**Question Set 1 - Easy**

**Question 1: who is the senior most employee based on job role**

**Query:**  select \* from employee

order BY levels desc

limit 3

**Steps:**

1. Look at all employees in the employee table.
2. Sort them by their job level, starting with the highest (senior-most).
3. Show only the top 3 senior employees.

**Question 2 : which contries have the most invoices**

**Query:**  SELECT COUNT(\*) AS e, billing\_country

FROM invoice

GROUP BY billing\_country

ORDER BY e DESC

**Steps:**

1. Count how many invoices were created for each country.
2. Group the results by each country to see totals per country.
3. Arrange the countries in order of invoice count, starting with the highest.

**Question 3: what are the top 3 values of the total invoice**

**Query:** SELECT \* FROM invoice

ORDER by total desc

limit 3

**Steps:**

1. Look at all the invoices in the invoice table.
2. Sort them by the total amount, starting with the biggest.
3. Show only the top 3 invoices with the highest amounts.

**Question 4: Which city has the best customers? We would like to throw a promotional Music**

**Festival in the city we made the most money. 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**

**Query:** SELECT billing\_city,SUM(total) AS InvoiceTotal

FROM invoice

GROUP BY billing\_city

ORDER BY InvoiceTotal DESC

LIMIT 1;

**Steps:**

1. Add up the total money made from invoices for each city.
2. Group the results by city to see totals for each city.
3. Arrange the cities by total money made, starting with the highest.
4. Show only the city where you made the most money.

**Question 5: Who is the best customer? The customer who has spent the most money will be**

**declared the best customer. Write a query that returns the person who has spent the**

**most money**

**Query:** SELECT customer.customer\_id, first\_name, last\_name, SUM(total) AS total\_spending

FROM customer

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

GROUP BY customer.customer\_id

ORDER BY total\_spending DESC

LIMIT 1;

**Steps:**

1. Combine the customer and invoice tables using their customer\_id.
2. Add up how much money each customer has spent.
3. Group the results by each customer.
4. Arrange the customers by how much they’ve spent, starting with the highest.
5. Show only the customer who spent the most.

**Question Set 2 - Moderate**

**Q1: List the email, first name, last name, and genre of all Rock music listeners, ordered by email.**

**Query:**

sql

Copy code

SELECT DISTINCT email AS Email, first\_name AS FirstName, last\_name AS LastName, genre.name AS Name

FROM customer

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

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

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

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

WHERE genre.name LIKE 'Rock'

ORDER BY email;

**Steps:**

1. **Find customers who listened to Rock music**:
   * Link the customer table to the invoice table using customer\_id to find what each customer purchased.
   * Use invoice\_line to see the individual tracks in each invoice.
   * Connect track to genre using genre\_id to identify tracks in the "Rock" genre.
2. **Select relevant details**: Get the customer’s email, first name, last name, and the genre name (Rock).
3. **Remove duplicates**: Use DISTINCT to avoid showing the same customer multiple times.
4. **Order alphabetically by email**: Sort the results using ORDER BY email.

**Q2: Find the top 10 artists (bands) with the most Rock songs.**

**Query:**

sql

Copy code

SELECT artist.artist\_id, artist.name, COUNT(artist.artist\_id) AS number\_of\_songs

FROM track

JOIN album ON album.album\_id = track.album\_id

JOIN artist ON artist.artist\_id = album.artist\_id

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

WHERE genre.name LIKE 'Rock'

GROUP BY artist.artist\_id

ORDER BY number\_of\_songs DESC

LIMIT 10;

**Steps:**

1. **Find Rock songs**:
   * Start with the track table to get song details.
   * Use album and artist to link tracks to their respective artists.
   * Filter tracks in the "Rock" genre using genre.
2. **Count songs by artist**: Use COUNT(artist.artist\_id) to calculate how many Rock songs each artist has.
3. **Group by artist**: Use GROUP BY artist.artist\_id to get one row per artist.
4. **Sort by song count**: Use ORDER BY number\_of\_songs DESC to rank artists starting with the most songs.
5. **Show the top 10 artists**: Use LIMIT 10 to display only the top 10 results.

**Q3: Find all songs longer than the average song length, sorted by length (longest first).**

**Query:**

sql

Copy code

SELECT name, milliseconds

FROM track

WHERE milliseconds > (

SELECT AVG(milliseconds) AS avg\_track\_length

FROM track

)

ORDER BY milliseconds DESC;

**Steps:**

1. **Calculate the average song length**: Use AVG(milliseconds) in a subquery to find the average duration of all songs.
2. **Find songs longer than the average**:
   * Compare each song’s length (milliseconds) with the average length calculated in the subquery.
   * Use WHERE milliseconds > (...) to filter only longer songs.
3. **Select relevant details**: Show the song’s name and its duration in milliseconds.
4. **Sort by length**: Use ORDER BY milliseconds DESC to list the longest songs first.

**Question Set 3 - Advance**

**Q1: How much has each customer spent on the best-selling artist?**

**Query:**

sql

Copy code

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

JOIN artist ON artist.artist\_id = album.artist\_id

GROUP BY 1

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 album 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;

**Steps:**

1. **Find the best-selling artist**:
   * Use a **Common Table Expression (CTE)** named best\_selling\_artist to calculate which artist has the highest total sales.
   * Join invoice\_line, track, album, and artist to get sales for each artist.
   * Group by artist\_id, calculate total sales, and pick the top-selling artist using ORDER BY and LIMIT 1.
2. **Calculate customer spending on the best-selling artist**:
   * Join invoice, customer, invoice\_line, track, and album to link purchases with artists.
   * Use the best\_selling\_artist CTE to filter transactions related to that artist.
   * Group by customer details and the artist’s name, summing up their spending (unit\_price \* quantity).
3. **Sort and display results**:
   * Order customers by how much they spent on the artist, from highest to lowest.

**Q2: What is the most popular music genre in each country?**

**Query:**

sql

Copy code

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;

**Steps:**

1. **Calculate genre popularity in each country**:
   * Use a **CTE** called popular\_genre to count how many times each genre was purchased (COUNT(invoice\_line.quantity)).
   * Join invoice\_line, invoice, customer, track, and genre to link countries with genres.
   * Group by country and genre to calculate purchases for each combination.
2. **Rank genres within each country**:
   * Use ROW\_NUMBER() to rank genres within each country by purchase count (highest first).
3. **Filter to get the top genre**:
   * In the main query, pick only rows where RowNo = 1 (the most popular genre in each country).

**Q3: Which customer spent the most on music in each country?**

**Query:**

sql

Copy code

WITH Customer\_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 Customer\_with\_country

WHERE RowNo <= 1;

**Steps:**

1. **Calculate customer spending in each country**:
   * Use a **CTE** called Customer\_with\_country to calculate how much each customer spent (SUM(total)).
   * Join invoice and customer to associate spending with customers and their countries.
   * Group by customer and country to compute total spending for each customer in each country.
2. **Rank customers within each country**:
   * Use ROW\_NUMBER() to rank customers by spending within each country (highest first).
3. **Filter to get the top spender**:
   * In the main query, select only rows where RowNo = 1 (the top spender in each country).

Top of Form

Bottom of Form