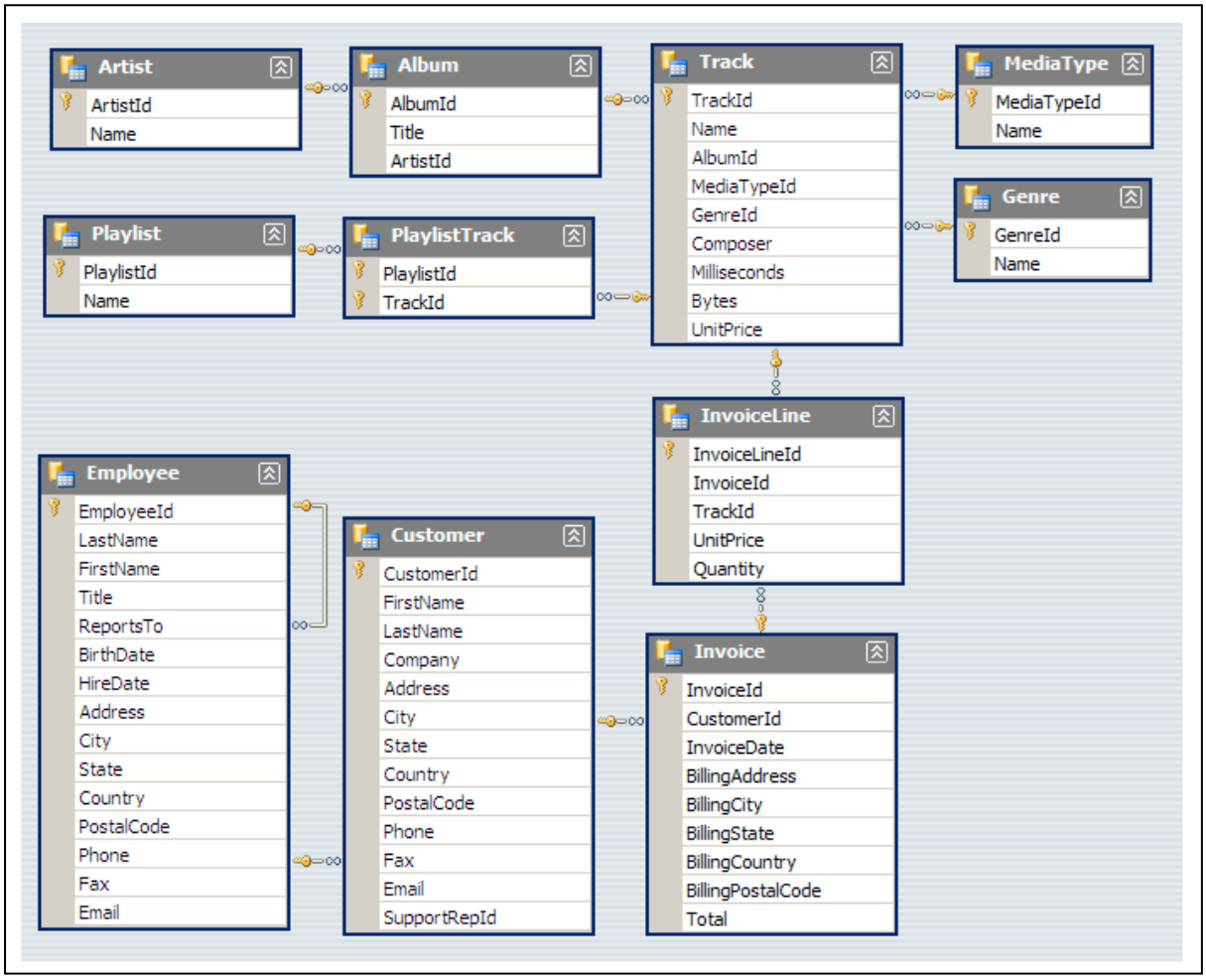
Music Store Analysis

Schema



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

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

SELECT employee_id,first_name,last_name,title,levels
FROM employee
Where levels= (Select Max(Levels) FROM employee)
```

Data Output | Messages | Notifications

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

Q2. Which TOP 5 Countries have the most Invoices?

```
7   --Q2. Which TOP 5 Countries have the most Invoices?
8
9   SELECT * FROM invoice;
10
11  SELECT billing_country,COUNT(total) AS Most_Invoices
12  FROM invoice
13  GROUP BY billing_country
14  ORDER BY Most_Invoices DESC
15  LIMIT 5;
```

Data Output | Messages | Notifications

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

Q3. What are top 3 values of total invoices?

```
17  --Q3.What are top 3 values of total invoices?
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

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.

```
--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

SELECT * FROM invoice;
SELECT billing_city, SUM(total) AS Billing_Total
FROM invoice
GROUP BY billing_city
ORDER BY Billing_Total DESC
LIMIT 5
```

Data Output | Messages | Notifications

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

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.

```
--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


SELECT customer.first_name, customer.last_name, SUM(total) AS Total_Money_spent
FROM customer
INNER JOIN invoice ON
customer.customer_id=invoice.customer_id
GROUP BY first_name,last_name
ORDER BY Total_Money_spent DESC
LIMIT 1
```

Data Output | Messages | Notifications

	first_name character	last_name character	total_money_spent double precision
1	R …	Madhav	144.54000000000002

Q6. 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.

```
--Q6. 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.

SELECT DISTINCT email,first_name,last_name,genre.name AS Genre
FROM genre
INNER JOIN Track ON
genre.genre_id=Track.genre_id
INNER JOIN invoice_line ON
Track.track_id=invoice_line.track_id
INNER JOIN invoice ON
invoice.invoice_id=invoice_line.invoice_id
INNER JOIN customer ON
customer.customer_id=invoice.customer_id
WHERE genre.name='Rock'
ORDER BY email ASC
```

Data Output | Messages | Notifications

	email character varying (50)	first_name character	last_name character	genre 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
6	daan_peeters@apple.be	Daan	Peeters	Rock
7	diego.gutierrez@yahoo.ar	Diego	Gutiérrez	Rock

Total rows: 59 of 59 | Query complete 00:00:00.057

Q7. 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.

```
--Q7. 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.artist_id, artist.name,COUNT(track.track_id)
FROM artist
INNER JOIN album ON artist.artist_id=album.artist_id
INNER JOIN track ON album.album_id=track.album_id
INNER JOIN genre ON track.genre_id=genre.genre_id
WHERE genre.name='Rock'
GROUP BY artist.artist_id, artist.name
ORDER BY COUNT(artist.artist_id) DESC
LIMIT 10
```

Data Output | Messages | Notifications

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

Q8. 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.

```sql
--Q8. 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.


SELECT name,milliseconds AS song_lengths
FROM track
Where milliseconds > (Select Avg(milliseconds) AS Average_song_length FROM track)
ORDER BY milliseconds DESC
```

Data Output | Messages | Notifications

	name character varying (150)	song_lengths 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

Total rows: 494 of 494 Query complete 00:00:00.099

Q9. Find how much amount spent by each customer on artists? Write a query to return customer name, artist name and total spent.

```sql
--Q9. 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_sales
    FROM invoice_line
    INNER JOIN track ON invoice_line.track_id=track.track_id
    INNER JOIN album ON track.album_id=album.album_id
    INNER JOIN artist ON album.artist_id=artist.artist_id
    GROUP BY artist.artist_id
    ORDER BY total_sales 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 invoice
INNER JOIN customer ON invoice.customer_id=customer.customer_id
INNER JOIN invoice_line ON invoice_line.invoice_id=invoice.invoice_id
INNER JOIN track ON invoice_line.track_id=track.track_id
INNER JOIN album ON track.album_id=album.album_id
INNER JOIN best_selling_artist ON best_selling_artist.artist_id=album.artist_id
GROUP BY customer.customer_id, customer.first_name,customer.last_name,best_selling_artist.artist_name
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

Total rows: 43 of 43 Query complete 00:00:00.103

Q10. 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 he maximum number of purchases is shared return all Genres.

```sql
--Q10. 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 purchases, customer.country,genre.name,genre.genre_id,
    ROW_NUMBER() OVER(PARTITION BY customer.country ORDER BY COUNT(invoice_line.quantity)DESC) AS Row_No
    FROM invoice_line
    INNER JOIN invoice ON invoice_line.invoice_id=invoice.invoice_id
    INNER JOIN customer ON customer.customer_id=invoice.customer_id
    JOIN track ON track.track_id = invoice_line.track_id
    JOIN genre ON genre.genre_id = track.genre_id
    GROUP BY customer.country,genre.name,genre.genre_id
    ORDER BY customer.country ASC, purchases DESC
)
SELECT * FROM popular_genre WHERE Row_No <= 1
```

Data Output | Messages | Notifications

	purchases bigint	country character varying (50)	name character varying (120)	genre_id character varying (50)	row_no 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

Total rows: 24 of 24 | Query complete 00:00:00.059 | Ln 13

Q11. 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

```sql
--Q11.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

WITH Customter_with_country AS (
        SELECT customer.customer_id,first_name,last_name,billing_country,SUM(total) AS total_spending,
        ROW_NUMBER() OVER(PARTITION BY billing_country ORDER BY SUM(total) DESC) AS RowNo
        FROM invoice
        JOIN customer ON customer.customer_id = invoice.customer_id
        GROUP BY customer.customer_id,first_name,last_name,billing_country
        ORDER BY billing_country ASC,total_spending DESC)
SELECT * FROM Customter_with_country WHERE RowNo <= 1
```

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

Total rows: 24 of 24 | Query complete 00:00:00.095