

# Structured Query Language project by Navjoth Singh on Music Store Data

## Question Set 1 – Easy (5 Questions)

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

### SQL Query

- select employee\_id, last\_name, first\_name, title, levels from employee
- order by levels desc limit 1

### Data Output

Dashboard Properties SQL Statistics Dependencies Dependents Processes Music\_Store\_Data\_Analysis/postgres@PostgreSQL 16\*

Music\_Store\_Data\_Analysis/postgres@PostgreSQL 16

No limit

Query Query History

1

-- SQL Project by Navjoth Singh

2

-- Music Store Data Analysis

3

|

4

-- Q1. Who is the senior most employee based on job title?

5

6

select employee\_id, last\_name, first\_name, title, levels from employee

7

order by levels desc limit 1

Data Output Messages Notifications

	employee_id [PK] character varying (50)	last_name character	first_name character	title character varying (50)	levels character varying (10)
1	9	Madan	Mohan	Senior General Manager	L7

Q2: Which countries have the most Invoices?

### SQL Query

- select cu.country, count(invoice\_id) as Count\_of\_invoices from invoice as inv
- full join customer as cu
- on cu.customer\_id = inv.customer\_id
- group by cu.country
- order by Count\_of\_invoices desc
- limit 5

### Data Output

The screenshot shows a PostgreSQL IDE interface. At the top, there are tabs for Dashboard, Properties, SQL, Statistics, Dependencies, Dependents, and Processes. The active tab is 'SQL', and the connection is 'Music\_Store\_Data\_Analysis/postgres@PostgreSQL 16\*'. Below the tabs, there is a toolbar with icons for file operations, query execution, and other functions. The main area is divided into 'Query' and 'Query History' sections. The 'Query' section contains the following SQL code:

```
1 -- SQL Project by Navjoth Singh
2 -- Music Store Data Analysis
3
4 -- Q2. Which countries have the most Invoices?
5
6 select cu.country, count(invoice_id) as Count_of_invoices from invoice as inv
7 full join customer as cu
8 on cu.customer_id = inv.customer_id
9 group by cu.country
10 order by Count_of_invoices desc
11 limit 5
```

Below the query, there are tabs for 'Data Output', 'Messages', and 'Notifications'. The 'Data Output' tab is active, showing a table with the following data:

	country character varying (50)	count_of_invoices bigint
1	USA	131
2	Canada	76
3	Brazil	61
4	France	50
5	Germany	41

Q3: Which artists have top 3 values of total invoice?

SQL Query

- SELECT MAX(ar.artist\_id) as artist\_id, MAX(ar.name) as artist\_name,
- ROUND(CAST(SUM(inv.total) AS numeric), 2) as Total\_invoice\_value
- FROM invoice AS inv
- JOIN invoice\_line ON inv.invoice\_id = invoice\_line.invoice\_id
- JOIN track ON invoice\_line.track\_id = track.track\_id
- JOIN album ON track.album\_id = album.album\_id
- JOIN artist AS ar ON album.artist\_id = ar.artist\_id
- GROUP BY ar.name
- ORDER BY Total\_invoice\_value DESC LIMIT 3

Data Output

Dashboard Properties SQL Statistics Dependencies Dependents Processes Music\_Store\_Data\_Analysis/postgres@PostgreSQL 16\*

Music\_Store\_Data\_Analysis/postgres@PostgreSQL 16

No limit

Query Query History

1 -- SQL Project by Navjoth Singh

2 -- Music Store Data Analysis

3 |

4 -- Q3. Which artists have top 3 values of total invoice?

5

6 SELECT MAX(ar.artist\_id) as artist\_id, MAX(ar.name) as artist\_name,

7 ROUND(CAST(SUM(inv.total) AS numeric), 2) as Total\_invoice\_value

8 FROM invoice AS inv

9 JOIN invoice\_line ON inv.invoice\_id = invoice\_line.invoice\_id --Joined with linked table

10 JOIN track ON invoice\_line.track\_id = track.track\_id --Joined with linked table

11 JOIN album ON track.album\_id = album.album\_id --Joined with linked table

12 JOIN artist AS ar ON album.artist\_id = ar.artist\_id --Joined with linked table

13 GROUP BY ar.name

14 ORDER BY Total\_invoice\_value DESC LIMIT 3

--

Data Output Messages Notifications

	artist_id text	artist_name text	total_invoice_value numeric
1	94	Jimi Hendrix	2623.50
2	51	Queen	2269.08
3	127	Red Hot Chili Peppers	1484.01

Q4: Which 3 cities have 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 3 cities that have the highest sum of invoice totals. Return both the city name & sum of all invoice totals?

SQL Query

- select max(cu.city) as City\_Name,
- round(cast(sum(inv.total)as numeric),2) as Total\_Invoice from invoice as inv
- join customer as cu on inv.customer\_id = cu.customer\_id
- group by cu.city
- order by Total\_Invoice Desc
- limit 3

Data Output

Dashboard Properties SQL Statistics Dependencies Dependents Processes Music\_Store\_Data\_Analysis/postgres@PostgreSQL 16\*

Music\_Store\_Data\_Analysis/postgres@PostgreSQL 16

No limit

Query Query History

1

-- SQL Project by Navjoth Singh

2

-- Music Store Data Analysis

3

4

-- Q4. Which 3 cities have the best customers?

5

-- We would like to throw a promotional Music Festival in the city we made the most money.

6

-- Write a query that returns 3 cities that have the highest sum of invoice totals.

7

-- Return both the city name & sum of all invoice totals?

8

9

select max(cu.city) as City\_Name,

10

round(cast(sum(inv.total)as numeric),2) as Total\_Invoice from invoice as inv

11

join customer as cu on inv.customer\_id = cu.customer\_id

12

group by cu.city

13

order by Total\_Invoice Desc

14

limit 3

Data Output Messages Notifications

	city_name text	total_invoice numeric
1	Prague	273.24
2	Mountain View	169.29
3	London	166.32

Question 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.

SQL Query

- select cu.customer\_id, cu.city, cu.country, cu.email, cu.phone,
- cu.first\_name || ' ' || cu.last\_name as full\_name,
- round(cast(sum(inv.total)as numeric),2) as Total\_Invoice from invoice as inv
- join customer as cu on inv.customer\_id = cu.customer\_id
- GROUP BY cu.customer\_id
- order by Total\_Invoice Desc
- limit 3

Data Output

Dashboard Properties SQL Statistics Dependencies Dependents Processes Music\_Store\_Data\_Analysis/postgres@PostgreSQL 16\*

Music\_Store\_Data\_Analysis/postgres@PostgreSQL 16

No limit

Query Query History

1

-- SQL Project by Navjoth Singh

2

-- Music Store Data Analysis

3

4

-- Q5. Who is the best customer? The customer who has spent the most money will be declared the best customer.

5

-- Write a query that returns the person who has spent the most money.

6

7

select cu.customer\_id, cu.city, cu.country, cu.email, cu.phone,

8

cu.first\_name || ' ' || cu.last\_name as full\_name,

9

round(cast(sum(inv.total)as numeric),2) as Total\_Invoice from invoice as inv

10

join customer as cu on inv.customer\_id = cu.customer\_id

11

GROUP BY cu.customer\_id

12

order by Total\_Invoice Desc

13

limit 3

14

Data Output Messages Notifications

	customer_id [PK] integer	city character varying (50)	country character varying (50)	email character varying (50)	phone character varying (50)	full_name text	total_invoice numeric
1	5	Prague	Czech Republic	r.madhav@jetbrains.com	+420 2 4172 5555	R Madhav	144.54
2	6	Prague	Czech Republic	hholy@gmail.com	+420 2 4177 0449	Helena Holý	128.70
3	46	Dublin	Ireland	hughoreilly@apple.ie	+353 01 6792424	Hugh O'Reilly	114.84

Question Set 2 – Moderate (3 Questions)

Q6: 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 Query

- SELECT distinct cu.email, cu.first\_name || ' ' || cu.last\_name as Full\_Name,
- genre.name, cu.country,cu.phone,cu.address
- FROM customer AS cu
- Join invoice as inv on cu.customer\_id = inv.customer\_id
- JOIN invoice\_line ON inv.invoice\_id = invoice\_line.invoice\_id
- JOIN track ON invoice\_line.track\_id = track.track\_id
- JOIN genre ON track.genre\_id = genre.genre\_id
- where genre.name in ('Rock')
- ORDER BY cu.email

Data Output

board Properties SQL Statistics Dependencies Dependents Processes Music\_Store\_Data\_Analysis/postgres@PostgreSQL 16\* Music\_Stor...

Music\_Store\_Data\_Analysis/postgres@PostgreSQL 16

No limit

Query Query History

1 -- SQL Project by Navjoth Singh

2 -- Music Store Data Analysis

3

4 -- Q6. Write query to return the email, first name, last name, & Genre of all Rock Music listeners.

5 -- Return your list ordered alphabetically by email starting with A?

6

7 SELECT distinct cu.email, cu.first\_name || ' ' || cu.last\_name as Full\_Name,

8 genre.name, cu.country,cu.phone,cu.address

9 FROM customer AS cu

10 Join invoice as inv on cu.customer\_id = inv.customer\_id

11 JOIN invoice\_line ON inv.invoice\_id = invoice\_line.invoice\_id

12 JOIN track ON invoice\_line.track\_id = track.track\_id

13 JOIN genre ON track.genre\_id = genre.genre\_id

14 where genre.name in ('Rock')

15 ORDER BY cu.email

Data Output Messages Notifications

	email character varying (50)	full_name text	name character varying (120)	country character varying (50)	phone character varying (50)	address character varying (120)
1	aaronmitchell@yahoo.ca	Aaron Mitchell	Rock	Canada	+1 (204) 452-6452	696 Osborne Street
2	alero@uol.com.br	Alexandre Rocha	Rock	Brazil	+55 (11) 3055-3278	Av. Paulista, 2022
3	astrid.gruber@apple.at	Astrid Gruber	Rock	Austria	+43 01 5134505	Rotenturmstraße 4, 1010 Innere Stadt



Q7: 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 Query

- select ar.artist\_id, ar.name , count(ge.name) as Count\_of\_rock\_music
- from artist as ar
- join album as al on al.artist\_id = ar.artist\_id
- join track as tr on tr.album\_id = al.album\_id
- join genre as ge on ge.genre\_id = tr.genre\_id
- where ge.name like 'Rock'
- group by ar.name, ar.artist\_id
- order by Count\_of\_rock\_music desc
- limit 10

### Data Output

The screenshot shows the PostgreSQL pgAdmin interface. The top navigation bar includes links for Dashboard, Properties, SQL, Statistics, Dependencies, Dependents, Processes, and a server connection named 'Music\_Store\_D...'. Below this, a dropdown menu shows the selected connection 'Music\_Store\_Data\_Analysis/postgres@PostgreSQL 16'. A toolbar with various icons for file operations, query execution, and settings is visible. The main area is divided into 'Query' and 'Query History' tabs. The 'Query' tab contains a SQL query that filters for rock music and orders artists by the number of tracks. The 'Data Output' tab shows the results of the query in a table format.

**Query:**

```

1  -- SQL Project by Navjoth Singh
2  -- Music Store Data Analysis
3
4  -- Q7. Let's invite the artists who have written the most rock music in our dataset.
5  -- Write a query that returns the Artist name and total track count of the top 10 rock bands?
6
7  select ar.artist_id, ar.name , count(ge.name) as Count_of_rock_music
8  from artist as ar
9  join album as al on al.artist_id = ar.artist_id
10 join track as tr on tr.album_id = al.album_id
11 join genre as ge on ge.genre_id = tr.genre_id
12 where ge.name like 'Rock'
13 group by ar.name, ar.artist_id
14 order by Count_of_rock_music desc
15 limit 10

```

**Data Output:**

	artist_id [PK] character varying (50)	name character varying (120)	count_of_rock_music bigint
1	22	Led Zeppelin	114
2	150	U2	112
3	58	Deep Purple	92

SQL Project by Navjoth Singh – 09-Oct.-2023 – PostgreSQL

Q8: 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 Query

- SELECT name, milliseconds, bytes, unit\_price
- FROM track
- WHERE milliseconds > (SELECT AVG(milliseconds) FROM track)
- ORDER BY milliseconds DESC
- limit 5

Data Output

Dashboard Properties SQL Statistics Dependencies Dependents Processes Music\_Store\_D... Music\_Store\_Data\_Analysis/postgres@PostgreSQL

Music\_Store\_Data\_Analysis/postgres@PostgreSQL 16

📁 📄 ⌵ ✎ ⌵ ⌵ No limit ⌵ 🎵 🎵 ⌵ ⌵ ⌵ ⌵ ⌵ ?

Query Query History

1 -- SQL Project by Navjoth Singh

2 -- Music Store Data Analysis

3

4 -- Q8. Return all the track names that have a song length longer than the average song length.

5 -- Return the Name and Milliseconds for each track. Order by the song length with the longest songs listed first?

6

7 SELECT name, milliseconds, bytes, unit\_price

8 FROM track

9 WHERE milliseconds > (SELECT AVG(milliseconds) FROM track)

10 ORDER BY milliseconds DESC

11 limit 5

12

Data Output Messages Notifications

⌵ 📄 ⌵ 📄 🗑️ 🗑️ 📄 ⌵ ⌵

	name character varying (150) 🔒	milliseconds integer 🔒	bytes integer 🔒	unit_price double precision 🔒
1	Occupation / Precipice	5286953	1054423946	1.99
2	Through a Looking Glass	5088838	1059546140	1.99
3	Greetings from Earth, Pt...	2960293	536824558	1.99
4	The Man With Nine Lives	2956998	577829804	1.99
5	Battlestar Galactica, Pt. 2	2956081	521387924	1.99



Question Set 3 - Advance (3 Questions)

Q9: Find how much amount spent by each customer on best-selling artists? Write a query to return customer name, artist name and total spent?

SQL Query

- with best\_selling\_artist as(
  - select art.artist\_id, art.name, round(cast(sum(invl.unit\_price\*invl.quantity) as numeric),2)
  - as total\_invoice from invoice as inv
  - join invoice\_line as invl on invl.invoice\_id = inv.invoice\_id
  - join track as trk on trk.track\_id = invl.track\_id
  - join album as alb on alb.album\_id = trk.album\_id
  - join artist as art on art.artist\_id = alb.artist\_id
  - group by 1,2
  - order by 3 Desc
  - limit 1)
- Select cu.customer\_id, cu.first\_name || ' ' || cu.last\_name as Full\_Name\_of\_Customer,
- max(art.name) as Artist\_Name, max(art.artist\_id) as artist\_id, round(cast(sum(invl.quantity\*invl.unit\_price ) as numeric),2)
- as total\_incoice from customer as cu
- join Invoice as inv on inv.customer\_id = cu.customer\_id
- join invoice\_line as invl on invl.invoice\_id = inv.invoice\_id
- join track as trk on trk.track\_id = invl.track\_id
- join album as alb on alb.album\_id = trk.album\_id
- join best\_selling\_artist as art on art.artist\_id = alb.artist\_id
- group by 1,2
- order by 5 Desc

Data Output

Dashboard Properties SQL Statistics Dependencies Dependents Processes Music\_Store\_Data\_Analysis/postgres@PostgreSQL 16\*

Music\_Store\_Data\_Analysis/postgres@PostgreSQL 16

No limit

Query Query History

1 -- SQL Project by Navjoth Singh

2 -- Music Store Data Analysis

3

4 -- Q9. Find how much amount spent by each customer on best-selling artists?

5 -- Write a query to return customer name, artist name and total spent?

6

7 with best\_selling\_artist as(

8     select art.artist\_id, art.name, round(cast(sum(invl.unit\_price\*invl.quantity) as numeric),2)

9     as total\_invoice from invoice as inv

10    join invoice\_line as invl on invl.invoice\_id = inv.invoice\_id

11    join track as trk on trk.track\_id = invl.track\_id

12    join album as alb on alb.album\_id = trk.album\_id

13    join artist as art on art.artist\_id = alb.artist\_id

Data Output Messages Notifications

	customer_id [PK] integer	full_name_of_customer text	artist_name text	artist_id text	total_incoice numeric
1	46	Hugh O'Reilly	Queen	51	27.72
2	38	Niklas Schröder	Queen	51	18.81
3	3	François Tremblay	Queen	51	17.82
4	34	João Fernandes	Queen	51	16.83

SQL Project by Navjoth Singh – 09-Oct.-2023 – PostgreSQL

Q10: 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 quantity 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 Query (This can be solve with the help of RowNumber() & RECURSIVE Query)

- with Popular\_music\_genre as (
  - select inv.billing\_country as country ,pmg.name as genre, count(invl.quantity) as count\_of\_purchase
  - from invoice as inv join invoice\_line as invl on invl.invoice\_id = inv.invoice\_id
  - join track as tr on tr.track\_id = invl.track\_id
  - join genre as pmg on pmg.genre\_id = tr.genre\_id
  - group by 1,2
  - order by 3 desc)
- select pmg.country as Country, pmg.genre as Genre,
- max(pmg.count\_of\_purchase) as Highest\_purchase\_of\_country
- from Popular\_music\_genre as pmg
- join (select pmg.country as gof\_name, max(pmg.count\_of\_purchase) as purchase
  - from Popular\_music\_genre as pmg group by 1 )
- as gof on gof.gof\_name = pmg.country
- where pmg.count\_of\_purchase = gof.purchase
- group by 1,2
- order by 3 Desc

Data Output

Dashboard Properties SQL Statistics Dependencies Dependents Processes Music\_Store\_Data\_Analysis/postgres@PostgreSQL 16\*

Music\_Store\_Data\_Analysis/postgres@PostgreSQL 16

Query

Query History

1

-- SQL Project by Navjoth Singh

2

-- Music Store Data Analysis

3

4

-- Q10. We want to find out the most popular music Genre for each country.

5

-- We determine the most popular genre as the genre with the highest quantity of purchases.

6

-- Write a query that returns each country along with the top Genre.

7

-- For countries where the maximum number of purchases is shared return all Genres?

8

9

with Popular\_music\_genre as (

10

select inv.billing\_country as country ,pmg.name as genre, count(invl.quantity) as count\_of\_purchase

11

from invoice as inv join invoice\_line as invl on invl.invoice\_id = inv.invoice\_id

12

join track as tr on tr.track\_id = invl.track\_id

13

join genre as pmg on pmg.genre id = tr.genre id

Data Output

Messages

Notifications

	country character varying (30)	genre character varying (120)	highest_purchase_of_country bigint
1	USA	Rock	561
2	Canada	Rock	333
3	France	Rock	211
4	Brazil	Rock	205

Q11: 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?

SQL Query

- with Popular\_music\_genre as (
  - select Sum(invl.unit\_price\*invl.quantity) as money\_spend ,
  - cu.first\_name || ' ' || cu.last\_name as full\_name,
  - cu.country,cu.customer\_id, Row\_Number() Over(partition by cu.country
  - order by Sum(invl.unit\_price\*invl.quantity) Desc) as Row\_No
  - from customer as cu join invoice as inv on cu.customer\_id = inv.customer\_id
  - join invoice\_line as invl on invl.invoice\_id = inv.invoice\_id
  - group by 2,3,4
  - order by 2 Asc, 1 Desc)
- select \* from Popular\_music\_genre as pmg
- where Row\_No = 1
- order by pmg.money\_spend Desc

Data Output

Dashboard Properties SQL Statistics Dependencies Dependents Processes Music\_Store\_Data\_Analysis/postgres@PostgreSQL 16\*

Music\_Store\_Data\_Analysis/postgres@PostgreSQL 16

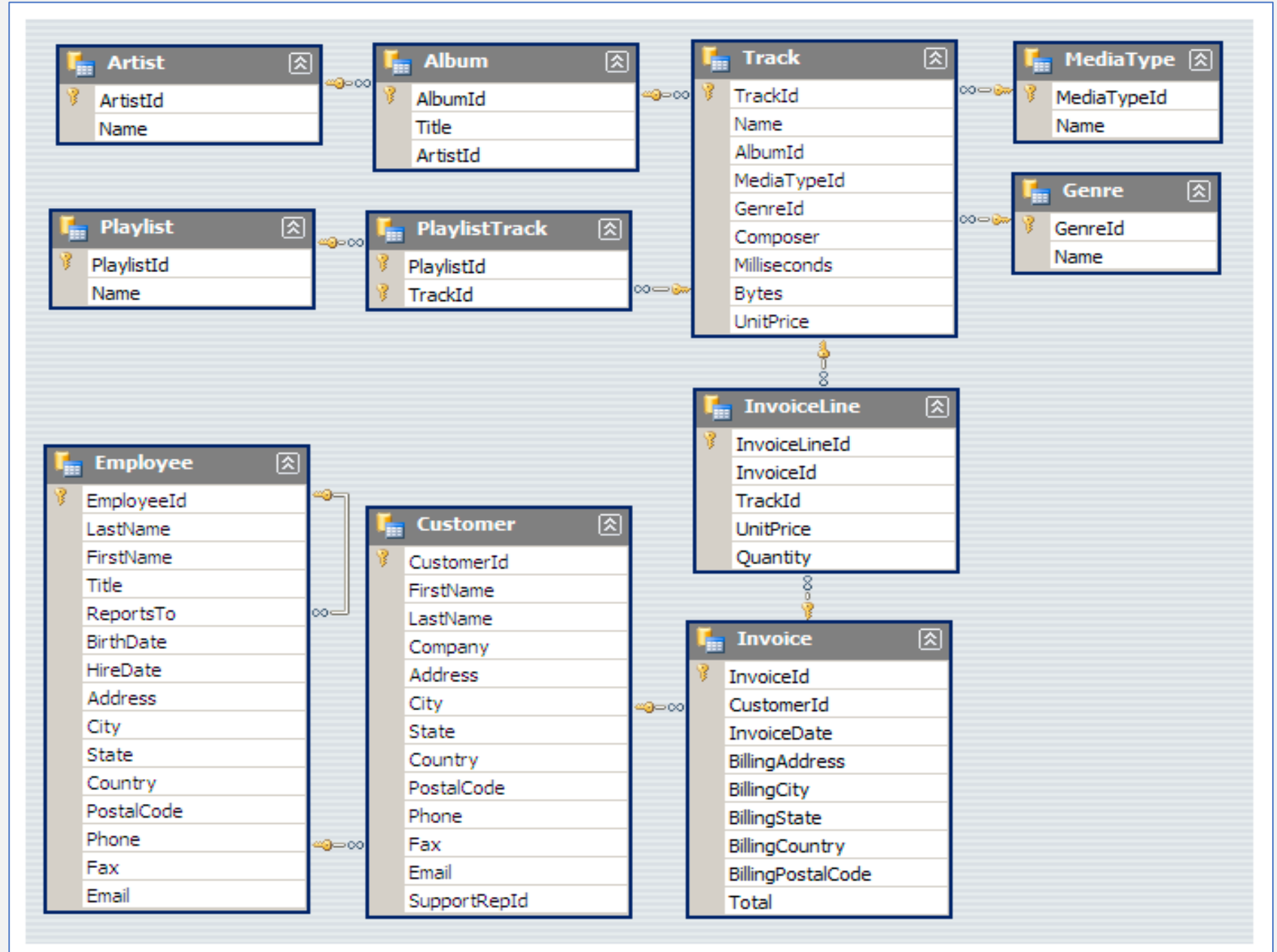
Query Query History

```
1 -- SQL Project by Navjoth Singh
2 -- Music Store Data Analysis
3
4 -- Q11. Write a query that determines the customer that has spent the most on music for each country.
5 -- Write a query that returns the country along with the top customer and how much they spent.
6 -- For countries where the top amount spent is shared, provide all customers who spent this amount?
7
8 with Popular_music_genre as (
9     select Sum(invl.unit_price*invl.quantity) as money_spend ,
10     cu.first_name || ' ' || cu.last_name as full_name,
11     cu.country,cu.customer_id, Row_Number() Over(partition by cu.country
12     order by Sum(invl.unit_price*invl.quantity) Desc) as Row_No
13     from customer as cu join invoice as inv on cu.customer id = inv.customer id
```

Data Output Messages Notifications

	money_spend double precision	full_name text	country character varying (50)	customer_id [PK] integer	row_no bigint
1	144.53999999999985	R Madhav	Czech Republic	5	1
2	114.83999999999978	Hugh O'Reilly	Ireland	46	1
3	111.86999999999979	Manoj Pareek	India	58	1
4	108.89999999999998	Luís Gonçalves	Brazil	1	1

## Schema Relationship Diagram Between 11 Tables



## Question Set 1 - Easy

1. Who is the senior most employee based on job title?
2. Which countries have the most Invoices?
3. Which artists have top 3 values of total invoice?
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?
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?

## Question Set 2 – Moderate

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?
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?
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?

### Question Set 3 – Advance

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