

Music store Data analysis

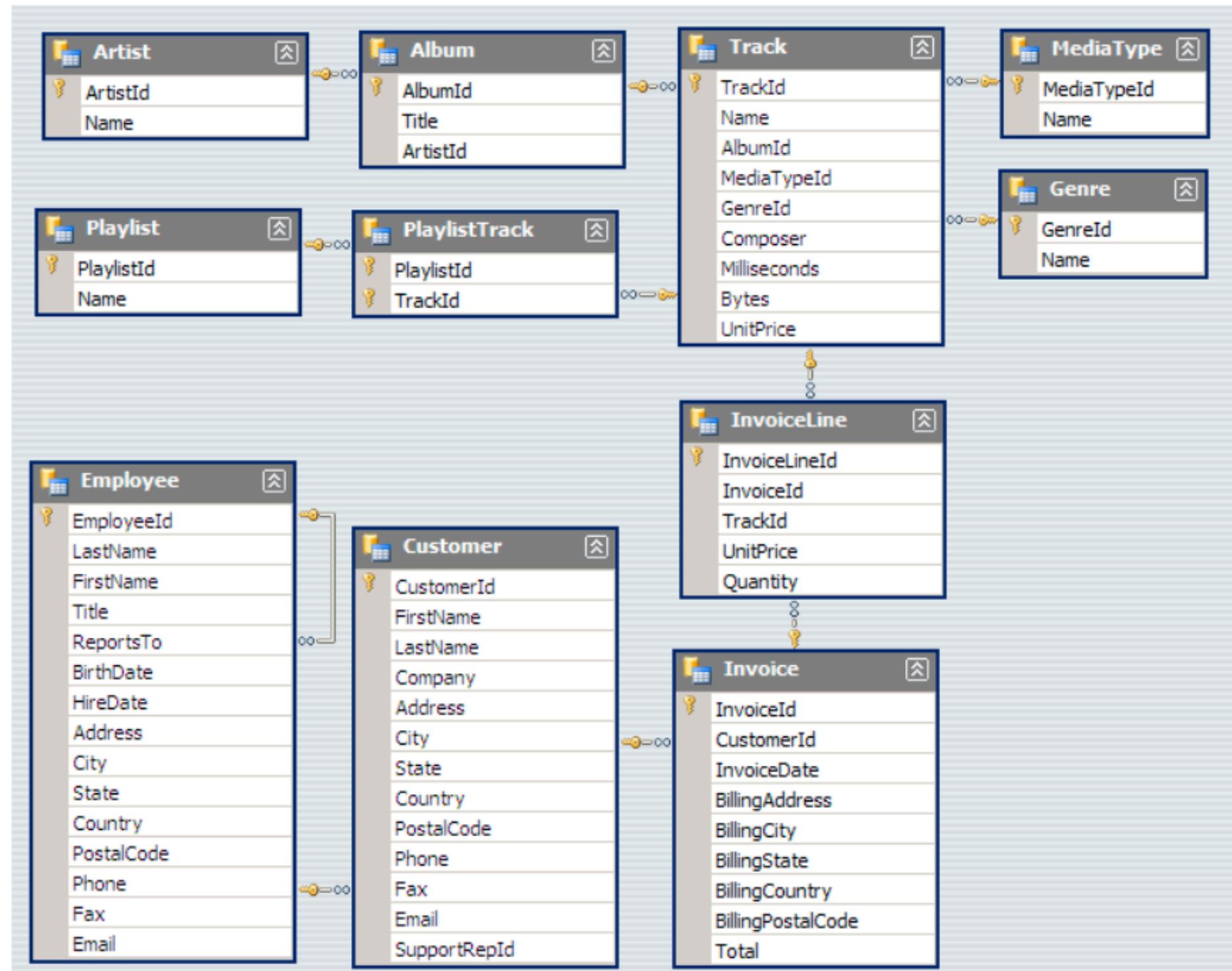
By - Asma Rahman

Introduction

A complete basic and advance level data analysis of a music store using various SQL tools including joins.

The project included solving multiplte analytical questions by creating SQL queries.

Schema



Q.1 who is the senior most employee based on job title?

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

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

Q.2 which countries have the most invoices?

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

	c bigint	billing_country character varying (30)
1	131	USA

Q.3 Select top 3 of total invoices

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

	total double precision	🔒
1	23.759999999999998	
2	19.8	
3	19.8	

Q.4 which city has the best customers? write a query that has one city that has the highest sum of invoice total. return both the city name and sum of invoice total.

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

billing_city character varying (30)	invoice_total double precision
Prague	273.24000000000007

Q.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 to the customer who has spent the most money.

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

	customer_id [PK] integer	first_name character	last_name character	total_invoice double precision
1	5	R	...	Madhav

total_invoice
double precision
144.54000000000002



moderate level

Q1. Write a query to return the email, first name, last name and genre of all rock music listeners ordered alphabetically.

```
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 = invoiceline.invoice_id  
where track_id(  
    Select track_id from track  
    join genre on track.genre_id= genre.genre_id  
    where genre.name like 'Rock')  
order by email;
```

	email character varying (50)	first_name character	last_name character
1	aaronmitchell@yahoo.ca	Aaron	Mitchell
2	alero@uol.com.br	Alexandre	Rocha
3	astrid.gruber@apple.at	Astrid	Gruber
4	bjorn.hansen@yahoo.no	Bjørn	Hansen
5	camille.bernard@yahoo.fr	Camille	Bernard
6	daan_peeters@apple.be	Daan	Peeters
7	diego.gutierrez@yahoo.ar	Diego	Gutiérrez

Q2. lets invite the artists who have written the most Rock music in our dataset. write a query that returns the Artist and total track counts of the top 10 rock bands.

```
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;
```

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

Q3. select all the track that have song lenght longer then the average song lenght. return name and millisecond for each track. order by lenght.

```
select "name", track_id , milliseconds
from track
where milliseconds > (select avg(milliseconds)
    from track)
group by track_id
order by milliseconds desc
```

	name character varying (150)	track_id [PK] integer	milliseconds integer
1	Occupation / Precipice	2820	5286953
2	Through a Looking Glass	3224	5088838
3	Greetings from Earth, Pt. 1	3244	2960293
4	The Man With Nine Lives	3242	2956998
5	Battlestar Galactica, Pt. 2	3227	2956081
6	Battlestar Galactica, Pt. 1	3226	2952702
7	Murder On the Rising Star	3243	2935894



Advance

Q1. Find how much amount spent by each customer on Artist. 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 amount
from invoice_line
join track on invoice_line.track_id= track.track_id
join album on track.album_id = album.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)
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;
```

	customer_id integer	first_name character	last_name character	artist_name character varying (120)	amount_spent double precision
1	46	Hugh	O'Reilly	Queen	27.71999999999985
2	38	Niklas	Schröder	... Queen	18.81
3	3	François	Tremblay	... Queen	17.82
4	34	João	Fernandes	... Queen	16.83000000000002
5	53	Phil	Hughes	... Queen	11.88
6	41	Marc	Dubois	... Queen	11.88
7	47	Lucas	Mancini	... Queen	10.89

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.

```
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)
    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
```

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



Thank you