

SQL PROJECT- MUSIC STORE DATA ANALYSIS

Question Set 1 – Easy

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

```
2  
3 SELECT * FROM employee  
4 ORDER BY levels desc  
5 limit 1;
```

2. Which countries have the most Invoices?

```
3 SELECT COUNT (*) as c,billing_country  
4 FROM invoice  
5 GROUP BY billing_country  
6 ORDER BY c DESC
```

3. What are top 3 values of total invoice?

```
9 SELECT TOTAL FROM INVOICE  
10 ORDER BY TOTAL DESC  
11 LIMIT 3
```

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.

```
14 select SUM (total) as invoice_total, billing_city  
15 from invoice  
16 group by billing_city  
17 order by invoice_total desc
```

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.

```
20 SELECT customer.customer_id, first_name, last_name, SUM(total) AS total_spending  
21 FROM customer  
22 JOIN invoice ON customer.customer_id = invoice.customer_id  
23 GROUP BY customer.customer_id  
24 ORDER BY total_spending DESC  
25 LIMIT 1
```

Question Set 2 – Moderate

1. 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.

```
27 SELECT DISTINCT email AS Email, first_name AS FirstName, last_name AS LastName, genre.name AS Name
28 FROM customer
29 JOIN invoice ON invoice.customer_id = customer.customer_id
30 JOIN invoice_line ON invoice_line.invoice_id = invoice.invoice_id
31 JOIN track ON track.track_id = invoice_line.track_id
32 JOIN genre ON genre.genre_id = track.genre_id
33 WHERE genre.name LIKE 'Rock'
34 ORDER BY email
```

2. 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

```
36 SELECT artist.artist_id, artist.name, COUNT(artist.artist_id) AS number_of_songs
37 FROM track
38 JOIN album ON album.album_id = track.album_id
39 JOIN artist ON artist.artist_id = album.artist_id
40 JOIN genre ON genre.genre_id = track.genre_id
41 WHERE genre.name LIKE 'Rock'
42 GROUP BY artist.artist_id
43 ORDER BY number_of_songs DESC
44 LIMIT 10;
```

3. 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.

```
46 SELECT name, milliseconds
47 FROM track
48 WHERE milliseconds > (
49     SELECT AVG(milliseconds) AS avg_track_length
50     FROM track )
51 ORDER BY milliseconds DESC;
```

Question Set 3 – Advance

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

```
--
53 v WITH best_selling_artist AS (
54     SELECT artist.artist_id AS artist_id, artist.name AS artist_name, SUM(invoice_line.unit_price*invoice_line.quantity) AS total_sales
55     FROM invoice_line
56     JOIN track ON track.track_id = invoice_line.track_id
57     JOIN album ON album.album_id = track.album_id
58     JOIN artist ON artist.artist_id = album.artist_id
59     GROUP BY 1
60     ORDER BY 3 DESC
61     LIMIT 1
62 )
63 SELECT c.customer_id, c.first_name, c.last_name, bsa.artist_name, SUM(il.unit_price*il.quantity) AS amount_spent
64 FROM invoice i
65 JOIN customer c ON c.customer_id = i.customer_id
66 JOIN invoice_line il ON il.invoice_id = i.invoice_id
67 JOIN track t ON t.track_id = il.track_id
68 JOIN album alb ON alb.album_id = t.album_id
69 JOIN best_selling_artist bsa ON bsa.artist_id = alb.artist_id
70 GROUP BY 1,2,3,4
71 ORDER BY 5 DESC;
```

2. 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.

```
74 v WITH popular_genre AS
75 (
76     SELECT COUNT(invoice_line.quantity) AS purchases, customer.country, genre.name, genre.genre_id,
77     ROW_NUMBER() OVER(PARTITION BY customer.country ORDER BY COUNT(invoice_line.quantity) DESC) AS RowNo
78     FROM invoice_line
79     JOIN invoice ON invoice.invoice_id = invoice_line.invoice_id
80     JOIN customer ON customer.customer_id = invoice.customer_id
81     JOIN track ON track.track_id = invoice_line.track_id
82     JOIN genre ON genre.genre_id = track.genre_id
83     GROUP BY 2,3,4
84     ORDER BY 2 ASC, 1 DESC
85 )
86 SELECT * FROM popular_genre WHERE RowNo <= 1
```

3. 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

```
89 WITH Customer_with_country AS (  
90     SELECT customer.customer_id, first_name, last_name, billing_country, SUM(total) AS total_spending,  
91     ROW_NUMBER() OVER(PARTITION BY billing_country ORDER BY SUM(total) DESC) AS RowNo  
92     FROM invoice  
93     JOIN customer ON customer.customer_id = invoice.customer_id  
94     GROUP BY 1,2,3,4  
95     ORDER BY 4 ASC, 5 DESC)  
96 SELECT * FROM Customer_with_country WHERE RowNo <= 1
```