Music Store Data Analysis

USING SQL

By Shivam Raj

SCHEMAS

ALBUM
AlbumId
Title
ArtistId

CUSTOMER
CustomerId
Firstname
LastName
Company
Address
City
State
Country
PostalCode
Phone
Fax
Email
SupportRepId

PLAYLIST TRACK
PlaylistId
TrackId

TRACK
TrackId
Name
AlbumId
MediaTypeId
GenreId
Composer
Milliseconds
Bytes
UnitPrice

EMPLOYEE
EmployeeId
LastName
FirstName
Title
ReportsTo
BirthDate
HireDate
Address
City
State
Country
PostalCode
Phone
Fax
Email

INVOICE LINE
InvoiceLineId
InvoiceId
TrackId
UnitPrice
Quantity

INVOICE
InvoiceId
CustomerId
InvoiceDate
BillingAddress
BillingCity
BillingState
BillingCountry
BillingPostalCode
Total

INVOICE LINE
InvoiceLineId
InvoiceId
TrackId
UnitPrice
Quantity

ARTIST
ArtistId
Name

PLAYLIST
PlaylistId
TrackId

GENRE
GenreId
Name

MEDIA TYPE
MediatypeId
Name

Introduction

In the ever-evolving world of music, understanding customer preferences and buying habits is crucial for online music stores to thrive.

Welcome to the SQL Project for analysing an online music store's database. This project will help the music store understand its business growth and customer preferences by answering various questions based on the data. The dataset includes information about employees, customers, invoices, tracks, and artists.

By performing SQL queries on this data, we will gain insights into key aspects of the business, such as employee hierarchy, customer behaviour, and popular music genres.

analysis 1

Query | Query History

```
--Q1: 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 (50)	last_name character	first_name character	title character varying (50)	reports_to character varying (30
1	9	Madan	Mohan	Senior General Manager	[null]

Total rows: 1 of 1 | Query complete 00:00:00.204

analysis 2

Query | Query History

```sql
--Q2: Which countries have the most Invoices?
select count(*) as c, billing_country
from invoice
group by billing_country
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
12	11	Spain
13	11	Finland
14	10	Australia
15	10	Netherlands
16	10	Sweden

Total rows: 24 of 24 | Query complete 00:00:00.065

analysis 3

Query Query History

```
--Q3: 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

analysis 4

Query | Query History

```sql
--Q4: 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
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
14	98.01	Madrid
15	98.01	Redmond

Total rows: 53 of 53 | Query complete 00:00:00.049

analysis 5

Query Query History

```
--Q5: 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

Total rows: 1 of 1 Query complete 00:00:00.066

analysis 6

Query | Query History

```
--Q6: Write query to return the email, first name, last name, & Genre of all Rock Music listners.
--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
6	daan_peeters@apple.be	Daan	Peeters
7	diego.gutierrez@yahoo.ar	Diego	Gutiérrez
8	dmiller@comcast.com	Dan	Miller
9	dominiquelefebvre@gmail.c...	Dominique	Lefebvre
10	edfrancis@yachoo.ca	Edward	Francis
11	eduardo@woodstock.com.br	Eduardo	Martins
12	ellie.sullivan@shaw.ca	Ellie	Sullivan

Total rows: 59 of 59 | Query complete 00:00:00.069

analysis 7

Query | Query History

```sql
--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.
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

Total rows: 10 of 10 | Query complete 00:00:00.068

analysis 8

Query | Query History

```
--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.
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
7	Murder On the Rising Star	2935894
8	Battlestar Galactica, Pt. 3	2927802
9	Take the Celestra	2927677
10	Fire In Space	2926593
11	The Long Patrol	2925008
12	The Magnificent Warriors	2924716
13	The Living Legend, Pt. 1	2924507
14	The Gun On Ice Planet Zero, Pt. 2	2924341

Total rows: 494 of 494 | Query complete 00:00:00.080

analysis 9

Query | Query History

```
--Q9: 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_artists 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 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 album alb on alb.album_id = t.album_id
join best_selling_artists bsa on bsa.artist_id = alb.artist_id
group by 1, 2, 3, 4
order by 5 desc;
```

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	Francois	Tremblay	Queen	17.82

Total rows: 43 of 43 | Query complete 00:00:00.043

analysis 10

Query | Query History

```
--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 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.
--Method-1:
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;
```

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
4	26	Belgium	Rock	1	1
5	205	Brazil	Rock	1	1
6	333	Canada	Rock	1	1
7	61	Chile	Rock	1	1

Total rows: 24 of 24 | Query complete 00:00:00.046

analysis 11

Query | Query History

```sql
--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 amout.
--Method-1:
with recursive
    customer_with_country as(
        select customer.customer_id, first_name, last_name, billing_country, sum(total) as total_spending
        from invoice
        join customer on customer.customer_id = invoice.customer_id
        group by 1, 2, 3, 4
        order by 2, 3 desc),

    country_max_spending as(
        select billing_country, max(total_spending) as max_spending
        from customer_with_country
        group by billing_country)

select cc.billing_country, cc.total_spending, cc.first_name, cc.last_name, cc.customer_id
from customer_with_country cc
join country_max_spending ms
on cc.billing_country = ms.billing_country
where cc.total_spending = ms.max_spending
order by 1;
```

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
2	Australia	81.18	Mark	Taylor	55

Total rows: 24 of 24 | Query complete 00:00:00.062

Thank You