



MUSIC STUDIO ANALYSIS

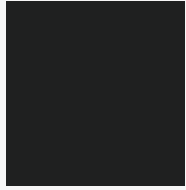
SQL PROJECT



HELLO,

Hello ,

My name is Nilam Gavate. In this SQL project, I conducted an in-depth analysis of a music store's dataset. The project aimed to extract meaningful insights that could help in understanding customer preferences, sales trends, and inventory management.

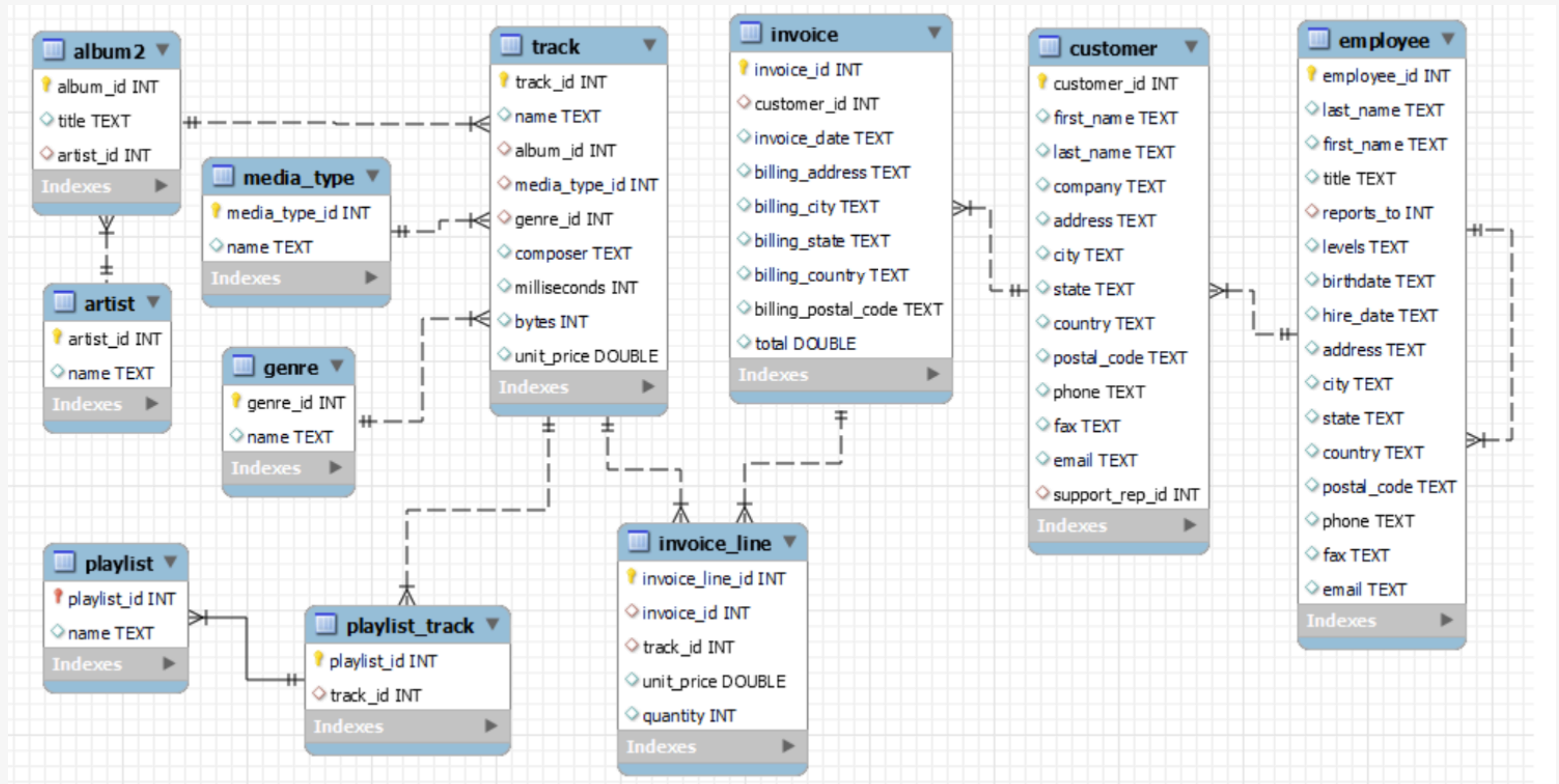


OBJECTIVE

- Analyze sales data to identify top-selling genres, artists, and albums.
- Determine customer purchasing patterns and preferences.
- Evaluate the effectiveness of marketing campaigns and promotions.
- Provide recommendations for inventory management and sales strategies.



Database





SQL

SQL

SQL

SQL

SQL

SQL

SQL

SQL

SQL

SQL

SQL

SQL

SQL

SQL

SQL

SQL

SQL

SQL

SQL

SQL

SQL



```
-- Who is the senior most employee based on job title
```

```
select * from employee
```

```
order by levels desc
```

```
limit 1;
```

Result Grid							
Filter Rows: <input type="text"/>							
Export: <input type="button" value="Export"/> Wrap Cell Content: <input type="checkbox"/>							
	employee_id	last_name	first_name	title	reports_to	levels	birthdate
▶	1	Adams	Andrew	General Manager	9	L6	18-02-1962 00:00



```
-- Which countries have the most Invoices?
```

```
SELECT
```

```
    COUNT(billing_country),
```

```
    billing_country AS most_invoices_country
```

```
FROM
```

```
    invoice
```

```
GROUP BY billing_country
```

```
ORDER BY most_invoices_country DESC;
```

Result Grid



Filter Rows:

Export:

	count(billing_country)	most_invoices_country
▶	131	USA
	28	United Kingdom
	10	Sweden
	11	Spain
	29	Portugal
	10	Poland
	9	Norway
	10	Netherlands
	9	Italy
	13	Ireland
	21	India
	10	Hungary
	41	Germany



-- What are top 3 values of total invoice?

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

	invoice_id	customer_id	invoice_date	billing_address	billing_city	billing_state
▶	183	42	2018-02-09 00:00:00	9, Place Louis Barthou	Bordeaux	None
	92	32	2017-07-02 00:00:00	696 Osborne Street	Winnipeg	MB
	526	5	2020-06-08 00:00:00	Klanova 9/506	Prague	None



```
1  -- Which city has the best customers?
2  -- We would like to throw a promotional Music Festival in the city we made the most money.
3  -- Write a query that returns one city that has the highest sum of invoice totals.
4  -- Return both the city name & sum of all invoice totals invoice
5
6 • select sum(total) as invoice_total, billing_city
7 from invoice
8 group by billing_city
9 order by invoice_total desc;
```

Result Grid			Filter Rows:	Export:	W
	invoice_total	billing_city			
▶	273.24000000000007	Prague			
	169.29	Mountain View			
	166.32	London			
	158.4	Berlin			
	151.47	Paris			
	129.69	SÃ£o Paulo			
	114.83999999999997	Dublin			
	111.86999999999999	Delhi			
	108.89999999999998	SÃ£o JosÃ© dos Campos			
	106.91999999999999	BrasÃ-lia			
	102.96000000000001	Lisbon			
	99.99	ordeaux			
	99.99	MontrÃ©al			
	98.01	Madrid			
	98.01	Redmond			



```
1  -- Who is the best customer?
2  -- The customer who has spent the most money will be declared the best customer.
3  -- Write a query that returns the person who has spent the most money
4
5  • SELECT
6      customer.customer_id,
7      customer.first_name,
8      customer.last_name,
9      SUM(invoice.total) AS spend_money
10 FROM
11     customer
12     JOIN
13     invoice ON customer.customer_id = invoice.customer_id
14 GROUP BY customer.customer_id , customer.first_name , customer.last_name
15 ORDER BY spend_money DESC
16 LIMIT 1;
```

Result Grid



Filter Rows:

Export:



Wrap Cell Content:



	customer_id	first_name	last_name	spend_money
▶	5	František	Wichterlovský	144.54000000000002



```
-- 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, genre.name AS Genre
```

```
FROM
```

```
    customer
```

```
    JOIN
```

```
    invoice ON invoice.customer_id = customer.customer_id
```

```
    JOIN
```

```
    invoice_line ON invoice_line.invoice_id = invoice.invoice_id
```

```
    JOIN
```

```
    track ON track.track_id = invoice_line.track_id
```

```
    JOIN
```

```
    genre ON genre.genre_id = track.genre_id
```

```
WHERE
```

```
    genre.name LIKE 'Rock'
```





```
ORDER BY email;
```

	email	first_name	last_name	Genre
▶	aaronmitchell@yahoo.ca	Aaron	Mitchell	Rock
	alero@uol.com.br	Alexandre	Rocha	Rock
	astrid.gruber@apple.at	Astrid	Gruber	Rock
	bjorn.hansen@yahoo.no	Björn	Hansen	Rock
	camille.bernard@yahoo.fr	Camille	Bernard	Rock
	daan_peeters@apple.be	Daan	Peeters	Rock
	diego.gutierrez@yahoo.ar	Diego	Gutiérrez	Rock



```
-- 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 number_of_songs  
FROM track  
JOIN album2 ON album2.album_id = track.album_id  
JOIN artist ON artist.artist_id = album2.artist_id  
JOIN genre ON genre.genre_id = track.genre_id  
WHERE genre.name LIKE 'Rock'  
GROUP BY artist.name , artist.artist_id  
ORDER BY number_of_songs DESC  
LIMIT 10;
```

Result Grid   Filter Rows: <input type="text"/> Export:  Wrap Cell Content: 			
	artist_id	name	number_of_songs
▶	1	AC/DC	18
	3	Aerosmith	15
	8	Audioslave	14
	22	Led Zeppelin	14
	4	Alanis Morissette	13
	5	Alice In Chains	12
	23	Frank Zappa & Captain Beefheart	9



```
-- 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.
```

SELECT

name, milliseconds

FROM

track

WHERE

milliseconds > (SELECT
AVG(milliseconds) AS avg_track_length
FROM
track)

ORDER BY milliseconds DESC;

Result Grid



Filter Rows:

Export:

	name	milliseconds
▶	How Many More Times	711836
	Advance Romance	677694
	Sleeping Village	644571
	You Shook Me(2)	619467
	Talkin' 'Bout Women Obviously	589531
	Stratus	582086
	No More Tears	555075



Music studio analysis



```
-- 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 album2 ON album2.album_id = track.album_id
    JOIN artist ON artist.artist_id = album2.artist_id
    GROUP BY 1,2
    ORDER BY total_sales 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 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 album2 alb ON alb.album_id = t.album_id
JOIN best_selling_artist bsa ON bsa.artist_id = alb.artist_id
GROUP BY c.customer_id, c.first_name, c.last_name, bsa.artist_name
ORDER BY amount_spent DESC;
```

	customer_id	first_name	last_name	artist_name	amount_spent
▶	54	Steve	Murray	AC/DC	17.82
	53	Phil	Hughes	AC/DC	10.89
	21	Kathy	Chase	AC/DC	10.89
	49	Stanisław	Wójcik	AC/DC	9.9
	1	Luís	Gonçalves	AC/DC	7.920000000000001
	24	Frank	Ralston	AC/DC	7.920000000000001
	31	Martha	Silk	AC/DC	3.96
	16	Frank	Harris	AC/DC	2.9699999999999998
	42	Wyatt	Girard	AC/DC	2.9699999999999998
	6	Helena	Holmér	AC/DC	2.9699999999999998
	38	Niklas	Schröder	AC/DC	2.9699999999999998
	35	Madalena	Sampaio	AC/DC	2.9699999999999998
	44	Terhi	Hämäläinen	AC/DC	2.9699999999999998
	9	Kara	Nielsen	AC/DC	1.98



```
-- 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) 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_genre WHERE RowNo <= 1;_
```

Result Grid						Filter Rows:	Export:	Write
	purchases	country	name	genre_id	RowNo			
▶	1	Argentina	Rock	1	1			
	18	Australia	Rock	1	1			
	6	Austria	Rock	1	1			
	5	Belgium	Rock	1	1			
	26	Brazil	Rock	1	1			
	57	Canada	Rock	1	1			
	7	Chile	Rock	1	1			
	14	Czech Republic	Rock	1	1			
	6	Denmark	Rock	1	1			
	6	Finland	Rock	1	1			
	26	France	Rock	1	1			
	28	Germany	Rock	1	1			
	4	Hungary	Rock	1	1			
	13	India	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 Customer_with_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_with_country WHERE RowNo <= 1;
```

Result Grid Filter Rows: Export: Wrap Cell Content:						
	customer_id	first_name	last_name	billing_country	total_spending	RowNo
▶	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	František	Wichterlov	Czech Republic	144.54000000000002	1
	9	Kara	Nielsen	Denmark	37.61999999999999	1
	44	Terhi	Hämäläinen	Finland	79.2	1
	42	Wyatt	Girard	France	99.99	1
	37	Fynn	Zimmermann	Germany	94.05000000000001	1
	45	Ladislav	Kovács	Hungary	78.21	1
	58	Manoj	Pareek	India	111.86999999999999	1
	46	Hugh	O'Reilly	Ireland	114.83999999999997	1
	47	Lucas	Mancini	Italy	50.49	1
	48	Johannes	Van der Berg	Netherlands	65.34	1
	4	Bjørn	Hansen	Norway	72.27000000000001	1



THANK YOU



in/nilam-gavate-6b14b51a4



nilamgavate@gmail.com