Data-Driven Crescendo: Crafting Harmony in Music Store Insights  
  
Embark on a riveting data analysis odyssey, transforming commonplace music store data into a compelling narrative of strategic brilliance. Delve into the intricate rhythms of customer behavior, inventory dynamics, demographic nuances, and pricing strategies. Uncover hidden patterns, enabling the creation of personalized experiences and orchestrating marketing symphonies that resonate with diverse audiences. This project is more than a routine data analysis; it's a symphony of innovation, offering a unique perspective that will captivate potential employers and set you apart in the competitive landscape of data analytics.Top of Form

**Skills in highlight: Data analytics, SQL**

**Tools: Mqsql workbench**

Questions from the data: LVL1

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

SELECT \*

FROM

employee

ORDER BY

levels

DESC

LIMIT 1;



1. Which countries have the most invoices?

SELECT COUNT(\*) AS c,

billing\_country

FROM

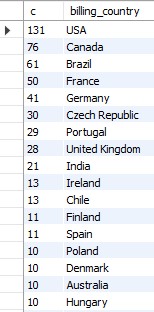
invoice

GROUP BY

billing\_country

ORDER BY

c DESC;



1. What are top 3 values of total invoice?

SELECT

total

FROM

invoice

ORDER BY

total DESC

LIMIT

3;

A screenshot of a computer

Description automatically generated

1. Which city has the best customers. Write a query that returns one city that has the highest sum of invoice totals. (Returning both the city name & sum of all invoice totals?)

SELECT

SUM(total) AS invoice\_total,

billing\_city

FROM

invoice

GROUP BY

billing\_city

ORDER BY

invoice\_total DESC

LIMIT 1;



1. Who is the best customer? The customer who has spent the most money will be declared the best customer.

SELECT

customer.customer\_id,

customer.first\_name,

customer.last\_name,

SUM(invoice.total) AS tot

FROM

customer

JOIN

invoice ON customer.customer\_id = invoice.customer\_id

GROUP BY

customer.customer\_id,

customer.first\_name,

customer.last\_name

ORDER BY

tot DESC

LIMIT

1;



Questions from the data: LVL2

1. Write query to return the email, first name, last name, & Genre of all Rock Music listeners.

SELECT DISTINCT

customer.email,

customer.first\_name,

customer.last\_name

FROM

customer

JOIN

invoice ON customer.customer\_id = invoice.customer\_id

JOIN

invoice\_line ON invoice.invoice\_id = invoice\_line.invoice\_id

WHERE

invoice\_line.track\_id IN (

SELECT

track\_id

FROM

track

JOIN

genre ON track.genre\_id = genre.genre\_id

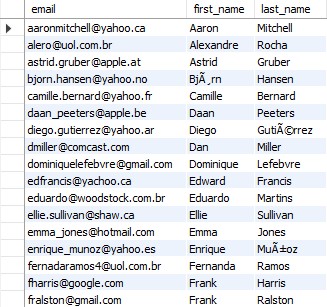
WHERE

genre.name LIKE 'Rock'

)

ORDER BY

customer.email;



1. 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

artist.artist\_id,

artist.name,

COUNT(artist.artist\_id) AS number\_of\_songs

FROM

track

JOIN

album2 ON album2.album\_id = track.album\_id

JOIN

artist ON artist.artist\_id = album2.artist\_id

JOIN

genre ON genre.genre\_id = track.genre\_id

WHERE

genre.name LIKE 'Rock'

GROUP BY

artist.artist\_id,

artist.name

ORDER BY

number\_of\_songs DESC

LIMIT

10;

A screenshot of a music list

Description automatically generated

1. 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

AVG(milliseconds) AS avg\_track\_length

FROM

track

)

ORDER BY

milliseconds DESC;

A screenshot of a computer

Description automatically generated

Questions from the data: LVL3

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

WITH best\_selling\_artist AS (

SELECT

artist.artist\_id AS artist\_id,

artist.name AS artist\_name,

SUM(invoice\_line.unit\_price \* invoice\_line.quantity) AS total\_sales

FROM

invoice\_line

JOIN

track ON track.track\_id = invoice\_line.track\_id

JOIN

album2 ON album2.album\_id = track.album\_id

JOIN

artist ON artist.artist\_id = album2.artist\_id

GROUP BY

1, 2

ORDER BY

3 DESC

LIMIT 1

)

SELECT

c.customer\_id,

c.first\_name,

c.last\_name,

bsa.artist\_name,

SUM(il.unit\_price \* il.quantity) AS amount\_spent

FROM

invoice i

JOIN

customer c ON c.customer\_id = i.customer\_id

JOIN

invoice\_line il ON il.invoice\_id = i.invoice\_id

JOIN

track t ON t.track\_id = il.track\_id

JOIN

album2 alb ON alb.album\_id = t.album\_id

JOIN

best\_selling\_artist bsa ON bsa.artist\_id = alb.artist\_id

GROUP BY

1, 2, 3, 4

ORDER BY

5 DESC;

A screenshot of a computer

Description automatically generated

1. 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

**Method 1 : CET**

WITH popular\_genre AS

(

SELECT

COUNT(invoice\_line.quantity) AS purchases,

customer.country,

genre.name,

genre.genre\_id,

ROW\_NUMBER() OVER(PARTITION BY customer.country ORDER BY COUNT(invoice\_line.quantity) DESC) AS RowNo

FROM

invoice\_line

JOIN

invoice ON invoice.invoice\_id = invoice\_line.invoice\_id

JOIN

customer ON customer.customer\_id = invoice.customer\_id

JOIN

track ON track.track\_id = invoice\_line.track\_id

JOIN

genre ON genre.genre\_id = track.genre\_id

GROUP BY

2,3,4

ORDER BY

2 ASC, 1 DESC

)

SELECT \* FROM popular\_genre WHERE RowNo <= 1;

A screenshot of a table

Description automatically generated

**Method 2: Recursive**

WITH RECURSIVE

sales\_per\_country AS (

SELECT

COUNT(\*) AS purchases\_per\_genre,

customer.country,

genre.name,

genre.genre\_id

FROM

invoice\_line

JOIN

invoice ON invoice.invoice\_id = invoice\_line.invoice\_id

JOIN

customer ON customer.customer\_id = invoice.customer\_id

JOIN

track ON track.track\_id = invoice\_line.track\_id

JOIN

genre ON genre.genre\_id = track.genre\_id

GROUP BY

2,3,4

ORDER BY

2

),

max\_genre\_per\_country AS (

SELECT

MAX(purchases\_per\_genre) AS max\_genre\_number,

country

FROM

sales\_per\_country

GROUP BY

2

ORDER BY

2

)

SELECT

sales\_per\_country.\*

FROM

sales\_per\_country

JOIN

max\_genre\_per\_country ON sales\_per\_country.country = max\_genre\_per\_country.country

WHERE

sales\_per\_country.purchases\_per\_genre = max\_genre\_per\_country.max\_genre\_number;

A screenshot of a computer

Description automatically generated

1. 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

**Method 1 : CET**

WITH Customter\_with\_country AS (

SELECT

customer.customer\_id,

first\_name,

last\_name,

billing\_country,

SUM(total) AS total\_spending,

ROW\_NUMBER() OVER(PARTITION BY billing\_country ORDER BY SUM(total) DESC) AS RowNo

FROM

invoice

JOIN

customer ON customer.customer\_id = invoice.customer\_id

GROUP BY

1, 2, 3, 4

ORDER BY

4 ASC, 5 DESC

)

SELECT

\*

FROM

Customter\_with\_country

WHERE

RowNo <= 1;

A screenshot of a computer

Description automatically generated

**Method 2: Recursive**

WITH RECURSIVE

customter\_with\_country AS (

SELECT

customer.customer\_id,

first\_name,

last\_name,

billing\_country,

SUM(total) AS total\_spending

FROM

invoice

JOIN

customer ON customer.customer\_id = invoice.customer\_id

GROUP BY

1,2,3,4

ORDER BY

2,3 DESC

),

country\_max\_spending AS (

SELECT

billing\_country,

MAX(total\_spending) AS max\_spending

FROM

customter\_with\_country

GROUP BY

billing\_country

)

SELECT

cc.billing\_country,

cc.total\_spending,

cc.first\_name,

cc.last\_name,

cc.customer\_id

FROM

customter\_with\_country cc

JOIN

country\_max\_spending ms ON cc.billing\_country = ms.billing\_country

WHERE

cc.total\_spending = ms.max\_spending

ORDER BY

1;

