# Music Store Analysis

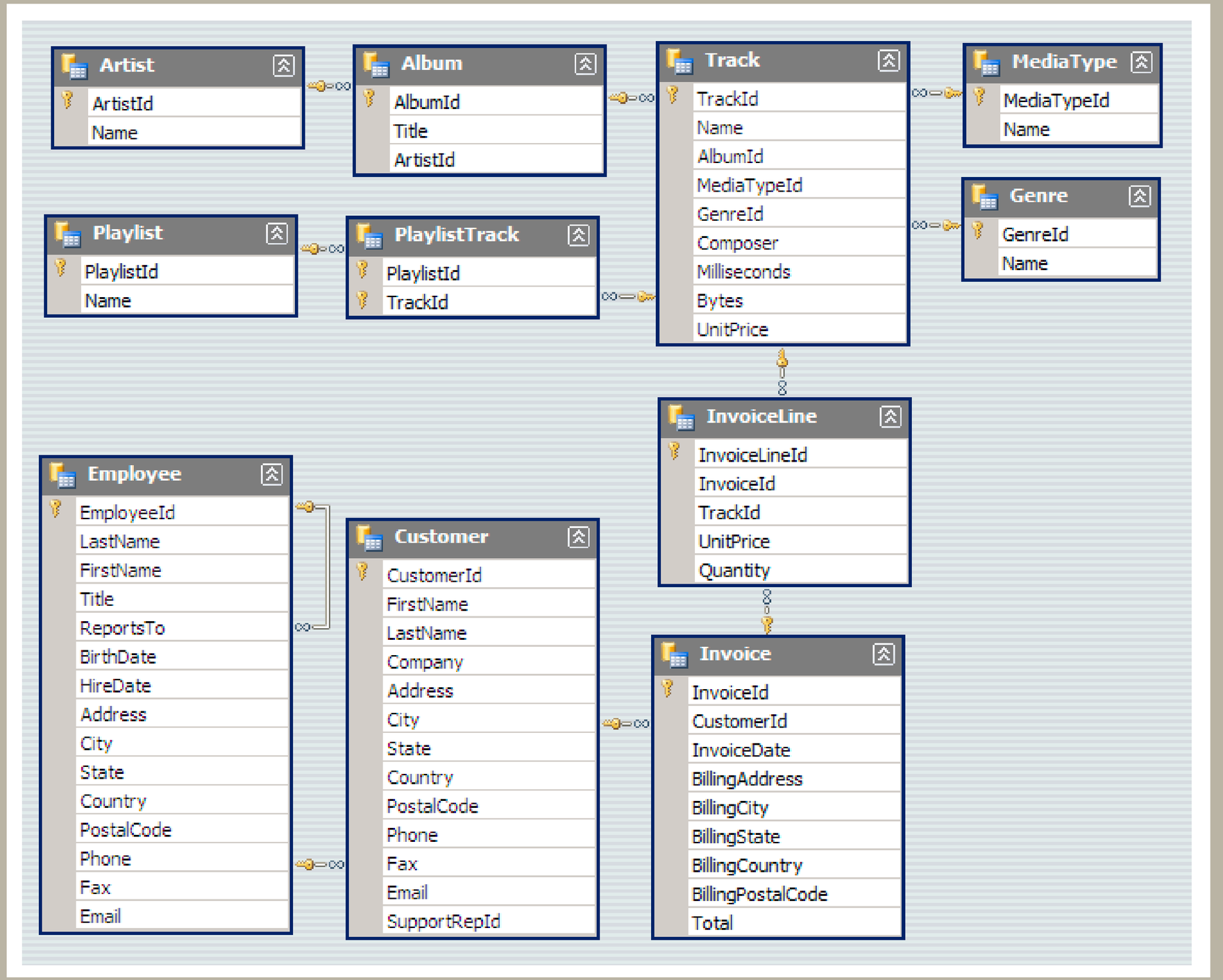
SQL Project

INTRODUCTION

# My name is vidhi jain. In this project, i have utilize SQL query to solve questions that were related to music store....

# OBJECTIVE

This music store wants to analyze its music playlist database to understand its business growth.


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

# Who is the senior most employee based on job title ?

```sql
select * from employee
order by levels desc
limit 1;
```

# Which countries have the most invoice ?

```sql
select billing_country, count(*) as total_invoice
from invoice
group by billing_country
order by total_invoice desc
```

# What are top 3 values of total invoice?

```
select total from invoice
order by total desc
limit 3
```

Who is the best customer? The customer who has spent the money will be declared the best customer. Write a query that returns the person who spent the most money.

```sql
select customer.customer_id, customer.first_name,customer.last_name,
sum(invoice.total) as total
from customer
join invoice on customer.customer_id = invoice.customer_id
group by customer.customer_id
order by total desc
limit 1;
```

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 and sum of all invoice totals.

```sql
select billing_city, sum(total) as c
from invoice
group by billing_city
order by c desc
limit 1;
```

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.

```sql
select artist.artist_id, artist.name, count(artist.artist_id) as number_of_songs
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 number_of_songs desc
limit 10;
```

# 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.

```sql
select distinct email, first_name, last_name
from customer
join invoice on customer.customer_id = invoice.customer_id
join invoice_line on invoice.invoice_id = invoice_line.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;
```

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
select name, milliseconds
from track
where milliseconds > (
    select avg(milliseconds) as avg_track_length
    from track)
order by milliseconds desc;
```

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

```sql
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 track.track_id = invoice_line.track_id
    join album on album.album_id = track.album_id
    join artist on artist.artist_id = album.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 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 best_selling_artist bsa on bsa.artist_id = alb.artist_id
group by 1,2,3,4
order by 5 desc;
```

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.

```sql
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 rowNo
    from invoice_line
    join invoice on invoice.invoice_id = invoice_line.invoice_id
    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 2,3,4
    order by 2 asc, 1 desc
)
select * from popular_genre where rowNo <= 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 is shared, provide all customers who spent this amount.

```
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...