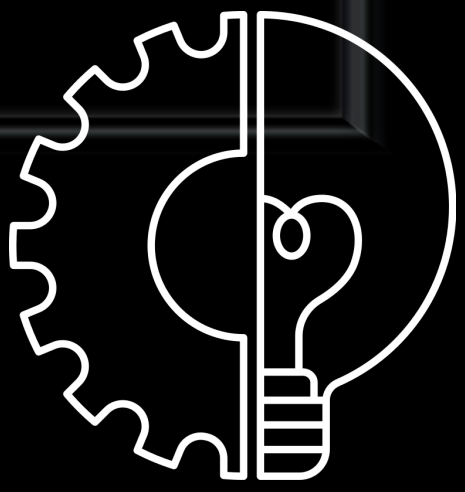
MUSIC STORE DATA ANALYSIS

USING SQL

PRESENTING THE GLIMPSE OF THE PROJECT

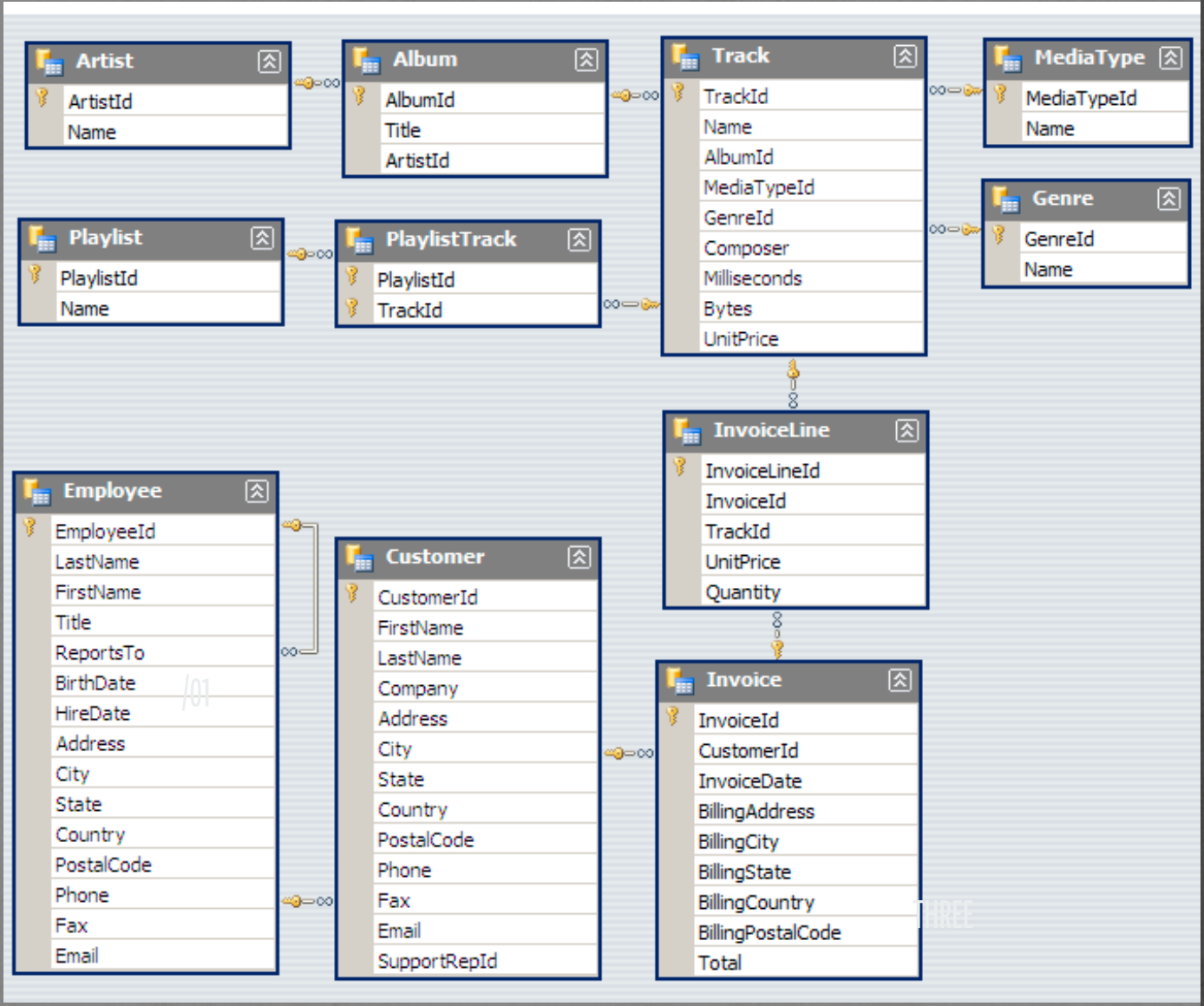
OBJECTIVE

PROJECT FOR FINDING OUT SOME USEFUL INSIGHTS USING SQL FOR BUSINESS GROWTH OF THE MUSIC STORE



UNDERSTANDING THE DATASET

THE MUSIC STORE DATASET CONTAINS 11 TABLES. SCHEMA TABLE FOR THIS DATASET IS SHOW BELOW



WHO IS THE SENIOR MOST EMPLOYEE BASED ON JOB TITLE?

Query | Query History | Scratch Pad

```
1  SELECT * FROM employee
2  ORDER BY levels Desc
3  LIMIT 1
```

Data Output | Messages | Notifications

	employee_id [PK] character varying (50)	last_name character (50)	first_name character (50)	title character varying (50)	reports_to character varying (30)	levels character varying (10)	birthdate timestamp wi
1	9	Madan	Mohan	Senior General Manager	[null]	L7	1961-01-26 0

WHICH COUNTRIES HAVE THE MOST INVOICES?

Query | Query History | Scratch Pad

```
1  Q2-Which country have the most invoices?
2
3  SELECT COUNT(*) AS c,billing_country
4  FROM invoice
5  GROUP BY billing_country
6  ORDER BY c desc
7
```

	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
12	11	Spain
13	11	Finland

WHAT ARE TOP 3 VALUES OF TOTAL INVOICE?

```sql
Q3- What are the top 3 values 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

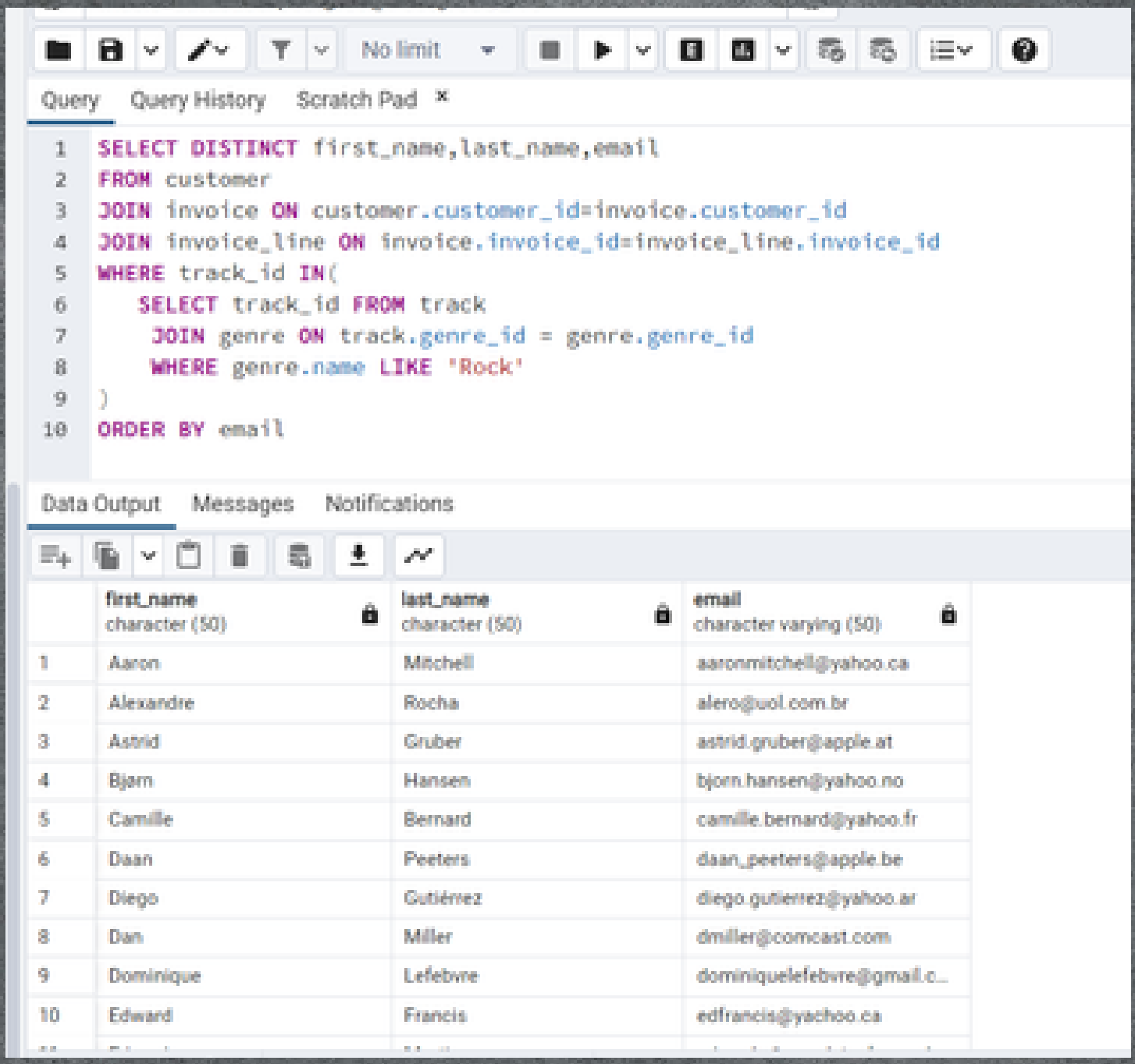
WHICH CITY HAS THE BEST CUSTOMERS? WE WOULD LIKE TO THROW A PROMOTIONAL MUSIC FESTIVAL IN THE CITY WE MADE THE MOST MONEY.
WRITE A QUERY THAT RETURNS ONE CITY THAT HAS THE HIGHEST SUM OF INVOICE TOTALS. RETURN BOTH THE CITY NAME & SUM OF ALL INVOICE TOTALS

```
2
3  SELECT SUM(total) AS invoice_totals,billing_city
4  FROM invoice
5  GROUP BY billing_city
6  ORDER BY invoice_totals desc
```

Data Output | Messages | Graph Visualiser | Notifications

	invoice_totals 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
8	111.86999999999999	Delhi
9	108.89999999999998	São José dos Campos
10	106.91999999999999	Brasília
11	102.96000000000001	Lisbon
12	99.99	Bordeaux
13	99.99	Montréal

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.



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

music_database/postgres@PostgreSQL 15

No limit

Query | Query History | Scratch Pad

```sql
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
```

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
7	51	Queen	45
8	142	The Rolling Stones	41
9	76	Creedence Clearwater Revival	40
10	52	Kiss	35

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.

Query | Query History | Scratch Pad

```
1  SELECT milliseconds,name
2  FROM track
3  WHERE milliseconds>(
4      SELECT AVG(milliseconds) AS average_time
5      FROM TRACK)
6  ORDER BY milliseconds desc
7
8
9
```

Data Output | Messages | Notifications

	milliseconds integer	name character varying (150)
1	5286953	Occupation / Precipice
2	5088838	Through a Looking Glass
3	2960293	Greetings from Earth, Pt. 1
4	2956998	The Man With Nine Lives
5	2956081	Battlestar Galactica, Pt. 2
6	2952702	Battlestar Galactica, Pt. 1
7	2935894	Murder On the Rising Star
8	2927802	Battlestar Galactica, Pt. 3
9	2927677	Take the Celestra
10	2926593	Fire In Space

FIND HOW MUCH AMOUNT SPENT BY EACH CUSTOMER ON ARTISTS? WRITE A QUERY TO RETURN CUSTOMER NAME, ARTIST NAME AND TOTAL SPENT

```sql
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 1
)
SELECT c.customer_id, c.first_name, c.last_name, bsa.artist_name, SUM(il.unit_price*il.quantity) AS am
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;
```

Output Messages Notifications

customer_id integer	first_name character (50)	last_name character (50)	artist_name character varying (120)	amount_spent double precision
46	Hugh	O'Reilly	Queen	27.719999999999985
38	Niklas	Schröder	Queen	18.81
3	François	Tremblay	Queen	17.82

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.

```
ITH 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

ELECT * FROM popular_genre WHERE RowNo <= 1
```

utput Messages Notifications

urchases igint	country character varying (50)	name character varying (120)	genre_id character varying (50)	rowno bigint
17	Argentina	Alternative & Punk	4	1
34	Australia	Rock	1	1
40	Austria	Rock	1	1
26	Belgium	Rock	1	1
205	Brazil	Rock	1	1
333	Canada	Rock	1	1
61	Chile	Rock	1	1
143	Czech Republic	Rock	1	1
24	Denmark	Rock	1	1

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 Customter_with_country AS (
        SELECT customer.customer_id,first_name,last_name,billing_country,SUM(total) AS total_spendin
        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 Customter_with_country WHERE RowNo <= 1
```

Output | Messages | Notifications

customer_id integer	first_name character (50)	last_name character (50)	billing_country character varying (30)	total_spending double precision	rowno bigint
56	Diego	Gutiérrez	Argentina	39.6	1
55	Mark	Taylor	Australia	81.18	1
7	Astrid	Gruber	Austria	69.3	1
8	Daan	Peeters	Belgium	60.38999999999999	1
1	Luís	Gonçalves	Brazil	108.89999999999998	1
3	François	Tremblay	Canada	99.99	1
57	Luis	Rojas	Chile	97.02000000000001	1
5	R	Madhav	Czech Republic	144.54000000000002	1
9	Kara	Nielsen	Denmark	37.61999999999999	1

PROJECT END

DONE BY MEENAL SAINI