SQL Music Store Analysis
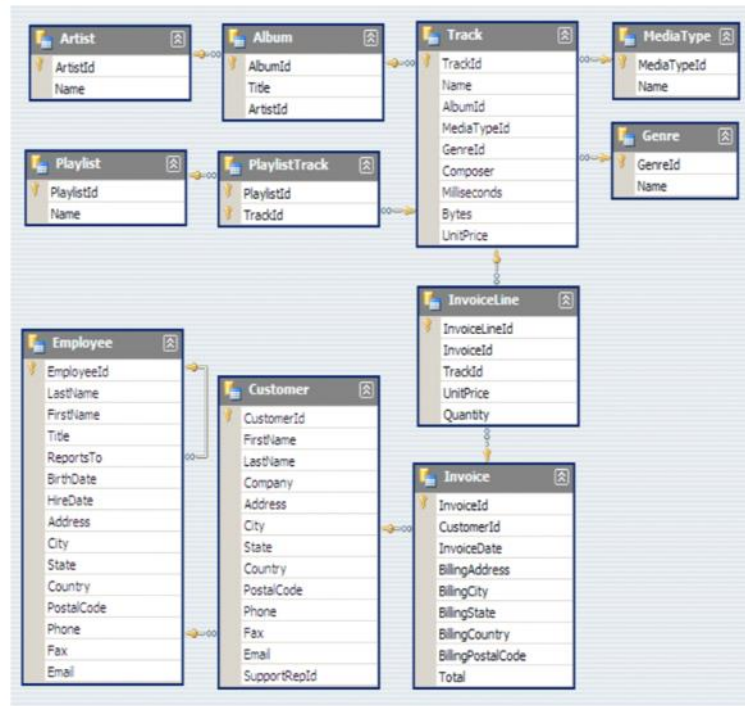
Friday, June 21, 2024   1:25 AM

SCHEMA:


```
1    -- Set 1
2
3    -- Q1: Who is the senior most employee based on job title?
4
5  ∨ SELECT title, first_name , last_name
6    FROM employee
7        ORDER BY levels DESC LIMIT 1 ;
```

Data Output   Messages   Notifications

title character varying (50)	first_name character	last_name character
1	Senior General Manager	Mohan	...	Madan	...

```
10    -- Q2: Which countries have the most Invoices?
11
12 ∨ SELECT  billing_country, COUNT(*) AS c
13    FROM invoice
14        GROUP BY billing_country ORDER BY c DESC ;
15
```

Data Output   Messages   Notifications

	billing_country character varying (30) 🔒	c bigint 🔒
1	USA	131
2	Canada	76
3	Brazil	61
4	France	50
5	Germany	41
6	Czech Republic	30
7	Portugal	29
8	United Kingdom	28
9	India	21
10	Chile	13
11	Ireland	13
12	Spain	11

Total rows: 24 of 24   Query complete 00:00:00.097

```
17    -- Q3: What are top 3 values of total invoice?
18
19 ∨ SELECT total
20    FROM invoice
21        ORDER BY total DESC
22            LIMIT 3;
```

Data Output   Messages   Notifications

	total double precision 🔒
1	23.759999999999998
2	19.8
3	19.8

```
25    -- Q4: Which city has the best customers?
26    --       We would like to throw a promotional Music Festival in the city we made the most money.
27    --       Write a query that returns one city that has the highest sum of invoice totals.
28    --       Return both the city name & sum of all invoice totals.
29
30 ∨ SELECT billing_city, SUM(total) AS s
31    FROM invoice
32        GROUP BY billing_city
33            ORDER BY s DESC LIMIT 1;
34    |
35
```

Data Output   Messages   Notifications

	billing_city character varying (30) 🔒	s double precision 🔒
1	Prague	273.24000000000007

```
35
36    -- Q5: Who is the best customer? The customer who has spent the most money will be declared the best customer.
37    --     Write a query that returns the person who has spent the most money.*/
38
39 v  SELECT c.customer_id, c.first_name, c.last_name, SUM(i.total) AS s
40     FROM customer c    INNER JOIN invoice i
41                    ON c.customer_id = i.customer_id
42                        GROUP BY c.customer_id
43                            ORDER BY s DESC LIMIT 1;
```

Data Output   Messages   Notifications

customer_id [PK] integer	first_name character	last_name character	s double precision
1	5	R	Madhav	144.54000000000002

```
47    -- SET 2
48
49    -- 1. Write query to return the email, first name, last name, & Genre of all Rock Music listeners.
50    --     Return your list ordered alphabetically by email starting with A
51
52 v  select distinct c.email , c.first_name , c.last_name , g.name
53     from customer c
54        INNER JOIN invoice i  ON c.customer_id = i.customer_id
55        INNER JOIN invoice_line il  ON  i.invoice_id = il.invoice_id
56        INNER JOIN track t   ON  il.track_id = t.track_id
57        INNER JOIN genre g   ON  t.genre_id = g.genre_id
58           where g.name = 'Rock'
59               order by c.email ;
60
```

Data Output   Messages   Notifications

	email character varying (50)	first_name character	last_name character	name character varying (120)
1	aaronmitchell@yahoo.ca	Aaron	Mitchell	Rock
2	alero@uol.com.br	Alexandre	Rocha	Rock
3	astrid.gruber@apple.at	Astrid	Gruber	Rock
4	bjorn.hansen@yahoo.no	Bjørn	Hansen	Rock
5	camille.bernard@yahoo.fr	Camille	Bernard	Rock

Total rows: 59 of 59   Query complete 00:00:00.087

```
62  -- 2. Let's invite the artists who have written the most rock music in our dataset.
63  --    Write a query that returns the Artist name and total track count of the top 10 rock bands.
64
65 ∨ select a.name , count(a.artist_id) as total_songs
66  from artist a
67      INNER JOIN album ab  ON a.artist_id = ab.artist_id
68      INNER JOIN track t   ON t.album_id  = ab.album_id
69      INNER JOIN genre g   ON t.genre_id = g.genre_id
70          where g.name = 'Rock'
71              group by a.name
72                  order by total_songs desc Limit 10 ;
```

Data Output   Messages   Notifications

	name character varying (120)	total_songs bigint
1	Led Zeppelin	114
2	U2	112
3	Deep Purple	92
4	Iron Maiden	81
5	Pearl Jam	54
6	Van Halen	52
7	Queen	45

Total rows: 10 of 10   Query complete 00:00:00.107

```
75  -- 3. Return all the track names that have a song length longer than the average song length.
76  --    Return the Name and Milliseconds for each track. Order by the song length with the longest songs listed firs
77
78 ∨ select name , milliseconds
79  from track
80      where milliseconds > ( select Avg(Milliseconds)
81                             from track )
82              order by milliseconds desc ;
```

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
6	Battlestar Galactica, Pt. 1	2952702
7	Murder On the Rising Star	2935894
8	Battlestar Galactica, Pt. 3	2927802
9	Take the Celestra	2927677
10	Fire In Space	2926503

Total rows: 494 of 494   Query complete 00:00:00.099   Ln 80, Col 49

```
86    -- Set 3
87
88    -- 1. Find how much amount spent by each customer on artists? Write a query to return customer name,
89    --    artist name and total spent
90
91  ∨ WITH best_selling_artist AS ( SELECT a.artist_id AS artist_id, a.name AS artist_name, SUM(il.unit_price * il.quant
92                           FROM invoice_line il
93                                INNER JOIN track t ON t.track_id = il.track_id
94                                INNER JOIN album ab ON ab.album_id = t.album_id
95                                INNER JOIN artist a ON a.artist_id = ab.artist_id
96                                    GROUP BY 1
97                                        ORDER BY total_sales DESC LIMIT 1
98                           )
99
100   SELECT c.customer_id, c.first_name, c.last_name, bsa.artist_name, SUM(il.unit_price * il.quantity) AS amount_spent
101   FROM invoice i
102       INNER JOIN customer c ON c.customer_id = i.customer_id
103       INNER JOIN invoice_line il ON il.invoice_id = i.invoice_id
104       INNER JOIN track t ON t.track_id = il.track_id
105       INNER JOIN album alb ON alb.album_id = t.album_id
106       INNER JOIN best_selling_artist bsa ON bsa.artist_id = alb.artist_id
107           GROUP BY 1,2,3,4
108               ORDER BY amount_spent DESC;
```

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
6	41	Marc	Dubois	Queen	11.88
7	47	Lucas	Mancini	Queen	10.89
8	33	Ellie	Sullivan	Queen	10.89
9	20	Dan	Miller	Queen	3.96
10	5	R	Madhav	Queen	3.96
11	23	John	Gordon	Queen	2.9699999999999998
12	54	Steve	Murray	Queen	2.9699999999999998
13	31	Martha	Silk	Queen	2.9699999999999998
14	16	Frank	Harris	Queen	1.98
15	17	Jack	Smith	Queen	1.98
16	24	Frank	Ralston	Queen	1.98
17	30	Edward	Francis	Queen	1.98
18	35	Madalena	Sampaio	Queen	1.98

Total rows: 43 of 43   Query complete 00:00:00.102

```sql
-- 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


WITH popular_genre AS ( SELECT COUNT(il.quantity) AS purchases, c.country, g.name, g.genre_id,
                               ROW_NUMBER() OVER( PARTITION BY c.country ORDER BY COUNT(il.quantity) DESC ) AS RN
                        FROM invoice_line il
                            INNER JOIN invoice i  ON i.invoice_id = il.invoice_id
                            INNER JOIN customer c ON c.customer_id = i.customer_id
                            INNER JOIN track t    ON t.track_id = il.track_id
                            INNER JOIN genre g  ON g.genre_id = t.genre_id
                                GROUP BY 2,3,4
                                    ORDER BY c.country , purchases DESC
                      )

SELECT *
FROM popular_genre
    WHERE RN <= 1 ;
```

Data Output Messages Notifications

	purchases bigint	country character varying (50)	name character varying (120)	genre_id character varying (50)	rn 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
6	333	Canada	Rock	1	1
7	61	Chile	Rock	1	1
8	143	Czech Republic	Rock	1	1
9	24	Denmark	Rock	1	1
10	46	Finland	Rock	1	1
11	211	France	Rock	1	1
12	194	Germany	Rock	1	1
13	44	Hungary	Rock	1	1
14	102	India	Rock	1	1
15	72	Ireland	Rock	1	1
16	35	Italy	Rock	1	1
17	33	Netherlands	Rock	1	1
18	40	Norway	Rock	1	1

Total rows: 24 of 24   Query complete 00:00:00.109

```sql
131
132    -- 3. Write a query that determines the customer that has spent the most on music for each country.
133    --    Write a query that returns the country along with the top customer and how much they spent.
134    --    For countries where the top amount spent is shared, provide all customers who spent this amount
135
136    WITH Customter_with_country AS ( SELECT c.customer_id,
137                                             c.first_name ,
138                                             c.last_name ,
139                                             i.billing_country,
140                                             SUM(i.total) AS total_spending,
141                                             ROW_NUMBER() OVER( PARTITION BY i.billing_country ORDER BY SUM(i.total) DESC ) AS RN
142                                    FROM invoice i
143                                        INNER JOIN customer c  ON c.customer_id = i.customer_id
144                                            GROUP BY 1,2,3,4
145                                                ORDER BY total_spending , RN DESC
146                                    )
147
148    SELECT *
149    FROM Customter_with_country
150        WHERE RN <= 1 ;
151
```

Data Output Messages Notifications

	customer_id integer	first_name character		last_name character		billing_country character varying (30)	total_spending double precision	rn bigint
1	9	Kara	...	Nielsen		Denmark	37.61999999999999	1
2	56	Diego	...	Gutiérrez		Argentina	39.6	1
3	47	Lucas	...	Mancini		Italy	50.49	1
4	8	Daan	...	Peeters		Belgium	60.38999999999999	1
5	48	Johannes	...	Van der Berg	...	Netherlands	65.34	1
6	7	Astrid	...	Gruber		Austria	69.3	1
7	4	Bjørn	...	Hansen		Norway	72.27000000000001	1
8	51	Joakim	...	Johansson	...	Sweden	75.24	1
9	49	Stanisław	...	Wójcik		Poland	76.22999999999999	1
10	45	Ladislav	...	Kovács		Hungary	78.21	1
11	44	Terhi	...	Hämäläinen	...	Finland	79.2	1
12	55	Mark	...	Taylor		Australia	81.18	1
13	37	Fynn	...	Zimmermann	...	Germany	94.05000000000001	1
14	57	Luis	...	Rojas		Chile	97.02000000000001	1
15	17	Jack	...	Smith		USA	98.01	1
16	50	Enrique	...	Muñoz		Spain	98.01	1
17	53	Phil	...	Hughes		United Kingdom	98.01	1
18	3	François	...	Tremblay	...	Canada	99.99	1

Total rows: 24 of 24   Query complete 00:00:00.082