MUSIC STORE ANALYSIS USING SQL
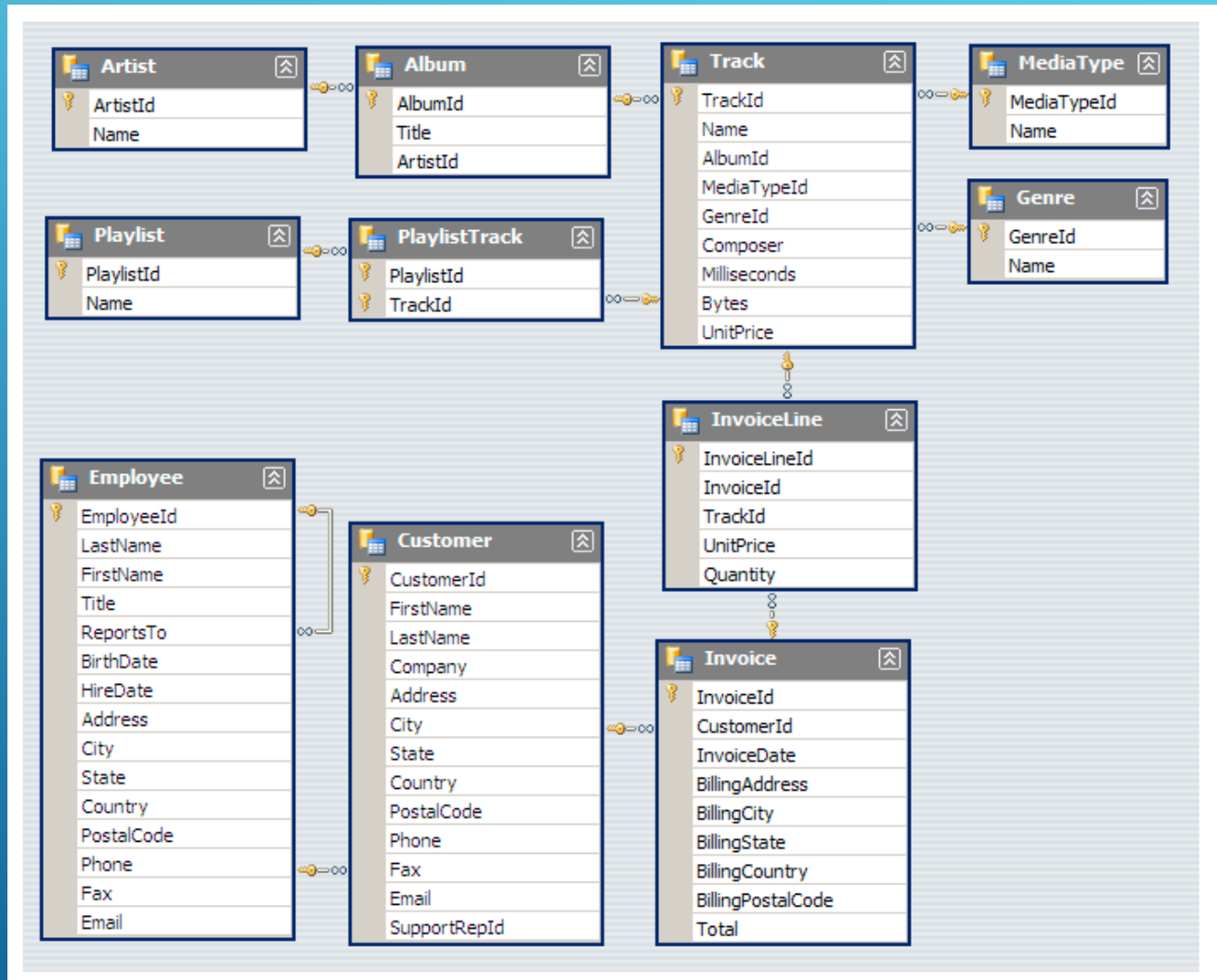
Database Schema :-

Questions & Answers

Easy Questions :-

Q1. Who is the senior most employee based on job title?

Query :

```
3  select * from employee
4  ORDER BY levels DESC
5  Limit 1;
```

Output :

	employee_id [PK] character varying (50)	last_name character	first_name character	title character varying (50)
1	9	Madan	Mohan	Senior General Manager

Q2. Which countries have the most invoices?

Query :

```
3  SELECT COUNT(*) AS c, billing_country
4  from invoice
5  GROUP BY billing_country
6  ORDER by c DESC;
```

Output :

	c bigint	billing_country character varying (30)
1	131	USA
2	76	Canada
3	61	Brazil
4	50	France
5	41	Germany

Q3. What are top 3 values of total invoice?

Query :

```
3  SELECT total from invoice
4  ORDER BY total DESC
5  LIMIT 3;
```

Output :

Data Output	Messages	N

	total double precision
1	23.759999999999998
2	19.8
3	19.8

Q4. 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 :

```
3  SELECT SUM(total) AS invoice_total, billing_city
4  from invoice
5  GROUP BY billing_city
6  ORDER BY invoice_total DESC;
```

Output :

Data Output	Messages	Notifications

	invoice_total double precision	billing_city character varying (30)
1	273.24000000000007	Prague
2	169.29	Mountain View
3	166.32	London
4	158.4	Berlin
5	151.47	Paris

Q5. 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 :

```
3   SELECT c.customer_id, c.first_name, c.last_name, SUM(i.total) AS total
4   FROM customer AS c
5   JOIN invoice AS i
6   ON c.customer_id = i.customer_id
7   GROUP BY c.customer_id
8   ORDER BY total DESC
9   LIMIT 1;
```

Output :

Data Output   Messages   Notifications

customer_id [PK] integer	first_name character	last_name character	total double precision
5	R	Madhav	144.54000000000002

Moderate Questions :-

Q1. 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.

Query :

```
3   SELECT DISTINCT email, first_name, last_name
4   FROM customer
5   JOIN invoice ON customer.customer_id = invoice.customer_id
6   JOIN invoice_line ON invoice.invoice_id = invoice_line.invoice_id
7   WHERE track_id IN(
8       SELECT track_id FROM track
9       JOIN genre ON track.genre_id = genre.genre_id
10      WHERE genre.name LIKE 'Rock'
11  )
12  ORDER BY email;
```

Output :

	email character varying (50)	first_name character	last_name character
1	aaronmitchell@yahoo.ca	Aaron	Mitchell
2	alero@uol.com.br	Alexandre	Rocha
3	astrid.gruber@apple.at	Astrid	Gruber
4	bjorn.hansen@yahoo.no	Bjørn	Hansen
5	camille.bernard@yahoo.fr	Camille	Bernard
6	daan_peeters@apple.be	Daan	Peeters
7	diego.gutierrez@yahoo.ar	Diego	Gutiérrez
8	dmiller@comcast.com	Dan	Miller
9	dominiquelefebvre@gmail.c...	Dominique	Lefebvre
10	edfrancis@yachoo.ca	Edward	Francis

Q2. 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.

Query :

```
3   SELECT artist.artist_id, artist.name, COUNT(artist.artist_id) AS no_of_songs
4   FROM track
5   JOIN album ON album.album_id = track.album_id
6   JOIN artist ON artist.artist_id = album.artist_id
7   JOIN genre ON genre.genre_id = track.genre_id
8   WHERE genre.name LIKE 'Rock'
9   GROUP BY artist.artist_id
10  ORDER BY no_of_songs DESC
11  LIMIT 10;
```

Output :

Data Output   Messages   Notifications

	artist_id [PK] character varying (50)	name character varying (120)	no_of_songs bigint
1	22	Led Zeppelin	114
2	150	U2	112
3	58	Deep Purple	92
4	90	Iron Maiden	81
5	118	Pearl Jam	54
6	152	Van Halen	52
7	51	Queen	45
8	142	The Rolling Stones	41
9	76	Creedence Clearwater Revival	40
10	52	Kiss	35

Q3. 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.

Query :

```
3   SELECT name,milliseconds
4   FROM track
5   where milliseconds >(
6       SELECT AVG(milliseconds) AS avg_len
7       FROM track
8   )
9   ORDER BY milliseconds DESC;
```

Output :

Data Output   Messages   Notifications

	name character varying (150)	milliseconds integer
1	Occupation / Precipice	5286953
2	Through a Looking Glass	5088838
3	Greetings from Earth, Pt. 1	2960293
4	The Man With Nine Lives	2956998
5	Battlestar Galactica, Pt. 2	2956081

Advance Questions :-

Q1. Find how much amount spent by each customer on artists? Write a query to return customer name, artist name and total spent.

Query :

```
3   WITH best_selling_artist AS(
4       SELECT artist.artist_id AS artist_id, artist.name AS artist_name,
5       SUM(invoice_line.unit_price*invoice_line.quantity) AS total_sales
6       FROM invoice_line
7       JOIN track ON track.track_id = invoice_line.track_id
8       JOIN album ON album.album_id = track.album_id
9       JOIN artist ON artist.artist_id = album.artist_id
10      GROUP BY 1
11      ORDER BY 3 DESC
12      LIMIT 1
13  )
14  SELECT c.customer_id, c.first_name, c.last_name, bsa.artist_name,
15  SUM(il.unit_price*il.quantity) AS amount_spent
16  FROM invoice i
17  JOIN customer c ON c.customer_id = i.customer_id
18  JOIN invoice_line il ON il.invoice_id = i.invoice_id
19  JOIN track t ON t.track_id = il.track_id
20  JOIN album alb ON alb.album_id = t.album_id
21  JOIN best_selling_artist bsa ON bsa.artist_id = alb.artist_id
22  GROUP BY 1,2,3,4
23  ORDER BY 5 DESC;
```

Output :

Data Output Messages Notifications

	customer_id integer	first_name character		last_name character		artist_name character varying (120)	amount_spent double precision
1	46	Hugh	...	O'Reilly	...	Queen	27.719999999999985
2	38	Niklas	...	Schröder	...	Queen	18.81
3	3	François	...	Tremblay	...	Queen	17.82
4	34	João	...	Fernandes	...	Queen	16.830000000000002
5	53	Phil	...	Hughes	...	Queen	11.88

Q2. 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.

Query :

```sql
3  WITH popular_genre AS
4  (
5      SELECT COUNT(invoice_line.quantity) AS purchases,
6      customer.country, genre.name, genre.genre_id,
7      ROW_NUMBER() OVER(
8          PARTITION BY customer.country
9          ORDER BY COUNT(invoice_line.quantity)DESC
10     ) AS RowNo
11     FROM invoice_line
12     JOIN invoice ON invoice.invoice_id = invoice_line.invoice_id
13     JOIN customer ON customer.customer_id = invoice.customer_id
14     JOIN track ON track.track_id = invoice_line.track_id
15     JOIN genre ON genre.genre_id = track.genre_id
16     GROUP BY 2,3,4
17     ORDER BY 2 ASC, 1 DESC
18 )
19 SELECT * FROM popular_genre WHERE  RowNo <= 1
```

Output :

Data Output Messages Notifications

	purchases bigint	country character varying (50)	name character varying (120)	genre_id character varying (50)	rowno bigint
1	17	Argentina	Alternative & Punk	4	1
2	34	Australia	Rock	1	1
3	40	Austria	Rock	1	1
4	26	Belgium	Rock	1	1
5	205	Brazil	Rock	1	1

Q2. 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.

Query :

```
3   WITH Customer_with_country AS (
4       SELECT customer.customer_id, first_name, last_name,
5       billing_country, SUM(total) AS total_apending,
6       ROW_NUMBER() OVER(
7           PARTITION BY billing_country
8           ORDER BY SUM(total) DESC
9       ) AS RowNo
10      FROM invoice
11      JOIN customer ON customer.customer_id = invoice.customer_id
12      GROUP BY 1,2,3,4
13      ORDER BY 4 ASC, 5 DESC
14  )
15  SELECT * FROM Customer_with_country
16  WHERE RowNo <=1
```

Output :

	customer_id integer	first_name character		last_name character		billing_country character varying (30)	total_apending double precision	rowno bigint
1	56	Diego	...	Gutiérrez	...	Argentina	39.6	1
2	55	Mark	...	Taylor	...	Australia	81.18	1
3	7	Astrid	...	Gruber	...	Austria	69.3	1
4	8	Daan	...	Peeters	...	Belgium	60.38999999999999	1
5	1	Luís	...	Gonçalves	...	Brazil	108.89999999999998	1