SQL PROJECT- MUSIC STORE DATA ANALYSIS

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

```
SELECT * FROM employee
ORDER BY levels desc
limit 1
```

o/p – emp_id = 9, first and last name = Mohan Madan, level = L7

2. Which countries have the most Invoices?

SELECT COUNT(*)AS c, billing_country (count no of invoices for each billing_country)

FROM invoice

GROUP BY billing_country(groups all the invoice rows **by country**, so you can count how many invoices belong to each country)

ORDER BY c desc (sorts the results by the count (c) in **descending order**, so the country with the **most invoices** appears at the top)

o/p – USA 131, Canada 76, Brazil 61

3. What are top 3 values of total invoice?

```
SELECT total FROM invoice
order by total desc
limit 3
```

o/p – 23.7599, 19.8, 19.8

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

```
SELECT SUM(total) AS invoice_total, billing_city (calculate sum of all invoice amounts for billing_city)
FROM invoice
GROUP BY billing_city (group data by each city)
ORDER BY invoice_total desc (results from highest to lowest)
```

o/p - invoice_total – 273.24, billing_city – Prague

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

```
SELECT customer.customer_id, customer.first_name, customer.last_name, SUM (invoice.total) AS total
from customer
join invoice on customer.customer_id = invoice.customer_id
group by customer.customer_id
order by total desc
limit 1
```

o/p – customer_id =5, total – 144.5, name - madhav

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

```
SELECT DISTINCT customer.customer_id, customer.first_name, customer.last_name (select customer_id first_name last_name from customer and distinct is used to ensure that each customer appears only once)
FROM customer
JOIN invoice ON customer.customer_id = invoice.customer_id
JOIN invoice_line ON invoice.invoice_id = invoice_line.invoice_id
JOIN track ON invoice_line.track_id = track.track_id
JOIN genre ON track.genre_id = genre.genre_id
WHERE genre.name LIKE 'Rock'
ORDER BY customer.last_name; (result is sorted alphabetically by customer’s last name)
```

o/p – customer_id = 12, first_name = Roberto, last_name = Almeida

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 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;
```

o/p – artist_id = 22, name = Led Zeppelin, no_of_songs = 114

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, miliseconds
FROM track
WHERE miliseconds > (
SELECT AVG(milliseconds) AS avg_track_length
FROM track)
ORDER BY miliseconds DESC;
```

o/p – 39359921210