Udacity SQL Project: Digital Music Store Challenge Questions

SQL: Question Set 1

Question 1

Which countries have the most Invoices?

Question 2

Which city has the best customers?

Question 3

Who is the best customer?

SQL: Question Set 2

Question 1

Use your query to return the email, first name, last name, and Genre of all **Rock** Music listeners (Rock & Roll would be considered a different category for this exercise). Return your list ordered alphabetically by email address starting with A.

```sql
SELECT BillingCountry,
       COUNT(InvoiceId) Invoices
FROM invoice
GROUP BY BillingCountry
ORDER BY Invoices DESC;


SELECT BillingCity,
       SUM(total)
FROM invoice
GROUP BY BillingCity
ORDER BY SUM(total) DESC
LIMIT 1;


SELECT c.CustomerId,
       c.FirstName,
       c.LastName,
       SUM(i.Total)
FROM customer c
JOIN invoice i ON c.CustomerId = i.CustomerId
GROUP BY c.CustomerId
ORDER BY SUM(i.Total) DESC
LIMIT 1;


SELECT DISTINCT c.Email,
                c.FirstName,
                c.LastName,
                g.name
FROM customer c
JOIN invoice i ON c.CustomerId = i.CustomerId
JOIN invoiceline iv ON i.InvoiceId = iv.InvoiceId
JOIN track t ON iv.TrackId = t.TrackId
JOIN Genre g ON t.GenreId = g.GenreId
WHERE g.GenreId = 1
ORDER BY c.email;
```

Question 2

Who is writing the rock music?

Now that we know that our customers love rock music, we can decide which musicians to invite to play at the concert.

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.

Question 3

First, find which artist has earned the most according to the **InvoiceLines**?

Now use this artist to find which customer spent the most on this artist.

For this query, you will need to use the **Invoice**, **InvoiceLine**, **Track**, **Customer**, **Album**, and **Artist** tables.

```sql
SELECT a.Name,
       COUNT(t.trackid) total_tracks
FROM artist a
RIGHT JOIN album ab ON a.ArtistId = ab.ArtistId
RIGHT JOIN track t ON ab.AlbumId = t.AlbumId
WHERE t.GenreId = 1
GROUP BY 1
ORDER BY 2 DESC
LIMIT 10;


SELECT a.name,
       ab.ArtistId,
       SUM(il.UnitPrice) AS amount_spent
FROM artist a
RIGHT JOIN album ab ON a.ArtistId = ab.ArtistId
RIGHT JOIN track t ON ab.AlbumId = t.AlbumId
RIGHT JOIN InvoiceLine il ON t.trackid = il.trackid
GROUP BY a.name,
         ab.ArtistId
ORDER BY amount_spent DESC
LIMIT 1;


SELECT a.name,
       c.FirstName,
       c.LastName,
       c.customerid,
       SUM(il.UnitPrice) AS amount_spent
FROM artist a
RIGHT JOIN album ab ON a.ArtistId = ab.ArtistId
RIGHT JOIN track t ON ab.AlbumId = t.AlbumId
RIGHT JOIN InvoiceLine il ON t.trackid = il.trackid
RIGHT JOIN invoice i ON il.invoiceid = i.invoiceid
RIGHT JOIN customer c ON i.CustomerId = c.customerid
WHERE ab.artistid = 90
GROUP BY a.name,
         c.FirstName,
         c.LastName,
         c.customerid
ORDER BY amount_spent DESC
LIMIT 5;
```

SQL: Question Set 3 - Subqueries

Question 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 number 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.

Question 2

Return all the track names that have a song length longer than the average song length. Though you could perform this with two queries. Imagine you wanted your query to update based on when new data is put in the database. Therefore, you do not want to hard code the average into your query. You only need the **Track** table to complete this query.

Return the **Name** and **Milliseconds** for each track. Order by the song length with the longest songs listed first.

Question 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.

You should only need to use the **Customer** and **Invoice** tables.

```sql
SELECT MAX(purchases) AS max_purchases,
       country,
       name
FROM
  (SELECT c.Country,
          g.name,
          COUNT(il.invoicelineid) AS purchases
   FROM Genre g
   LEFT JOIN Track t ON g.genreid = t.genreid
   LEFT JOIN InvoiceLine il ON t.trackid = il.trackid
   LEFT JOIN invoice i ON il.invoiceid = i.invoiceid
   LEFT JOIN Customer c ON i.customerid = c.customerid
   GROUP BY c.Country,
            g.name
   ORDER BY c.country) AS Table1
WHERE country NOT LIKE 'NULL'
GROUP BY country,
         name
ORDER BY country,
         max_purchases DESC;
```

```sql
WITH GenrePurchases AS
  (SELECT c.Country,
          g.name,
          COUNT(il.invoicelineid) AS purchases
   FROM Genre g
   LEFT JOIN Track t ON g.genreid = t.genreid
   LEFT JOIN InvoiceLine il ON t.trackid = il.trackid
   LEFT JOIN Invoice i ON il.invoiceid = i.invoiceid
   LEFT JOIN Customer c ON i.customerid = c.customerid
   WHERE c.Country IS NOT NULL
   GROUP BY c.Country,
            g.name),
     MaxPurchases AS
  (SELECT Country,
          MAX(purchases) AS max_purchases
   FROM GenrePurchases
   GROUP BY Country)
SELECT gp.Country,
       gp.name,
       gp.purchases
FROM GenrePurchases gp
JOIN MaxPurchases mp ON gp.Country = mp.Country
AND gp.purchases = mp.max_purchases
ORDER BY gp.Country,
         gp.name;
```

```sql
SELECT Name,
       Milliseconds
FROM track
WHERE Milliseconds >
    (SELECT AVG(Milliseconds)
     FROM Track)
ORDER BY Milliseconds DESC;

WITH Customer_Purchases AS
  (SELECT c.FirstName,
          c.LastName,
          c.Country,
          SUM(i.Total) amount
   FROM customer c
   JOIN invoice i ON c.CustomerId = i.CustomerId
   GROUP BY c.FirstName,
            c.LastName,
            c.Country),
     MaxPurchases AS
  (SELECT MAX(amount) AS Max_amount,
          Country
   FROM Customer_Purchases
   GROUP BY Country)
SELECT cp.country,
       cp.FirstName,
       cp.LastName,
       mp.Max_amount
FROM Customer_Purchases cp
JOIN MaxPurchases mp ON cp.country = mp.country
AND cp.amount = mp.max_amount
ORDER BY mp.max_amount DESC;
```