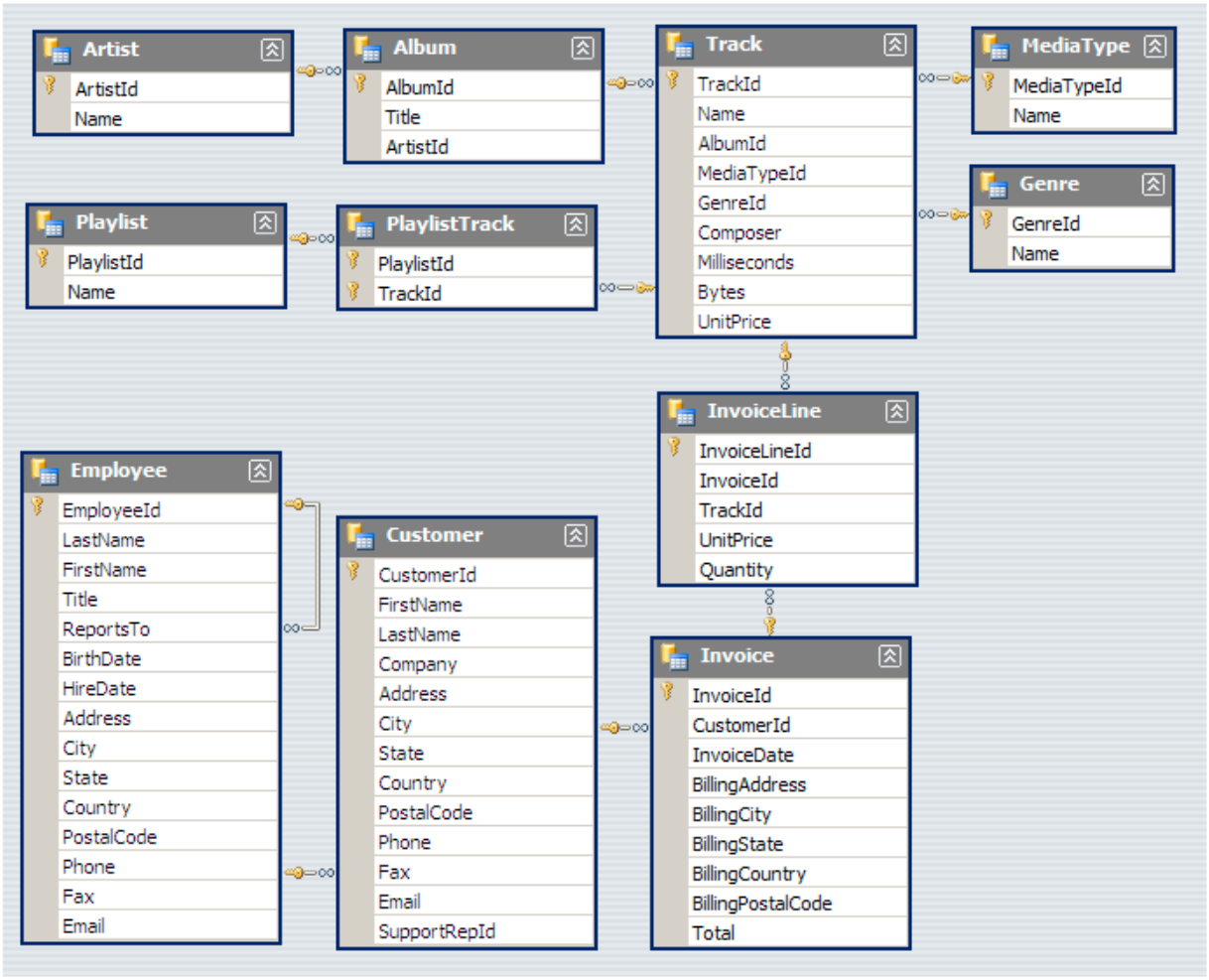
DIGITAL MUSIC STORE ANALYSIS PROJECT USING PostgreSQL

OBJECTIVE: Analyze the music store database and answer certain questions to help the store understand its business growth.

SCHEMA DIAGRAM: ***Below figure explains the connections between different tables of the databases using the foreign key.***



Queries answered:

Q-1: Senior most employee on basis of job title?

```
SELECT first_name, last_name, title FROM employee
WHERE title='Senior General Manager';
```

Q-2: Country having the most invoices?

```
SELECT billing_country,COUNT(invoice_id) FROM invoice
GROUP BY billing_country
ORDER BY COUNT(invoice_id) DESC
LIMIT 1;
```

Q-3: Top 3 values of total invoice?

```
SELECT total FROM invoice
ORDER BY total DESC
LIMIT 3;
```

Q-4: City having highest sum of invoice totals?

```
SELECT billing_city,SUM(total) FROM invoice
GROUP BY billing_city
ORDER BY SUM(total) DESC
LIMIT 1;
```

Q-5: Customer spending the most money?

```
SELECT first_name,last_name,customer_id FROM customer
WHERE customer_id IN (SELECT customer_id FROM invoice
                      GROUP BY customer_id
                      ORDER BY SUM(total) DESC
                      LIMIT 1)
```

Q-6: Customers listening to rock music?

```
SELECT DISTINCT first_name,last_name,email,genre.name FROM invoice
INNER JOIN customer
ON customer.customer_id=invoice.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 genre
ON genre.genre_id=track.genre_id
WHERE genre.name='Rock'
ORDER BY email
```

Q-7: Artist with most rock music?

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

Q-8: Track having length greater than average length?

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

Q-9: Query to return amount spent by each customer on artists?

```
SELECT
first_name,last_name,artist.name,SUM(quantity*invoice_line.unit_Price)  FROM
customer
JOIN invoice ON
customer.customer_id=invoice.customer_id
JOIN invoice_line ON
invoice_line.invoice_id=invoice.invoice_id
JOIN track ON
track.track_id=invoice_line.track_id
JOIN album ON
album.album_id=track.album_id
JOIN artist ON
album.artist_id=artist.artist_id
GROUP BY first_name,last_name,artist.name
HAVING artist.name='Queen'
ORDER BY SUM(quantity*invoice_line.unit_Price)DESC
```

Q-10: Countries with their top genre (having highest amount of purchases)?

```
WITH temp AS (SELECT country,genre.name,COUNT(*),ROW_NUMBER() OVER
(PARTITION BY country ORDER BY COUNT(*) DESC)

AS r FROM customer

JOIN invoice ON

customer.customer_id=invoice.customer_id

JOIN invoice_line ON

invoice_line.invoice_id=invoice.invoice_id

JOIN track ON

track.track_id=invoice_line.track_id

JOIN genre ON

genre.genre_id=track.genre_id


GROUP BY country,genre.name

ORDER BY country,count DESC


)

SELECT * FROM temp

WHERE r=1;
```

Q-11: Customer that has spent the most on music in each country?

```
WITH c as (

SELECT first_name,last_name,country,
```

```
SUM(total),RANK() OVER(PARTITION BY country ORDER BY SUM(total) desc)
FROM customer

INNER JOIN invoice on customer.customer_id=invoice.customer_id

GROUP BY first_name,last_name,country

ORDER BY country


)

SELECT first_name,last_name,country FROM c

where rank=1;
```

CONCLUSION: Through the above different queries we have understood different scenarios like the regions where the sales are maximum, country wise favorite genre, categories of customers spending the most money in their country to better understand the customer preferences and focusing more on areas where sale is maximum and also on areas where sale is not so good so that focus can be made on those areas.