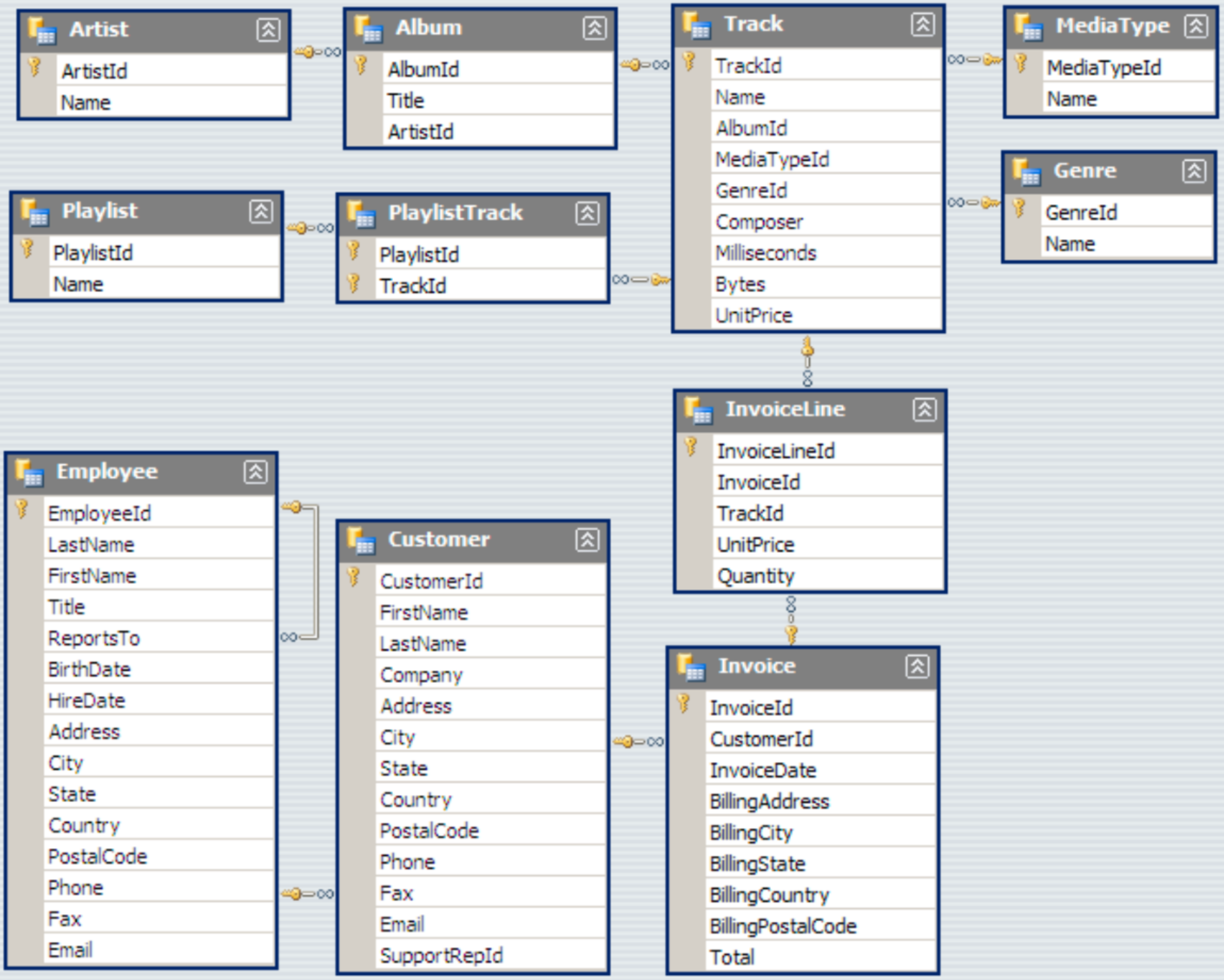
Digital Music Store Analysis

- Analysis Project using SQL
- Easy to Advance Level

Objective

The objective of this project is to teach beginners how to analyze a music playlist database using SQL and to help the store understand its business growth by answering simple questions using the insights gained from the data analysis.

Schema


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

Question

Set -1

Easy Level

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

sol:
    select * from employee
    order by levels desc
    limit 1;

-- Q2: Which country have the more Invoices?

sol:
    select count(*) as c, billing_country from invoice
    group by billing_country
    order by c desc
    limit 1;

-- Q3: What are the top 3 values to total invoice?

sol:
    select total from invoice
    order by total desc
    limit 3;
```

```
-- Q4: Which city has the best customer?
--     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.

sol:
    select sum(total) as Invoice_total, billing_city from invoice
    group by billing_city
    order by Invoice_total desc
    limit 1;
```

```
-- Q5: Who is the best customer? The customer who has spend the
--     most money will be declared the best customer.
--     Write a query that return the person who had spend the most money.

sol:
    select a.customer_id,a.first_name,a.last_name,a.city,
    sum(b.total) as amt_spend from customer a
    join invoice b on a.customer_id= b.customer_id
    group by a.customer_id
    order by amt_spend desc
    limit 1;
```

Question

Set -2

Medium Level

```sql
-- Q1: Write query to return the email,first name,last name,& Genre
--     of all Rock Music listener.
--     Return your list ordered alphabetically by email starting with A.

sol:
select a.email,a.first_name,a.last_name from customer a
join invoice b on a.customer_id=b.customer_id
join invoice_line c on b.invoice_id = c.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;
```

```
-- Q2: Lets 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.

sol:
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;
```

```
-- Q3: 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.

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

Question

Set -3

Hard Level

```sql
-- Q1: Find how much amount spent by each customer on artists?
--     write a query to return customer name,artist name and total spend
sol:
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 2
)
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;
```

```sql
-- Q2: 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

sol:
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
```

```sql
-- Q3: Write a query that determines the customer that has spend 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.

sol:
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
```