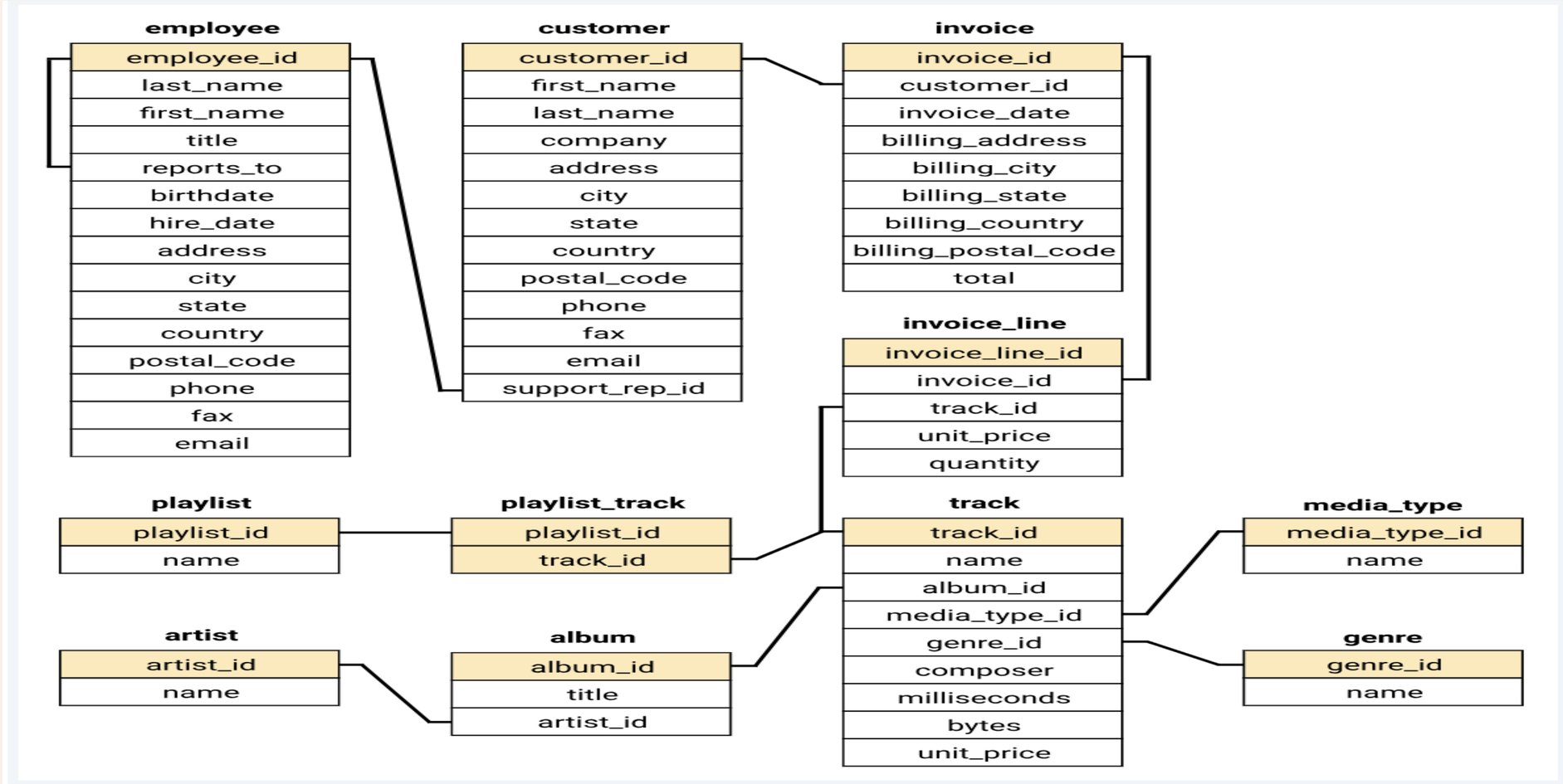
MUSIC STORE ANALYSIS

Table of Contents

OBJECTIVE

Analyze the music playlist database and help the store understand its business growth by providing answers to the provided questions.

DATABASE SCHEMA



Who is the senior most employee based on job title?

```
7
8
9   --1. Who is the senior most employee based on job title?
10  select * from employee
11  order by levels desc
12  limit 1;
13
```

Data Output | Messages | Notifications

	employee_id [PK] character varying (50)	last_name character	first_name character	title character varying (50)	reports_to character varying (30)	levels character varying (10)	birthdate timestamp without time zone	hire_date timestamp
1	9	Madan	Mohan	Senior General Manager	[null]	L7	1961-01-26 00:00:00	2016-01-14

Which countries have the most Invoices?

```
13
14  --2. Which countries have the most Invoices?
15  select count(*), billing_country from invoice
16  group by billing_country
17  order by count(*) desc
18  limit 1;
19
```

Data Output | Messages | Notifications

	count bigint	billing_country character varying (30)
1	131	USA

What are top 3 values of total invoice?

```
--3. What are top 3 values of total invoice?
select invoice_id, total from invoice
order by total desc
limit 3;
```

Data Output | Messages | Notifications

	invoice_id [PK] integer	total double precision
1	183	23.759999999999998
2	92	19.8
3	31	19.8

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

```
--4. Which city has the best customers? We would like to throw a promotional Music Festival in the city we made the most mo
select billing_city, sum(total) from invoice
group by billing_city
order by sum(total) desc
limit 1;
```

Data Output | Messages | Notifications

	billing_city character varying (30)	sum double precision
1	Prague	273.24000000000007

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

```
34
35  --5. Who is the best customer? The customer who has spent the most money will be declared the best customer. Write a query
36  select cu.first_name, cu.last_name, sum(i.total) from customer cu join invoice i on cu.customer_id=i.customer_id
37  group by cu.first_name, cu.last_name
38  order by sum(i.total) desc
39  limit 1
40
```

Data Output | Messages | Notifications

	first_name character	last_name character	sum double precision
1	R ...	Madhav	144.54000000000002

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 | Query History

Scratch Pad

```
--Write query to return the email, first name, last name, & Genre of all Rock Music listeners. Return your list ordered alp
select DISTINCT cu.email, cu.first_name, cu.last_name from customer cu
JOIN invoice i on cu.customer_id=i.customer_id
JOIN invoice_line il on i.invoice_id=il.invoice_id
JOIN track tr on tr.track_id=il.track_id
JOIN genre ge on ge.genre_id= tr.genre_id
WHERE ge.name='Rock'
ORDER BY cu.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
9	dominiquelefebvre@gmail.c...	Dominique	Lefebvre
10	edfrancis@yachoo.ca	Edward	Francis
11	eduardo@woodstock.com.br	Eduardo	Martins
12	ellie.sullivan@shaw.ca	Ellie	Sullivan
13	emma_jones@hotmail.com	Emma	Jones

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 | Query History | Scratch Pad

```
51  --Let's invite the artists who have written the most rock music in our dataset. Write a query that returns the Artist name
52  SELECT a.artist_id, a.name,COUNT(a.artist_id) AS num_of_songs from artist a
53  JOIN album al on al.artist_id=a.artist_id
54  JOIN track tr on tr.album_id=al.album_id
55  JOIN genre ge on ge.genre_id=tr.genre_id
56  WHERE ge.name='Rock'
57  GROUP BY a.artist_id
58  ORDER BY num_of_songs desc
59  LIMIT 10;
```

Data Output | Messages | Notifications

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

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

```
--Return all the track names that have a song length longer than the average song length.  Return the Name and Milliseconds
SELECT name, milliseconds
FROM track
WHERE milliseconds > (
    select AVG(milliseconds) from track
)
ORDER BY milliseconds desc;
```

	name character varying (150)	milliseconds integer
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

Total rows: 494 of 494 | Query complete 00:00:00.097 | Ln 53, Col 1

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

Query | Query History | Scratch Pad

```
--Find how much amount spent by each customer on top selling artist? Write a query to return customer name, artist name and
with top_selling_artist AS (
    SELECT a.artist_id as artist_id, a.name as artist_name, SUM(il.unit_price * il.quantity) as total_spent
    FROM artist a
    JOIN album al on al.artist_id=a.artist_id
    JOIN track tr on tr.album_id=al.album_id
    JOIN invoice_line il on il.track_id=tr.track_id
    group by a.artist_id
    order by total_spent desc
    limit 1
)
SELECT c.customer_id, c.first_name, c.last_name, tsa.artist_name, SUM(il.unit_price * il.quantity) as total_spent
FROM invoice i
JOIN customer c ON c.customer_id = i.customer_id
JOIN invoice_line il ON il.invoice_id = i.invoice_id
JOIN track t ON t.track_id = il.track_id
JOIN album alb ON alb.album_id = t.album_id
JOIN top_selling_artist tsa ON tsa.artist_id = alb.artist_id
GROUP BY c.customer_id, c.first_name, c.last_name, tsa.artist_name
ORDER BY c.customer_id;
```

Data Output | Messages | Notifications

	customer_id integer	first_name character	last_name character	artist_name character varying (120)	total_spent double precision
1	1	Luís	Gonçalves	Queen	1.98
2	3	François	Tremblay	Queen	17.82
3	5	R	Madhav	Queen	3.96
4	6	Helena	Holý	Queen	0.99

Total rows: 43 of 43 | Query complete 00:00:00.072 | Ln 81, Col 24

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

```
82
83  --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 pur
84  WITH popular_genre AS
85  (
86      SELECT COUNT(il.quantity) AS purchases, c.country, g.name, g.genre_id,
87      RANK() OVER(PARTITION BY c.country ORDER BY COUNT(il.quantity) DESC) AS Rank_num
88      FROM invoice_line il
89      JOIN invoice i ON i.invoice_id = il.invoice_id
90      JOIN customer c ON c.customer_id = i.customer_id
91      JOIN track t ON t.track_id = il.track_id
92      JOIN genre g ON g.genre_id = t.genre_id
93      GROUP BY c.country, g.name, g.genre_id
94      ORDER BY c.country ASC, count(il.quantity) DESC
95  )
96  SELECT country, name, purchases FROM popular_genre WHERE Rank_num <= 1
97
```

Data Output | Messages | Notifications

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

Total rows: 24 of 24 | Query complete 00:00:00.095 | Ln 84, Col 1

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

music_database/postgres@PostgreSQL 16

No limit

Query | Query History

```
87  WITH RECURSIVE
88      customter_spend AS (
89          SELECT customer.customer_id,first_name,last_name, country,SUM(total) AS total_spend
90          FROM invoice
91          JOIN customer ON customer.customer_id = invoice.customer_id
92          GROUP BY customer.customer_id,first_name,last_name, country
93          ORDER BY 2,3 DESC),
94
95      max_spending AS(
96          SELECT country, MAX(total_spend) AS max_spending
97          FROM customter_spend
98          GROUP BY country)
99
100 SELECT cs.country, cs.first_name, cs.last_name, cs.total_spend, cs.customer_id
101 FROM customter_spend cs
102 JOIN max_spending ms
103 ON cs.country = ms.country
104 WHERE cs.total_spend = ms.max_spending
105 ORDER BY cs.country;
```

Data Output | Messages | Notifications

	country character varying (50)	first_name character	last_name character	total_spend double precision	customer_id [PK] integer
1	Argentina	Diego	Gutiérrez	39.6	56
2	Australia	Mark	Taylor	81.18	55
3	Austria	Astrid	Gruber	69.3	7
4	Belgium	Daan	Peeters	60.38999999999999	8
5	Brazil	Luís	Gonçalves	108.89999999999998	1

Total rows: 24 of 24 | Query complete 00:00:00.112 | Ln 105, Col 21

Thank You