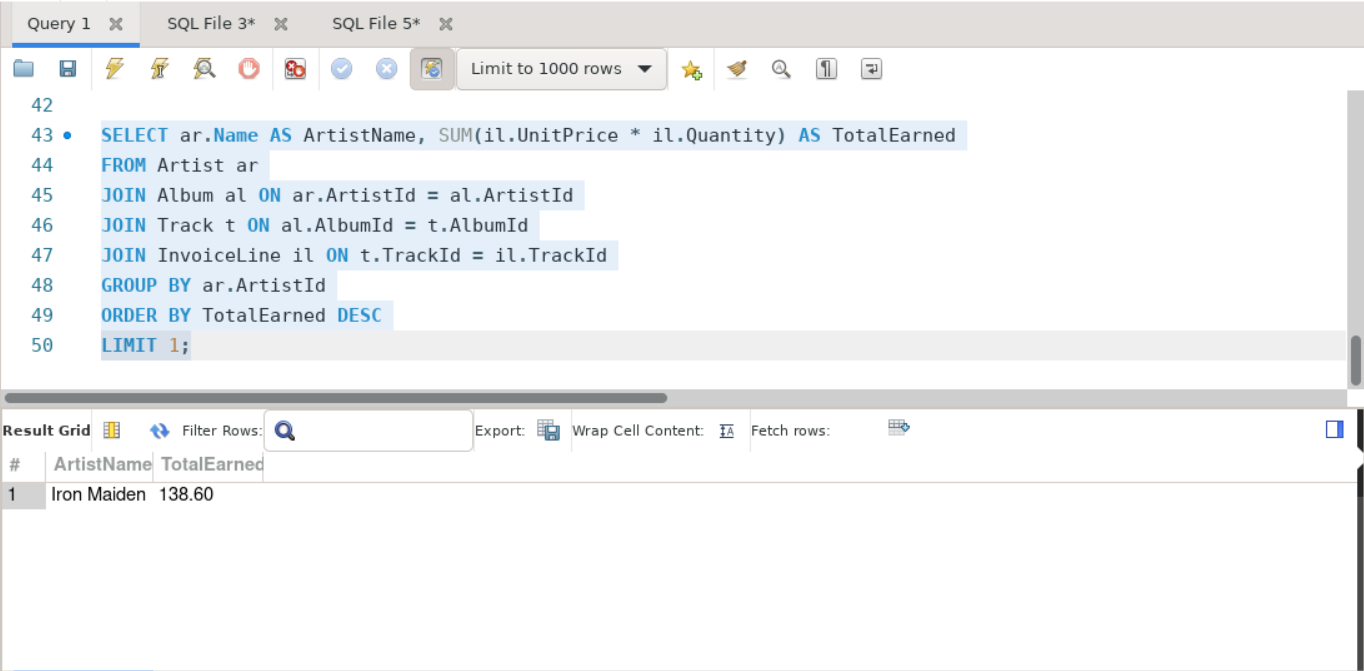
JOIN Track t ON al.AlbumId = t.AlbumId

JOIN InvoiceLine il ON t.TrackId = il.TrackId

GROUP BY ar.ArtistId

ORDER BY TotalEarned DESC

LIMIT 1;



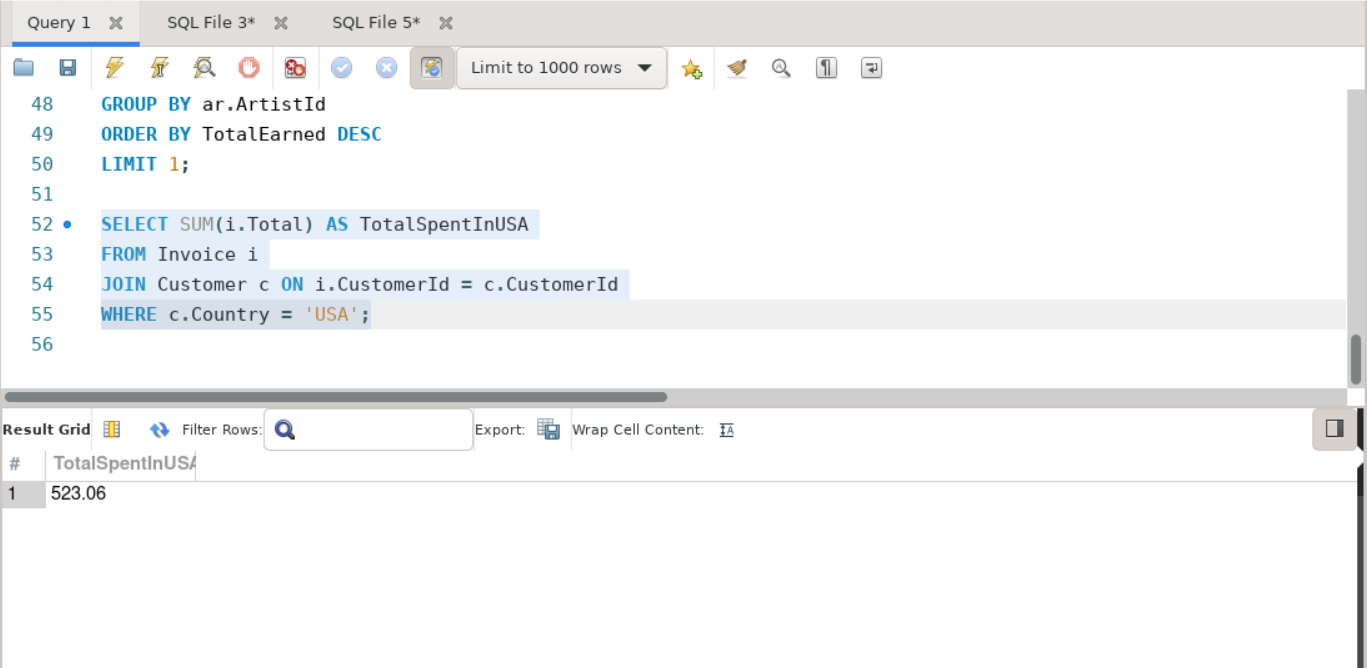
1. Encontrar cuánto gastaron en total en estados unidos en compras

SELECT SUM(i.Total) AS TotalSpentInUSA

FROM Invoice i

JOIN Customer c ON i.CustomerId = c.CustomerId

WHERE c.Country = 'USA';



1. Encontrar cuánto gastaron los usuarios por género.

SELECT g.Name AS Genre, SUM(i.Total) AS TotalSpent

FROM Invoice i

JOIN Customer c ON i.CustomerId = c.CustomerId

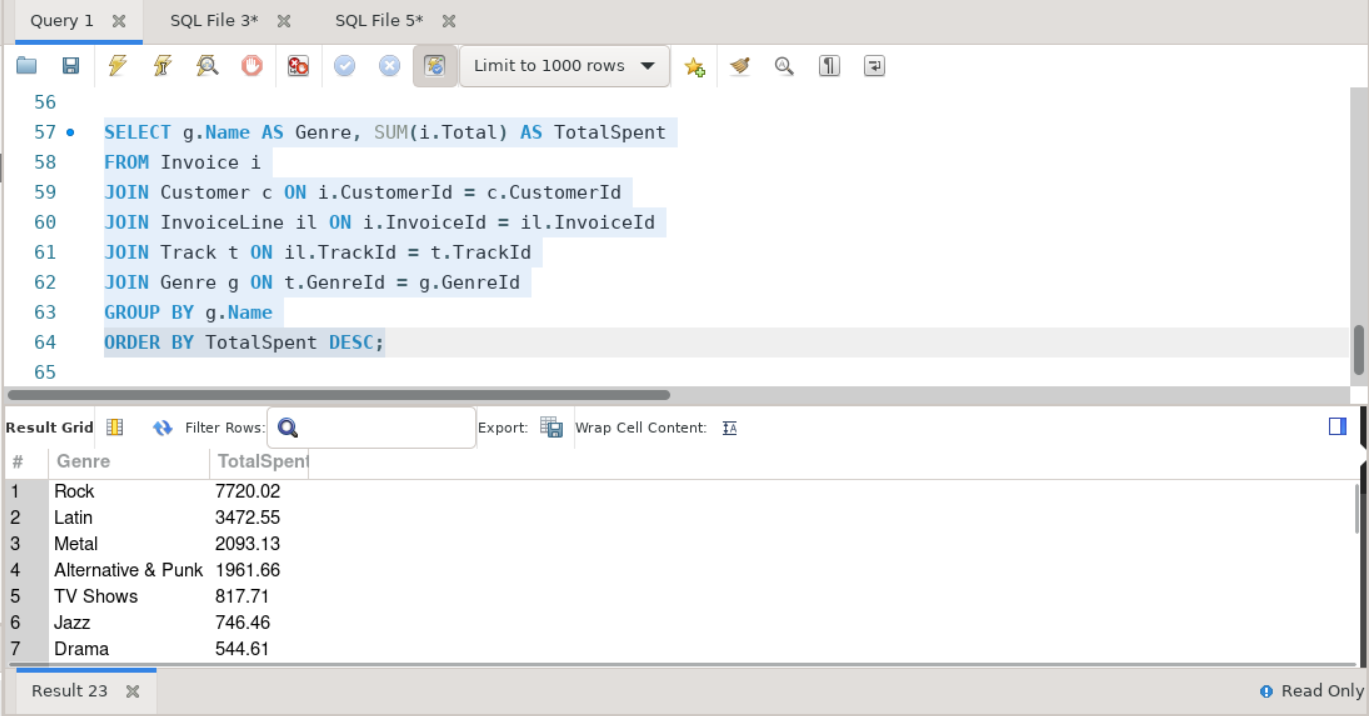
JOIN InvoiceLine il ON i.InvoiceId = il.InvoiceId

JOIN Track t ON il.TrackId = t.TrackId

JOIN Genre g ON t.GenreId = g.GenreId

GROUP BY g.Name

ORDER BY TotalSpent DESC;



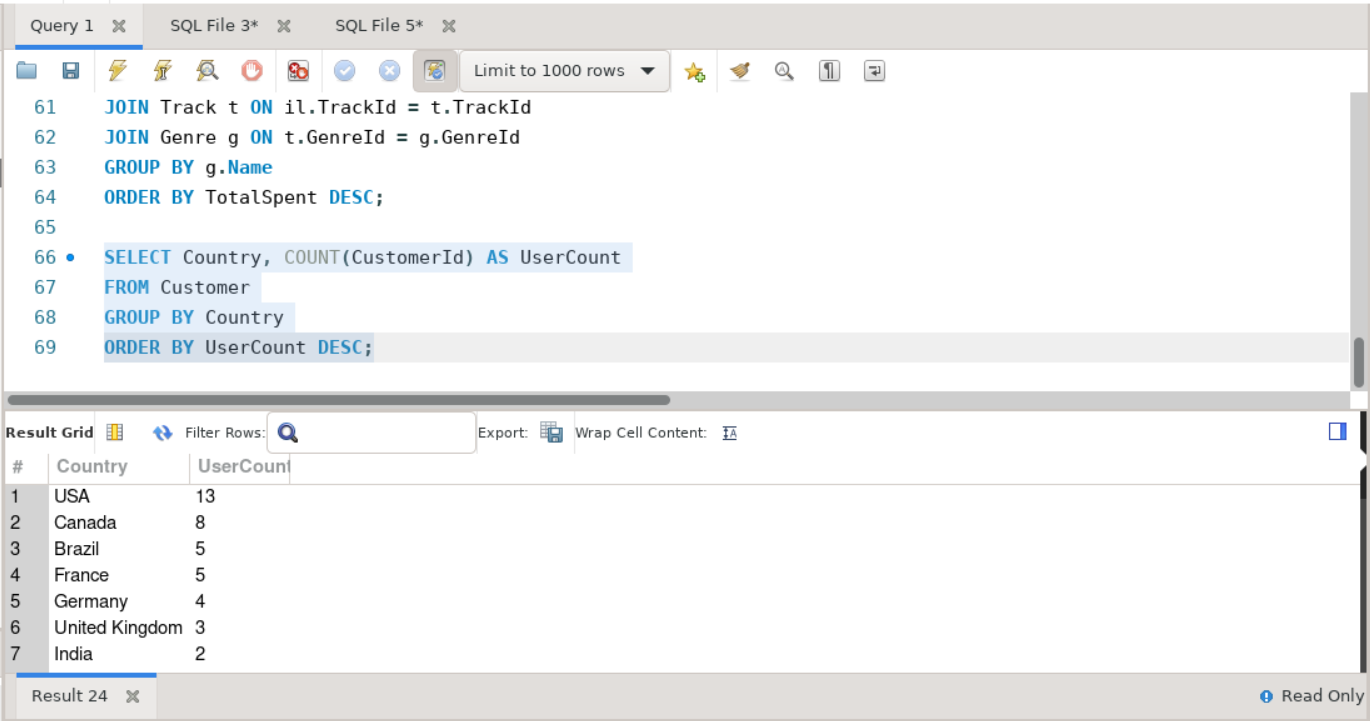
1. Generar una tabla con el conteo de usuarios por cada país.

SELECT Country, COUNT(CustomerId) AS UserCount

FROM Customer

GROUP BY Country

ORDER BY UserCount DESC;



1. Encontrar cuántas canciones hay por cada género.

SELECT g.Name AS Genre, COUNT(t.TrackId) AS SongCount

FROM Genre g

JOIN Track t ON g.GenreId = t.GenreId

GROUP BY g.Name

ORDER BY SongCount DESC;

