

SQL ANALYSIS



MUSIC STORE

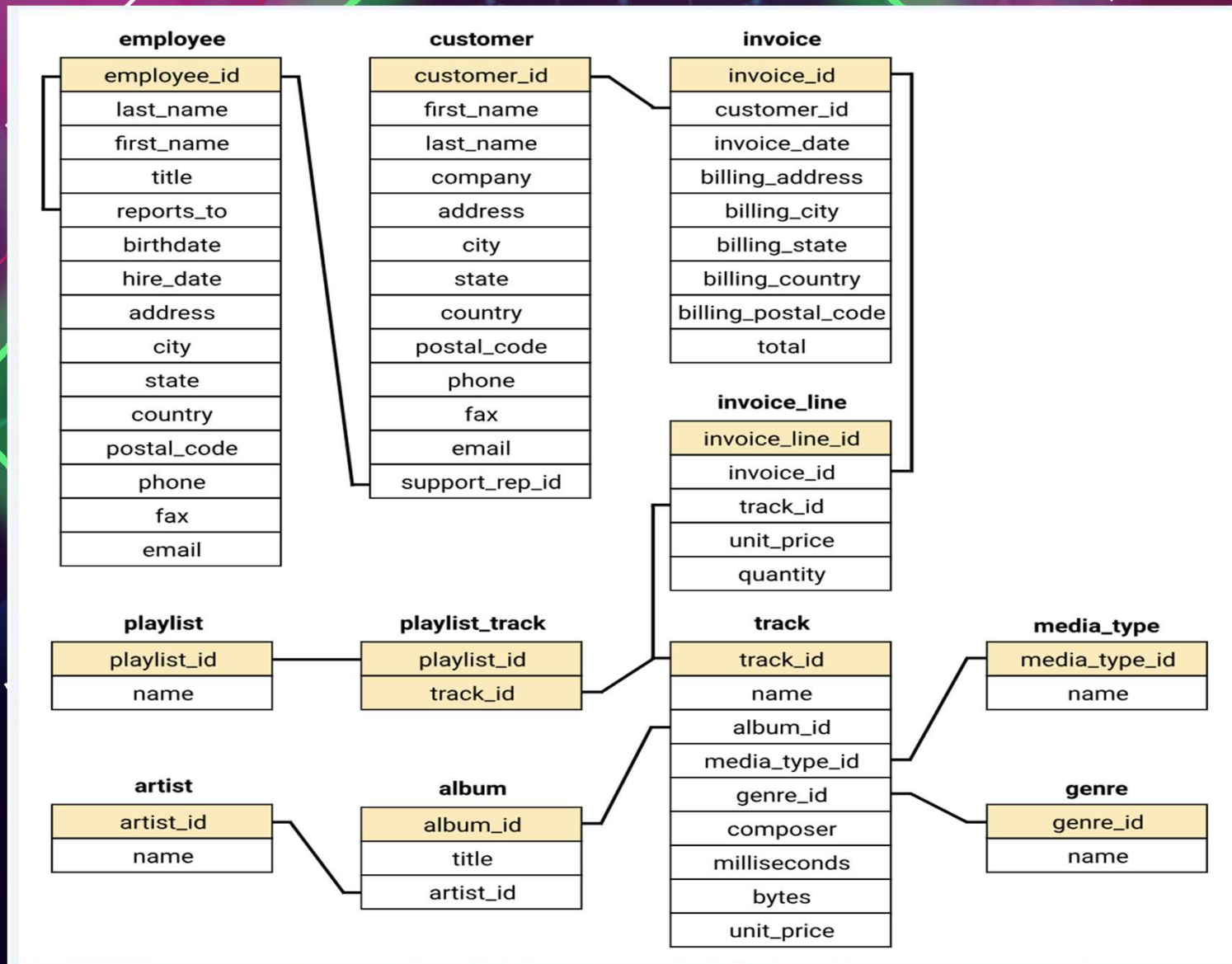
BY ANKITA KUMARI

PROBLEM STATEMENT

A music retailer aims to identify key factors for enhancing its business performance, diagnosing existing operational challenges, identifying high-value customer segments, and obtaining valuable insights for innovative marketing and initiatives.

We will be using SQL to solve these problems

SCHEMA



Query Query History

```
1  /* Music_Store_Analysis */
2
3  /* Q1: Who is the senior most employee based on job title? */
4
5  ✓ SELECT title, last_name, first_name
6     FROM employee
7     ORDER BY levels DESC
8     LIMIT 1;
9
```

Data Output Messages Notifications



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

Total rows: 1 of 1

Query complete 00:00:00.096

Query Query History

```
12
13  /* Q2: Which countries have the most Invoices? */
14
15  ✓ SELECT COUNT(*) AS c, billing_country
16     FROM invoice
17     GROUP BY billing_country
18     ORDER BY c DESC
19     LIMIT 1;
20
21
```

Data Output Messages Notifications



	c bigint	billing_country character varying (30)
1	131	USA

Total rows: 1 of 1

Query complete 00:00:00.089

Query Query History

```
21
22  /* Q3: What are top 3 values of total invoice? */
23
24  ✓ SELECT total
25     FROM invoice
26     ORDER BY total DESC
27     LIMIT 3 ;
28
29
```

Data Output Messages Notifications



	total double precision	🔒
1	23.759999999999998	
2		19.8
3		19.8

Total rows: 3 of 3 | Query complete 00:00:00.096

Query Query History

```

32  ✓ /* Q4: Which city has the best customers? We would like to throw a promotional Music Festival in the
33      city we made the most money.
34      Write a query that returns one city that has the highest sum of invoice totals.
35      Return both the city name & sum of all invoice totals */
36
37  ✓ SELECT billing_city,SUM(total) AS InvoiceTotal
38      FROM invoice
39      GROUP BY billing_city
40      ORDER BY InvoiceTotal DESC
41      LIMIT 1;
42
43

```

Data Output Messages Notifications



	billing_city character varying (30) 🔒	invoicetotal double precision 🔒
1	Prague	273.240000000000007

Total rows: 1 of 1 | Query complete 00:00:00.142

Query Query History

```

43
44 ✓ /* Q5: Who is the best customer? The customer who has spent the most money will be declared the best customer.
45 Write a query that returns the person who has spent the most money.*/
46
47 ✓ SELECT customer.customer_id, first_name, last_name, SUM(total) AS total_spending
48 FROM customer
49 JOIN invoice ON customer.customer_id = invoice.customer_id
50 GROUP BY customer.customer_id
51 ORDER BY total_spending DESC
52 LIMIT 1;
53
54

```

Data Output Messages Notifications



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

Total rows: 1 of 1 Query complete 00:00:00.217

Query Query History

```

56
57 /* Question Set 2 - Moderate */
58
59 ✓ /* Q1: Write query to return the email, first name, last name, & Genre of all Rock Music listeners.
60 Return your list ordered alphabetically by email starting with A. */
61
62 ✓ SELECT distinct c.email, c.first_name, c.last_name
63 FROM customer c
64 join invoice i on i.customer_id = c.customer_id
65 join invoice_line li on li.invoice_id = i.invoice_id
66 where li.track_id in (
67     select t.track_id from track t
68     join genre g on g.genre_id = t.genre_id
69     where g.name = 'Rock'
70 )
71 order by c.email

```

Data Output Messages Notifications



	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

Total rows: 59 of 59 Query complete 00:00:00.101

Query Query History

```
73 v /* Q2: Let's invite the artists who have written the most rock music in our dataset.
74 Write a query that returns the Artist name and total track count of the top 10 rock bands. */
75
76 SELECT artist.artist_id, artist.name, COUNT(artist.artist_id) AS number_of_songs
77 FROM track
78 JOIN album ON album.album_id = track.album_id
79 JOIN artist ON artist.artist_id = album.artist_id
80 JOIN genre ON genre.genre_id = track.genre_id
81 WHERE genre.name LIKE 'Rock'
82 GROUP BY artist.artist_id
83 ORDER BY number_of_songs DESC
84 LIMIT 10;
85
```

Data Output Messages Notifications



	artist_id [PK] character varying (50)	name character varying (120)	number_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

Total rows: 10 of 10

Query complete 00:00:00.102

```

85
86
87 /* Q3: Return all the track names that have a song length longer than the average song length.
88 Return the Name and Milliseconds for each track. Order by the song length with the longest songs listed first.
89
90 select name, milliseconds from track
91 where milliseconds > (select avg(milliseconds) from track)
92 order by milliseconds desc
93
94

```

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
Total rows: 494 of 494 Query complete 00:00:00.105		

Query Query History

```

102 /* Q1: Find how much amount spent by each customer on artists? Write a query to return customer name,
103 artist name and total spent */
104
105 WITH best_selling_artist AS (
106     SELECT artist.artist_id AS artist_id, artist.name AS artist_name,
107     SUM(invoice_line.unit_price*invoice_line.quantity) AS total_sales
108     FROM invoice_line
109     JOIN track ON track.track_id = invoice_line.track_id
110     JOIN album ON album.album_id = track.album_id
111     JOIN artist ON artist.artist_id = album.artist_id
112     GROUP BY 1
113     ORDER BY 3 DESC
114     LIMIT 1
115 )
116 SELECT c.customer_id, c.first_name, c.last_name, bsa.artist_name, SUM(il.unit_price*il.quantity) AS amount_spent
117 FROM invoice i
118 JOIN customer c ON c.customer_id = i.customer_id
119 JOIN invoice_line il ON il.invoice_id = i.invoice_id
120 JOIN track t ON t.track_id = il.track_id
121 JOIN album alb ON alb.album_id = t.album_id
122 JOIN best_selling_artist bsa ON bsa.artist_id = alb.artist_id
123 GROUP BY 1,2,3,4
124 ORDER BY 5 DESC;
125

```

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

Total rows: 43 of 43 Query complete 00:00:00.106

Query

Query History

```

127  /* Q2: We want to find out the most popular music Genre for each country. We determine the most
128  popular genre as the genre with the highest amount of purchases. Write a query that returns each
129  country along with the top Genre. For countries where the maximum number of purchases is shared return
130  all Genres. */

133  WITH RECURSIVE
134  sales_per_country AS(
135      SELECT COUNT(*) AS purchases_per_genre, customer.country, genre.name, genre.genre_id
136      FROM invoice_line
137      JOIN invoice ON invoice.invoice_id = invoice_line.invoice_id
138      JOIN customer ON customer.customer_id = invoice.customer_id
139      JOIN track ON track.track_id = invoice_line.track_id
140      JOIN genre ON genre.genre_id = track.genre_id
141      GROUP BY 2,3,4
142      ORDER BY 2
143  ),
144  max_genre_per_country AS (SELECT MAX(purchases_per_genre) AS max_genre_number, country
145      FROM sales_per_country
146      GROUP BY 2
147      ORDER BY 2)

149  SELECT sales_per_country.*
150  FROM sales_per_country
151  JOIN max_genre_per_country ON sales_per_country.country = max_genre_per_country.country
152  WHERE sales_per_country.purchases_per_genre = max_genre_per_country.max_genre_number;
153

```

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

Total rows: 43 of 43

Query complete 00:00:00.106

Query Query History

```

155
156 ▾ /* Q3: Write a query that determines the customer that has spent the most on music for each country.
157 Write a query that returns the country along with the top customer and how much they spent.
158 For countries where the top amount spent is shared, provide all customers who spent this amount. */
159
160
161 ▾ WITH Customer_with_country AS (
162     SELECT customer.customer_id,first_name,last_name,billing_country,SUM(total) AS total_spending,
163     ROW_NUMBER() OVER(PARTITION BY billing_country ORDER BY SUM(total) DESC) AS RowNo
164     FROM invoice
165     JOIN customer ON customer.customer_id = invoice.customer_id
166     GROUP BY 1,2,3,4
167     ORDER BY 4 ASC,5 DESC)
168 SELECT * FROM Customer_with_country WHERE RowNo <= 1
169
170
171

```

Data Output Messages Notifications



	customer_id integer	first_name character	last_name character	billing_country character varying (30)	total_spending 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	60.389999999999999	1
5	1	Luís	Gonçalves	Brazil	108.89999999999998	1
6	3	François	Tremblay	Canada	99.99	1
7	57	Luis	Rojas	Chile	97.020000000000001	1

Total rows: 24 of 24 Query complete 00:00:00.165



THANK
YOU