**Learners have to come up with a Report to support the answers to the following questions and suggestions**

Objective Questions

1. Does any table have missing values or duplicates? If yes how would you handle it ?

Query-

-- Checking duplicate and null values in table album--

-- Duplicate--

Select \* from album

Select album\_id, COUNT(album\_id) AS count FROM album GROUP BY album\_id HAVING count > 1;

SELECT title, COUNT(title) AS count FROM album GROUP BY title HAVING count > 1;

SELECT artist\_id, COUNT(artist\_id) AS count FROM album GROUP BY artist\_id HAVING count > 1;

-- Null --

SELECT album\_id, count(\*) FROM album WHERE album\_id IS NULL group by album\_id;

SELECT title, count(\*) FROM album WHERE title IS NULL group by title;

SELECT artist\_id, count(\*) FROM album WHERE artist\_id IS NULL group by artist\_id;

-- Checking duplicate and null values in table artist--

-- Duplicate--

Select \* from artist

Select artist\_id, COUNT(artist\_id) AS count FROM artist GROUP BY artist\_id HAVING count > 1;

Select name, COUNT(name) AS count FROM artist GROUP BY name HAVING count > 1;

-- Null --

SELECT artist\_id, count(\*) FROM artist WHERE artist\_id IS NULL group by artist\_id;

SELECT name, count(\*) FROM artist WHERE name IS NULL group by name;

-- Checking duplicate and null values in table customer--

-- Duplicate--

Select \* from customer

Select customer\_id, COUNT(customer\_id) AS count FROM customer GROUP BY customer\_id HAVING count > 1;

Select first\_name, COUNT(first\_name) AS count FROM customer GROUP BY first\_name HAVING count > 1;

Select last\_name, COUNT(last\_name) AS count FROM customer GROUP BY last\_name HAVING count > 1;

Select company, COUNT(company) AS count FROM customer GROUP BY company HAVING count > 1;

Select address, COUNT(address) AS count FROM customer GROUP BY address HAVING count > 1;

Select city, COUNT(city) AS count FROM customer GROUP BY city HAVING count > 1;

Select state, COUNT(state) AS count FROM customer GROUP BY state HAVING count > 1;

Select country, COUNT(country) AS count FROM customer GROUP BY country HAVING count > 1;

Select postal\_code, COUNT(postal\_code) AS count FROM customer GROUP BY postal\_code HAVING count > 1;

Select phone, COUNT(phone) AS count FROM customer GROUP BY phone HAVING count > 1;

Select fax, COUNT(fax) AS count FROM customer GROUP BY fax HAVING count > 1;

Select email, COUNT(email) AS count FROM customer GROUP BY email HAVING count > 1;

Select support\_rep\_id, COUNT(support\_rep\_id) AS count FROM customer GROUP BY support\_rep\_id HAVING count > 1;

-- Null --

SELECT customer\_id, count(\*) FROM customer WHERE customer\_id IS NULL group by customer\_id;

SELECT first\_name, count(\*) FROM customer WHERE first\_name IS NULL group by first\_name;

SELECT last\_name, count(\*) FROM customer WHERE last\_name IS NULL group by last\_name;

SELECT company, count(\*) FROM customer WHERE company IS NULL group by company;

SELECT address, count(\*) FROM customer WHERE address IS NULL group by address;

SELECT city, count(\*) FROM customer WHERE city IS NULL group by city;

SELECT state, count(\*) FROM customer WHERE state IS NULL group by state;

SELECT country, count(\*) FROM customer WHERE country IS NULL group by country;

SELECT postal\_code, count(\*) FROM customer WHERE postal\_code IS NULL group by postal\_code;

SELECT phone, count(\*) FROM customer WHERE phone IS NULL group by phone;

SELECT fax, count(\*) FROM customer WHERE fax IS NULL group by fax;

SELECT email, count(\*) FROM customer WHERE email IS NULL group by email;

SELECT support\_rep\_id, count(\*) FROM customer WHERE support\_rep\_id IS NULL group by support\_rep\_id;

-- Checking duplicate and null values in table employee--

-- Duplicate--

Select \* from employee

Select employee\_id, COUNT(employee\_id) AS count FROM employee GROUP BY employee\_id HAVING count > 1;

Select last\_name, COUNT(last\_name) AS count FROM employee GROUP BY last\_name HAVING count > 1;

Select first\_name, COUNT(first\_name) AS count FROM employee GROUP BY first\_name HAVING count > 1;

Select title, COUNT(title) AS count FROM employee GROUP BY title HAVING count > 1;

Select reports\_to, COUNT(reports\_to) AS count FROM employee GROUP BY reports\_to HAVING count > 1;

Select birthdate, COUNT(birthdate) AS count FROM employee GROUP BY birthdate HAVING count > 1;

Select hire\_date, COUNT(hire\_date) AS count FROM employee GROUP BY hire\_date HAVING count > 1;

Select address, COUNT(address) AS count FROM employee GROUP BY address HAVING count > 1;

Select city, COUNT(city) AS count FROM employee GROUP BY city HAVING count > 1;

Select state, COUNT(state) AS count FROM employee GROUP BY state HAVING count > 1;

Select country, COUNT(country) AS count FROM employee GROUP BY country HAVING count > 1;

Select postal\_code, COUNT(postal\_code) AS count FROM employee GROUP BY postal\_code HAVING count > 1;

Select phone, COUNT(phone) AS count FROM employee GROUP BY phone HAVING count > 1;

Select fax, COUNT(fax) AS count FROM employee GROUP BY fax HAVING count > 1;

Select email, COUNT(email) AS count FROM employee GROUP BY email HAVING count > 1;

-- Null --

SELECT employee\_id, count(\*) FROM employee WHERE employee\_id IS NULL group by employee\_id;

SELECT last\_name, count(\*) FROM employee WHERE last\_name IS NULL group by last\_name;

SELECT first\_name, count(\*) FROM employee WHERE first\_name IS NULL group by first\_name;

SELECT title, count(\*) FROM employee WHERE title IS NULL group by title;

SELECT reports\_to, count(\*) FROM employee WHERE reports\_to IS NULL group by reports\_to;

SELECT birthdate, count(\*) FROM employee WHERE birthdate IS NULL group by birthdate;

SELECT hire\_date, count(\*) FROM employee WHERE hire\_date IS NULL group by hire\_date;

SELECT address, count(\*) FROM employee WHERE address IS NULL group by address;

SELECT city, count(\*) FROM employee WHERE city IS NULL group by city;

SELECT state, count(\*) FROM employee WHERE state IS NULL group by state;

SELECT country, count(\*) FROM employee WHERE country IS NULL group by country;

SELECT postal\_code, count(\*) FROM employee WHERE postal\_code IS NULL group by postal\_code;

SELECT phone, count(\*) FROM employee WHERE phone IS NULL group by phone;

SELECT fax, count(\*) FROM employee WHERE fax IS NULL group by fax;

SELECT email, count(\*) FROM employee WHERE email IS NULL group by email;

-- Checking duplicate and null values in table genre--

-- Duplicate--

Select \* from genre

Select genre\_id, COUNT(genre\_id) AS count FROM genre GROUP BY genre\_id HAVING count > 1;

SELECT name, COUNT(name) AS count FROM genre GROUP BY name HAVING count > 1;

-- Null --

SELECT genre\_id, count(\*) FROM genre WHERE genre\_id IS NULL group by genre\_id;

SELECT name, count(\*) FROM genre WHERE name IS NULL group by name;

-- Checking duplicate and null values in table invoice--

-- Duplicate--

Select \* from invoice

Select invoice\_id, COUNT(invoice\_id) AS count FROM invoice GROUP BY invoice\_id HAVING count > 1;

Select customer\_id, COUNT(customer\_id) AS count FROM invoice GROUP BY customer\_id HAVING count > 1;

Select invoice\_date, COUNT(invoice\_date) AS count FROM invoice GROUP BY invoice\_date HAVING count > 1;

Select billing\_address, COUNT(billing\_address) AS count FROM invoice GROUP BY billing\_address HAVING count > 1;

Select billing\_city, COUNT(billing\_city) AS count FROM invoice GROUP BY billing\_city HAVING count > 1;

Select billing\_state, COUNT(billing\_state) AS count FROM invoice GROUP BY billing\_state HAVING count > 1;

Select billing\_country, COUNT(billing\_country) AS count FROM invoice GROUP BY billing\_country HAVING count > 1;

Select billing\_postal\_code, COUNT(billing\_postal\_code) AS count FROM invoice GROUP BY billing\_postal\_code HAVING count > 1;

Select total, COUNT(total) AS count FROM invoice GROUP BY total HAVING count > 1;

-- Null --

SELECT invoice\_id, count(\*) FROM invoice WHERE invoice\_id IS NULL group by invoice\_id;

SELECT customer\_id, count(\*) FROM invoice WHERE customer\_id IS NULL group by customer\_id;

SELECT invoice\_date, count(\*) FROM invoice WHERE invoice\_date IS NULL group by invoice\_date;

SELECT billing\_address, count(\*) FROM invoice WHERE billing\_address IS NULL group by billing\_address;

SELECT billing\_city, count(\*) FROM invoice WHERE billing\_city IS NULL group by billing\_city;

SELECT billing\_state, count(\*) FROM invoice WHERE billing\_state IS NULL group by billing\_state;

SELECT billing\_country, count(\*) FROM invoice WHERE billing\_country IS NULL group by billing\_country;

SELECT billing\_postal\_code, count(\*) FROM invoice WHERE billing\_postal\_code IS NULL group by billing\_postal\_code;

SELECT total, count(\*) FROM invoice WHERE total IS NULL group by total;

-- Checking duplicate and null values in table invoice\_line--

-- Duplicate--

Select \* from invoice\_line

Select invoice\_line\_id, COUNT(invoice\_line\_id) AS count FROM invoice\_line GROUP BY invoice\_line\_id HAVING count > 1;

Select invoice\_id, COUNT(invoice\_id) AS count FROM invoice\_line GROUP BY invoice\_id HAVING count > 1;

Select track\_id, COUNT(track\_id) AS count FROM invoice\_line GROUP BY track\_id HAVING count > 1;

Select unit\_price, COUNT(unit\_price) AS count FROM invoice\_line GROUP BY unit\_price HAVING count > 1;

Select quantity, COUNT(quantity) AS count FROM invoice\_line GROUP BY quantity HAVING count > 1;

-- Null --

SELECT invoice\_line\_id, count(\*) FROM invoice\_line WHERE invoice\_line\_id IS NULL group by invoice\_line\_id;

SELECT invoice\_id, count(\*) FROM invoice\_line WHERE invoice\_id IS NULL group by invoice\_id;

SELECT track\_id, count(\*) FROM invoice\_line WHERE track\_id IS NULL group by track\_id;

SELECT unit\_price, count(\*) FROM invoice\_line WHERE unit\_price IS NULL group by unit\_price;

SELECT quantity, count(\*) FROM invoice\_line WHERE quantity IS NULL group by quantity;

-- Checking duplicate and null values in table media\_type--

-- Duplicate--

Select \* from media\_type

Select media\_type\_id, COUNT(media\_type\_id) AS count FROM media\_type GROUP BY media\_type\_id HAVING count > 1;

Select name, COUNT(name) AS count FROM media\_type GROUP BY name HAVING count > 1;

-- Null --

SELECT media\_type\_id, count(\*) FROM media\_type WHERE media\_type\_id IS NULL group by media\_type\_id;

SELECT name, count(\*) FROM media\_type WHERE name IS NULL group by name;

-- Checking duplicate and null values in table playlist--

-- Duplicate--

Select \* from playlist

Select playlist\_id, COUNT(playlist\_id) AS count FROM playlist GROUP BY playlist\_id HAVING count > 1;

Select name, COUNT(name) AS count FROM playlist GROUP BY name HAVING count > 1;

-- Null --

SELECT playlist\_id, count(\*) FROM playlist WHERE playlist\_id IS NULL group by playlist\_id;

SELECT name, count(\*) FROM playlist WHERE name IS NULL group by name;

-- Checking duplicate and null values in table playlist\_track--

-- Duplicate--

Select \* from playlist\_track

Select playlist\_id, COUNT(playlist\_id) AS count FROM playlist\_track GROUP BY playlist\_id HAVING count > 1;

Select track\_id, COUNT(track\_id) AS count FROM playlist\_track GROUP BY track\_id HAVING count > 1;

-- Null --

SELECT playlist\_id, count(\*) FROM playlist\_track WHERE playlist\_id IS NULL group by playlist\_id;

SELECT track\_id, count(\*) FROM playlist\_track WHERE track\_id IS NULL group by track\_id;

-- Checking duplicate and null values in table track--

-- Duplicate--

Select \* from track

Select track\_id, COUNT(track\_id) AS count FROM track GROUP BY track\_id HAVING count > 1;

Select name, COUNT(name) AS count FROM track GROUP BY name HAVING count > 1;

Select album\_id, COUNT(album\_id) AS count FROM track GROUP BY album\_id HAVING count > 1;

Select media\_type\_id, COUNT(media\_type\_id) AS count FROM track GROUP BY media\_type\_id HAVING count > 1;

Select genre\_id, COUNT(genre\_id) AS count FROM track GROUP BY genre\_id HAVING count > 1;

Select composer, COUNT(composer) AS count FROM track GROUP BY composer HAVING count > 1;

Select milliseconds, COUNT(milliseconds) AS count FROM track GROUP BY milliseconds HAVING count > 1;

Select bytes, COUNT(bytes) AS count FROM track GROUP BY bytes HAVING count > 1;

Select unit\_price, COUNT(unit\_price) AS count FROM track GROUP BY unit\_price HAVING count > 1;

-- Null --

SELECT track\_id, count(\*) FROM track WHERE track\_id IS NULL group by track\_id;

SELECT name, count(\*) FROM track WHERE name IS NULL group by name;

SELECT album\_id, count(\*) FROM track WHERE album\_id IS NULL group by album\_id;

SELECT media\_type\_id, count(\*) FROM track WHERE media\_type\_id IS NULL group by media\_type\_id;

SELECT genre\_id, count(\*) FROM track WHERE genre\_id IS NULL group by genre\_id;

SELECT composer, count(\*) FROM track WHERE composer IS NULL group by composer;

SELECT milliseconds, count(\*) FROM track WHERE milliseconds IS NULL group by milliseconds;

SELECT bytes, count(\*) FROM track WHERE bytes IS NULL group by bytes;

SELECT unit\_price, count(\*) FROM track WHERE unit\_price IS NULL group by unit\_price;

To find the missing values or duplicates-

Table: album

|  |  |  |
| --- | --- | --- |
| Column name | Null | Duplicate |
| album\_id | 0 | 0 |
| title | 0 | 1 |
| artist\_id | 0 | 57 |

Explanation-

* No null value found in any column of table album.
* 1 duplicate value found in title column- Minha Historia
* 57 duplicate rows found in artist\_id

Analysis-

* Though the title was repeated twice, but the artist were different as it has two different artist id. So the it cant be ignored.
* 57 distinct artist id have been repeated but the title is different, so it cant be ignored. It means different artist sang different songs.
* We need to consider all the duplicate values because the title, and artist id are not coinciding, so none of the values can be ignored.

Table: artist

|  |  |  |
| --- | --- | --- |
| Column name | Null | Duplicate |
| artist\_id | 0 | 0 |
| name | 0 | 0 |

Explanation-

* No duplicate or null value found.

Analysis-

* No null or duplicate value was found.
* If found, we can use: Case statements or Coalesce to remove nulls. For duplicates, we can match the primary key and if it is same, we can combine the rows as one.

Table: customer

|  |  |  |
| --- | --- | --- |
| Column name | Null | Duplicate |
| customer\_id | 0 | 0 |
| first\_name | 0 | 3 |
| last\_name | 0 | 0 |
| company | 49 | 0 |
| address | 0 | 0 |
| city | 0 | 6 |
| state | 29 | 3 |
| country | 0 | 9 |
| postal code | 4 | 0 |
| phone | 1 | 0 |
| fax | 47 | 0 |
| email | 0 | 0 |
| support\_rep\_id | 0 | 3 |

Explanation-

* Duplicate values were found in first name, city, state, country, support rep id. Since none of them is a primary key and it is possible to have similar information of first name, city, state and country, so it cannot be combined or ignored.
* Null values were found in company, state and fax.

Analysis-

* It is very common to have same first name, also for city, state, and country, many people with same geographic details prefer same song, so we cant combine the data, it has to be treated as different rows.
* For null values in company, state, and fax, we can use coalesce to remove null and add values like- No information available etc.

Table: employee

|  |  |  |
| --- | --- | --- |
| Column name | Null | Duplicate |
| employee\_id | 0 | 0 |
| last\_name | 0 | 0 |
| first\_name | 0 | 0 |
| title | 0 | 2 |
| reports\_to | 1 | 3 |
| birth\_date | 0 | 0 |
| hire\_date | 0 | 0 |
| address | 0 | 0 |
| city | 0 | 2 |
| state | 0 | 1 |
| country | 0 | 1 |
| postal\_code | 0 | 0 |
| phone | 0 | 1 |
| fax | 0 | 0 |
| email | 0 | 0 |

Explanation-

* Duplicate values were found in title, reports to, city, state, country, and phone. Since the table contains employee data, it is possible to have multiple employees at a single designation, multiple employees reporting to same manager, and also people hailing from same city, state or country.
* Null value was found in reports\_to column. The person is the general manager and is not reporting to anyone, hence it has null value in reports\_to.

Analysis-

* It is very common to have same title, city, state, country and phone in the same company.
* For the null value, we can use coalesce and add any value like ‘Not reporting to anyone’, for better clarity.

Table: genre

|  |  |  |
| --- | --- | --- |
| Column name | Null | Duplicate |
| genre\_id | 0 | 0 |
| name | 0 | 0 |

Explanation-

* No duplicate or null value found.

Analysis-

* No null or duplicate value was found.
* This shows no genre were repeated.
* If found, we can use: Case statements or Coalesce to remove nulls. For duplicates, we can match the primary key and if it is same, we can combine the rows as one.

Table: invoice

|  |  |  |
| --- | --- | --- |
| Column name | Null | Duplicate |
| invoice\_id | 0 | 0 |
| customer\_id | 0 | 59 |
| invoice\_date | 0 | 100 |
| billing\_address | 0 | 59 |
| billing\_city | 0 | 53 |
| billing\_state | 0 | 26 |
| billing\_country | 0 | 24 |
| billing\_postal\_code | 0 | 56 |
| total | 0 | 19 |

Explanation-

* The primary key, ie, invoice id has no duplicates but rest of the coulmns have duplicates, this shows same customers have made more than 1 purchase.
* No null value present.

Analysis-

* Same customers have made more than 1 purchase which lead to dup-licate values. The difference in count of duplicate values of billing\_address, billing\_city, billing\_state, billing\_country, billing\_postal\_code is because none has been added instead of Null values.
* The company has provided none in place of null values.

Table: invoice\_line

|  |  |  |
| --- | --- | --- |
| Column name | Null | Duplicate |
| invoice\_line\_id | 0 | 0 |
| invoice\_id | 0 | 576 |
| track\_id | 0 | 1000 |
| unit\_price | 0 | 1 |
| quantity | 0 | 1 |

Explanation-

* The primary key, ie, invoice\_line\_id has no duplicates but rest of the coulmns have duplicates. Invoice\_id has a total of 576 duplicates but no invoice\_line\_id has been repeated it means it is for different line of product. Track\_id has duplicates but has been duplicated for different invoice\_line\_id. All the product has the same cost and only 1 item has been sold.
* No null value present.

Analysis-

* The primary key is not repeated, which shows that it belongs to different line of product.
* No null value found.

Table: media\_type

|  |  |  |
| --- | --- | --- |
| Column name | Null | Duplicate |
| media\_type\_id | 0 | 0 |
| name | 0 | 0 |

Explanation-

* No duplicate or null value found.

Analysis-

* No null or duplicate value was found.
* If found, we can use: Case statements or Coalesce to remove nulls. For duplicates, we can match the primary key and if it is same, we can combine the rows as one.

Table: playlist

|  |  |  |
| --- | --- | --- |
| Column name | Null | Duplicate |
| playlist\_id | 0 | 0 |
| name | 0 | 4 |

Explanation-

* No null value found.
* The primary key, ie, the playlist\_id has no duplicates but title of the playlist has duplicate

Analysis-

* No null value was found.
* Since the primary key is distinct, all the title will be treated as different rows.

Table: playlist\_track

|  |  |  |
| --- | --- | --- |
| Column name | Null | Duplicate |
| playlist\_id | 0 | 12 |
| track\_id | 0 | 1000 |

Explanation-

* No null value found.
* This consist of different playlist with their playlist\_id which contains different tracks.

Analysis-

* No null value was found.
* Different playlist contains different tracks. Playlist are duplicated because many tracks belong to same playlist and vice versa.

Table: track

|  |  |  |
| --- | --- | --- |
| Column name | Null | Duplicate |
| track\_id | 0 | 0 |
| name | 0 | 206 |
| album\_id | 0 | 265 |
| media\_type\_id | 0 | 5 |
| genre\_id | 0 | 24 |
| composer | 1 | 288 |
| milliseconds | 0 | 381 |
| bytes | 0 | 2 |
| unit\_price | 0 | 2 |

Explanation-

* No null value found.
* This table consists of information about different tracks. Name, album\_id, media\_type\_id, genre\_id, composer, milliseconds, bytes, and unit\_price columns has duplicate values.

Analysis-

* No null value was found.
* Name, album\_id, media\_type\_id, genre\_id, composer, milliseconds, bytes, and unit\_price columns has duplicate values but since the primary key track\_id has no duplicates hence it will be treated as different values and will not be merged.

1. Find the top-selling tracks and top artist in the USA and identify their most famous genres.

Query-

SELECT

t.name AS Track\_Name,

ar.name AS Artist\_Name,

g.name AS Genre,

SUM(il.quantity) AS Total\_Quantity\_Sold

FROM track t

JOIN album al ON t.album\_id = al.album\_id

JOIN artist ar ON al.artist\_id = ar.artist\_id

JOIN genre g ON t.genre\_id = g.genre\_id

JOIN invoice\_line il ON t.track\_id = il.track\_id

JOIN invoice i ON il.invoice\_id = i.invoice\_id

JOIN customer c ON i.customer\_id = c.customer\_id

WHERE c.country = 'USA'

GROUP BY t.name, ar.name, g.name

ORDER BY Total\_Quantity\_Sold DESC

LIMIT 10;

Method-

* Join the tables- track, album, artist, genre, invoice\_line, invoice, customer
* Filter the customer country to USA
* Group the result by tracks, artist and genre
* Order by total quantity sold in desc order to show top sellers
* Put a limit of 10

Analysis-



Insights-

* Rock is the most popular genre being sold in the USA.
* War Pig with 6 units by Cake is the most sold track in the USA.
* Artists like Nirvana, The Door appear more than once in the list, which shows their high popularity in the USA.

1. What is the customer demographic breakdown (age, gender, location) of Chinook's customer base?

Query-

-- \*\*Query 1 \*\* --

SELECT CONCAT(first\_name, ' ', last\_name) AS full\_name, city, state, country

FROM customer

GROUP BY first\_name, last\_name, city, state, country;

-- \*\*Query 2 \*\* --

SELECT country, COUNT(\*) AS customer\_count

FROM customer

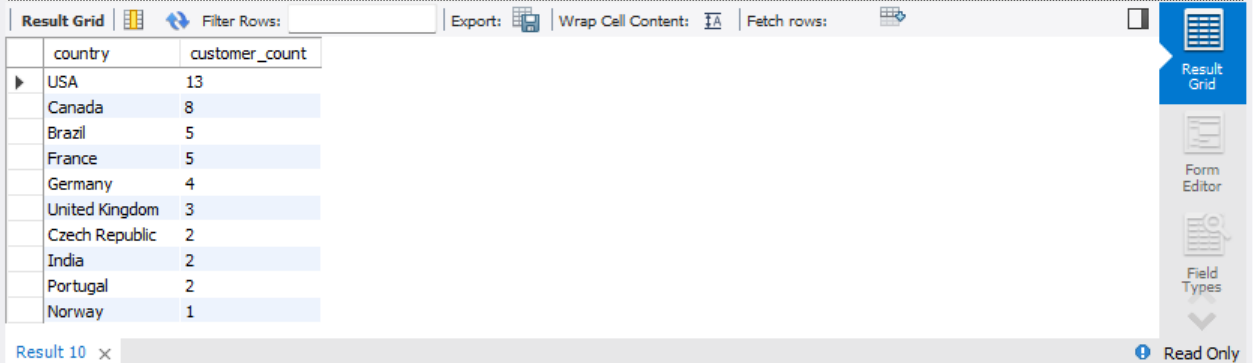
GROUP BY country

order by customer\_count desc

limit 10;

Method-

* Since no date of birth and gender is provided, we can only make the breakdown with the name, city, state and country.
* There are two queries written, the first query gives the detail information of the customer, its full name, city, state and country, whereas the second query tells us count of customer in each country. In the second query arrange the count of customers in descending order by appling the limit of 10.

Analysis-

Insights-

* The majority of customer base of chinook is from the USA following behind with Canada, Brazil and France.

1. Calculate the total revenue and number of invoices for each country, state, and city:

Query

Query 1- Total Revenue by Country, State, and City:

SELECT billing\_country, billing\_state, billing\_city, SUM(total) AS total\_revenue

FROM invoice

GROUP BY billing\_country, billing\_state, billing\_city

ORDER BY total\_revenue DESC limit 10;

Query 2- Number of invoices for each country, state, and city. --

SELECT billing\_country, billing\_state, billing\_city, COUNT(invoice\_id) AS number\_of\_invoices

FROM invoice

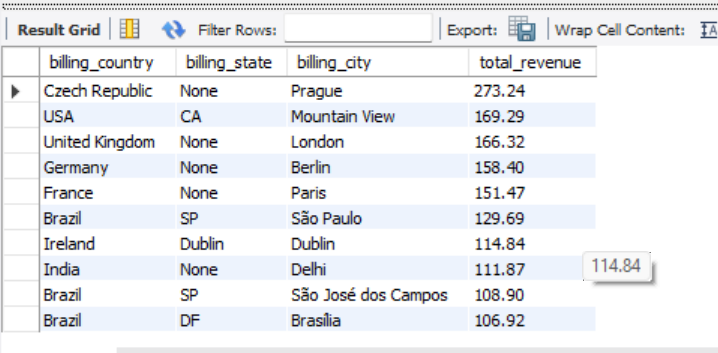
GROUP BY billing\_country, billing\_state, billing\_city

ORDER BY number\_of\_invoices DESC LIMIT 10;

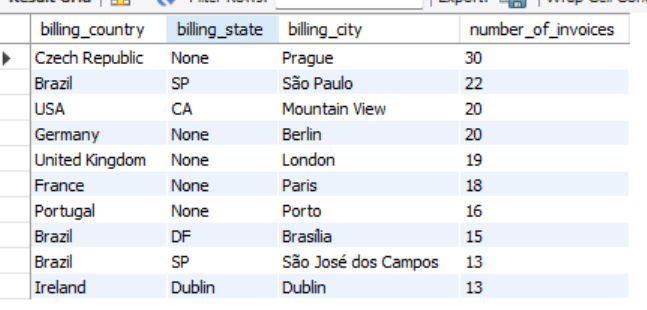
Method-

* Two separate queries have been used for total revenue and the number of invoices
* Query 1- Total Revenue
* Query 2- Number of invoices
* Billing country, state, and city have been selected with the sum of the total column to find out the total revenue for each country, state and city. The result has been grouped as per billing country, state and city and has been arranged in descending order of the sum to find the highest revenues with a limit of 10.
* Billing country, state, and city have been selected with the count of invoice id to find out the number of invoices for each country, state and city. The result has been grouped as per billing country, state and city and has been arranged in descending order of the count to find the highest number of invoices with a limit of 10.

Analysis-

Query 1- Total Revenue by Country, State, and City:

Query 2- Number of invoices for each country, state, and city.



Insights-

* Czech Republic (City- Prague), USA (State- CA, City- Mountain View), United Kingdom (City- London) has the highest total revenue.
* Czech Republic (City- Prague), Brazil (State- SP, City- Sao Paulo), USA (State- CA, City- Mountain View) has the highest number of invoices.

1. Find the top 5 customers by total revenue in each country

Query-

SELECT c.customer\_id, concat(c.first\_name,' ',c.last\_name) as full\_name, c.country,

SUM(i.total) AS total\_revenue

FROM customer c

JOIN invoice i ON c.customer\_id = i.customer\_id

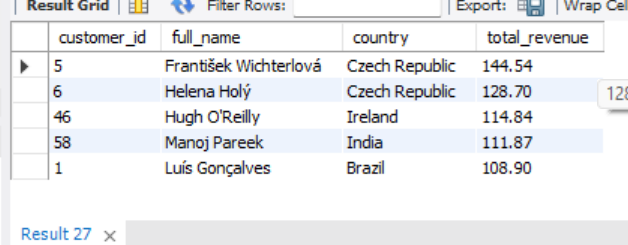
GROUP BY c.customer\_id,concat(c.first\_name,' ',c.last\_name), c.country

ORDER BY total\_revenue DESC, c.country

limit 5;

Method-

* Identify the required tables- Customer and invoice
* We will join both tables on customer id.
* Total revenue will be taken from invoice table and customer id, country, first and last name will be taken from customer table.
* The result has to be grouped on customer id, country and full name for better clarity.
* As per the question limit has been set to 5 and the revenue has been arranged in descending order to obtain the customer which has generated the highest revenue.

Analysis-

Insights-

* Customer with customer id 5 of country Czech Republic has generated the highest revenue.
* Customer with customer id 6 of country Czech Republic has generated the second highest revenue.
* Customer with customer id 46 of country Ireland has generated the third highest revenue.
* Customer with customer id 58 of country India has generated the fourth highest revenue.
* Customer with customer id 1 of country Brazil has generated the fifth highest revenue.

1. Identify the top-selling track for each customer

Query-

WITH Cte AS (

SELECT c.customer\_id, c.first\_name, c.last\_name, t.track\_id, t.name AS track\_name,

SUM(il.unit\_price \* il.quantity) AS total\_revenue,

ROW\_NUMBER() OVER (PARTITION BY c.customer\_id ORDER BY SUM(il.unit\_price \* il.quantity) DESC) AS revenue\_rank

FROM customer c

JOIN invoice i ON c.customer\_id = i.customer\_id

JOIN invoice\_line il ON i.invoice\_id = il.invoice\_id

JOIN track t ON il.track\_id = t.track\_id

GROUP BY c.customer\_id, c.first\_name, c.last\_name, t.track\_id, t.name

)

SELECT customer\_id, first\_name, last\_name, track\_id, track\_name, total\_revenue

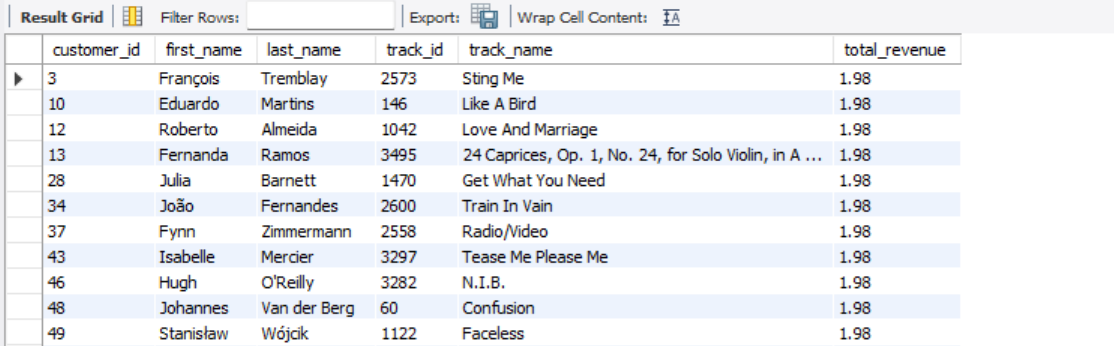
FROM Cte

WHERE revenue\_rank = 1

ORDER BY total\_revenue desc, customer\_id;

Method-

* Start by identifying the necessary tables.
* We will use common table expression for ease of work.
* Join the invoice\_line table with the invoice table using the invoice\_id to obtain the corresponding customer\_id.
* Next, link this result with the customer table via the customer\_id to retrieve customer details.
* Then, join this combined data with the track table using the track\_id to get track-related information.
* Group the data by customer\_id and track\_id to analyze track purchases per customer.
* Calculate the total quantity sold for each track by summing the quantity column.
* Assign ranks to the tracks for each customer based on total quantity sold, and sort it in descending order.
* Finally, filter the result to show only the top track (highest sales) for each customer.
* In the end order it by total revenue in descending order.

Analysis*:*

Insights-

* Customer 3 has purchased track 2573 twice, making it their top-selling track.
* Customer 10 has purchased track 146 twice, making it their top-selling track.
* Customer 12 has purchased track 1042 twice, making it their top-selling track.
* Customer 13 has purchased track 3495 twice, making it their top-selling track.
* Customer 28 has purchased track 1470 twice, making it their top-selling track.
* Customer 34 has purchased track 2600 twice, making it their top-selling track.
* Customer 37 has purchased track 2558 twice, making it their top-selling track.
* Customer 43 has purchased track 3297 twice, making it their top-selling track.
* Customer 46 has purchased track 3282 twice, making it their top-selling track.
* Customer 48 has purchased track 60 twice, making it their top-selling track.
* Customer 49 has purchased track 1122 twice, making it their top-selling track.

1. Are there any patterns or trends in customer purchasing behavior (e.g., frequency of purchases, preferred payment methods, average order value)?

Query-

Frequency of purchases-

SELECT c.customer\_id,c.first\_name,c.last\_name,

COUNT(i.invoice\_id) AS purchase\_frequency

FROM customer c

JOIN invoice i ON c.customer\_id = i.customer\_id

GROUP BY c.customer\_id

ORDER BY purchase\_frequency DESC

limit 10;

Preferred payment methods-

No information Provided for payment method

Average order value-

SELECT c.customer\_id,c.first\_name,c.last\_name,

Round(AVG(i.total)) AS average\_order\_value

FROM customer c

JOIN invoice i ON c.customer\_id = i.customer\_id

GROUP BY c.customer\_id

ORDER BY average\_order\_value DESC

limit 10;

Method-

-- Frequency of purchases –

* Identify the table containing required information and join it.
* Table customer and invoice has been joined on customer id.
* For calculating frequency, count of invoice id has been taken.
* Grouping is done on customer id.
* Records are sort in descending order as per frequency to show the highest on the top.

-- Preferred payment methods –

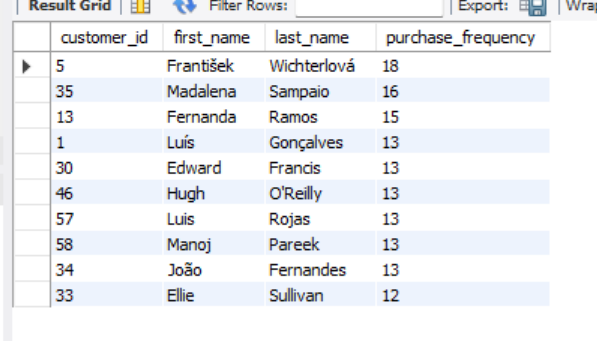
* No information provided for payment method

-- Average order value –

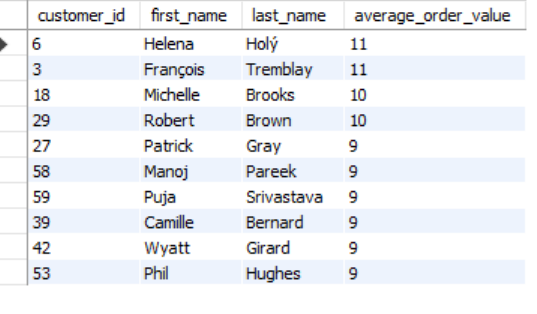
* Identify the table containing required information and join it.
* Table customer and invoice has been joined on customer id.
* For calculating average, average of total value has been taken from invoice table, it is rounded off to avoid complex values.
* Grouping is done on customer id.
* Records are sort in descending order as per average of total to show the highest on the top.

Average-

-- Frequency of purchases –



-- Average order value –



Insights-

-- Frequency of purchases –

* A pattern can be seen in the frequency.
* As, customer with customer id 5 purchases a lot of tracks with a frequency of 18 followed for customer with customer id 35 and 13 with frequency of 16 and 15 respectively.

-- Average order value –

* A pattern can be seen in the average.
* Customer with customer id 6 and 13 have the highest average value of 11

1. What is the customer churn rate?

Query-

Calculate the oldest date/year-

SELECT c.customer\_id, MIN(i.invoice\_date) AS first\_invoice\_date

FROM customer c

JOIN invoice i ON c.customer\_id = i.customer\_id

GROUP BY c.customer\_id

limit 1

Calculate the recent date/year -

SELECT c.customer\_id, MAX(i.invoice\_date) AS first\_invoice\_date

FROM customer c

JOIN invoice i ON c.customer\_id = i.customer\_id

GROUP BY c.customer\_id

limit 1

Count Unique Customers in 2017-

SELECT COUNT(DISTINCT customer\_id) AS customer\_count\_2017

FROM invoice

WHERE YEAR(invoice\_date) = 2017;

Count Unique Customers in 2020-

SELECT COUNT(DISTINCT customer\_id) AS customer\_count\_2020

FROM invoice

WHERE YEAR(invoice\_date) = 2020;

**Churn Rate**-

SELECT COUNT(DISTINCT customer\_id) AS churned\_customers

FROM invoice

WHERE YEAR(invoice\_date) = 2017

AND customer\_id NOT IN (

SELECT DISTINCT customer\_id

FROM invoice

WHERE YEAR(invoice\_date) = 2020

);

Method-

* The first query shows at which date/year the purchase began.
* The next query shows at which date/year the purchase began.
* The next query shows the unique customer id in 2017 (oldest year)
* The next query shows the unique customer id in 2020 (latest year)
* The last quesy shows the count of unique customer id which were lost in 2020

Formula used-

Customer Churn rate-

(Lost customer/Total customer at the start of period)X 100

= (1/58) \* 100 = 1.72

Insights-

* The churn rate is low, which indicates high customer retention.
* This shows that only 1 customer who was part of the purchase in 2017 is not a part of it in 2020.

1. Calculate the percentage of total sales contributed by each genre in the USA and identify the best-selling genres and artists.

Query-

WITH GenreSales AS (

SELECT g.name AS genre,

SUM(il.unit\_price \* il.quantity) AS total\_sales

FROM Invoice i

JOIN Invoice\_Line il ON i.invoice\_id = il.invoice\_id

JOIN Track t ON il.track\_id = t.track\_id

JOIN Genre g ON t.genre\_id = g.genre\_id

JOIN Customer c ON i.customer\_id = c.customer\_id

WHERE c.country = 'USA'

GROUP BY g.name

),

TotalSales AS (

SELECT SUM(total\_sales) AS total\_sales\_amount

FROM GenreSales)

SELECT gs.genre,gs.total\_sales,

(gs.total\_sales / ts.total\_sales\_amount \* 100) AS sales\_percentage

FROM GenreSales gs

CROSS JOIN TotalSales ts

ORDER BY gs.total\_sales DESC

LIMIT 10;

Method- WHERE c.country = 'USA'

* The Genre\_Sales CTE calculates the total sales for each genre in the USA by summing up the product of unit\_price and quantity for each genre.
* The Total\_Sales CTE calculates the total sales amount across all genres by summing the values from the Genre\_Sales CTE.
* The final SELECT query calculates what percentage of total sales comes from each genre by dividing genre sales by total sales and multiplying by 100.
* The results are sorted in descending order of sales to emphasize the top-performing genres.

Analysis-



Insights-

* The genre Rock, bags the highest sales of $555.36 with the percentage of 53.377.
* The genre Alternative & Punk, bags the second highest sales of $128.70 with the percentage of 12.36.
* The genre Metal, bags the third highest sales of $122.76 with the percentage of 11.79.

1. Find customers who have purchased tracks from at least 3 different genres

Query-

SELECT c.customer\_id, concat(c.first\_name," ", c.last\_name) as Full\_name,

COUNT(DISTINCT g.genre\_id) AS genre\_count

FROM customer c

JOIN invoice i ON c.customer\_id = i.customer\_id

JOIN invoice\_line il ON i.invoice\_id = il.invoice\_id

JOIN track t ON il.track\_id = t.track\_id

JOIN genre g ON t.genre\_id = g.genre\_id

GROUP BY c.customer\_id, c.first\_name,c.last\_name

HAVING genre\_count >= 3

limit 10;

Method-

* To identify the required tables. Join customer, invoice, invoice line, track and genre table.
* Use count distinct of genre id to calculate the total number of genre purchased by each customer.
* Group the results by customer name, customer id and genre.
* Filter the results to get information about only those customers who purchased atleast 3 genre.
* Limit the result upto 10.

Analysis-



1. Rank genres based on their sales performance in the USA

Query-

SELECT g.name AS genre,

RANK() OVER (ORDER BY SUM(il.unit\_price \* il.quantity) DESC) AS sales\_rank

FROM genre g

JOIN track t ON g.genre\_id = t.genre\_id

JOIN invoice\_line il ON t.track\_id = il.track\_id

JOIN invoice i ON il.invoice\_id = i.invoice\_id

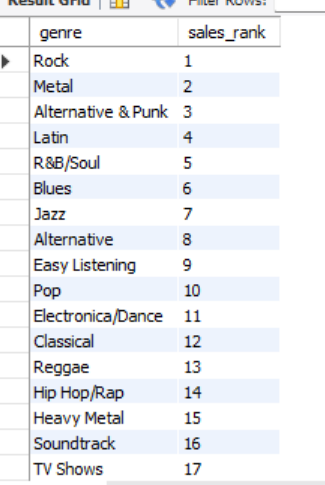
GROUP BY g.name

ORDER BY sales\_rank;

Method-

* Identify the required tables- Invoice,track, genre.
* Use where clause to filter the result of USA.
* Order the result by descending order of sales

Analysis-



Insights-

* Rock secured the first rank in terms of descending order of total sales.
* Metal secured the second rank in terms of descending order of total sales.
* Alternative & Punk secured the third rank in terms of descending order of total sales.
* Latin secured the fourth rank in terms of descending order of total sales.

1. Identify customers who have not made a purchase in the last 3 months

Query-

SELECT c.customer\_id, concat(c.first\_name," ", c.last\_name) as full\_name

FROM customer c

WHERE c.customer\_id NOT IN (

SELECT DISTINCT i.customer\_id

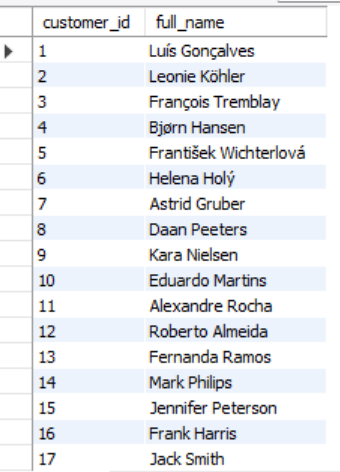
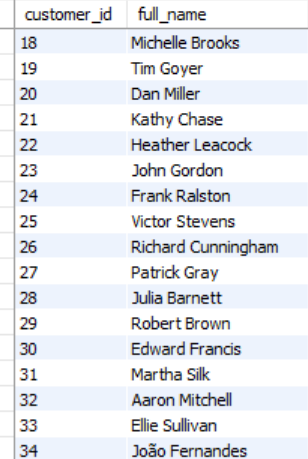
FROM invoice i

WHERE i.invoice\_date >= DATE\_SUB(CURRENT\_DATE, INTERVAL 3 MONTH)

);

Method-

* Select customer id and customer name from customers table which are not present in the list of customer id who have made purchase in the recent 3 months.

Analysis-

Insights-

* A total of 59 customers have not made any purchase in the last 3 months, which is a significantly high number.

Subjective Questions

1. Recommend the three albums from the new record label that should be prioritised for advertising and promotion in the USA based on genre sales analysis.

Query-

SELECT g.name AS genre\_name,

SUM(il.unit\_price \* il.quantity) AS total\_sales

FROM invoice\_line il

JOIN invoice i ON il.invoice\_id = i.invoice\_id

JOIN track t ON il.track\_id = t.track\_id

JOIN album a ON t.album\_id = a.album\_id

JOIN genre g ON t.genre\_id = g.genre\_id

WHERE i.billing\_country = 'USA'

GROUP BY g.name

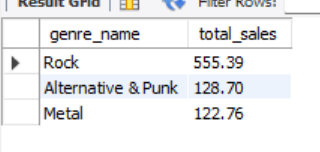
ORDER BY total\_sales DESC

LIMIT 3;

Method-

* Identify the required tables.
* Join invoice\_line, invoice, track, and genre table.
* Filter the country as USA
* Use the aggregate filter SUM to add the multiplication of unit price and quantity as total sales.
* Group the required data with the genre name.
* Arrange the result in the descending order of total sales.
* Put a limit of 3 to show only 3 records.

Analysis-



Insights-

The three albums from the new record label that should be prioritised for advertising and promotion in the USA based on genre sales analysis are-

* The genre Rock has the highest sales
* The genre Alternative & Punk has the second highest sales
* The genre Metal has the third highest sales

Promoting these 3 will gain more profit and will yield best results as per the current market trend in USA.

1. Determine the top-selling genres in countries other than the USA and identify any commonalities or differences.

Query-

WITH Genre\_Sales AS (

SELECT i.billing\_country AS Country,g.name AS Genre,

SUM(il.unit\_price \* il.quantity) AS TotalSales

FROM invoice\_line il

JOIN invoice i ON il.invoice\_id = i.invoice\_id

JOIN track t ON il.track\_id = t.track\_id

JOIN genre g ON t.genre\_id = g.genre\_id

WHERE i.billing\_country != 'USA'

GROUP BY i.billing\_country, g.name),

Top\_Genres AS (

SELECT Country,Genre,TotalSales,

ROW\_NUMBER() OVER (PARTITION BY Country ORDER BY TotalSales DESC) AS row\_no

FROM Genre\_Sales)

SELECT Country,Genre,TotalSales

FROM Top\_Genres

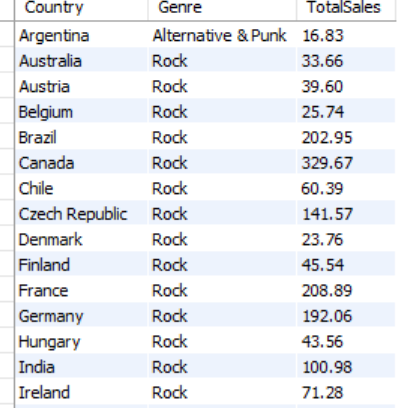
WHERE row\_no = 1

ORDER BY Country;

Method-

* Use 2 common table expressions. First as genre\_sales to analyse total sales for each country except for USA. Second as top\_genres to assign a rank to each genre within a country.
* In the end use select statement to select all genres from each country except USA with rank 1.

Analysis-





Commonalities-

* Rock can be seen as a top genre for many countries, indicating broad scope throughout international market

Insights-

* Rock should be a key focus for global marketing due to its widespread popularity.
* Leverage Rock's global appeal to drive international music promotion strategies.

1. Customer Purchasing Behavior Analysis: How do the purchasing habits (frequency, basket size, spending amount) of long-term customers differ from those of new customers? What insights can these patterns provide about customer loyalty and retention strategies?

Query-

First purchase date of each customer-

SELECT c.customer\_id, MIN(i.invoice\_date) AS first\_purchase\_date

FROM customer c

JOIN invoice i ON c.customer\_id = i.customer\_id

GROUP BY c.customer\_id;

Divide the customer into long term and new customer-

SELECT c.customer\_id,

CASE WHEN MIN(i.invoice\_date) < DATE\_SUB(CURDATE(), INTERVAL 1 YEAR) THEN 'Long\_term'ELSE 'New'

END AS customer\_type

FROM customer c

JOIN invoice i ON c.customer\_id = i.customer\_id

GROUP BY c.customer\_id;

Purchasing Frequency-

SELECT ct.customer\_type,c.customer\_id,COUNT(i.invoice\_id) AS purchase\_frequency

FROM (SELECT c.customer\_id,

CASE WHEN MIN(i.invoice\_date) < DATE\_SUB(CURDATE(), INTERVAL 1 YEAR) THEN 'Long-term'ELSE 'New'

END AS customer\_type

FROM customer c

JOIN invoice i ON c.customer\_id = i.customer\_id

GROUP BY c.customer\_id) ct

JOIN customer c ON ct.customer\_id = c.customer\_id

JOIN invoice i ON c.customer\_id = i.customer\_id

GROUP BY ct.customer\_type, c.customer\_id;

Basket Size-

SELECT ct.customer\_type,c.customer\_id,ROUND(AVG(il.quantity)) AS average\_basket\_size

FROM (SELECT c.customer\_id,

CASE WHEN MIN(i.invoice\_date) < DATE\_SUB(CURDATE(), INTERVAL 1 YEAR) THEN 'Long-term'ELSE 'New'

END AS customer\_type

FROM customer c

JOIN invoice i ON c.customer\_id = i.customer\_id

GROUP BY c.customer\_id) ct

JOIN customer c ON ct.customer\_id = c.customer\_id

JOIN invoice i ON c.customer\_id = i.customer\_id

JOIN invoice\_line il ON i.invoice\_id = il.invoice\_id

GROUP BY ct.customer\_type, c.customer\_id;

Spending Amount-

SELECT ct.customer\_type,c.customer\_id,SUM(i.total) AS total\_spending

FROM (SELECT c.customer\_id,

CASE WHEN MIN(i.invoice\_date) < DATE\_SUB(CURDATE(), INTERVAL 1 YEAR) THEN 'Long-term'ELSE 'New'

END AS customer\_type

FROM customer c

JOIN invoice i ON c.customer\_id = i.customer\_id

GROUP BY c.customer\_id) ct

JOIN customer c ON ct.customer\_id = c.customer\_id

JOIN invoice i ON c.customer\_id = i.customer\_id

GROUP BY ct.customer\_type, c.customer\_id;

Method-

* Calculate the first purchase date of each customer.
* Divide the database into Lomg term and new customer.
* Analyse its frequency, basket size and spending amount.

Analysis-

* Purchasing Frequency- **Long-Term customers** buy more often, suggesting higher engagement and trust. **New customers** on the other hand have less frequency may be due to them exploring the options.
* Basket Size- **Long-Term customers** often have a larger average basket size, reflecting their familiarity with and trust in the product range. **New customers** may initially have a smaller basket size, but they could potentially increase it as they become more comfortable with the offerings.
* Spending Amount- **Long-term customers** contribute more to total spending**.**

Insights-

* Focus should be on customer retention.
* Loyalty or reward points to be offers.

1. Product Affinity Analysis: Which music genres, artists, or albums are frequently purchased together by customers? How can this information guide product recommendations and cross-selling initiatives?

Query-

Genre Pairs-

SELECT g1.name AS genre1, g2.name AS genre2, COUNT(\*) AS frequency

FROM invoice\_line il1

JOIN track t1 ON il1.track\_id = t1.track\_id

JOIN genre g1 ON t1.genre\_id = g1.genre\_id

JOIN invoice\_line il2 ON il1.invoice\_id = il2.invoice\_id

JOIN track t2 ON il2.track\_id = t2.track\_id

JOIN genre g2 ON t2.genre\_id = g2.genre\_id

WHERE il1.invoice\_line\_id < il2.invoice\_line\_id AND g1.name <> g2.name

GROUP BY g1.name, g2.name

ORDER BY frequency DESC

LIMIT 10;

Artist Pairs-

SELECT a1.name AS artist1, a2.name AS artist2, COUNT(\*) AS frequency

FROM invoice\_line il1

JOIN track t1 ON il1.track\_id = t1.track\_id

JOIN album al1 ON t1.album\_id = al1.album\_id

JOIN artist a1 ON al1.artist\_id = a1.artist\_id

JOIN invoice\_line il2 ON il1.invoice\_id = il2.invoice\_id

JOIN track t2 ON il2.track\_id = t2.track\_id

JOIN album al2 ON t2.album\_id = al2.album\_id

JOIN artist a2 ON al2.artist\_id = a2.artist\_id

WHERE il1.invoice\_line\_id < il2.invoice\_line\_id AND a1.name <> a2.name

GROUP BY a1.name, a2.name

ORDER BY frequency DESC

LIMIT 10;

Album pair-

SELECT al1.title AS album1, al2.title AS album2, COUNT(\*) AS frequency

FROM invoice\_line il1

JOIN track t1 ON il1.track\_id = t1.track\_id

JOIN album al1 ON t1.album\_id = al1.album\_id

JOIN invoice\_line il2 ON il1.invoice\_id = il2.invoice\_id

JOIN track t2 ON il2.track\_id = t2.track\_id

JOIN album al2 ON t2.album\_id = al2.album\_id

WHERE il1.invoice\_line\_id < il2.invoice\_line\_id

AND al1.title <> al2.title

GROUP BY al1.title, al2.title

ORDER BY frequency DESC

LIMIT 10;

Method-

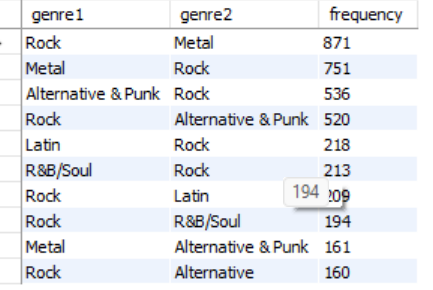
* Identify the genre pair, artist pair and album pairs.
* Calculate its frequency

Analysis-

* Genre pairs

Rock and Metal have been purchased for about 871 times. It shows high favourability in customers.

Rock has been repeated a lot of times, further investments can be made in rock genre.



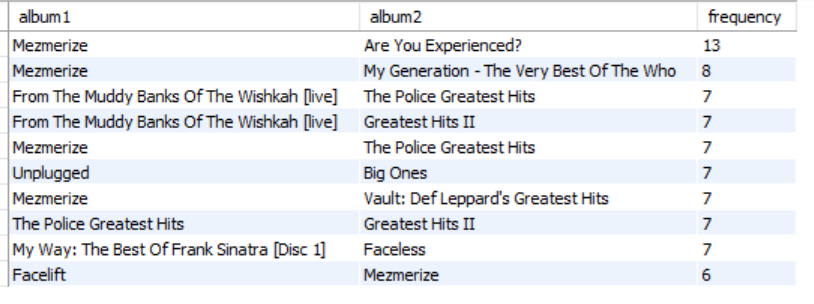
* Artists Pair

System of a down and Jimi Hendrix have been together purchased for 13 times.



* Album Pair

Albums like Mezmerize and Are you experienced has been brought together for 13 times shows customer preferences.

Insights-

* Discounts and combo deals on frequently purchased genres and albums can be used as a good advertising strategy worldwide.

Cross selling initiatives-

* Offer bundles on artists who are purchased together.

Product recommendations-

* Use high frequency purchases as a guide for new customers with proper data to authenticate it.

1. Regional Market Analysis: Do customer purchasing behaviors and churn rates vary across different geographic regions or store locations? How might these correlate with local demographic or economic factors?

Query-

Customer purchasing behavior-

SELECT c.country, c.city, COUNT(DISTINCT i.invoice\_id) AS total\_purchases,

SUM(il.unit\_price \* il.quantity) AS total\_spending,

(Round(AVG(il.unit\_price \* il.quantity))) AS avg\_spending\_per\_purchase,

COUNT(DISTINCT c.customer\_id) AS total\_customers

FROM customer c

JOIN invoice i ON c.customer\_id = i.customer\_id

JOIN invoice\_line il ON i.invoice\_id = il.invoice\_id

GROUP BY c.country, c.city

ORDER BY total\_spending DESC

LIMIT 5;

Churn rates-

WITH customer\_last\_purchase AS (

SELECT c.customer\_id,MAX(i.invoice\_date) AS last\_purchase\_date,c.country,c.city

FROM customer c

JOIN invoice i ON c.customer\_id = i.customer\_id

GROUP BY c.customer\_id, c.country, c.city

)

SELECT country,city,COUNT(\*) AS total\_customers,

SUM(CASE WHEN DATEDIFF(CURDATE(), last\_purchase\_date) > 365 THEN 1 ELSE 0 END) AS churned\_customers,

(SUM(CASE WHEN DATEDIFF(CURDATE(), last\_purchase\_date) > 365 THEN 1 ELSE 0 END) / COUNT(\*)) \* 100 AS churn\_rate

FROM customer\_last\_purchase

GROUP BY country, city

ORDER BY churn\_rate DESC

limit 5;

Method-

* Identify the customer purchasing behaviour- Segment the country-city wise data of purchases, their speding and also the number of customer with each country-city.
* Calculate the churn rates- Measure purchasing frequency, average spending, and churn rates for each region.

Analysis-

Purchasing behaviour-

* **Total Purchases & Spending**: Regions like the **USA and Canada** show high total spending, likely due to larger customer bases and higher disposable incomes.
* **Average Spend per Purchase**: Higher average spending in certain regions suggests **greater willingness to pay** for music content.

Churn Rates

* High Churn Regions: Some European cities show higher churn, indicating a need for improved engagement or localized campaigns.
* Low Churn Regions: Areas with low churn may reflect effective retention practices that could be replicated elsewhere

Correlation with Demographics & Economics

* Demographics: Younger regions tend to favor genres like Pop and Hip-Hop, influencing buying behavior.
* Economics: Wealthier regions often show higher spend and lower churn, driven by stronger purchasing power.

Insights-

* Targeted Marketing: Customize campaigns by region — premium promotions in affluent areas, discounts in high-churn markets.
* Customer Retention: Implement region-specific loyalty offers, localized content, and tailored support.
* Product Localization: Align the catalog with regional music tastes, featuring popular genres and artists per location.

1. Customer Risk Profiling: Based on customer profiles (age, gender, location, purchase history), which customer segments are more likely to churn or pose a higher risk of reduced spending? What factors contribute to this risk?

Query-

Customers with No Recent Purchases-

SELECT c.customer\_id, c.first\_name, c.last\_name, c.country, MAX(i.invoice\_date) AS last\_purchase\_date

FROM customer c

JOIN invoice i ON c.customer\_id = i.customer\_id

GROUP BY c.customer\_id, c.first\_name, c.last\_name, c.country

HAVING MAX(i.invoice\_date) < DATE\_SUB(CURDATE(), INTERVAL 1 YEAR);

Customers by Spending Patterns-

SELECT c.customer\_id, c.first\_name, c.last\_name, c.country,

COUNT(i.invoice\_id) AS purchase\_count,

SUM(i.total) AS total\_spent,

AVG(i.total) AS avg\_spent\_per\_purchase,

MAX(i.invoice\_date) AS last\_purchase\_date

FROM customer c

JOIN invoice i ON c.customer\_id = i.customer\_id

GROUP BY c.customer\_id, c.first\_name, c.last\_name, c.country

ORDER BY total\_spent DESC

LIMIT 5;

Customer Segments by Location-

SELECT c.country,

COUNT(c.customer\_id) AS total\_customers,

COUNT(i.invoice\_id) AS total\_purchases,

SUM(i.total) AS total\_revenue,

AVG(i.total) AS avg\_revenue\_per\_purchase,

MAX(i.invoice\_date) AS last\_purchase\_date

FROM customer c

JOIN invoice i ON c.customer\_id = i.customer\_id

GROUP BY c.country

ORDER BY total\_revenue DESC

LIMIT 5;

Purchase Frequency and Recency-

SELECT c.customer\_id, c.first\_name, c.last\_name, c.country,

COUNT(i.invoice\_id) AS purchase\_count,

DATEDIFF(CURDATE(), MAX(i.invoice\_date)) AS days\_since\_last\_purchase,

CASE

WHEN DATEDIFF(CURDATE(), MAX(i.invoice\_date)) > 365 THEN 'High Risk'

WHEN DATEDIFF(CURDATE(), MAX(i.invoice\_date)) BETWEEN 180 AND 365 THEN 'Medium Risk'

ELSE 'Low Risk'

END AS risk\_profile

FROM customer c

JOIN invoice i ON c.customer\_id = i.customer\_id

GROUP BY c.customer\_id, c.first\_name, c.last\_name, c.country

Limit 5;

Method-

* Determine customer with no recent purchase, which shows chured customers, spending pattern of the customer, customer segments by location, customer purchase Frequency and Recency.

Analysis-

Factors-

* Customer with no recent purchase- They pose high threat of churned customers
* Spending pattern of the customer- Reduction in spending pattern may lead to losses.
* Customer segments by location- Locations like USA, Canada, Brazil may have a higher concentration of at-risk customers.
* Customer purchase Frequency and Recency- Customers with no recent activity are more likely to churn

1. Customer Lifetime Value Modeling: How can you leverage customer data (tenure, purchase history, engagement) to predict the lifetime value of different customer segments? This could inform targeted marketing and loyalty program strategies. Can you observe any common characteristics or purchase patterns among customers who have stopped purchasing?

Query-

Customer tenure-

SELECT c.customer\_id, c.first\_name, c.last\_name, c.country, MIN(i.invoice\_date) AS first\_purchase\_date, MAX(i.invoice\_date) AS last\_purchase\_date, COUNT(i.invoice\_id) AS purchase\_count, SUM(i.total) AS total\_spent

FROM customer c

JOIN invoice i ON c.customer\_id = i.customer\_id

GROUP BY c.customer\_id;

Purchase history-

SELECT i.customer\_id, il.invoice\_line\_id, il.track\_id, il.unit\_price, il.quantity, t.album\_id, t.genre\_id

FROM invoice\_line il

JOIN invoice i ON il.invoice\_id = i.invoice\_id

JOIN track t ON il.track\_id = t.track\_id;

Engagement-

SELECT t.track\_id, a.album\_id, ar.artist\_id, ar.name AS artist\_name

FROM track t

JOIN album a ON t.album\_id = a.album\_id

JOIN artist ar ON a.artist\_id = ar.artist\_id;

Method-

* Calculate customer tenure- Identify by find the first and last purchase date, how many purchases were made in between them and the total spent.
* Calculate customer purchase history- Identify the purchase history by using the invoice line table. How many times the same or different customer purchased the same or different product.
* Calculate engagements- Present the combination of track id, album id, artist id and name

Insights-

* Customers with **longer tenures** and **frequent purchases** generally have higher Customer Lifetime Value.

1. If data on promotional campaigns (discounts, events, email marketing) is available, how could you measure their impact on customer acquisition, retention, and overall sales?

Query-

Customer Acquisition-

SELECT MIN(i.invoice\_date) AS first\_purchase\_date,

COUNT(DISTINCT i.customer\_id) AS new\_customers

FROM invoice i

JOIN customer c ON i.customer\_id = c.customer\_id

WHERE i.invoice\_date BETWEEN '2018-01-01' AND '2018-01-31'

GROUP BY i.customer\_id

HAVING first\_purchase\_date BETWEEN '2018-01-01' AND '2018-01-31';

Customer retention-

SELECT COUNT(DISTINCT i2.invoice\_id) AS repeat\_purchases,COUNT(DISTINCT i1.customer\_id) AS total\_customers

FROM invoice i1

JOIN invoice i2 ON i1.customer\_id = i2.customer\_id

WHERE i1.invoice\_date BETWEEN '2018-01-01' AND '2018-01-31'AND i2.invoice\_date > '2018-01-31';

Overall sales comparison-

SELECT SUM(CASE WHEN i.invoice\_date BETWEEN '2017-12-01' AND '2017-12-31' THEN i.total ELSE 0 END) AS before\_campaign\_sales,

SUM(CASE WHEN i.invoice\_date BETWEEN '2018-01-01' AND '2018-01-31' THEN i.total ELSE 0 END) AS during\_campaign\_sales,

SUM(CASE WHEN i.invoice\_date BETWEEN '2020-02-01' AND '2020-02-28' THEN i.total ELSE 0 END) AS after\_campaign\_sales

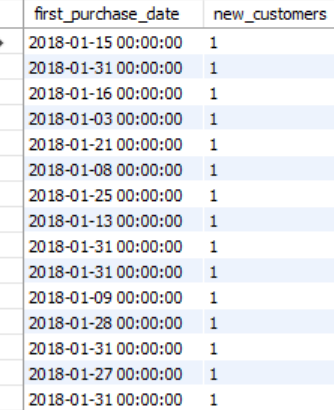
FROM invoice i;

Method-

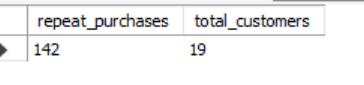
* Customer Acquisition- Write a query to find the MIN date which is the promotion start date, count of the customers that made a purchase during that promotion period, filter the period using where clause, also group the result using customer id.
* Customer retention- Write a query who made purchase in January and also later to show retention. Join is performed with the same table on customer id.
* Overall sales comparison- A query is written from invoice table, using case statement to calculate the sales before, after and during promotional periods.

Analysis

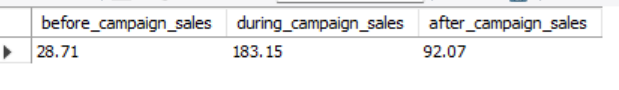
* Customer Acquisition-



* Customer retention-



* Overall sales comparison-



Recommendation-

* Enhance customer retention tactics
* Implement post campaign check ups
* Proper monitoring of sales
* Note which campaign led to the most sales generation

1. How would you approach this problem, if the objective and subjective questions weren't given?

Query-

Customer Acquisition-

SELECT COUNT(DISTINCT customer\_id) AS new\_customers

FROM invoice

WHERE invoice\_date BETWEEN '2018-01-01' AND '2018-01-31'

AND customer\_id NOT IN (SELECT customer\_id FROM invoice

WHERE invoice\_date < '2018-01-01'

);

Customer retention-

SELECT COUNT(DISTINCT i1.customer\_id) AS retained\_customers

FROM invoice i1

JOIN invoice i2 ON i1.customer\_id = i2.customer\_id

WHERE i1.invoice\_date < '2018-01-01'

AND i2.invoice\_date BETWEEN '2018-01-01' AND '2018-01-31';

Overall sales comparison-

SELECT

SUM(CASE WHEN invoice\_date BETWEEN '2017-12-01' AND '2017-12-31' THEN total ELSE 0 END) AS sales\_before,

SUM(CASE WHEN invoice\_date BETWEEN '2018-01-01' AND '2018-01-31' THEN total ELSE 0 END) AS sales\_during\_campaign,

SUM(CASE WHEN invoice\_date BETWEEN '2018-02-01' AND '2018-02-28' THEN total ELSE 0 END) AS sales\_after

FROM invoice;

Understand the concept-

* The first thing would be to understand the business, what is really happening, and what needs to be done.

Objectives-

* Increase sales performance, understand metrics, customer retention, and lower churn rate.

Explore and Analyse the dataset-

* Use of SQL queries to find out the sales, churn rate, customer retention rate, region wise sales analysis and steps taken to enhance it.
* Join multiple tables to find complex data and their resulting solutions.
* Use aggregate functions like SUM, COUNT, MIN, MAX to find the overall sales, customer frequency, tenure etc.

Derive Insights-

* Which genre performed well, in which state, city or country. Promotion campaigns or customer preference what lead to it.
* Which album is most preferred?
* Combination sets of most preferred artists, album or genre sets.
* Rock is the most preferred genre is many countries.
* Combine tracks or album or genre with rock as an offer to increase sales.
* Using customer purchase trend as a major factor for increasing sales/ campaigns/ promotions.

Suggest Recommendations-

* Launch region specific campaigns.
* Promote employee motivation schemes to enhance sales and healthy competition
* Promote underperforming album/genre/artist/track.

1. How can you alter the "Albums" table to add a new column named "ReleaseYear" of type INTEGER to store the release year of each album?

Query-

ALTER TABLE album

ADD COLUMN ReleaseYear INTEGER;

Select \* from album;

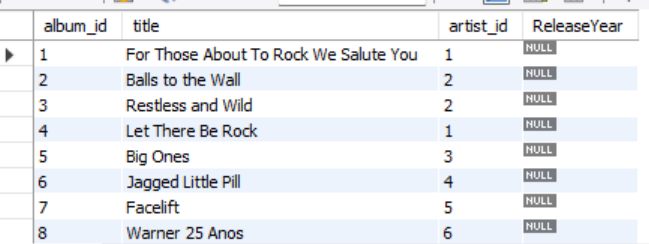
Method-

* Use query-

ALTER TABLE album

ADD COLUMN ReleaseYear INTEGER;

Analysis-



1. Chinook is interested in understanding the purchasing behavior of customers based on their geographical location. They want to know the average total amount spent by customers from each country, along with the number of customers and the average number of tracks purchased per customer. Write an SQL query to provide this information.

Query-

WITH customer\_spending AS (

SELECT c.customer\_id, c.country, SUM(i.total) AS total\_spent, COUNT(il.track\_id) AS total\_tracks

FROM customer c

JOIN invoice i ON c.customer\_id = i.customer\_id

JOIN invoice\_line il ON i.invoice\_id = il.invoice\_id

GROUP BY c.customer\_id, c.country

),

Country\_summary AS (

SELECT country, COUNT(customer\_id) AS number\_of\_customers,

ROUND(AVG(total\_spent))AS avg\_total\_spent,

ROUND(AVG(total\_tracks)) AS avg\_tracks\_per\_customer

FROM customer\_spending

GROUP BY country

)SELECT country, number\_of\_customers, avg\_total\_spent, avg\_tracks\_per\_customer

FROM Country\_summary

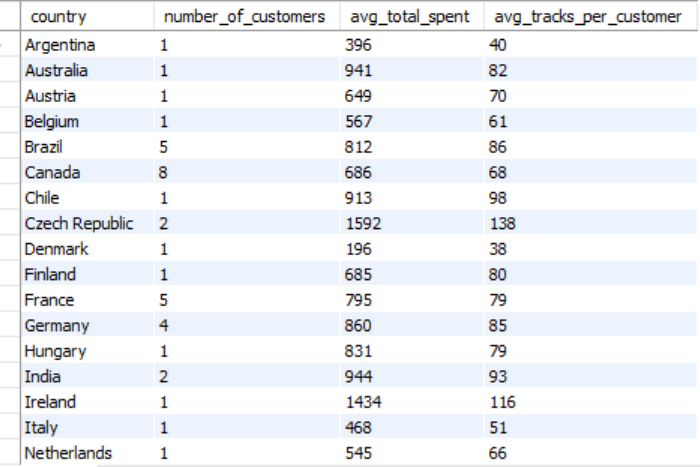
ORDER BY country;

Method-

To find-

* Average total amount spent by customers from each country.
* Number of customers
* Average number of tracks purchased per customer

Analysis-



High- Value Markets

* Germany and Canada  
  Customers from these regions exhibit high average spending, indicating strong market potential for premium offerings and loyalty programs.

Largest Market, Untapped Potential

* United States  
  Despite having the largest customer base, the average spend per customer is just moderate.

Low Engagement Region

* Brazil  
  Shows lower average tracks purchased per customer, suggesting less engagement.

Stable Market

* France  
  Displays balanced customer count and consistent purchasing, indicating a steady market with room for gradual growth.