MUSIC STORE ANALYSIS USING SQL

BY SANA ARSHAD

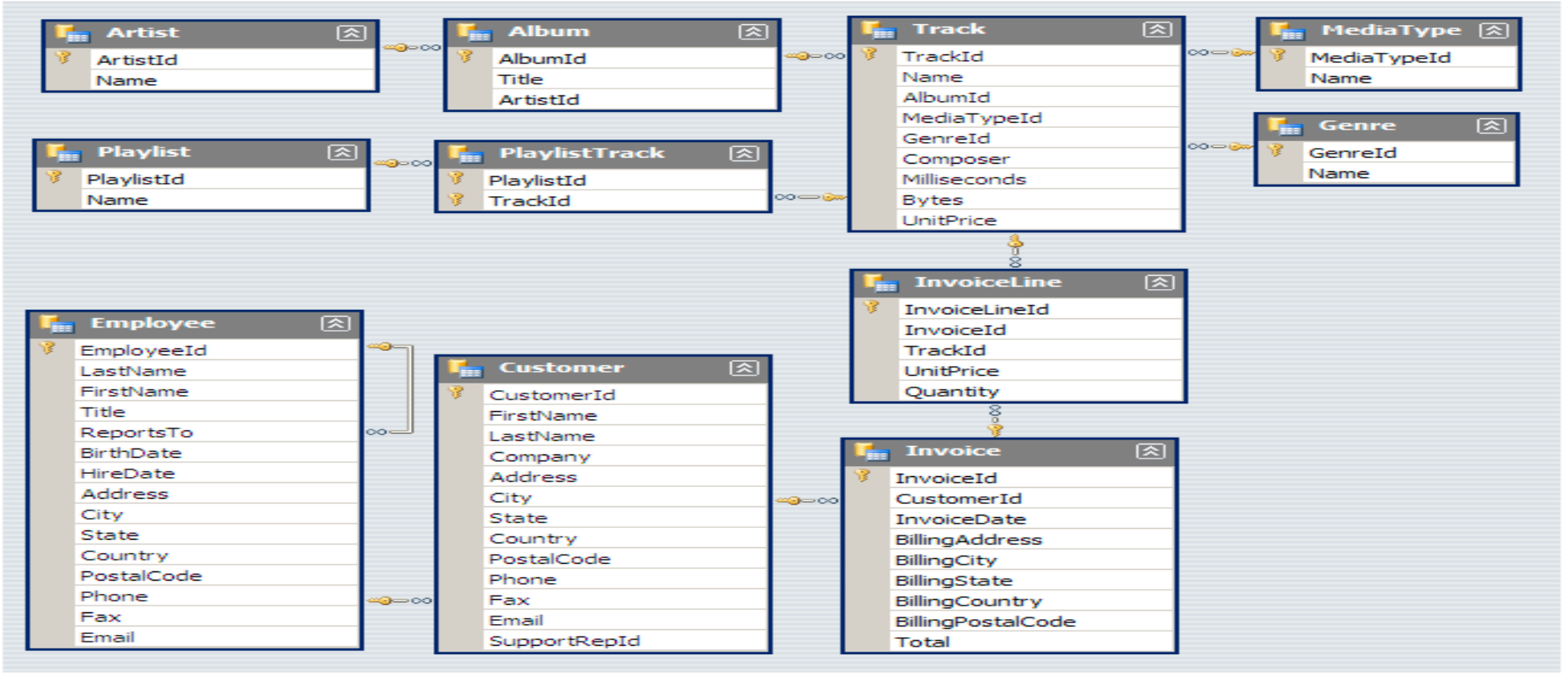
OBJECTIVES

· To help music store to overcome their challenges and help them to grow their business

FUNCTIONS USED:

- Joins in SQL
- Where clause
- Aggregate functions
- Group by clause
- Order by clause
- Limit in SQL
- CTEs

MUSIC STORE DATA BASE SCHEMEA



Query | Query History

```
1  -- 1--Who is the senior most employee based on job title?
2  select * from employee
3  order by levels desc limit 1
4
```

Data Output | Messages | Notifications

	employee_id [PK] character varying (50)	last_name character	first_name character	title character varying (50)	reports_to character varying (30)	levels character varying
1	9	Madan	Mohan	Senior General Manager	[null]	L7

Music_store/postgres@PostgreSQL 16*

Music_store/postgres@PostgreSQL 16

No limit

Query | Query History

```
1  -- 2-. Which countries have the most Invoices?
2  select count(*) as c,billing_country from
3      invoice group by billing_country
4  order by c desc
```

Data Output | Messages | Notifications

	c bigint	billing_country character varying (30)
1	131	USA
2	76	Canada
3	61	Brazil
4	50	France
5	41	Germany
6	30	Czech Republic
7	29	Portugal
8	28	United Kingdom
9	21	India
10	13	Chile
11	13	Ireland

Total rows: 24 of 24 | Query complete 00:00:00.101

Music_store/postgres@PostgreSQL 16*

Music_store/postgres@PostgreSQL 16

No limit

Query | Query History

```sql
-- 3. What are top 3 values of total invoice?
select total from invoice
    order by total desc limit 3
```

Data Output | Messages | Notifications

	total double precision
1	23.759999999999998
2	19.8
3	19.8

No limit

Query | Query History

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

Data Output | Messages | Notifications

	invoice_total double precision	billing_city character varying (30)
1	273.24000000000007	Prague
2	169.29	Mountain View
3	166.32	London
4	158.4	Berlin
5	151.47	Paris
6	129.69	São Paulo
7	114.83999999999997	Dublin

Query | Query History

```
--5Who 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
select customer.customer_id,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

```

Data Output | Messages | Notifications

	customer_id [PK] integer	last_name character	total double precision
1	5	Madhav	144.54000000000002

Music_store/postgres@PostgreSQL 16*

Music_store/postgres@PostgreSQL 16

No limit

Query | Query History

```sql
---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
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
```

Data Output | Messages | Notifications

	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

Total rows: 59 of 59 | Query complete 00:00:00.102

No limit

Query | Query History

```
2  --7---Let's invite the artists who have written the most rock music in our dataset. Write a
3  --query that returns the Artist name and total track count of the top 10 rock bands-
4  select artist.artist_id,artist.name, count(artist.artist_id) as number_of_songs
5  from track
6  join album on album.album_id=track.album_id
7  join artist on artist.artist_id=album.artist_id
8  join genre on genre.genre_id=track.genre_id
9  where genre.name like 'Rock'
10 group by artist.artist_id
11 order by number_of_songs desc
12 limit 10;
```

Data Output | Messages | Notifications

	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

Total rows: 10 of 10 | Query complete 00:00:00.105

No limit

Query | Query History

```
2  --7---Let's invite the artists who have written the most rock music in our dataset. Write a
3  --query that returns the Artist name and total track count of the top 10 rock bands-
4  select artist.artist_id,artist.name, count(artist.artist_id) as number_of_songs
5  from track
6  join album on album.album_id=track.album_id
7  join artist on artist.artist_id=album.artist_id
8  join genre on genre.genre_id=track.genre_id
9  where genre.name like 'Rock'
10 group by artist.artist_id
11 order by number_of_songs desc
12 limit 10;
```

Data Output | Messages | Notifications

	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

Total rows: 10 of 10 | Query complete 00:00:00.105

Query | Query History

```
---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 firs

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

Data Output | Messages | Notifications

	name character varying (150)	milliseconds integer
1	Occupation / Precipice	5286953
2	Through a Looking Glass	5088838
3	Greetings from Earth, Pt. 1	2960293
4	The Man With Nine Lives	2956998
5	Battlestar Galactica, Pt. 2	2956081
6	Battlestar Galactica, Pt. 1	2952702

Total rows: 494 of 494 | Query complete 00:00:00.112

Query | Query History

```sql
--Find how much amount spent by each customer on artists? 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 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
```

Data Output | Messages | Notifications

	customer_id integer	first_name character	last_name character	artist_name character varying (120)	amount_spent double precision
1	46	Hugh	O'Reilly	Queen	27.719999999999985
2	38	Niklas	Schröder	Queen	18.81
3	3	François	Tremblay	Queen	17.82
4	34	João	Fernandes	Queen	16.830000000000002
5	53	Phil	Hughes	Queen	11.88
6	41	Marc	Dubois	Queen	11.88

Music_store/postgres@PostgreSQL 16

No limit

Query | Query History

```
1  --We want to find out the most popular music Genre for each country. We determine the
2  --most popular genre as the genre with the highest amount of purchases. Write a query
3  --that returns each country along with the top Genre. For countries where the maximum
4  --number of purchases is shared return all
5  with popular_genre as(
6      select count(invoice_line.quantity) as purchases,customer.country,genre.name,genre.genre_id,row_numbe
7      over(partition by customer.country order by count(invoice_line.quantity) desc) RowNo
8      from invoice_line
9      join invoice on invoice_line.invoice_id=invoice.invoice_id
10     join customer on customer.customer_id=invoice.customer_id
11     join track on track.track_id=invoice_line.track_id
12     join genre on genre.genre_id=track.genre_id
13     group by 2,3,4
14     order by 2 asc,1 desc)
15     select * from popular_genre where RowNo<=1
```

Data Output | Messages | Notifications

	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

Total rows: 24 of 24 | Query complete 00:00:00.070

No limit

Query | Query History

```
1  --Write a query that determines the customer that has spent the most on music for each
2  --country. Write a query that returns the country along with the top customer and how
3  --much they spent. For countries where the top amount spent is shared, provide all
4  --customers who spent this amount
5  with recursive customer_with_country as(
6      select customer.customer_id,first_name,last_name,billing_country,sum(total) as total_spending
7      from invoice
8      join customer on customer.customer_id=invoice.customer_id
9      group by 1,2,3,4
10     order by 2,3 desc),
11     country_max_spending as(
12     select billing_country,max(total_spending) as max_spending
13     from customer_with_country
14     group by billing_country)
15     select customer_with_country.billing_country,customer_with_country.total_spending,
16     customer_with_country.first_name,customer_with_country.last_name,customer_with_country.customer_id
17 from customer_with_country join
```

Data Output | Messages | Notifications

	billing_country character varying (30)	total_spending double precision	first_name character	last_name character	customer_id integer
1	Argentina	39.6	Diego	Gutiérrez	56

Total rows: 59 of 59 | Query complete 00:00:00.079 | Ln 2

INSIGHTS GATHERED:

- - Adam Andrew is the oldest employee working in company.
  - Customers from Prague has the Spent the highest amount of money.
  - Customer named Franteik has spent the most amount of money to buy music.
  - Number of Songs sung by different Artists with their names and artist ID.
  - Rock is the most popular Genre amongst all countries.

THANK YOU