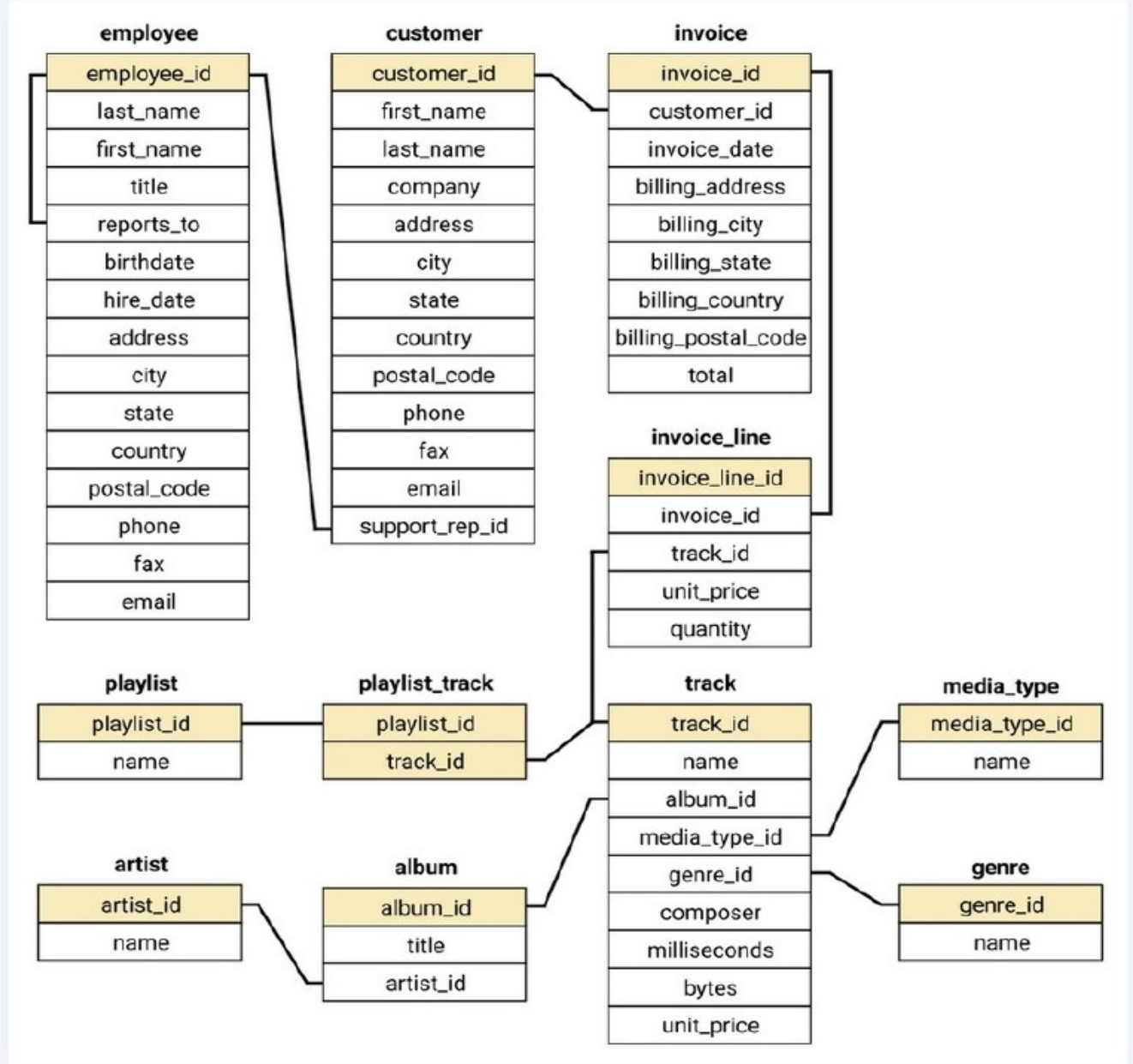
Music Store Sales Analysis
SQL case study

Created by
Tareq Al Hossain Tomal

Schema Diagram



Query | Query History

```sql
/* Question Set 1 */

/* Q1 : Who is the most senior employee based on job title? */


SELECT first_name, last_name, title, employee_id, Levels FROM employee
ORDER BY levels DESC
LIMIT 1;

```

Data Output | Messages | Notifications

	first_name character	last_name character	title character varying (50)	employee_id [PK] character varying (50)	levels character varying (10)
1	Mohan	Madan	Senior General Manager	9	L7

Query | Query History

```sql
/* Q2 : Which country has the most invoice? */

SELECT COUNT(*) AS total_invoice, billing_country FROM invoice
GROUP BY billing_country
ORDER BY total_invoice DESC
LIMIT 1;

```

Data Output | Messages | Notifications

	total_invoice bigint	billing_country character varying (30)
1	131	USA

Query | Query History

```
18
19  /* Q3 : What are top 3 values of total invoices?*/
20
21  SELECT total FROM invoice
22  ORDER BY total DESC
23  LIMIT 3;
24
25
```

Data Output | Messages | Notifications

	total double precision
1	23.759999999999998
2	19.8
3	19.8

```
26  /* Q4 : Which city has the best customers? We would like to throw a promotional Music Festival in the city
27        we made the most money. 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 customer.city, SUM(total) FROM customer
31  JOIN invoice ON customer.customer_id = invoice.customer_id
32  GROUP BY Customer.city
33  ORDER BY SUM(invoice.total) DESC
34  LIMIT 1;
```

Data Output | Messages | Notifications

	city character varying (50)	sum double precision
1	Prague	273.24000000000007

Query | Query History

```
38  /* Q5 : Who is the best customer? The customer who has spent the most money will be declared the best customer.
39  Write a query that returns the person who has spent the most money */
40
41  SELECT customer.customer_id, CONCAT(customer.first_name, customer.last_name), SUM(invoice.total) as total
42  FROM customer
43  LEFT JOIN invoice
44  ON customer.customer_id = invoice.customer_id
45  group by customer.customer_id
46  order by total desc
47  LIMIT 1;
48
49
```

Data Output | Messages | Notifications

	customer_id [PK] integer	concat text	total double precision
1	5	R Madhav ...	144.54000000000002

Query | Query History

```
50                          /* Question Set 2 */
51
52  /* Q1: Write query to return the email, first name, last name & genre of all Rock Music listeners.
53  Return your list ordered alphabatically by email starting with A. */
54
55  SELECT DISTINCT first_name, last_name, email
56  FROM customer
57  JOIN invoice ON customer.customer_id = invoice.customer_id
58  JOIN invoice_line ON invoice.invoice_id = invoice_line.invoice_id
59  WHERE track_id IN(
60      SELECT track_id
61      FROM track
62      JOIN genre
63      ON genre.genre_id = track.genre_id
64      WHERE genre.Name LIKE 'Rock')
65  ORDER BY email
66
```

Data Output | Messages | Notifications

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

Total rows: 59 of 59 | Query complete 00:00:00.078 | Ln 61, Col 12

```sql
/* 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*/

SELECT artist.name, COUNT(track_id) AS number_of_songs FROM artist
JOIN album ON artist.artist_id = album.artist_id
JOIN track ON album.album_id = track.album_id
WHERE track_id IN
    (SELECT track_id
    FROM track
    JOIN genre ON track.genre_id = genre.genre_id
    WHERE genre.name LIKE 'Rock')
GROUP BY artist.name
ORDER BY number_of_songs desc
LIMIT 10
```

	name character varying (120)	number_of_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
8	The Rolling Stones	41
9	Creedence Clearwater Revival	40
10	Kiss	35

```sql
/* 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 sogs listed first*/

SELECT name, milliseconds FROM track
WHERE milliseconds > (
                    SELECT AVG(milliseconds) AS avg_track_length
                    FROM track
                    )
ORDER BY milliseconds DESC
```

	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	2926593
11	The Long Patrol	2925008
12	The Magnificent Warriors	2924716
13	The Living Legend, Pt. 1	2924507
14	The Gun On Ice Planet Zero, Pt. 2	2924341
15	The Hand of God	2924007
16	Experiment In Terra	2923548
17	War of the Gods, Pt. 2	2923381
18	The Living Legend, Pt. 2	2923298

Query | Query History

```sql
                            /* Question Set 3 */

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

WITH best_selling_artist AS(
    SELECT artist.artist_id AS artist_id, artist.name AS artist_name,
    SUM(invoice_line.unit_price * invoice_line.quantity) AS total_spent
    FROM invoice_line
    JOIN track ON invoice_line.track_id = track.track_id
    JOIN album ON track.album_id = album.album_id
    JOIN artist ON album.artist_id = artist.artist_id
    GROUP BY artist.artist_id
    ORDER BY total_spent DESC
    LIMIT 1)

SELECT customer.customer_id, customer.first_name, customer.last_name, best_selling_artist.artist_name,
SUM(invoice_line.unit_price * invoice_line.quantity) AS amount_spent
FROM customer
JOIN invoice ON customer.customer_id = invoice.customer_id
JOIN invoice_line ON invoice.invoice_id = invoice_line.invoice_id
JOIN track ON invoice_line.track_id = track.track_id
JOIN album ON track.album_id = album.album_id
JOIN best_selling_artist ON album.artist_id = best_selling_artist.artist_id
GROUP BY 1,2,3,4
ORDER BY 5 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
19	36	Hannah	Schneider	Queen	1.98
20	11	Alexandre	Rocha	Queen	1.98
21	8	Daan	Peeters	Queen	1.98
22	42	Wyatt	Girard	Queen	1.98
23	44	Terhi	Hämäläinen	Queen	1.98
24	1	Luís	Gonçalves	Queen	1.98
25	48	Johannes	Van der Berg	Queen	1.98
26	49	Stanisław	Wójcik	Queen	1.98
27	52	Emma	Jones	Queen	1.98
28	57	Luis	Rojas	Queen	1.98

Query | Query History

```sql
/* 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. */

WITH popular_genre AS(

SELECT COUNT(invoice_line.quantity) AS purchase, customer.country, genre.genre_id, genre.name,
ROW_NUMBER() OVER(PARTITION BY customer.country ORDER BY COUNT(invoice_line.quantity) DESC) AS row_number
FROM invoice_line
JOIN invoice ON invoice_line.invoice_id = invoice.invoice_id
JOIN customer ON invoice.customer_id = customer.customer_id
JOIN track ON invoice_line.track_id = track.track_id
JOIN genre ON track.genre_id = genre.genre_id

GROUP BY 2,3,4
ORDER BY 2 ASC, 1 DESC)

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

Data Output | Messages | Notifications

	purchase bigint	country character varying (50)	genre_id character varying (50)	name character varying (120)	row_number bigint
1	17	Argentina	4	Alternative & Punk	1
2	34	Australia	1	Rock	1
3	40	Austria	1	Rock	1
4	26	Belgium	1	Rock	1
5	205	Brazil	1	Rock	1
6	333	Canada	1	Rock	1
7	61	Chile	1	Rock	1
8	143	Czech Republic	1	Rock	1

Query | Query History

```
146
147 /* Q3: Write a query that determines the customer that has spent the most on music for each country.
148 Write a query that returns the country along with the top customer and how much they spent.
149 For countries where the top amount spent is shared, provide all customers who spent this amount*/
150
151 WITH customer_with_country AS
152
153 (SELECT customer.customer_id, first_name, last_name, billing_country, SUM(total) AS total_spending,
154 ROW_NUMBER () OVER(PARTITION BY billing_country ORDER BY SUM(total) DESC) AS RowNo
155 FROM invoice
156 JOIN customer ON invoice.customer_id = customer.customer_id
157 GROUP BY 1,2,3,4
158 ORDER BY 4 ASC, 5 DESC)
159
160 SELECT * FROM customer_with_country
161 WHERE RowNo <=1
162
```

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	60.38999999999999	1
5	1	Luís	Gonçalves	Brazil	108.89999999999998	1
6	3	François	Tremblay	Canada	99.99	1
7	57	Luis	Rojas	Chile	97.02000000000001	1
8	5	R	Madhav	Czech Republic	144.54000000000002	1
9	9	Kara	Nielsen	Denmark	37.61999999999999	1
10	44	Terhi	Hämäläinen	Finland	79.2	1
11	42	Wyatt	Girard	France	99.99	1
12	37	Fynn	Zimmermann	Germany	94.05000000000001	1