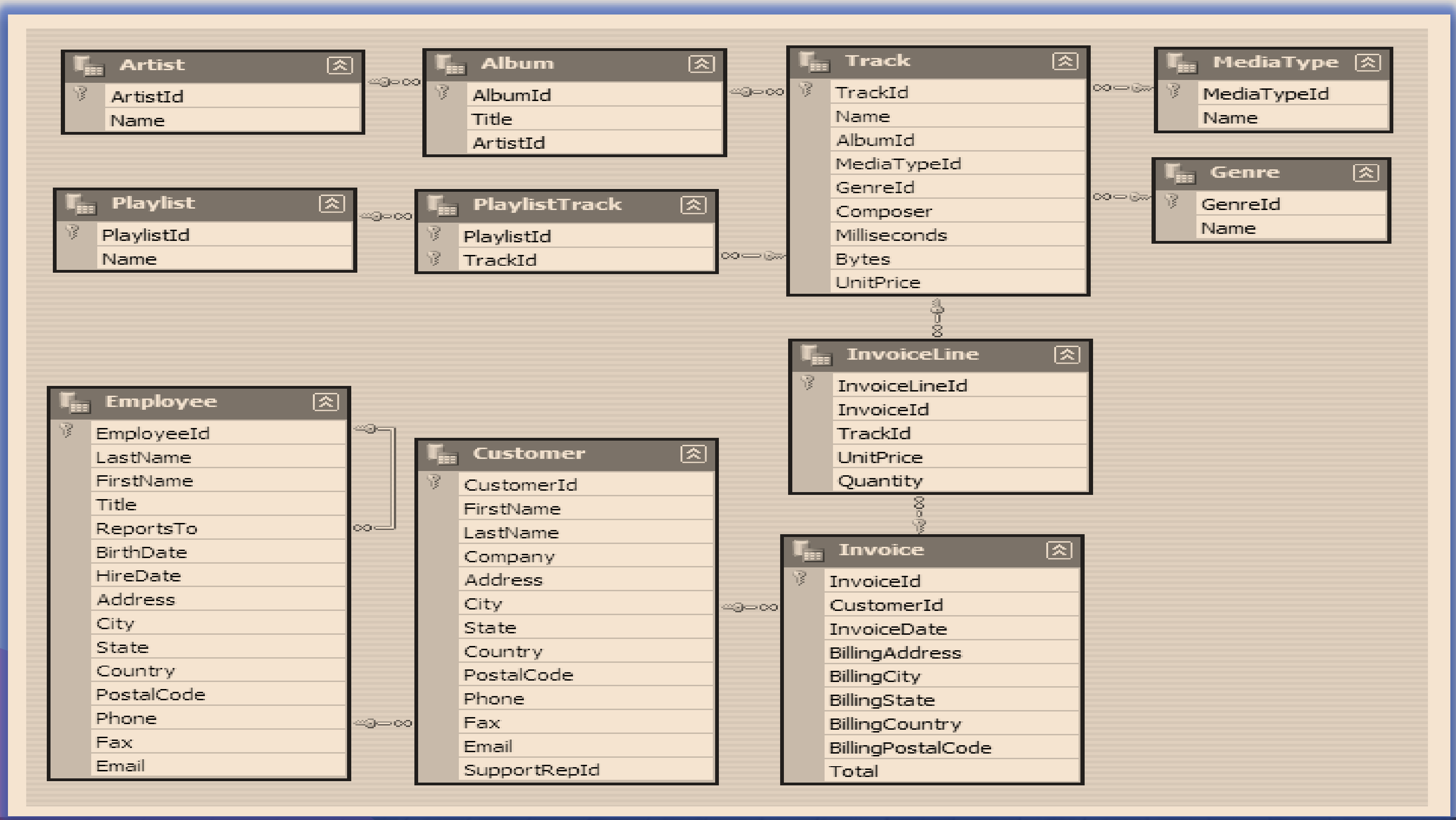
MUSIC STORE ANALYSIS-
SQL PROJECT

DESCRIPTION

- The Music Store Project is a data analysis initiative designed to extract valuable insights from a music store database using PostgreSQL. The project leverages the PgAdmin4 tool to query and analyze data related to employees, customers, invoices, tracks, albums, artists, and genres.The following descriptions outline the objectives and outcomes of the SQL queries developed for this project:

SCHEMA- MUSIC STORE DATABASE



Artist
- ArtistId
- Name

Album
- AlbumId
- Title
- ArtistId

Track
- TrackId
- Name
- AlbumId
- MediaTypeId
- GenreId
- Composer
- Milliseconds
- Bytes
- UnitPrice

MediaType
- MediaTypeId
- Name

Genre
- GenreId
- Name

Playlist
- PlaylistId
- Name

PlaylistTrack
- PlaylistId
- TrackId

InvoiceLine
- InvoiceLineId
- InvoiceId
- TrackId
- UnitPrice
- Quantity

Employee
- EmployeeId
- LastName
- FirstName
- Title
- ReportsTo
- BirthDate
- HireDate
- Address
- City
- State
- Country
- PostalCode
- Phone
- Fax
- Email

Customer
- CustomerId
- FirstName
- LastName
- Company
- Address
- City
- State
- Country
- PostalCode
- Phone
- Fax
- Email
- SupportRepId

Invoice
- InvoiceId
- CustomerId
- InvoiceDate
- BillingAddress
- BillingCity
- BillingState
- BillingCountry
- BillingPostalCode
- Total

PROJECT ANALYSIS

--1. Who is the senior most employee based on job title?

Code:

```
SELECT *
FROM employee
ORDER BY levels desc
LIMIT 1;
```

--2. Which countries have the most Invoices?

Code:

```
SELECT COUNT(*) AS invoice, billing_country FROM invoice
GROUP BY billing_country
ORDER BY invoice DESC;
```

--3. What are top 3 values of total invoice?

Code:

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

--4. 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?

Code:

```
SELECT billing_city, SUM(total) as highest_total

FROM invoice

GROUP BY billing_city

ORDER BY highest_total DESC;
```

--5. 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?

Code:

```
SELECT c.customer_id, c.first_name,c.last_name,SUM(i.total) as total_invoice

FROM customer c

JOIN invoice i ON c.customer_id = i.customer_id

GROUP BY c.customer_id

ORDER BY total_invoice DESC

LIMIT 1;
```

--6. 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?

Code:

```
SELECT DISTINCT c.email,c.first_name,c.last_name FROM
Customer c
JOIN invoice i ON c.customer_id = i.customer_id
JOIN invoice_line il ON i.invoice_id =il.invoice_id
WHERE track_id IN
(
SELECT track_id FROM track
JOIN genre ON track.genre_id = genre.genre_id
WHERE genre.name LIKE 'Rock’
)
ORDER BY email;
```

--7. 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?

Code:

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

--8. 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?

Code:

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

--9. Find how much amount spent by each customer on artists? Write a query to return customer name, artist name and total spent?

Code:

```
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
        JOIN track ON invoice_line.track_id = track.track_id
        JOIN album alb ON alb.album_id = track.album_id
        JOIN artist ON artist.artist_id = alb.artist_id
        GROUP  BY 1
        ORDER BY 3 DESC LIMIT 1)
SELECT c.customer_id, c.first_name, c.last_name, bsa.artist_name,
SUM(il.unit_price* il.quantity) as amount_spent
from invoice i
```

```
JOIN customer c ON i.customer_id = c.customer_id

JOIN invoice_line il ON il.invoice_id = i.invoice_id

JOIN track ON il.track_id = track.track_id

JOIN album alb ON alb.album_id = track.album_id

JOIN best_selling_artist bsa ON bsa.artist_id = alb.artist_id

GROUP BY 1,2, 3, 4

ORDER BY 5 DESC;
```

--10. 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?

Code:

```
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)
RowNo
        FROM invoice_line
JOIN invoice  ON invoice_line.invoice_id = invoice.invoice_id
JOIN customer ON customer.customer_id = invoice.customer_id
JOIN track  ON track.track_id = invoice_line.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 RowNo<=1;
```

--11.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?

Code:

```
with customer_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 1,2,3,4
ORDER BY 4 ASC,5 DESC)
SELECT * FROM customer_with_country
WHERE RowNo <= 1;
```

THANK YOU

SANGEETA YADAV

Sangeeta.yadavvv@gmail.com