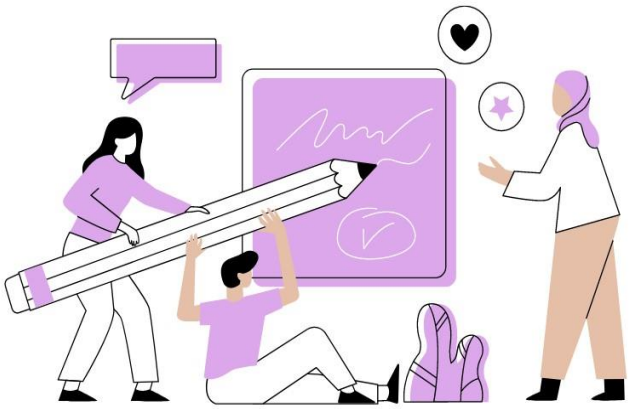
MUSIC STORE DATA ANALYSIS

SQL PROJECT

PORTFOLIO

BY:-RISHAB BHATTACHARYA

TABLE OF CONTENTS



MUSIC STORE DATA ANALYSIS PORTFOLIO PROJECT USING SQL

Properties SQL Dependents Processes music_databas... music_database/postgres@PostgreSQL 15*

music_database/postgres@PostgreSQL 15

No limit

Query Query History

```sql
-- 1)Who is the senior most employee based on job title ?

SELECT * FROM employee
ORDER BY levels DESC
LIMIT 1
```

Data Output Messages Notifications

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

Properties SQL Dependents Processes music_databas... music_database/postgres@PostgreSQL 15*

music_database/postgres@PostgreSQL 15

No limit

Query Query History

```sql
-- 2) Which countries have the most invoices?

SELECT COUNT(*) AS total_count, billing_country
FROM invoice
GROUP BY billing_country
ORDER BY total_count DESC
LIMIT 5
```

Data Output Messages Notifications

	total_count bigint	billing_country character varying
1	131	USA
2	76	Canada
3	61	Brazil
4	50	France
5	41	Germany

Properties SQL Dependents Processes music_databas... music_database/post

music_database/postgres@PostgreSQL 15

No limit

Query Query History

```
1  -- 3) What are top 3 values of total invoice?
2
3  SELECT total FROM invoice
4  ORDER BY total DESC
5  LIMIT 3
```

Data Output Messages Notifications

	total double precision
1	23.759999999999998
2	19.8
3	19.8

Properties SQL Dependents Processes music_databas... music_database/postgres@PostgreSQL 15*

music_database/postgres@PostgreSQL 15

No limit

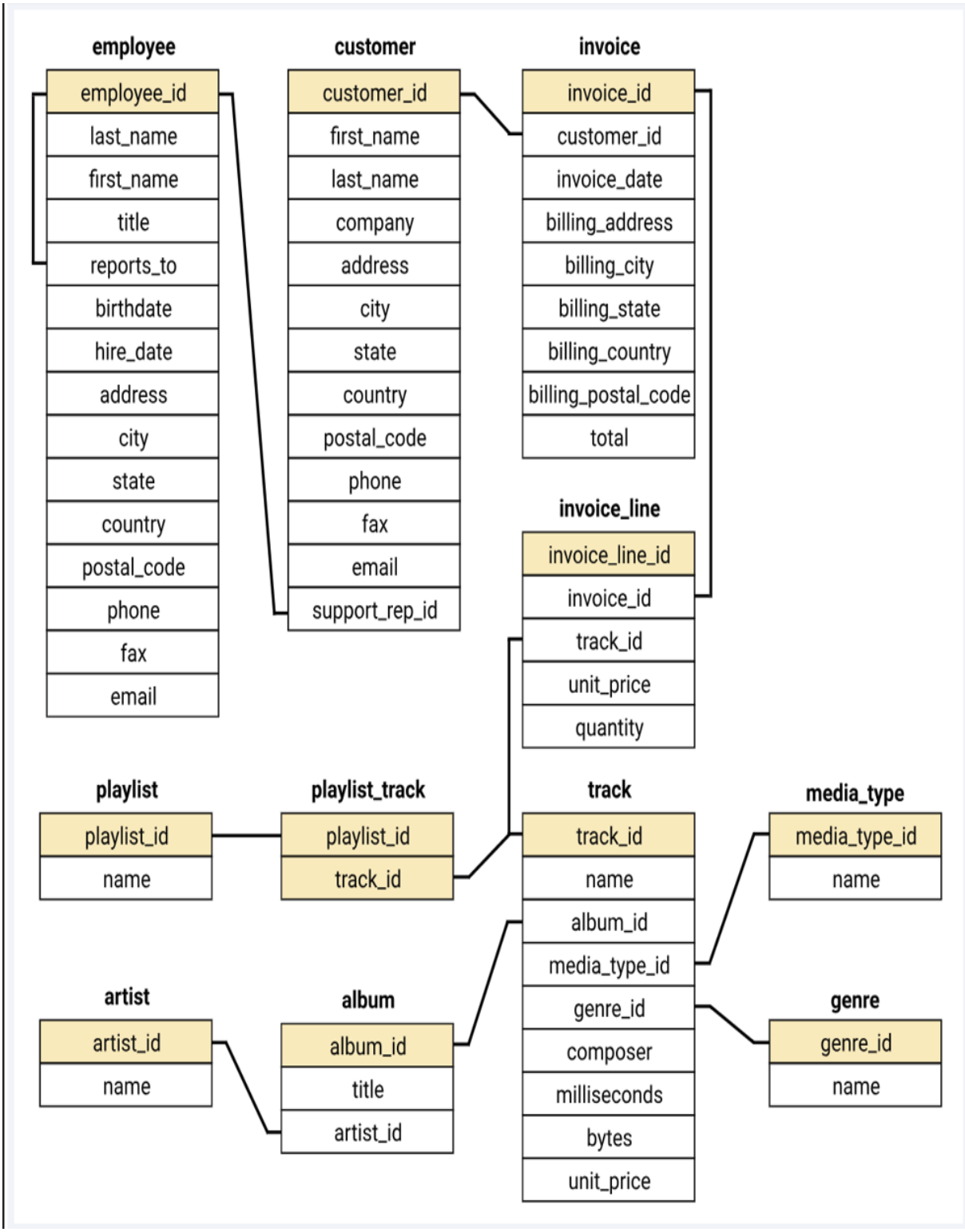
Query Query History

```
1  -- 4) Which city has the best customers ? we would like to throw a promotional music festival in the city
2  -- We made the most money. Write a query that returns one city that has highest sum of invoice totals.
3  -- Retrun both the city name & sum of all invoice totals
4
5  SELECT SUM(total) AS invoice_total, billing_city
6  FROM invoice
7  GROUP BY billing_city
8  ORDER BY invoice_total DESC
9  LIMIT 5
```

Data Output Messages Notifications

	invoice_total double precision	billing_city character varying
1	273.24000000000007	Prague
2	169.29	Mountain View
3	166.32	London
4	158.4	Berlin
5	151.47	Paris

MUSIC STORE DATABASE SCHEMA



Properties SQL Dependents Processes music_databas... music_database/postgres@PostgreSQL 15*

music_database/postgres@PostgreSQL 15

No limit

Query Query History

```sql
-- 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.

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
```

Data Output Messages Notifications

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

Properties SQL Dependents Processes music_database/postgres@PostgreSQL 15*

music_database/postgres@PostgreSQL 15

No limit

Query Query History

```sql
-- 6) Write a query to return the email, frist 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
LIMIT 5;
```

Data Output Messages Notifications

	email character varying	first_name character	last_name character
1	[redacted]	Aaron	Mitchell
2	[redacted]	Alexandre ...	Rocha
3	[redacted]	Astrid	Gruber
4	[redacted]	Bjørn	Hansen ...
5	[redacted]	Camille ...	Bernard

Properties SQL Dependents Processes music_database/postgres@PostgreSQL 15*

music_database/postgres@PostgreSQL 15

No limit

Query Query History

```sql
-- 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.

SELECT artist.artist_id, artist.name,
COUNT(artist.artist_id) AS Total_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 Total_number_of_songs DESC
LIMIT 10;
```

Data Output Messages Notifications

	artist_id [PK] character varying	name character varying	total_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
8	142	The Rolling Stones	41
9	76	Creedence Clearwater Revival	40
10	52	Kiss	35

Properties SQL Dependents Processes music_database/postgres@PostgreSQL 15*

music_database/postgres@PostgreSQL 15

No limit

Query Query History

```sql
-- 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 songs legth with the longest songs
-- listed first.

SELECT name, milliseconds
FROM track
WHERE milliseconds > (SELECT AVG(milliseconds) AS avg_track_length
                      FROM track)
ORDER BY milliseconds DESC
LIMIT 5;
```

Data Output Messages Notifications

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

Properties SQL Dependents Processes music_database/postgres@PostgreSQL 15*

music_database/postgres@PostgreSQL 15

No limit

Query Query History

```sql
-- 9) Find how much amount spent by each customers 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_spent
    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 AS i
JOIN customer AS c ON c.customer_id = i.customer_id
JOIN invoice_line AS il ON il.invoice_id = i.invoice_id
JOIN track AS t ON t.track_id = il.track_id
JOIN album AS alb ON alb.album_id = t.album_id
JOIN best_selling_artist AS bsa ON bsa.artist_id = alb.artist_id
GROUP BY 1, 2, 3, 4
ORDER BY 5 DESC
LIMIT 5;
```

Data Output Messages Notifications

	customer_id integer	first_name character	last_name character	artist_name character varying	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	41	Marc	Dubois	Queen	11.88

Properties SQL Dependents Processes music_database/postgres@PostgreSQL 15*

music_database/postgres@PostgreSQL 15

No limit

Query Query History

```sql
-- 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 amount of purchases)

WITH popular_music_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_music_genre WHERE RowNo <= 1
LIMIT 8;
```

Data Output Messages Notifications

	purchases bigint	country character varying	name character varying	genre_id character varying	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
8	143	Czech Republic	Rock	1	1

Properties SQL Dependents Processes music_database/postgres@PostgreSQL 15*

music_database/postgres@PostgreSQL 15

No limit

Query Query History

```sql
-- 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

WITH customer_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_country WHERE RowNo <= 1
LIMIT 8;
```

Data Output Messages Notifications

	customer_id integer	first_name character	last_name character	billing_country character varying	total_spending double precision	rowno bigint
1	56	Diego	Gutiérrez	Argentina	39.6	1
2	55	Mark	Taylor	Australia	81.18	1
3	7	Astrid	Gruber	Austria	69.3	1
4	8	Daan	Peeters	Belgium	60.38999999999999	1
5	1	Luís	Gonçalves	Brazil	108.89999999999998	1
6	3	François	Tremblay	Canada	99.99	1
7	57	Luis	Rojas	Chile	97.02000000000001	1
8	5	R	Madhav	Czech Republic	144.54000000000002	1