# **Chinook Database project**

**Explore the Music Store Data Using SQL!**

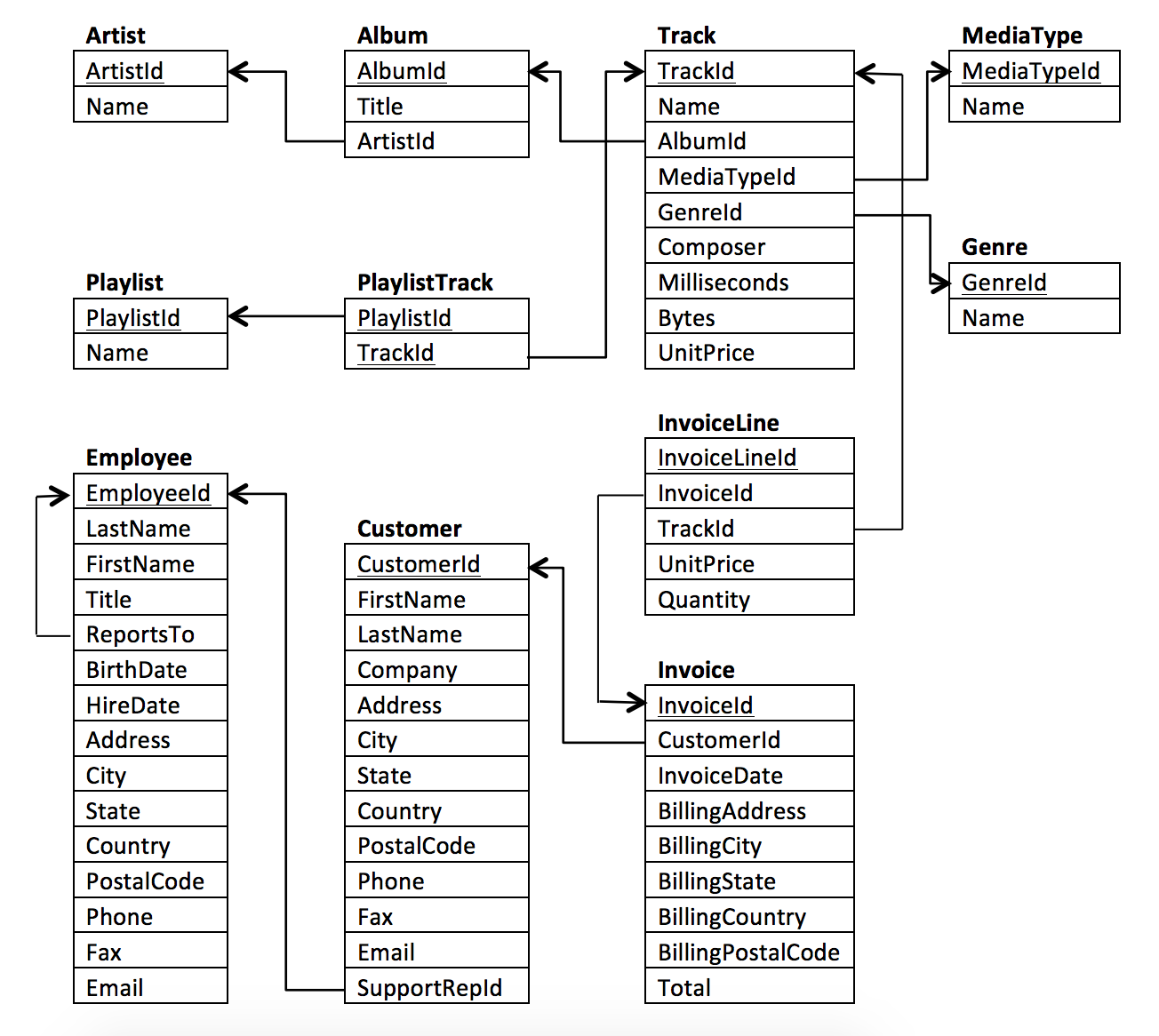
**Project Overview**  
In this project, I executed queries on the Chinook database. Chinook contains information about a music store.

**Objectives:**

* Utilize fundamental SQL functionalities
* Answer questions using accurate queries
* Discover interesting insights using JOINs and subqueries
* Transfer data from one platform to another
* Enjoy the process

**My Role**  
In this project, I analyzed the Chinook data to understand their store's products, customers, employees, and account information.

**Database**  
The schema of the Chinook database is provided below.



**Part 1**

#### **Question 1: Which countries have the most invoices?**

My query:

SELECT BillingCountry,

COUNT (BillingCountry) as Invoices

FROM da-nfactorial.chinook.invoice

GROUP BY(BillingCountry)

ORDER BY Invoices DESC

#### **Question 2: Which city has generated the most income?**

My query:

SELECT BillingCity,

SUM (Total) as TotalAmount

FROM da-nfactorial.chinook.invoice

GROUP BY(BillingCity)

ORDER BY TotalAmount DESC

#### **Question 3:Who is the best client? Which client spent the most money?**

My query:

SELECT i.CustomerID as CustomerID,

c.FirstName,

c.LastName,

COUNT(i.InvoiceID) as Invoices,

SUM (Total) as TotalAmount

FROM da-nfactorial.chinook.invoice i

JOIN da-nfactorial.chinook.invoiceline il

ON i.InvoiceID = il.InvoiceId

LEFT JOIN da-nfactorial.chinook.customer c

ON i.CustomerId = c.CustomerId

GROUP BY

i.CustomerID,

c.FirstName,

c.LastName

ORDER BY TotalAmount DESC

#### **Question 4:Identify every clent that listens to rock music**

My query:

SELECT c.Email as Email,

c.FirstName as FtN,

c.LastName as LtN,

g.Name as Genre,

FROM da-nfactorial.chinook.customer c

LEFT JOIN da-nfactorial.chinook.invoice i

ON c.CustomerID = i.CustomerId

LEFT JOIN da-nfactorial.chinook.invoiceline il

ON i.InvoiceId = il.InvoiceId

LEFT JOIN da-nfactorial.chinook.track t

ON il.TrackId = t.TrackId

LEFT JOIN da-nfactorial.chinook.genre g

ON t.GenreId = g.GenreId

WHERE g.GenreID = 1

GROUP BY

Email,

FtN,

LtN,

Genre

ORDER BY Email ASC

#### **QUestion 5: Identify top-10 artists or groups who published the most rock music.**

My query:

SELECT ar.Name as Name,

COUNT(t.TrackID) as Tracks,

g.Name as Genre,

FROM da-nfactorial.chinook.track t

LEFT JOIN da-nfactorial.chinook.album al

ON t.AlbumId = al.AlbumId

LEFT JOIN da-nfactorial.chinook.genre g

ON t.GenreId = g.GenreId

LEFT JOIN da-nfactorial.chinook.artist ar

ON al.ArtistId = ar.ArtistId

WHERE g.GenreID = 1

GROUP BY

Name,

Genre

ORDER BY Tracks DESC

#### **Question 6: Which artist has earned the most money? Which client has spent the most on this artist’s music?**

My query:

SELECT ar.Name as Name,

SUM(il.quantity\*il.UnitPrice) as TotalIncome,

FROM da-nfactorial.chinook.invoiceline il

LEFT JOIN da-nfactorial.chinook.track t

ON il.TrackId = t.TrackId

LEFT JOIN da-nfactorial.chinook.album al

ON t.AlbumId = al.AlbumId

LEFT JOIN da-nfactorial.chinook.artist ar

ON al.ArtistId = ar.ArtistId

LEFT JOIN da-nfactorial.chinook.genre g

ON t.GenreId = g.GenreId

GROUP BY

Name

ORDER BY TotalIncome DESC

LIMIT 1

-----------------------------------------------------------

SELECT

c.CustomerID as CustomerId,

c.FirstName as Name,

c.LastName as Surname,

ar.Name as ArtistName,

SUM(il.quantity\*il.UnitPrice) as TotalSpent,

FROM da-nfactorial.chinook.customer c

LEFT JOIN da-nfactorial.chinook.invoice i

ON c.CustomerId = i.CustomerId

LEFT JOIN da-nfactorial.chinook.invoiceline il

ON i.InvoiceId = il.InvoiceId

LEFT JOIN da-nfactorial.chinook.track t

ON il.trackid = t.trackid

LEFT JOIN da-nfactorial.chinook.album al

ON t.albumid = al.albumid

LEFT JOIN da-nfactorial.chinook.artist ar

ON al.ArtistId = ar.ArtistId

WHERE ar.Name = "Iron Maiden"

GROUP BY

CustomerId,

Name,

Surname,

ArtistName

ORDER BY TotalSpent DESC  
LIMIT 1

### **Question 7: Which music genre is the most popular? Which genre has generated the most sales?**

My query:

WITH GenreSales AS (

SELECT

i.BillingCountry as Country,

g.Name as Genre,

COUNT(i.InvoiceID) as AmountSold

FROM da-nfactorial.chinook.invoice i

LEFT JOIN da-nfactorial.chinook.invoiceline il

ON i.InvoiceId = il.InvoiceId

LEFT JOIN da-nfactorial.chinook.track t

ON il.TrackId = t.TrackId

LEFT JOIN da-nfactorial.chinook.album al

ON t.AlbumId = al.AlbumId

LEFT JOIN da-nfactorial.chinook.genre g

ON t.GenreId = g.GenreId

GROUP BY

Country,

Genre

),

TopGenre AS(

SELECT

Country,

Genre,

AmountSold,

RANK () OVER (PARTITION BY COUNTRY ORDER BY AmountSold DESC) AS GenreNumber FROM GenreSales

)

SELECT

TG.Country,

TG.Genre,

TG.AmountSold

FROM TopGenre TG

WHERE GenreNumber = 1

GROUP BY

TG.Country,

TG.Genre,

TG.AmountSold

ORDER BY TG.COUNTRY ASC

#### **Вопрос 8: Which songs have the longest duration? Identify the songs that have above-average duration**

My query:

SELECT

Name as Track,Milliseconds as Duration

FROM da-nfactorial.chinook.track

WHERE Milliseconds > (SELECT AVG(Milliseconds) as AVGDuration

FROM da-nfactorial.chinook.track)

GROUP BY Track,Duration

ORDER BY Duration DESC

#### **Question 9:**

Write a query to determine the customer who spent the most on music for each country. Write a query that returns the country along with the top spender(s) and the amount they spent. For countries with multiple customers who spent the maximum amount, list all customers who spent that amount.

My query:

WITH Table1 as (SELECT

c.FirstName || ' ' || c.LastName as FullName,

i.BillingCountry as Country,

SUM(i.Total) as TotalSpent

FROM da-nfactorial.chinook.customer c

JOIN da-nfactorial.chinook.invoice i

ON c.CustomerID = i.CustomerID

GROUP BY Fullname,

Country

ORDER BY Country ASC),

Table2 AS (

SELECT FullName, Country, TotalSpent,

RANK () OVER (PARTITION BY Country ORDER BY TotalSpent DESC) AS Ranking FROM Table1

)

SELECT Fullname, Country, TotalSpent

FROM Table2

WHERE Ranking = 1

GROUP BY 1,2,3

ORDER BY 2 ASC

## **Part 2**

**Define four questions about Chinook that you want to answer based on data analysis.**

* Write SQL queries to obtain the data needed to successfully answer your questions.
* Visualize the obtained data (using histograms or other charts) that answer your questions.
* Explain the answers in 1-2 sentences.
* The questions you ask are up to you, but all four queries should include JOINs and AGGREGATION.

**Question 1:** Which artists created the 10 most popular albums and what are their genres? Popularity is determined by the number of purchases of songs included in the album.

My query and graph with explanation:

SELECT ar.Name,

al.Title,

g.Name as Genre,

COUNT(i.InvoiceID) as TimerPurchased

FROM da-nfactorial.chinook.invoiceline i

JOIN da-nfactorial.chinook.track t

ON t.TrackID = i.TrackID

JOIN da-nfactorial.chinook.album al

ON t.AlbumID=al.AlbumID

JOIN da-nfactorial.chinook.artist ar

ON al.ArtistID = ar.ArtistID

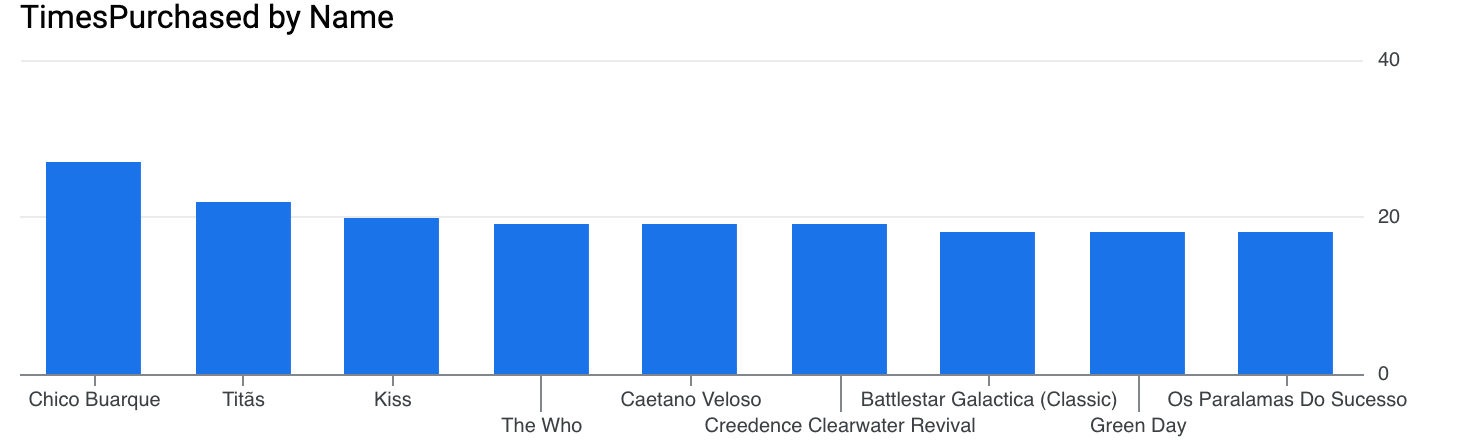
JOIN da-nfactorial.chinook.genre g

ON t.GenreID = g.GenreID

GROUP BY 1,2,3

ORDER BY 4 DESC

LIMIT 10



This query identifies the top 10 albums by sales and the artists who created these albums. Leading the list is artist Chico Buarque with the Latin album Minha História, which has 27 track sales. Next are Acústico by artist Titãs with 22 sales and Greatest Kiss by Kiss with 20 sales. Rounding out the list are albums by artists Os Paralamas do Sucesso, Creedence Clearwater Revival, Green Day, and Battlestar Galactica, each with 18 sales.

**Question 2:** What are the top 3 most popular genres in each playlist? Popularity of a genre is determined by the number of songs of that genre in the playlist.

My query and explanation with graph:

WITH Top3 as (SELECT

PlaylistID,

GenreCount,

RANK() OVER(PARTITION BY PlaylistID ORDER BY GenreCount DESC) as Ranking,

Genre as Genre1,

LEAD(Genre) OVER(PARTITION BY PlaylistID ORDER BY GenreCount DESC) as Genre2,

LEAD(Genre,2) OVER(PARTITION BY PlaylistID ORDER BY GenreCount DESC) as Genre3,

FROM (

SELECT

g.Name as Genre,

pt.PlaylistID,

COUNT(pt.TrackID) AS GenreCount

FROM

da-nfactorial.chinook.playlisttrack pt

JOIN

da-nfactorial.chinook.track t ON pt.TrackID = t.TrackID

JOIN

da-nfactorial.chinook.genre g ON t.GenreID = g.GenreID

GROUP BY

g.Name,

pt.PlaylistID

) AS subquery

ORDER BY

PlaylistID,

Ranking)

SELECT PlaylistID,

Genre1 as Top1Genre,

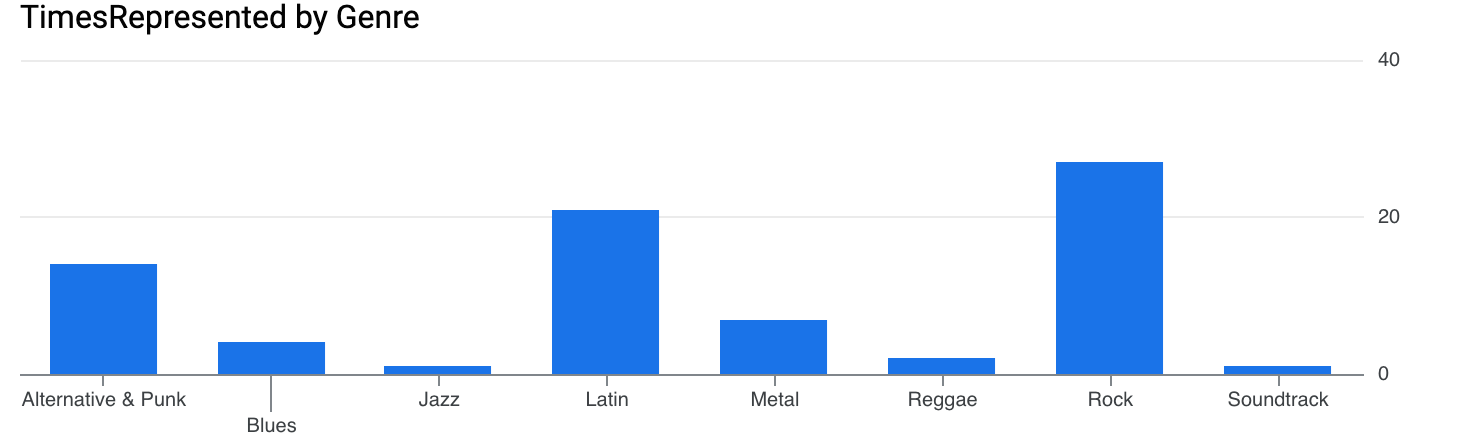
Genre2 as Top2Genre,

Genre3 as Top3Genre

FROM Top3

WHERE Ranking = 1

ORDER BY 1 ASC



This chart shows the number of times a particular genre appears in the top 3 genres of playlists (for this chart, the code had to be slightly modified).

**Question 3:** Which month is the most profitable and which month had the lowest sales? Also, what is the average order amount, the cost of the most expensive and the cheapest orders, and their quantities?

My query and graph with explanation:

SELECT

DATE\_TRUNC(InvoiceDate, month) AS Month,

SUM(Total) AS TotalSales,

AVG(Total) as AvgPerInvoice,

COUNT(DISTINCT(InvoiceId)) as Invoices,

MAX(Total) as BiggestPurchase,

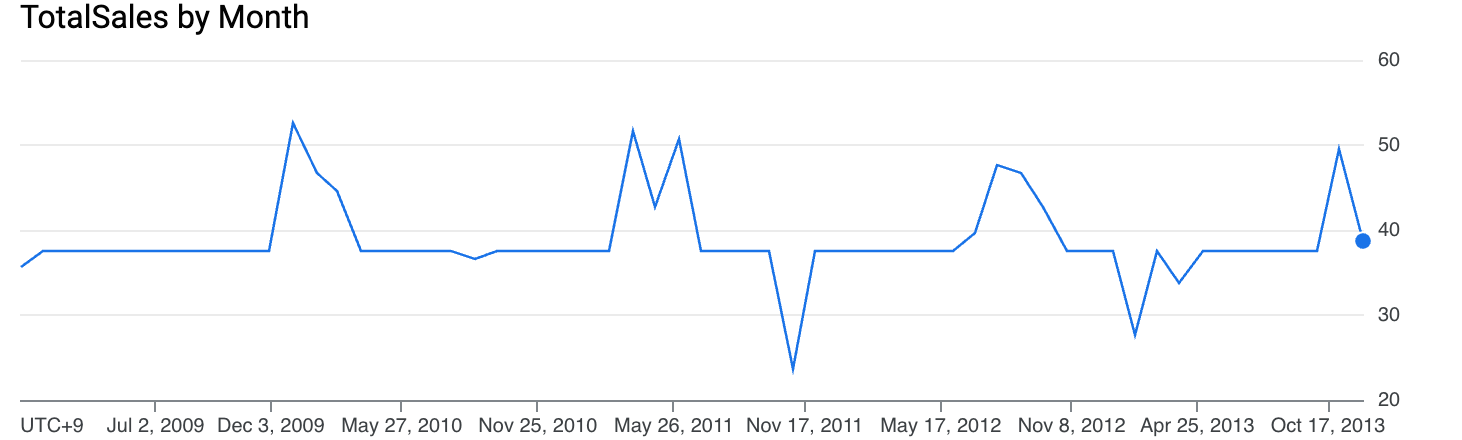
MIN(Total) as SmallestPurchase

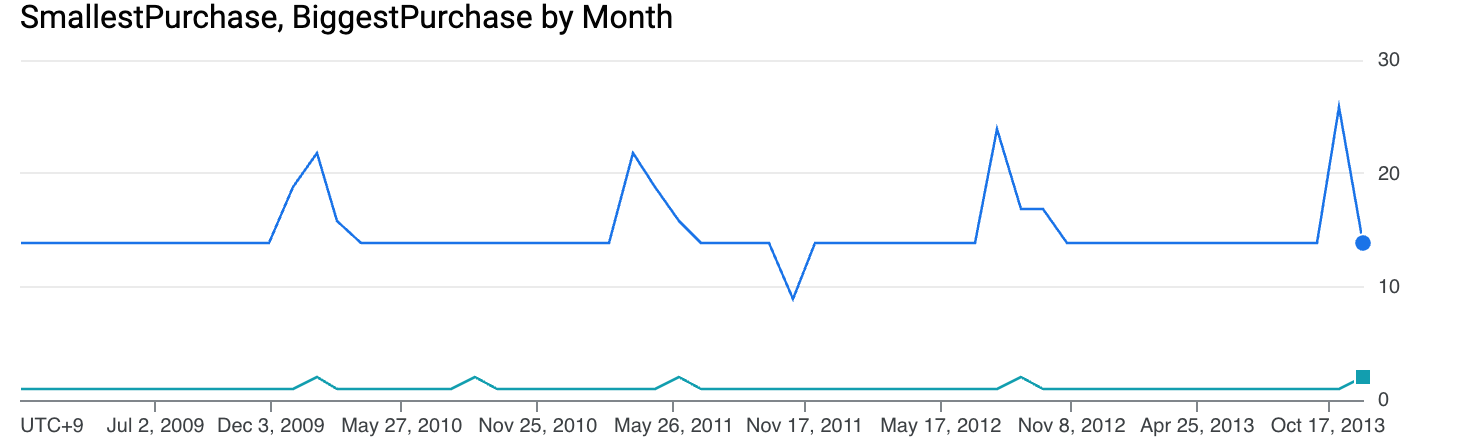
FROM da-nfactorial.chinook.invoice

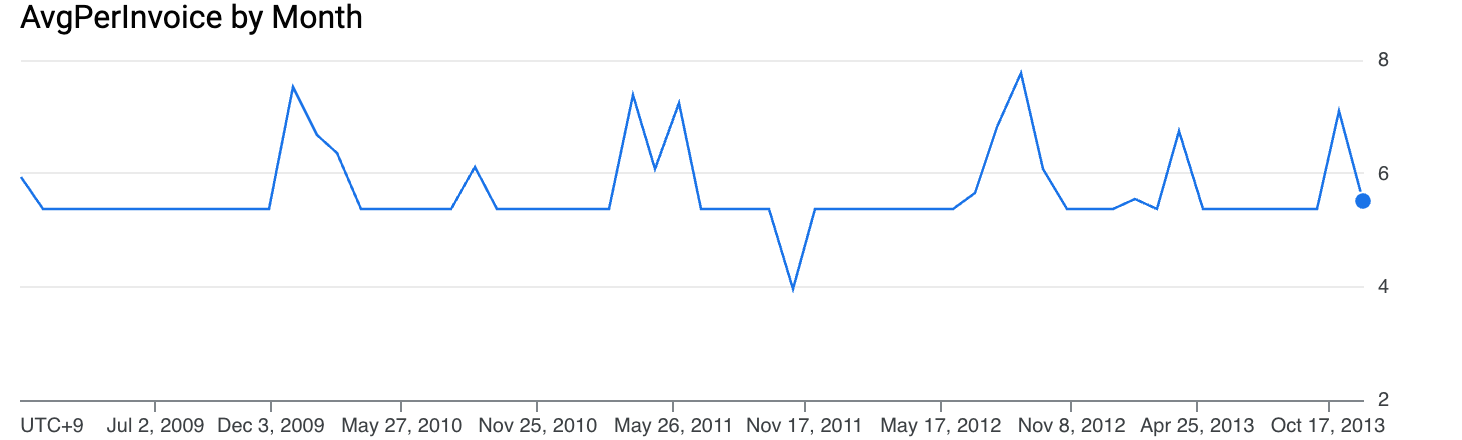
GROUP BY 1

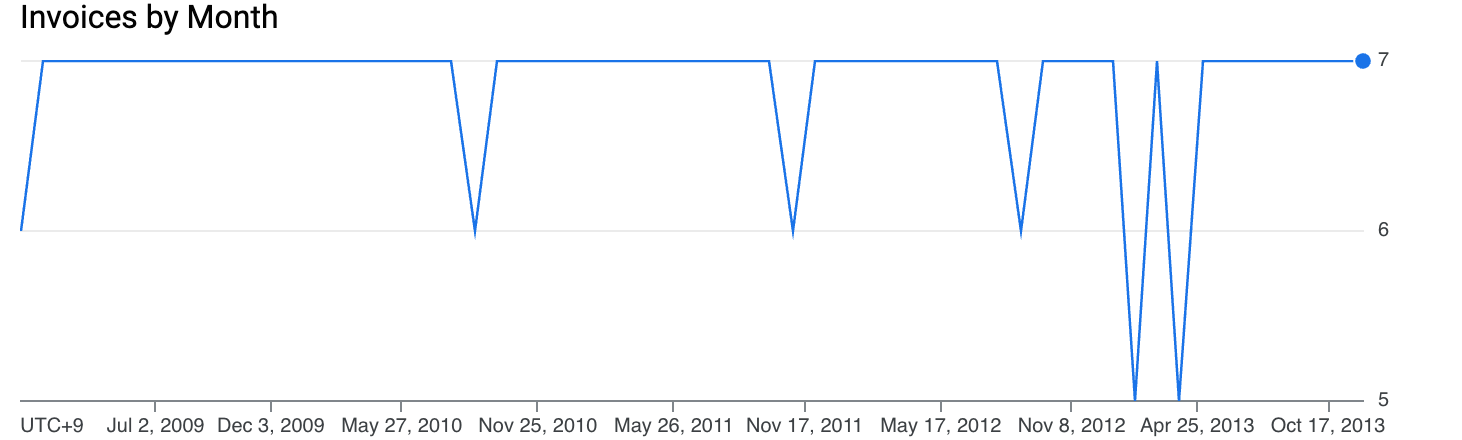
ORDER BY 1 ASC

ORDER BY TotalSales DESC









**Question 4:** How many clients does each Sales Rep have, and how much do they spend in total?

My query with graph and explanation:

SELECT Employee,

COUNT(DISTINCT(CustomerId)) as Clients,

SUM(TotalSpent) as CLientsSpendings

FROM (SELECT

c.CustomerID as CustomerId,

c.FirstName as Name,

c.LastName as Surname,

e.EmployeeID as Employee,

SUM(il.quantity\*il.UnitPrice) as TotalSpent,

FROM da-nfactorial.chinook.customer c

LEFT JOIN da-nfactorial.chinook.invoice i

ON c.CustomerId = i.CustomerId

LEFT JOIN da-nfactorial.chinook.invoiceline il

ON i.InvoiceId = il.InvoiceId

LEFT JOIN da-nfactorial.chinook.track t

ON il.trackid = t.trackid

LEFT JOIN da-nfactorial.chinook.album al

ON t.albumid = al.albumid

LEFT JOIN da-nfactorial.chinook.artist ar

ON al.ArtistId = ar.ArtistId

LEFT JOIN da-nfactorial.chinook.employee e

ON c.SupportRepID = e.EmployeeID

GROUP BY

CustomerId,

Name,

Surname,

EmployeeID

ORDER BY TotalSpent DESC)

GROUP BY 1

ORDER BY 3 DESC

