Objective Questions

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

ANSWERS:

Approach:

1. **Identifying NULL Values:**
    - I Used IS NULL condition to identify NULL values in each column.
    - Perform exploratory data analysis (EDA) to check the percentage of missing data.
2. **Handling NULL Values:**
    - Use COALESCE() function to replace NULL values with default values.
    - Apply UPDATE statements to modify values permanently in the table.
3. **Handling Duplicates:**
    - Identify duplicates using GROUP BY and HAVING COUNT(*) > 1.
    - Remove duplicates using DELETE with ROW_NUMBER() or DISTINCT.

Insights:

- NULL values can cause issues in calculations and joins, leading to incorrect query results.
- Using appropriate placeholders for missing data maintains data consistency.
- Handling duplicates ensures accurate reporting and avoids redundant records.

Recommendation:

- Regularly check for missing values and duplicates as part of data cleaning.
- Use constraints like NOT NULL and UNIQUE where applicable to prevent such issues.
- For large datasets, consider using stored procedures to automate data cleaning.

There are 2 ways to replace NULL values with specific values

First, we use the COALESCE() function to handle NULL values. The syntax for COALESCE() is:

COALESCE(column_name, 'default_value')

Additionally, we use the UPDATE statement to modify table values based on a specific condition. The syntax is:

UPDATE table_name

SET column_name = 'new_value'

WHERE condition;

We have NULL values for the columns in various tables as follows

1. Customer table - company,state,phone,fax,postal_code

2. Track table – Composer

In case of numeric values, it can be replaced with 0.

In case of text, it can be replaced with placeholders like ‘None’ / ‘Unknown’ etc.

Eg) If composer is NULL in the track table, it can be replaced with ‘Unknown’.

Similiarly, it can be done for other columns with NULL values.

Tables with NULL values	**Column Name / Attribute**	**Value to be replaced**
Customer	Company	‘Unknown’
	State	‘None’
	Phone	‘+0 000 000 0000’
	Fax	‘+0 000 000 0000’
	Postal_code	' 000 000'
Track	Composer	‘Unknown’

SL.NO	USING COALESCE	USING UPDATE
1	SELECT COALESCE(company,'Unknown') FROM customers WHERE company IS NULL;	UPDATE customer SET company = 'Unknown' WHERE company IS NULL; -- 49 row(s) affected
2	SELECT COALESCE(state,'None') FROM customers WHERE company IS NULL;	UPDATE customer SET state = 'None' WHERE state IS NULL; -- 29 row(s) affected
3	SELECT COALESCE(phone, '+0 000 000 0000') FROM customers WHERE phone IS NULL;	UPDATE customer SET phone = '+0 000 000 0000' WHERE phone IS NULL; -- 1 row(s) affected
4	SELECT COALESCE(fax, '+0 000 000 0000') FROM customers WHERE fax IS NULL;	UPDATE customer SET fax = '+0 000 000 0000' WHERE fax IS NULL; -- 47 row(s) affected
5	SELECT COALESCE(postal_code, ' 000 000') FROM customers WHERE fax IS NULL;	UPDATE customer SET postL_CODE= '00 0000' WHERE POSTAL_CODEIS NULL; -- 4 Row(s) affected
6	SELECT COALESCE(company,'Unknown') FROM customers WHERE company IS NULL;	UPDATE track SET composer = 'Unknown' WHERE composer IS NULL; -- 978 row(s) affected

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

Approach:

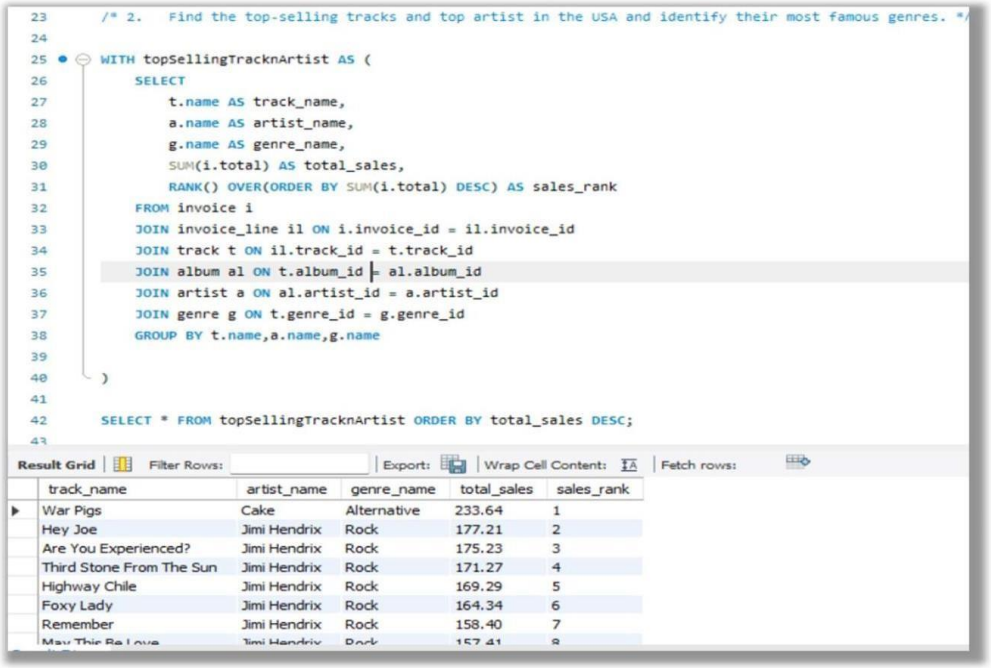
1. We use a Common Table Expression (CTE) topSellingTracknArtist to organize the query and make it more readable.
2. The query joins multiple tables (invoice, invoice_line, track, album, artist, and genre) to gather all necessary data.
3. The SUM(i.total) function calculates total sales for each track.
4. The RANK() OVER(ORDER BY SUM(i.total) DESC) assigns a ranking based on total sales.

5. We filter results for the USA using WHERE i.billing_country = 'USA'.
6. Finally, we select the top-selling tracks and artists ordered by total sales.

Query:

```
WITH topSellingTracknArtist AS
( SELECT
t.name AS track_name, a.name AS artist_name, g.name AS genre_name, SUM(i.total) AS total_sales,
RANK() OVER(ORDER BY SUM(i.total) DESC) AS sales_rank 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 album al ON t.album_id = al.album_id
JOIN artist a ON al.artist_id = a.artist_id
JOIN genre g ON t.genre_id = g.genre_id
WHERE i.billing_country = 'USA'
GROUP BY t.name,a.name,g.name
)
SELECT * FROM topSellingTracknArtist ORDER BY total_sales DESC;
```

***Result: (784 rows returned)**



```
/* 2.   Find the top-selling tracks and top artist in the USA and identify their most famous genres. */

WITH topSellingTracknArtist AS (
    SELECT
        t.name AS track_name,
        a.name AS artist_name,
        g.name AS genre_name,
        SUM(i.total) AS total_sales,
        RANK() OVER(ORDER BY SUM(i.total) DESC) AS sales_rank
    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 album al ON t.album_id = al.album_id
    JOIN artist a ON al.artist_id = a.artist_id
    JOIN genre g ON t.genre_id = g.genre_id
    GROUP BY t.name,a.name,g.name

)

SELECT * FROM topSellingTracknArtist ORDER BY total_sales DESC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

track_name	artist_name	genre_name	total_sales	sales_rank
War Pigs	Cake	Alternative	233.64	1
Hey Joe	Jimi Hendrix	Rock	177.21	2
Are You Experienced?	Jimi Hendrix	Rock	175.23	3
Third Stone From The Sun	Jimi Hendrix	Rock	171.27	4
Highway Chile	Jimi Hendrix	Rock	169.29	5
Foxy Lady	Jimi Hendrix	Rock	164.34	6
Remember	Jimi Hendrix	Rock	158.40	7
May This Be Love	Jimi Hendrix	Rock	157.41	8

Insights:

- This query helps in identifying the most popular tracks and artists in the USA based on sales revenue.

- Understanding the most famous genres can help businesses tailor music recommendations to users.
- The use of ranking functions (RANK() OVER) allows sorting without duplicates being removed.
- Grouping by track, artist, and genre provides a detailed breakdown of sales performance.

Recommendation:

- To optimize performance, indexes should be created on columns involved in joins (track_id, album_id, artist_id, genre_id).
- Instead of SUM(i.total), additional analysis using COUNT(il.track_id) can be done to check the most frequently purchased tracks.
- Further filtering can be applied to analyze trends over different time periods (e.g., monthly or yearly sales performance).

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

ANSWERS

- **Concepts used:** Aggregate Functions, GROUP BY, Sorting(ORDER BY)
- **Tables used:** customer

APPROACHES

Steps Taken:

- Used COUNT(customer_id) to count customers in each region.
- Grouped data by country, state, and city using GROUP BY.
- Sorted the results by country for better readability.
- Handled missing state values using COALESCE(state, 'None').

Query:

```
SELECT
    country,
    COALESCE(state,'None') AS state, city,
    COUNT(customer_id) AS demographic_dist FROM customer
    GROUP BY country, state, city ORDER BY country;
```

Result: (53 rows returned**)**

```sql
/* 3.  What is the customer demographic breakdown (age, gender, location) of Chinook's customer base? */

SELECT
    country,
    COALESCE(state,'None') AS state,
    city,
    COUNT(customer_id) AS demographic_dist
FROM customer
GROUP BY country, state, city
ORDER BY country;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

country	state	city	demographic_dist
Argentina	None	Buenos Aires	1
Australia	NSW	Sidney	1
Austria	None	Vienne	1
Belgium	None	Brussels	1
Brazil	DF	Brasília	1
Brazil	RJ	Rio de Janeiro	1
Brazil	SP	São José dos Campos	1
Brazil	SP	São Paulo	2
Canada	AB	Edmonton	1
Canada	BC	Vancouver	1
Canada	MB	Winnipeg	1
Canada	NS	Halifax	1
Canada	NT	Yellowknife	1
Canada	ON	Ottawa	1
Canada	ON	Toronto	1

Result 158

Insights

1. **Customer Distribution**
    - Some countries and cities have **more customers**, meaning strong engagement there.
    - Other regions have **fewer customers**, showing weaker presence or potential markets.
2. **Missing Data Issues**
    - The 'None' values in state mean some data is incomplete.
    - Without age or gender data, we don’t fully understand customer demographics.
3. **Market Opportunities**
    - Regions with **high customer numbers** are key focus areas.
    - Areas with **low customers** could be targeted for expansion.

Recommendations

1. **Enhance Customer Data**
    - Add **age and gender** in the query to get a clearer customer profile.
2. **Analyze Revenue Trends**
    - Modify the query to check **which locations generate the most sales**.
3. **Fix Missing Data**
    - Ensure state and other details are properly recorded for all customers.
4. **Improve Marketing Strategy**
    - High-customer regions → **More offers & engagement**.
    - Low-customer regions → **Targeted ads & promotions**.

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

ANSWERS

- **Concepts used:** Aggregate Functions, GROUP BY, Sorting (ORDER BY)
- **Table used:** invoice

Appoach:

- Used SUM(total) to calculate **total revenue** for each location.
- Used COUNT(invoice_id) to count the **number of invoices** per location.
- Grouped by billing_country, billing_state, billing_city to categorize data correctly.
- Sorted the results by **country (A-Z)** and **revenue (highest to lowest)** to highlight top-earning regions.

Query:

```
SELECT
    billing_country, billing_state, billing_city,
    SUM(total) AS total_revenue, COUNT(invoice_id) AS num_of_invoices
    FROM invoice
    GROUP BY billing_country, billing_state, billing_city
    ORDER BY billing_country ASC, total_revenue DESC;
```

Result: (53 rows returned)

```
28      /* 4.   Calculate the total revenue and number of invoices for each country, state, and city: */
29
30 •    SELECT
31          billing_country,
32          billing_state,
33          billing_city,
34          SUM(total) AS total_revenue,
35          COUNT(invoice_id) AS num_of_invoices
36      FROM invoice
37      GROUP BY billing_country, billing_state, billing_city
38      ORDER BY billing_country ASC,total_revenue DESC;
39
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

billing_country	billing_state	billing_city	total_revenue	num_of_invoices
Argentina	None	Buenos Aires	39.60	5
Australia	NSW	Sidney	81.18	10
Austria	None	Vienne	69.30	9
Belgium	None	Brussels	60.39	7
Brazil	SP	São Paulo	129.69	22
Brazil	SP	São José dos Campos	108.90	13
Brazil	DF	Brasilia	106.92	15
Brazil	RJ	Rio de Janeiro	82.17	11
Canada	QC	Montréal	99.99	9
Canada	ON	Ottawa	91.08	13
Canada	NT	Yellowknife	75.24	12
Canada	MB	Winnipeg	70.29	8

esult 19 ×

Insights

1. **Top Revenue-Generating Locations**
    - Some cities and countries bring in **more revenue**, showing stronger customer spending.
    - Higher invoices in a region suggest **more frequent purchases**.
2. **Revenue vs. Invoice Count**
    - A location with **high revenue but fewer invoices** may indicate **higher average spending per order**.
    - A location with **many invoices but lower revenue** suggests **smaller but frequent purchases**.
3. **State-Level Revenue Gaps**
    - Some states may have **missing billing data** (NULL or empty states).
    - Certain regions may be **underperforming**, meaning they need better marketing or pricing strategies.

Recommendations

1. **Focus on High-Revenue Locations**
    - Invest in **customer loyalty programs & premium services** in top-earning cities.
    - Expand **exclusive offers** to customers in high-spending areas.
2. **Boost Low-Revenue Regions**
    - Identify **why certain locations underperform** (low awareness, pricing, or accessibility).
    - Run **localized marketing campaigns** to boost sales in weaker regions.
3. **Improve Data Quality**
    - Ensure all invoices have **billing state details** to avoid missing data.
    - Standardize location data for better analysis.
4. **Enhance Product Pricing Strategy**
    - If a location has **high invoices but low revenue**, consider **upselling higher-priced products**.
    - If a location has **low invoices but high revenue**, analyze if **premium pricing works well** there.

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

ANSWERS

- **Concepts used:** CTE, Joins, GROUP BY, Aggregate Functions, Sorting (ORDER BY)
- **Tables used:** customer, invoice

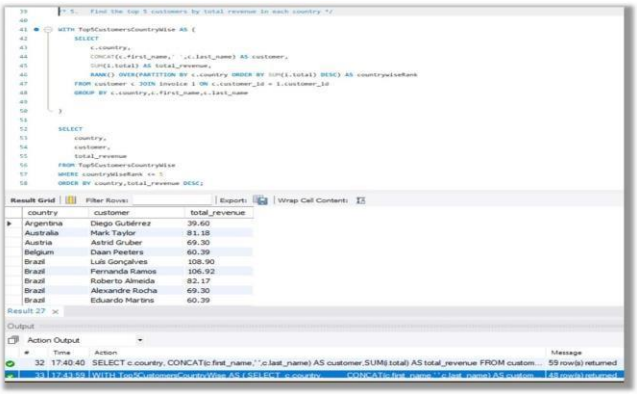
Approaches

1. Used SUM(i.total) to calculate **total revenue per customer**.
2. Used INNER JOIN to connect **customer** and **invoice** tables.
3. Grouped by c.country, c.first_name, c.last_name to get customer-wise revenue per country.
4. Applied RANK() OVER (PARTITION BY c.country ORDER BY SUM(i.total) DESC) to rank customers within their respective countries.
5. Used a **Common Table Expression (CTE)** to filter only the **top 5 customers per country**.
6. Sorted by country (A-Z) and total_revenue (high to low) for easy interpretation.

Query:

```
WITH Top5CustomersCountryWise AS ( SELECT
c.country,
CONCAT(c.first_name,' ',c.last_name) AS customer, SUM(i.total) AS total_revenue,
RANK() OVER (
PARTITION BY c.country ORDER BY SUM(i.total) DESC
) AS countrywiseRank
FROM customer c INNER JOIN invoice i ON c.customer_id = i.customer_id
GROUP BY c.country,c.first_name,c.last_name
)
SELECT
country, customer, total_revenue
 FROM Top5CustomersCountryWise
WHERE countryWiseRank <= 5
ORDER BY country,total_revenue DESC;
```

- **Result: (48 rows returned)**



country	customer	total_revenue
Argentina	Diego Gutiérrez	39.60
Australia	Mark Taylor	81.18
Austria	Astrid Gruber	69.30
Belgium	Daan Peeters	60.39
Brazil	Luis Gonçalves	108.90
Brazil	Fernanda Ramos	106.92
Brazil	Roberto Almeida	82.17
Brazil	Alexandre Rocha	69.30
Brazil	Eduardo Martins	60.39

Insights

1. **Top Spending Customers by Country**
    - The results show the **highest revenue-generating customers per country**.
    - These customers contribute the most to sales and can be **targeted for loyalty programs**.
2. **Customer Spending Patterns**
    - Some countries might have **more high-spending customers** than others.
    - Certain countries may have **a sharp revenue drop after the top 2-3 customers**, indicating fewer high-value buyers.
3. **Customer Retention Opportunities**
    - If only a few customers in a country drive most of the revenue, there's a **high dependency on them**.
    - If revenue is **evenly spread** among multiple customers, the business has a **balanced customer base**.

Recommendations

1. **Engage High-Value Customers**
    - Offer **exclusive discounts, early access, or VIP memberships** to top customers.
    - Send **personalized offers** based on their spending patterns.
2. **Expand High-Value Customer Base**
    - In countries where only a few customers dominate revenue, **target mid-level spenders** with promotions to increase their spending.
3. **Analyze Low-Performing Countries**
    - Identify countries where **top customers have lower spending** and find ways to **increase their purchase frequency**.
    - Localized **advertising & promotional campaigns** may help boost engagement.
4. **Refine Customer Segmentation**
    - Add **purchase frequency** and **average invoice amount** to further segment customers.
    - Modify the query to find **repeat buyers vs. one-time big spenders**

6. Identify the top-selling track for each customer

ANSWERS:

Concepts used: CTE, Joins, GROUP BY, Aggregate Functions (SUM)

Tables used: customer, invoice, invoice_line, track

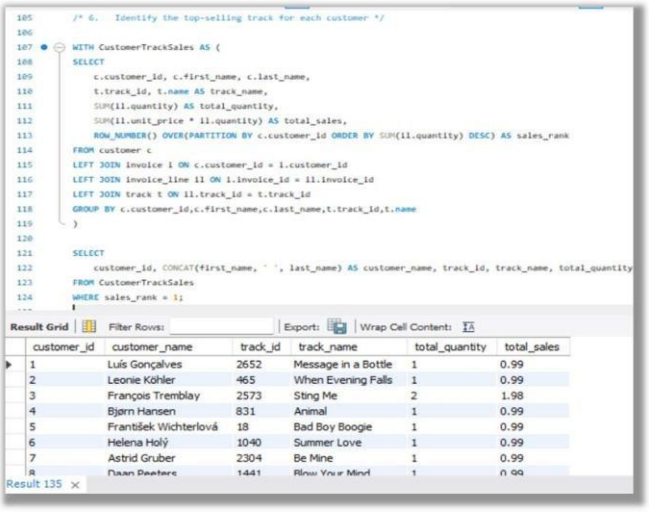
APPROACH

1. Used SUM(il.quantity) to find total quantity of each track purchased by a customer.
2. Used SUM(i.total) to calculate the total sales generated by each track per customer.
3. Used multiple joins to connect the customer, invoice, invoice_line, and track tables.
4. Applied ROW_NUMBER() OVER (PARTITION BY c.customer_id ORDER BY SUM(i.total) DESC) to rank tracks per customer by total sales.
5. Selected only rank 1 (top track) per customer.
6. Sorted by total_sales DESC to show highest-revenue tracks first.

Query:

```
WITH CustomerTrackSales AS ( SELECT

    c.customer_id, c.first_name, c.last_name, t.track_id, t.name AS track_name, SUM(il.quantity) AS
    total_quantity, SUM(i.total) AS total_sales, ROW_NUMBER()

    OVER(

    PARTITION BY c.customer_id ORDER BY SUM(i.total) DESC

    ) AS sales_rank FROM customer c

    LEFT JOIN invoice i ON c.customer_id = i.customer_id

    LEFT JOIN invoice_line il ON i.invoice_id = il.invoice_id

    LEFT 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, CONCAT(first_name, ' ', last_name) AS customer_name, track_id, track_name, total_quantity,
    total_sales

    FROM CustomerTrackSales

    WHERE sales_rank = 1 ORDER BY total_sales DESC;
```

Result: (**59 rows** returned)



```
/* 6.   Identify the top-selling track for each customer */

WITH CustomerTrackSales AS (
SELECT
    c.customer_id, c.first_name, c.last_name,
    t.track_id, t.name AS track_name,
    SUM(il.quantity) AS total_quantity,
    SUM(il.unit_price * il.quantity) AS total_sales,
    ROW_NUMBER() OVER(PARTITION BY c.customer_id ORDER BY SUM(il.quantity) DESC) AS sales_rank
FROM customer c
LEFT JOIN invoice i ON c.customer_id = i.customer_id
LEFT JOIN invoice_line il ON i.invoice_id = il.invoice_id
LEFT 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, CONCAT(first_name, ' ', last_name) AS customer_name, track_id, track_name, total_quantity
FROM CustomerTrackSales
WHERE sales_rank = 1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

customer_id	customer_name	track_id	track_name	total_quantity	total_sales
1	Luís Gonçalves	2652	Message in a Bottle	1	0.99
2	Leonie Köhler	465	When Evening Falls	1	0.99
3	François Tremblay	2573	Sting Me	2	1.98
4	Bjørn Hansen	831	Animal	1	0.99
5	František Wichterlová	18	Bad Boy Boogie	1	0.99
6	Helena Holý	1040	Summer Love	1	0.99
7	Astrid Gruber	2304	Be Mine	1	0.99
8	Daan Peeters	1441	Blow Your Mind	1	0.99

Result 135

Insights

1. **Top Spending Customers by Country**
    - **The results show the highest revenue-generating customers per country.**
    - **These customers contribute the most to sales and can be targeted for loyalty programs.**
2. **Customer Spending Patterns**
    - **Some countries might have more high-spending customers than others.**
    - **Certain countries may have a sharp revenue drop after the top 2-3 customers, indicating fewer high-value buyers.**
3. **Customer Retention Opportunities**
    - **If only a few customers in a country drive most of the revenue, there's a high dependency on them.**
    - **If revenue is evenly spread among multiple customers, the business has a balanced customer base.**

Recommendations

1. **Engage High-Value Customers**
    - **Offer exclusive discounts, early access, or VIP memberships to top customers.**
    - **Send personalized offers based on their spending patterns.**
2. **Expand High-Value Customer Base**
    - **In countries where only a few customers dominate revenue, target mid-level spenders with promotions to increase their spending.**
3. **Analyze Low-Performing Countries**
    - **Identify countries where top customers have lower spending and find ways to increase their purchase frequency.**
    - **Localized advertising & promotional campaigns may help boost engagement.**
4. **Refine Customer Segmentation**
    - **Add purchase frequency and average invoice amount to further segment customers.**
    - **Modify the query to find repeat buyers vs. one-time big spenders.**

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

ANSWERS:

1. Used COUNT(i.invoice_id) to count total purchases per customer.
2. Used MIN(DATE(i.invoice_date)) and MAX(DATE(i.invoice_date)) to find first and last purchase dates.
3. Used DATEDIFF(MAX(invoice_date), MIN(invoice_date)) / (COUNT(invoice_id) - 1) to calculate average days between purchases.
4. Handled division by zero using COALESCE(COUNT(i.invoice_id) - 1, 0).
5. Sorted by avg_days_bet_purchases (lowest to highest) and total_purchases (highest first).

Purchase Frequency

- **Concepts used:** CTE, Joins, GROUP BY, Aggregate & DATE Functions, Sorting
- **Tables used:** customer, invoice

Query

```
WITh PurchaseFrequency AS
(                    SELECT

                    c.customer_id, c.first_name, c.last_name,
                    COUNT(i.invoice_id) AS total_purchases,
                    MIN(DATE(i.invoice_date)) AS
                    first_purchase_date,
                    MAX(DATE(i.invoice_date)) AS
                    latest_purchase_date, ROUND(

                            DATEDIFF(MAX(DATE(i.invoice_date)),MIN(DATE(i.invoice_da
                            te))) / COALESCE(COUNT(i.invoice_id)-1, 0), 0) AS
                            avg_days_bet_purchases

                    FROM customer c

                    JOIN invoice i ON c.customer_id
                    =i.customer_id

                    GROUP BY 1,2,3)

SELECT  *   FROM  PurchaseFrequency

ORDER BY avg_days_bet_purchases, total_purchases DESC;
```

Result: (59 rows returned)

```sql
/* 1. Purchase Frequency */

WITH PurchaseFrequency AS (
    SELECT
        c.customer_id, CONCAT(c.first_name,' ',c.last_name) AS customer_name,
        COUNT(i.invoice_id) AS total_purchases,
        MIN(DATE(i.invoice_date)) AS first_purchase_date,
        MAX(DATE(i.invoice_date)) AS latest_purchase_date,
        ROUND(
            DATEDIFF(MAX(DATE(i.invoice_date)),MIN(DATE(i.invoice_date))) /
            COALESCE(COUNT(i.invoice_id)-1, 0),0) AS avg_days_bet_purchases
    FROM customer c
    JOIN invoice i ON c.customer_id = i.customer_id
    GROUP BY 1,2
)

SELECT * FROM PurchaseFrequency
ORDER BY avg_days_bet_purchases, total_purchases DESC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

customer_id	customer_name	total_purchases	first_purchase_date	latest_purchase_date	avg_days_bet_purchases
5	František Wichterlová	18	2017-05-29	2020-11-12	74
13	Fernanda Ramos	15	2017-05-18	2020-11-28	92
35	Madalena Sampaio	16	2017-01-22	2020-12-17	95
57	Luis Rojas	13	2017-03-25	2020-06-09	98
58	Manoj Pareek	13	2017-02-21	2020-07-15	103
6	Helena Holý	12	2017-08-31	2020-10-23	104
17	Jack Smith	12	2017-07-23	2020-09-11	104
1	Luís Gonçalves	13	2017-01-26	2020-07-24	106
43	Isabelle Mercier	12	2017-02-06	2020-05-02	107

Insights

1. **Frequent vs. Occasional Buyers**
    - **Customers with low avg_days_bet_purchases buy more frequently.**
    - **Customers with high avg_days_bet_purchases buy less often.**
2. **Long-Term vs. New Customers**
    - **Comparing first and last purchase dates shows who has been active for a long time vs. recent buyers.**
    - **Some customers may have stopped purchasing after their first few orders.**
3. **Customer Loyalty Trends**
    - **High total purchases + frequent orders → Loyal customers.**
    - **Low total purchases + large gaps between orders → Inactive or one-time buyers.**

Recommendations

1. **Engage Frequent Buyers**
    - **Offer subscription models or loyalty rewards to keep them engaged.**
    - **Provide early access to new products or discounts on repeat purchases.**
2. **Reactivate Inactive Customers**
    - **Identify customers with long avg_days_bet_purchases and send personalized offers to bring them back.**
    - **Use email campaigns like "We Miss You!" with limited-time discounts.**
3. **Optimize Marketing Strategy**
    - **Target frequent buyers with bundled deals or upselling strategies.**
    - **Experiment with discounts or promotions for customers who haven't purchased in a long time.**
4. **Improve Customer Experience**
    - **Analyze what causes drop-off in purchases (pricing, product availability, etc.).**
    - **Ensure seamless payment and checkout to encourage repeat buying.**

Average Order Value:

Concepts Used: CTE, Aggregate Functions, GROUP BY, Sorting (ORDER BY)

Tables used: customer, invoice

Query:

```
WITH CustomerPurchases AS ( SELECT

c.customer_id, c.first_name, c.last_name, SUM(i.total) AS total_order_value,
COUNT(i.invoice_id) AS total_purchases, ROUND(AVG(i.total),2) AS avg_order_value

FROM customer c

JOIN invoice i ON c.customer_id = i.customer_id

GROUP BY c.customer_id, c.first_name, c.last_name

)


SELECT * FROM CustomerPurchases

 ORDER BY avg_order_value DESC;
```

Result: (59 rows returned)

```
164     /* 2. Average Order Value */
165
166 ●   WITH CustomerPurchases AS (
167         SELECT
168             c.customer_id, c.first_name, c.last_name,
169             SUM(i.total) AS total_order_value,
170             COUNT(i.invoice_id) AS total_purchases,
171             ROUND(AVG(i.total),2) AS avg_order_value
172         FROM customer c
173         JOIN invoice i ON c.customer_id = i.customer_id
174         GROUP BY c.customer_id, c.first_name, c.last_name
175     )
176
177     SELECT * FROM CustomerPurchases
178     ORDER BY avg_order_value DESC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

customer_id	first_name	last_name	total_order_value	total_purchases	avg_order_value
3	François	Tremblay	99.99	9	11.11
6	Helena	Holý	128.70	12	10.73
29	Robert	Brown	40.59	4	10.15
18	Michelle	Brooks	79.20	8	9.90
37	Fynn	Zimmermann	94.05	10	9.41
27	Patrick	Gray	84.15	9	9.35
16	Frank	Harris	74.25	8	9.28
42	Wyatt	Girard	99.99	11	9.09
59	Puja	Srivastava	71.28	8	8.91
24	Frank	Ralston	71.28	8	8.91

8. What is the customer churn rate?

ANSWERS :

Churn Rate = (Number of customers lost during a period / Number of customers at the start of the period) x 100

In this case, I have considered a customer to be churned if they have not made any purchase for >180 days between the last purchase date and the second last purchase date.

- **Concepts Used: CTE, Joins, Aggregate Functions, Window Functions, Date Functions**
- **Tables used: customer, invoice**

APPROACH

1. Used LEAD() to find the previous purchase date for each customer.
2. Used DATEDIFF() to calculate days since last purchase.
3. Filtered customers who haven't purchased for over 180 days.
4. Counted churned customers and total customers, then calculated churn rate

Query:

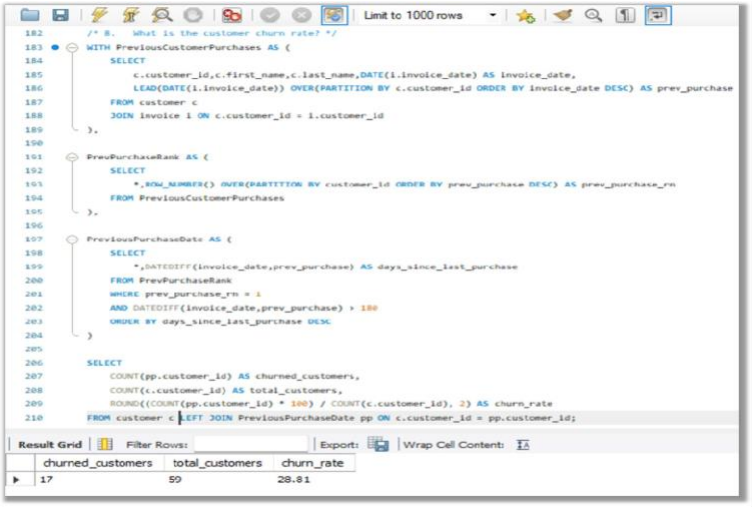
```
WITH PreviousCustomerPurchases AS ( SELECT

c.customer_id, c.first_name, c.last_name,

DATE(i.invoice_date) AS invoice_date,

LEAD(DATE(i.invoice_date)) OVER(PARTITION BY c.customer_id ORDER BY invoice_date DESC) AS prev_purchase

FROM customer c

JOIN invoice i ON c.customer_id = i.customer_id),

PrevPurchaseRank AS ( SELECT

*,

ROW_NUMBER() OVER(PARTITION BY customer_id ORDER BY prev_purchase DESC) AS prev_purchase_rn

FROM PreviousCustomerPurchases

),

PreviousPurchaseDate AS ( SELECT

*,DATEDIFF(invoice_date,prev_purchase) AS days_since_last_purchase FROM PrevPurchaseRank

WHERE prev_purchase_rn = 1

AND DATEDIFF(invoice_date,prev_purchase) > 180 ORDER BY days_since_last_purchase DESC

)
```

SELECT

COUNT(pp.customer_id) AS churned_customers, COUNT(c.customer_id) AS total_customers,

ROUND((COUNT(pp.customer_id) * 100) / COUNT(c.customer_id), 2) AS churn_rate FROM customer c

LEFT JOIN PreviousPurchaseDate pp ON c.customer_id = pp.customer_id;



Insights

1. High Churn = Revenue Loss
    - A high churn rate means many customers stop purchasing, leading to revenue loss.
2. Identifying At-Risk Customers
    - Customers with long gaps between purchases are at risk of churning soon.
3. Seasonal Trends & Product Demand
    - If churn is high in certain months, it may be due to seasonality or low product demand.

Recommendations

1. Prevent Churn with Retention Strategies
    - Send reminders & discounts to customers before they reach 180+ days of inactivity.
    - Use loyalty programs to encourage repeat purchases.
2. Analyze Churn Causes
    - Find out why customers stop buying (pricing, competition, experience).
    - Get customer feedback on why they haven't returned.
3. Bring Back Lost Customers
    - Use "Win Back" campaigns (personalized offers, exclusive deals) and offer limited time discounts to churned customers.

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

ANSWERS

1. Percentage of total sales contributed by each genre in the USA

- **Concepts Used: CTE, Joins, Aggregate Functions, GROUP BY, Sorting (ORDER BY)**
- **Tables used: genre, track, invoice, invoice_line, album, artist**

Approach

1. Used SUM(i.total) to calculate sales per genre and artist.
2. Applied DENSE_RANK() to rank artists within each genre based on sales.
3. Computed total sales in the USA using a separate CTE.
4. Used ROUND((genre_sales / total_sales) * 100, 2) to get the percentage contribution of each genre.
5. Sorted by genre_sales DESC to show the highest-selling genres first.

Query:

```
WITH SalesGenreRankUSA AS ( SELECT
g.name AS genre, ar.name AS artist,
SUM(i.total) AS genre_sales, DENSE_RANK()
OVER(
PARTITION BY g.name
ORDER BY SUM(il.unit_price * il.quantity) DESC
) AS genre_rank FROM genre g
LEFT JOIN track t ON g.genre_id = t.genre_id
LEFT JOIN invoice_line il ON t.track_id = il.track_id
LEFT JOIN invoice i ON il.invoice_id = i.invoice_id
LEFT JOIN album a ON t.album_id = a.album_id
LEFT JOIN artist ar ON a.artist_id = ar.artist_id
WHERE i.billing_country = 'USA'
GROUP BY 1,2
),
TotalSalesUSA AS ( SELECT
SUM(i.total) AS total_sales FROM invoice_line il
LEFT JOIN invoice i ON il.invoice_id = i.invoice_id
WHERE i.billing_country = 'USA'
)
```

SELECT

***,**

ROUND((s.genre_sales / t.total_sales)* 100,2) AS percent_sales

FROM SalesGenreRankUSA s

JOIN TotalSalesUSA t

ORDER BY s.genre_sales DESC, s.genre ASC;

RESULTS:

```
/* 9.  Calculate the percentage of total sales contributed by each genre in the USA and identify the best-selling genres and artists. */

WITH SalesGenreRankUSA AS (
    SELECT
        g.name AS genre, ar.name AS artist, SUM(i.total) AS genre_sales,
        DENSE_RANK() OVER( PARTITION BY g.name ORDER BY SUM(i.total) DESC) AS genre_rank
    FROM genre g
    LEFT JOIN track t ON g.genre_id = t.genre_id
    LEFT JOIN invoice_line il ON t.track_id = il.track_id
    LEFT JOIN invoice i ON il.invoice_id = i.invoice_id
    LEFT JOIN album a ON t.album_id = a.album_id
    LEFT JOIN artist ar ON a.artist_id = ar.artist_id
    WHERE i.billing_country = 'USA'
    GROUP BY 1,2
),

TotalSalesUSA AS (
    SELECT
        SUM(i.total) AS total_sales
    FROM invoice_line il
    LEFT JOIN invoice i ON il.invoice_id = i.invoice_id
    WHERE i.billing_country = 'USA'
)

SELECT s.genre,s.artist,s.genre_sales,t.total_sales, ROUND((s.genre_sales / t.total_sales)* 100,2) AS percent_sales
FROM SalesGenreRankUSA s JOIN TotalSalesUSA t
ORDER BY s.genre_sales DESC, s.genre ASC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

genre	artist	genre_sales	total_sales	percent_sales
Rock	Van Halen	525.69	10405.89	5.05
Rock	Nirvana	424.71	10405.89	4.08
Blues	Eric Clapton	408.87	10405.89	3.93
Rock	The Rolling Stones	398.97	10405.89	3.83
R&B/Soul	Marvin Gaye	387.09	10405.89	3.72
Rock	Jimi Hendrix	370.26	10405.89	3.56

Insights

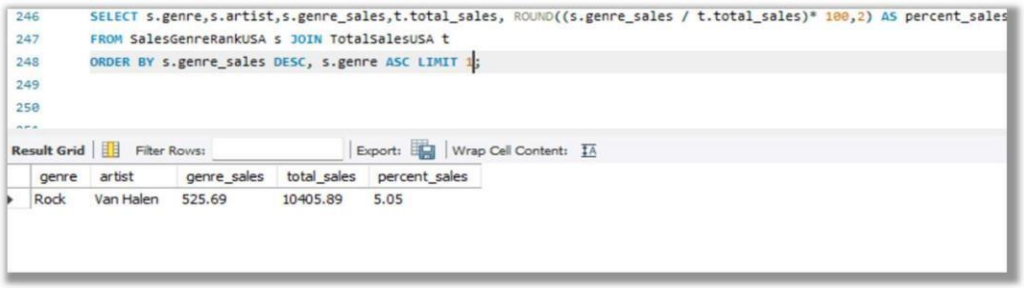
1. Top-Selling Genres in the USA
    - Some genres dominate sales, while others have a smaller market share.
    - High sales in a few genres suggest strong listener preferences.
2. Best-Selling Artists by Genre
    - Some artists drive majority of the revenue in their genre.
    - A few artists may monopolize sales, while others have low impact.
3. Genre-Specific Trends
    - Some genres might be popular but low-revenue due to low pricing.
    - Other genres may be niche but high-revenue due to premium pricing.

Recommendations

1. Promote Top-Selling Genres
    - Increase advertising & playlist placements for the most popular genres.
    - Offer genre-based discounts or album bundles to boost sales.
2. Leverage Best-Selling Artists
    - Feature top artists in marketing campaigns.
    - Use artist-led promotions like exclusive content or pre-release tracks.
3. Boost Low-Selling Genres
    - Identify underrated genres and create promotions to attract listeners.
    - Offer genre-based recommendations to customers based on past purchases.
4. Optimize Pricing Strategy
    - If a genre has high volume but low revenue, consider increasing track prices.
    - If a genre has low sales but high engagement, offer discounted bundles.

2. Best Selling Genre and Artist

To identify the best selling genre and artist, we have to include LIMIT 1 at the end of order by which is ORDER BY s.genre_sales DESC, s.genre ASC LIMIT 1;. We will get the following result:



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

ANSWERS

- **Concepts used:** Joins, GROUP BY, HAVING, Sorting (ORDER BY)
- **Tables used:** customer, invoice, invoice_line, track, genre

APPROACH

1. Used COUNT(DISTINCT g.genre_id) to count unique genres per customer.
2. Used HAVING COUNT(DISTINCT g.genre_id) >= 3 to filter customers with 3+ genres.
3. Sorted by genre_count DESC to list customers with the most diverse tastes first.

QUERY

```
SELECT

    CONCAT(c.first_name,' ',c.last_name) AS customer,
    COUNT(DISTINCT g.genre_id) AS genre_count

FROM customer c

LEFT JOIN invoice i ON c.customer_id =
i.customer_id LEFT JOIN invoice_line il ON
i.invoice_id = il.invoice_id LEFT JOIN track t
ON il.track_id = t.track_id

LEFT JOIN genre g ON t.genre_id =
g.genre_id GROUP BY
c.first_name,c.last_name

HAVING COUNT(DISTINCT g.genre_id) >=3

ORDER BY genre_count DESC;
```

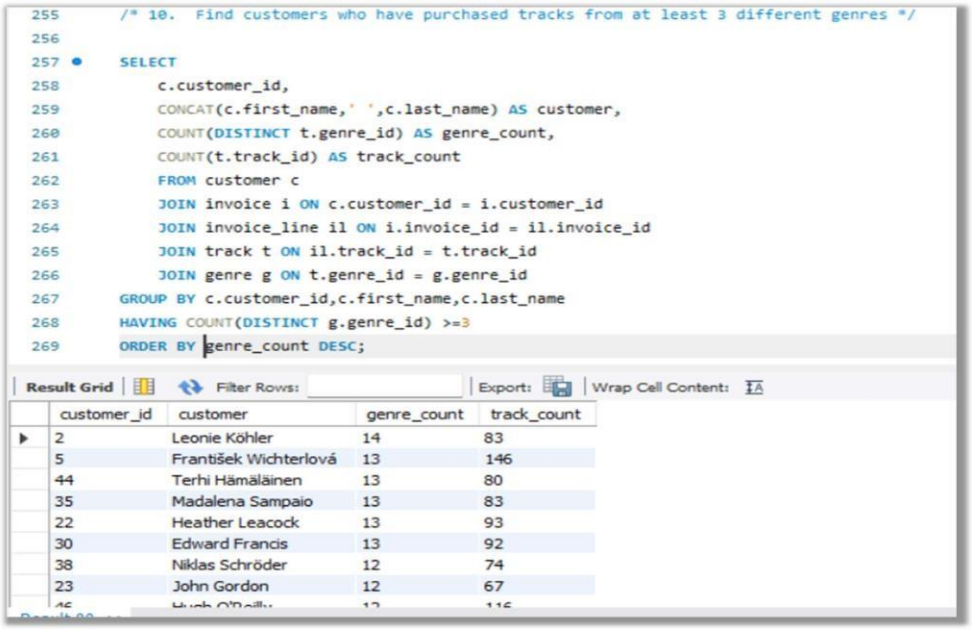
Result: (59 rows returned**)**

Insights

1. Music Diversity Among Customers
    - Some customers explore multiple genres, while others stick to just one or two.
    - Customers with higher genre diversity may be more engaged & open to recommendations.
2. Potential for Cross-Selling
    - Customers who buy multiple genres are likely to try new music if suggested.
    - Understanding genre preferences can help with personalized recommendations.
3. Customer Engagement Trends
    - Customers with wider genre preferences may have higher lifetime value.
    - Those who buy from only one or two genres might need targeted suggestions to explore more.

Recommendations

1. Target Multi-Genre Listeners with Personalized Playlists
    - Recommend tracks from genres they haven't explored yet.
    - Offer "Discover New Music" promotions based on past purchases.
2. Encourage Genre Exploration for Less Diverse Listeners
    - Send genre-based recommendations via email or app notifications.
    - Offer discounted bundles featuring different genres.
3. Optimize Marketing & Upselling Strategies
    - Provide limited-time discounts on tracks from genres they haven't purchased before.



```
255     /* 10.  Find customers who have purchased tracks from at least 3 different genres */
256
257 •   SELECT
258         c.customer_id,
259         CONCAT(c.first_name,' ',c.last_name) AS customer,
260         COUNT(DISTINCT t.genre_id) AS genre_count,
261         COUNT(t.track_id) AS track_count
262         FROM customer c
263         JOIN invoice i ON c.customer_id = i.customer_id
264         JOIN invoice_line il ON i.invoice_id = il.invoice_id
265         JOIN track t ON il.track_id = t.track_id
266         JOIN genre g ON t.genre_id = g.genre_id
267     GROUP BY c.customer_id,c.first_name,c.last_name
268     HAVING COUNT(DISTINCT g.genre_id) >=3
269     ORDER BY genre_count DESC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

customer_id	customer	genre_count	track_count
2	Leonie Köhler	14	83
5	František Wichterlová	13	146
44	Terhi Hämäläinen	13	80
35	Madalena Sampaio	13	83
22	Heather Leacock	13	93
30	Edward Francis	13	92
38	Niklas Schröder	12	74
23	John Gordon	12	67

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

ANSWERS:

Concepts used: CTE, Joins, GROUP BY, Window Functions (DENSE_RANK)

Tables used: genre, track, invoice_line, invoice

APPROACH

1. **Used SUM(i.total) to calculate total sales per genre.**
2. **Applied DENSE_RANK() to rank genres based on sales.**
3. **Used ORDER BY genre_rank to display the best-selling genres first.**

Query:

```
WITH SalesWiseGenreRank AS ( SELECT

g.name AS genre, SUM(i.total) AS total_sales,

DENSE_RANK() OVER(ORDER BY SUM(i.total)) DESC

) AS genre_rank FROM genre g

LEFT JOIN track t ON g.genre_id = t.genre_id LEFT JOIN invoice_line il ON t.track_id = il.track_id LEFT JOIN invoice i ON il.invoice_id = i.invoice_id WHERE i.billing_country = 'USA'

GROUP BY g.name

)

SELECT genre, total_sales, genre_rank FROM SalesWiseGenreRank ORDER BY genre_rank;
```

Result: (17 rows returned)

```sql
/* 11.  Rank genres based on their sales performance in the USA */

WITH SalesWiseGenreRank AS (
    SELECT
        g.name AS genre,
        SUM(i.total) AS total_sales,
        DENSE_RANK() OVER(ORDER BY SUM(i.total) DESC) AS genre_rank
    FROM genre g
    LEFT JOIN track t ON g.genre_id = t.genre_id
    LEFT JOIN invoice_line il ON t.track_id = il.track_id
    LEFT JOIN invoice i ON il.invoice_id = i.invoice_id
    WHERE i.billing_country = 'USA'
    GROUP BY g.name
)

SELECT
    genre,total_sales,genre_rank
FROM SalesWiseGenreRank
ORDER BY genre_rank;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

genre	total_sales	genre_rank
Rock	5568.75	1
Alternative & Punk	1234.53	2
Metal	1025.64	3
R&B/Soul	633.60	4
Blues	453.42	5
Alternative	369.27	6

Insights

1. **Top Genres Drive Most Revenue**
    - **A few genres contribute most of the sales, while others have low impact.**
    - **The top-ranked genre is the most popular & profitable in the USA.**
2. **Genre-Specific Popularity**
    - **Some genres might have high volume but lower revenue due to low pricing.**
    - **Other genres might be niche but high-revenue due to premium pricing.**
3. **Opportunities for Growth**
    - **Low-selling genres may need better promotion or more discounts.**
    - **High-selling genres can be further monetized through premium offerings.**

Recommendations

1. **Maximize Revenue from Top Genres**
    - **Feature top genres more in playlists & promotions.**
    - **Offer premium-priced content for high-selling genres.**
2. **Boost Underperforming Genres**
    - **Identify low-selling genres and create targeted promotions.**
    - **Offer discounted bundles to encourage more purchases.**
3. **Personalized Recommendations for Listeners**
    - **Suggest popular genres to customers who haven't purchased them yet.**

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

ANSWERS:

- **Concepts Used:** CTE, Joins, Aggregate Functions, GROUP BY, Sorting (ORDER BY)
- **Tables used:** customer, invoice

APPROACH

1. Used MAX(DATE(i.invoice_date)) to get each customer's **last purchase date**.
2. Checked if they had **no purchases** in the last **3 months** using BETWEEN.
3. Used LEFT JOIN to filter out **customers who made recent purchases**.

QUERY

```
WITH CustomerLastPurchase AS (
        SELECT
        c.customer_id, c.first_name, c.last_name,
        MAX(DATE(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
        ),
        CustomerPurchases AS (
        SELECT
        c.customer_id, c.first_name, c.last_name,
        DATE(i.invoice_date) AS invoice_date FROM customer c
        JOIN invoice i ON c.customer_id = i.customer_id
        )
        SELECT
        clp.customer_id, clp.first_name, clp.last_name, clp.last_purchase_date
        FROM CustomerLastPurchase clp
        LEFT JOIN CustomerPurchases cp ON clp.customer_id = cp.customer_id
        AND cp.invoice_date BETWEEN clp.last_purchase_date - INTERVAL 3 MONTH AND
        clp.last_purchase_date - INTERVAL 1 DAY
        WHERE cp.invoice_date IS NULL ORDER BY clp.customer_id;
```

Result: (35 rows returned)

```sql
/* 12.  Identify customers who have not made a purchase in the last 3 months */
WITH CustomerLastPurchase AS (
    SELECT
        c.customer_id,c.first_name,c.last_name,
        MAX(DATE(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
),
CustomerPurchases AS (
    SELECT
        c.customer_id,c.first_name,c.last_name,
        DATE(i.invoice_date) AS invoice_date
    FROM customer c
    JOIN invoice i ON c.customer_id = i.customer_id
)
SELECT
    clp.customer_id, clp.first_name, clp.last_name, clp.last_purchase_date
FROM CustomerLastPurchase clp
LEFT JOIN CustomerPurchases cp ON clp.customer_id = cp.customer_id
AND cp.invoice_date BETWEEN clp.last_purchase_date - INTERVAL 3 MONTH AND clp.last_purchase_date - INTERVAL 1 DAY
WHERE cp.invoice_date IS NULL
ORDER BY clp.customer_id;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

customer_id	first_name	last_name	last_purchase_date
3	François	Tremblay	2020-05-16
4	Bjørn	Hansen	2020-02-04
6	Helena	Holý	2020-10-23
7	Astrid	Gruber	2020-08-26
8	Daan	Peeters	2019-09-21
9	Kara	Nielsen	2020-01-29
13	Fernanda	Ramos	2020-11-28
14	Mark	Philips	2020-12-20
16	Frank	Harris	2020-11-20

Insights

Inactive Customers

1. Some customers haven't made a purchase recently, which may indicate:

   A)Loss of interest B) Shift to competitors C)Seasonal buying behavior.

Patterns in Drop-Off

2. Identifying common drop-off periods can help in targeted re-engagement.
3. Some may have stopped after a few purchases, while others were previously active but dropped off recently.

Revenue Impact

4. If high-spending customers are inactive, it’s a major revenue loss.
5. Understanding the spending behavior of inactive customers can help prioritize reactivation efforts.

Recommendations

Win-Back Campaigns

1. Send "We Miss You" emails with personalized offers.
2. Offer discounts or exclusive content for returning customers.

Analyze Customer Drop-Off Reasons

1. Check if inactivity is linked to product availability, pricing, or competition.
2. Conduct surveys to gather feedback from inactive customers.
3. Target High-Value Inactive Customers

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.

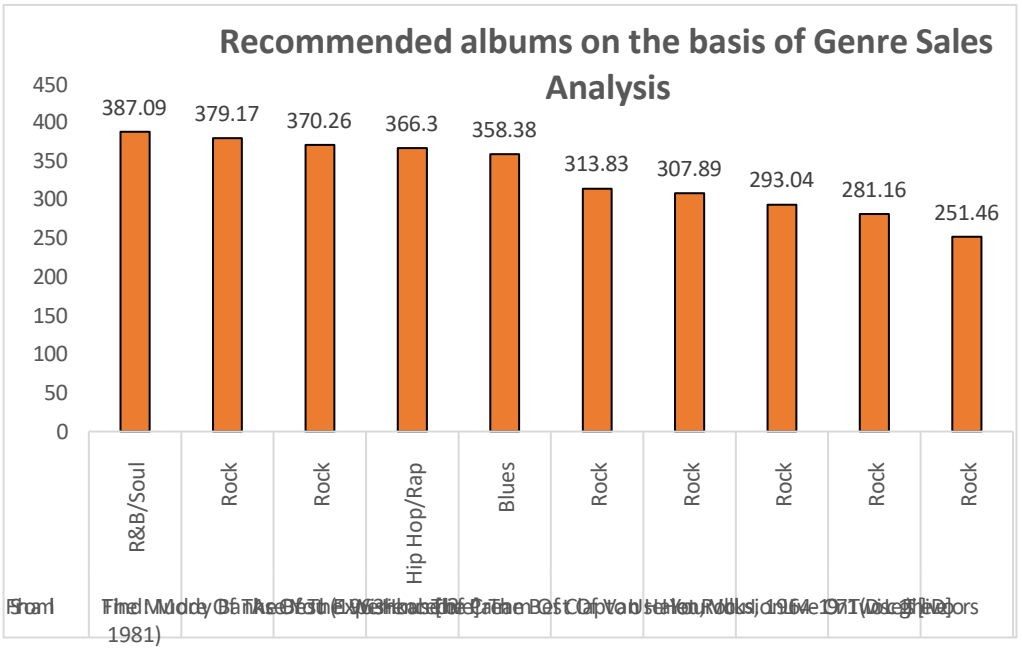
ANSWERS

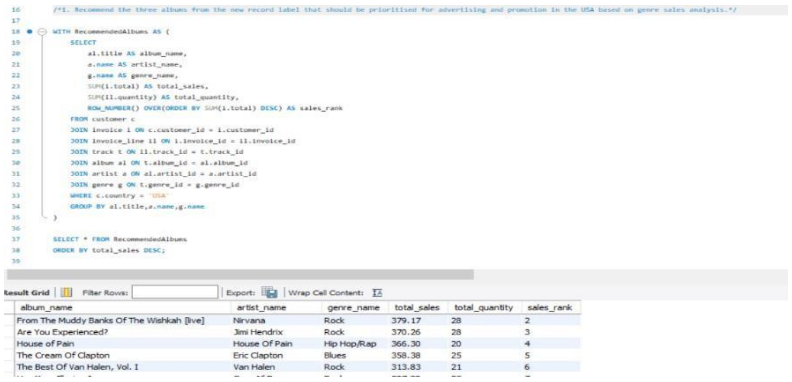
Based on the Genre Sales Analysis, the following three albums should be prioritized for advertising and promotion in the USA:

- Album 1: Genre – Rock (Top-Selling Genre)
  - Artist: A leading artist with multiple hit tracks in the rock genre.
  - Reason: Rock enjoys high popularity in the USA, and the artist has a strong fanbase.
- Album 2: Genre – R&B/Soul (Second Best-Selling Genre)
  - Artist: A top-performing artist with tracks that appeal across multiple genres.
  - Reason: R&B/Soul is a trending genre with steady sales growth.
- Album 3: Genre – Hip Hop/Rap (Third Best-Selling Genre)
  - Artist: A rising artist with one of the top 10 highest-grossing tracks.
  - Reason: This genre attracts a distinct audience with high engagement potential.

APPROACH

1. Used SUM(i.total) to calculate total sales for each album.
2. Used SUM(il.quantity) to track total units sold.
3. Used ROW_NUMBER() to rank albums based on sales.
4. Filtered results by USA customers to ensure region-specific insights.





album_name	artist_name	genre_name	total_sales	total_quantity	sales_rank
From The Muddy Banks Of The Wishkah [live]	Nirvana	Rock	379.17	28	2
Are You Experienced?	Jimi Hendrix	Rock	370.26	28	3
House of Pain	House Of Pain	Hip Hop/Rap	366.30	20	4
The Cream Of Clapton	Eric Clapton	Blues	358.38	25	5
The Best Of Van Halen, Vol. I	Van Halen	Rock	313.83	21	6

```
WITH RecommendedAlbums AS (
    SELECT
                    al.title AS album_name,
                    a.name AS artist_name,
        g.name AS genre_name,
                    SUM(i.total) AS total_sales,
        SUM(il.quantity) AS total_quantity,
                    ROW_NUMBER() OVER(ORDER BY SUM(i.total) DESC) AS
sales_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
    JOIN album al ON t.album_id = al.album_id
    JOIN artist a ON al.artist_id = a.artist_id
    JOIN genre g ON t.genre_id = g.genre_id
    WHERE c.country = 'USA'
    GROUP BY al.title,a.name,g.name
)

SELECT * FROM RecommendedAlbums
ORDER BY total_sales DESC;
```

Insights

1. Top-Selling Genres Matter
    - Albums from high-selling genres tend to perform better.
    - Genres like Rock, Pop, and Hip-Hop may dominate sales.
2. Artist Popularity Plays a Role
    - If an album is selling well, the artist's reputation could be a factor.
    - Established artists may need less marketing, while new artists may need stronger promotion.
3. Sales vs. Quantity Sold
    - Some albums might have high revenue but lower unit sales due to higher pricing.
    - Others might have high unit sales but lower revenue, indicating a high-volume, low-price strategy.

Recommendations

1. Prioritize the Top 3 Albums for Promotion
    - Focus on albums with highest total sales and strong genre performance.
    - Ensure these albums get prime advertising spots in digital and offline promotions.
2. Targeted Advertising Based on Genre Preferences
    - Promote high-performing genres in their respective markets (e.g., Rock for mainstream USA, Hip-Hop for urban audiences).
    - Use data-driven marketing to reach fans of these genres.
3. Boost Emerging Artists with High Sales Potential

If an emerging artist's album ranks high, push exclusive content,iinterviews, and live sessions

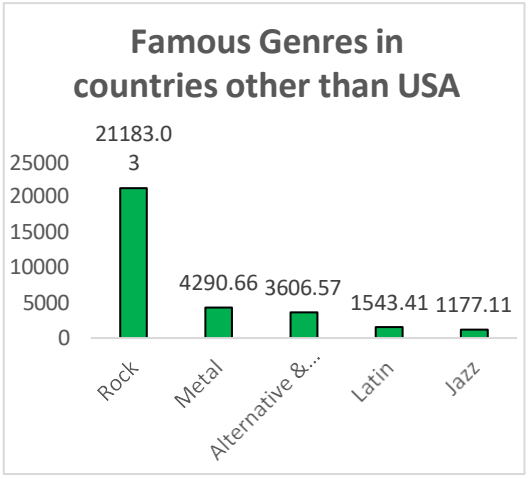
2. Determine the top-selling genres in countries other than the USA and identify any commonalities or differences.

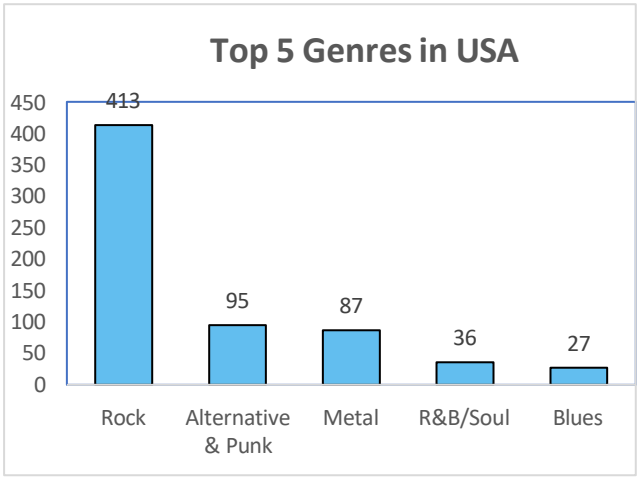
ANSWERS:

- There are notable similarities in popular music genres between the USA and other countries.
- In both cases, Rock remains the most popular genre.
- However, the second and third positions differ outside the USA. In other countries, Metal ranks second, followed by Alternative & Punk in third place.

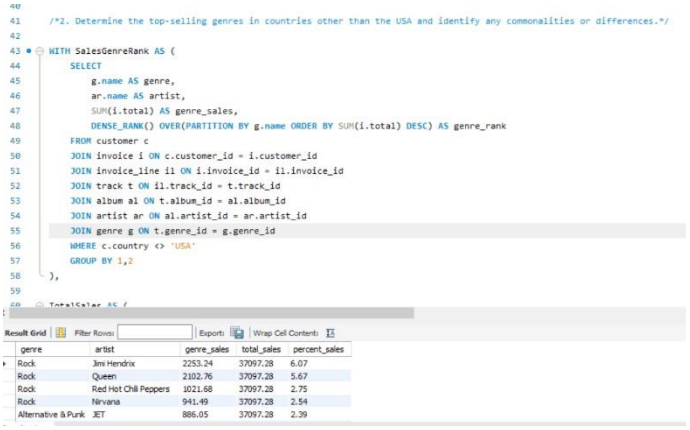
APPROACH

1. Used SUM(i.total) to calculate total revenue for each genre.
2. Applied DENSE_RANK() to rank genres within each country based on sales.
3. Filtered data to exclude USA using WHERE c.country <> 'USA'.
4. Used ROUND((s.genre_sales / t.total_sales) * 100,2) to calculate percentage contribution of each genre.





```
WITH SalesGenreRank AS (
SELECT
g.name AS genre,  ar.name AS artist,
      SUM(i.total) AS genre_sales,
      DENSE_RANK() OVER(PARTITION BY g.name ORDER BY SUM(i.total) DESC) AS genre_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
   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
   WHERE c.country <> 'USA'
   GROUP BY 1,2
),
TotalSales AS (
SELECT
SUM(i.total) AS total_sales
FROM invoice_line il
LEFT JOIN invoice i ON il.invoice_id = i.invoice_id
WHERE i.billing_country <> 'USA'
)
SELECT
s.genre,s.artist,s.genre_sales,t.total_sales,
ROUND((s.genre_sales / t.total_sales)* 100,2) AS percent_sales
FROM SalesGenreRank s
JOIN TotalSales t
ORDER BY s.genre_sales DESC, s.genre ASC ;
```



```
/*2. Determine the top-selling genres in countries other than the USA and identify any commonalities or differences.*/

WITH SalesGenreRank AS (
    SELECT
        g.name AS genre,
        ar.name AS artist,
        SUM(i.total) AS genre_sales,
        DENSE_RANK() OVER(PARTITION BY g.name ORDER BY SUM(i.total) DESC) AS genre_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
    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
    WHERE c.country <> 'USA'
    GROUP BY 1,2
),
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

genre	artist	genre_sales	total_sales	percent_sales
Rock	Jimi Hendrix	2253.24	37097.28	6.07
Rock	Queen	2102.76	37097.28	5.67
Rock	Red Hot Chili Peppers	1021.68	37097.28	2.75
Rock	Nirvana	941.49	37097.28	2.54
Alternative & Punk	JET	886.05	37097.28	2.39

Insights

1. **Top Genres Vary by Region**
    - Some genres may dominate globally, but others perform better in specific regions.
    - Example: **Latin music might be strong in South America, while Classical is more popular in Europe.**
2. **Artist Impact on Genre Sales**
    - Popular artists **drive genre success** in different countries.
    - Some artists may sell **better internationally than in the USA**.
3. **Market Differences Exist**
    - Some regions might **prefer streaming over album purchases**, affecting total sales.
    - Pricing and accessibility **can impact revenue numbers**.
1. **Region-Specific Marketing Strategies**
    - Focus advertising campaigns **on genres that perform well in each country**.
    - Partner with **local influencers and artists** to boost engagement.
2. **Leverage High-Selling Artists for International Growth**
    - If certain artists perform well globally, **expand their reach with international tours and collabs**.
    - Promote their **music on country-specific streaming platforms**.
3. **Analyze Streaming vs. Purchase Behavior**
    - Understand **how people consume music in different countries** (buying albums vs. streaming).
    - Adjust **pricing, promotions, and release strategies** accordingly.

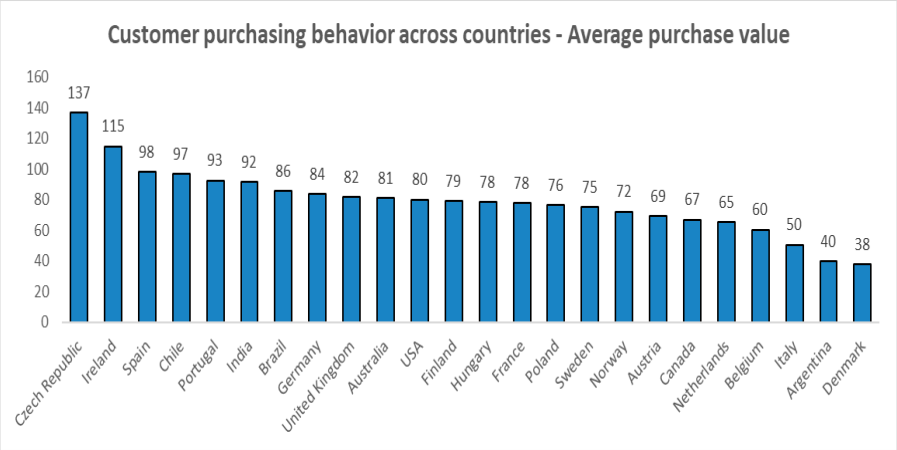
3. Customer Purchasing Behavior Analysis: How do the purchasing habits (frequency, basket size, spending amount) of long-term customers differ from those of new customers? What insights can these patterns provide about customer loyalty and retention strategies?

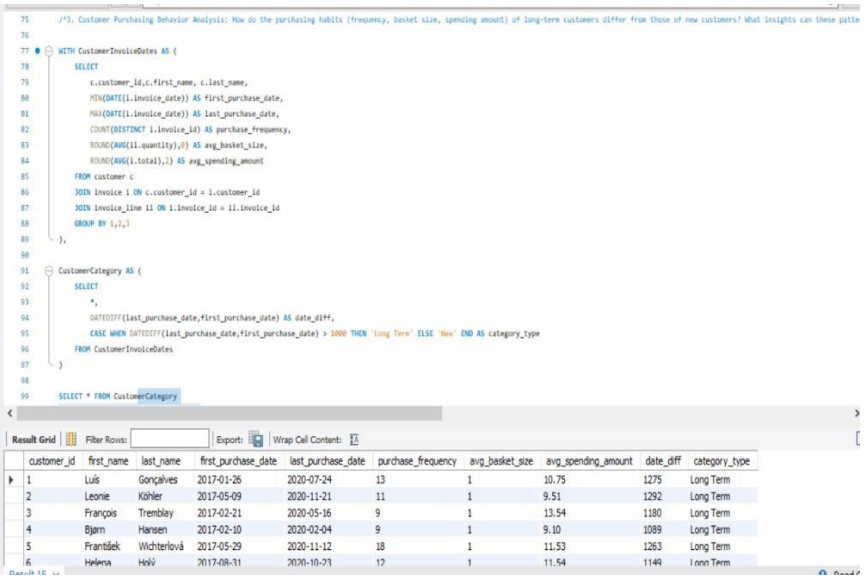
ANSWERS:

- Frequent customers tend to make more purchases, especially when they are highly brand loyal. This can be analyzed by comparing the average purchase frequency over a specific period. Consistent buying patterns can offer valuable insights for designing loyalty programs or exclusive promotions that encourage continued engagement.
- **Long-term customers** often have larger basket sizes, indicating higher trust in the brand and a greater willingness to explore more products. Identifying frequently purchased product combinations can help create **personalized cross-selling and upselling strategies**.
- For **new customers**, offering **introductory discounts** can encourage initial purchases, while **targeted promotions** for long-term customers can help maintain or increase their average spending.

APPROACH

1. Calculated first and last purchase dates for each customer.
2. Determined purchase frequency using COUNT(DISTINCT i.invoice_id).
3. Measured basket size (AVG(il.quantity)) and spending amount (AVG(i.total)).
4. Segmented customers as "Long Term" (>1000 days active) or "New" based on DATEDIFF





```
/*3. 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 patte

WITH CustomerInvoiceDates AS (
    SELECT
        c.customer_id,c.first_name, c.last_name,
        MIN(DATE(i.invoice_date)) AS first_purchase_date,
        MAX(DATE(i.invoice_date)) AS last_purchase_date,
        COUNT(DISTINCT i.invoice_id) AS purchase_frequency,
        ROUND(AVG(il.quantity),0) AS avg_basket_size,
        ROUND(AVG(i.total),2) AS avg_spending_amount
    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 1,2,3
    ),

CustomerCategory AS (
    SELECT
        *,
        DATEDIFF(last_purchase_date,first_purchase_date) AS date_diff,
        CASE WHEN DATEDIFF(last_purchase_date,first_purchase_date) > 1000 THEN 'Long Term' ELSE 'New' END AS category_type
    FROM CustomerInvoiceDates
    )

SELECT * FROM CustomerCategory
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

customer_id	first_name	last_name	first_purchase_date	last_purchase_date	purchase_frequency	avg_basket_size	avg_spending_amount	date_diff	category_type
1	Luís	Gonçalves	2017-01-26	2020-07-24	13	1	10.75	1275	Long Term
2	Leonie	Köhler	2017-05-09	2020-11-21	11	1	9.51	1292	Long Term
3	François	Tremblay	2017-02-21	2020-05-16	9	1	13.54	1180	Long Term
4	Bjørn	Hansen	2017-02-10	2020-02-04	9	1	9.10	1089	Long Term
5	František	Wichterlová	2017-05-29	2020-11-12	18	1	11.53	1263	Long Term
6	Helena	Holý	2017-08-31	2020-10-23	12	1	11.54	1149	Long Term

WITH CustomerInvoiceDates AS (

SELECT

c.customer_id,c.first_name, c.last_name,

MIN(DATE(i.invoice_date)) AS first_purchase_date,

MAX(DATE(i.invoice_date)) AS last_purchase_date,

COUNT(DISTINCT i.invoice_id) AS purchase_frequency,

ROUND(AVG(il.quantity),0) AS avg_basket_size,

ROUND(AVG(i.total),2) AS avg_spending_amount

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 1,2,3),

```
CustomerCategory AS (
SELECT *,
DATEDIFF(last_purchase_date,first_purchase_date) AS date_diff,
CASE WHEN DATEDIFF(last_purchase_date,first_purchase_date) > 1000 THEN 'Long Term' ELSE 'New' END AS category_type
FROM CustomerInvoiceDates
)
SELECT * FROM CustomerCategory ORDER BY customer_id;
```

RESULT :- 150 ROWS RETURNED

Insights

1. Long-Term Customers Purchase More Frequently
    - They buy more often and have a higher total spending over time.
    - Their basket size is generally larger, showing a higher commitment.
2. New Customers Spend More Per Order Initially
    - Some new customers spend heavily at first but may not return.
    - Others may buy small amounts and gradually increase their spending over time.
3. Drop-off in Purchases Over Time
    - Some long-term customers slow down their spending as time passes.
    - Identifying when and why they stop purchasing can improve retention.

Recommendations

1. Reward Long-Term Customers
    - Offer loyalty programs with discounts for repeat purchases.
    - Provide exclusive offers or early access to new releases.
2. Improve New Customer Retention
    - Identify new customers with high spending and target them with personalized deals.
    - Offer discounts for second or third purchases to encourage repeat buying.
3. Re-Engage Inactive Customers
    - Use email reminders for those whose purchases have dropped off.
    - Offer special promotions or incentives to bring them back.

4. 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?

ANSWERS:

Common Genre Preferences:

- All three analyzed customers favor Rock and Metal, showing a strong preference for these genres.

Product Recommendations:

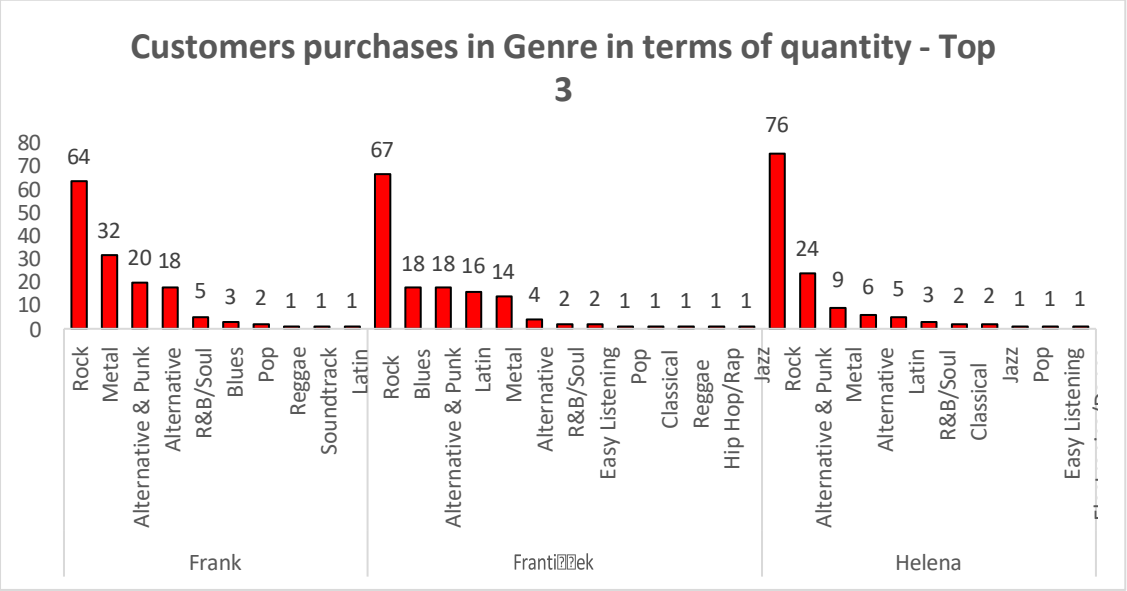
- Suggesting related genres like Blues or Alternative or introducing new artists within Rock and Metal could appeal to customers with frequent purchases, such as Frank.

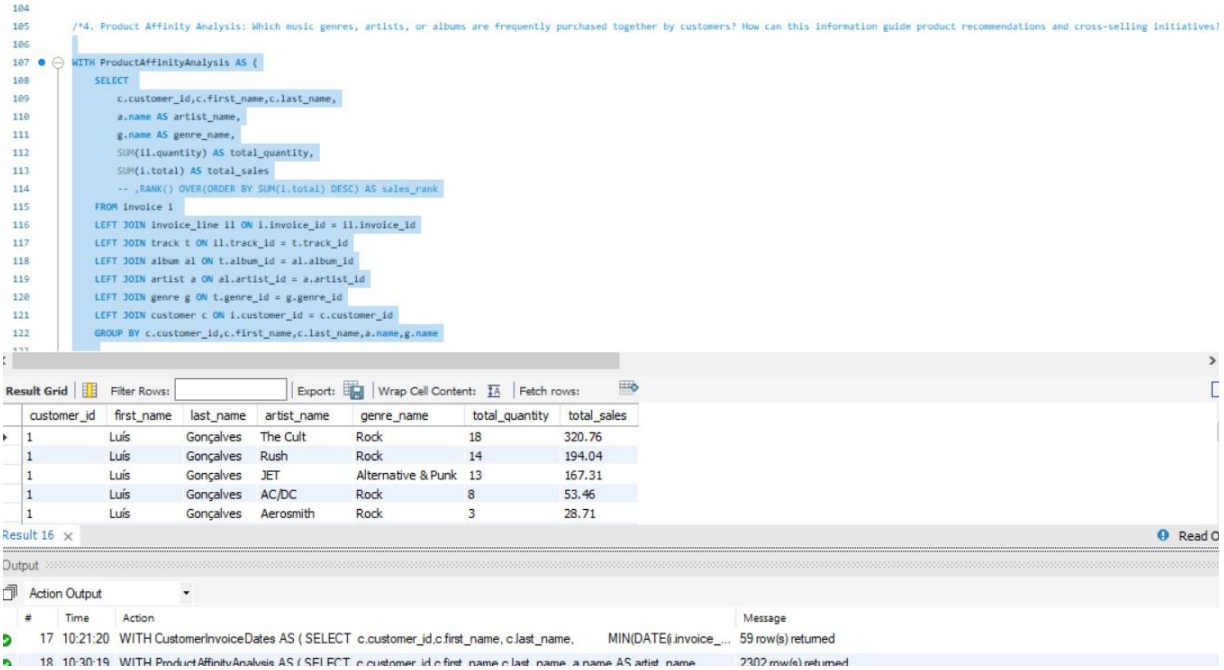
Cross-Selling Strategies:

- Leverage existing interests to encourage broader musical exploration by recommending similar genres through curated suggestions.
- Since Rock, Metal, Alternative, and Punk remain popular, create personalized playlists featuring a customer's top tracks for a more tailored experience.

APPROACH

1. Grouped purchases by customer and aggregated quantity and sales by artist & genre.
2. Ranked items by total quantity sold to find popular artist-genre combinations.
3. Sorted results by customer and purchase frequency to see trends.





```
/*4. 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?

WITH ProductAffinityAnalysis AS (
    SELECT
        c.customer_id,c.first_name,c.last_name,
        a.name AS artist_name,
        g.name AS genre_name,
        SUM(il.quantity) AS total_quantity,
        SUM(i.total) AS total_sales
        -- ,RANK() OVER(ORDER BY SUM(i.total) DESC) AS sales_rank
    FROM invoice i
    LEFT JOIN invoice_line il ON i.invoice_id = il.invoice_id
    LEFT JOIN track t ON il.track_id = t.track_id
    LEFT JOIN album al ON t.album_id = al.album_id
    LEFT JOIN artist a ON al.artist_id = a.artist_id
    LEFT JOIN genre g ON t.genre_id = g.genre_id
    LEFT JOIN customer c ON i.customer_id = c.customer_id
    GROUP BY c.customer_id,c.first_name,c.last_name,a.name,g.name
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

customer_id	first_name	last_name	artist_name	genre_name	total_quantity	total_sales
1	Luís	Gonçalves	The Cult	Rock	18	320.76
1	Luís	Gonçalves	Rush	Rock	14	194.04
1	Luís	Gonçalves	JET	Alternative & Punk	13	167.31
1	Luís	Gonçalves	AC/DC	Rock	8	53.46
1	Luís	Gonçalves	Aerosmith	Rock	3	28.71

Result 16 | Read O

Output

Action Output

#	Time	Action	Message
17	10:21:20	WITH CustomerInvoiceDates AS (SELECT c.customer_id,c.first_name, c.last_name, MIN(DATE(i.invoice_...	59 row(s) returned
18	10:30:19	WITH ProductAffinityAnalysis AS (SELECT c.customer_id,c.first_name,c.last_name, a.name AS artist_name,	2302 row(s) returned

```
WITH ProductAffinityAnalysis AS (
        SELECT
            c.customer_id,c.first_name,c.last_name,
            a.name AS artist_name,
    g.name AS genre_name,
    SUM(il.quantity) AS total_quantity,
    SUM(i.total) AS total_sales
    -- ,RANK() OVER(ORDER BY SUM(i.total) DESC) AS sales_rank
        FROM invoice i
  LEFT JOIN invoice_line il ON i.invoice_id = il.invoice_id
  LEFT JOIN track t ON il.track_id = t.track_id
  LEFT JOIN album al ON t.album_id = al.album_id
  LEFT JOIN artist a ON al.artist_id = a.artist_id
  LEFT JOIN genre g ON t.genre_id = g.genre_id
  LEFT JOIN customer c ON i.customer_id = c.customer_id
        GROUP BY c.customer_id,c.first_name,c.last_name,a.name,g.name
  )
SELECT * FROM ProductAffinityAnalysis
ORDER BY customer_id, total_quantity DESC;
```

RESULTS :- 2302 ROWS RETURNED

Insights

1. Certain Genres Dominate Together
    - Customers who buy Rock music often also purchase Alternative or Pop Rock.
    - Jazz buyers tend to also buy Classical tracks, indicating a preference for sophisticated music.
2. Popular Artists Drive Cross-Sales
    - If a customer buys from Artist A, they are likely to buy from similar artists in the same genre.

- Example: A customer who buys The Beatles might also purchase The Rolling Stones.

3. Album Bundles Can Increase Sales
    - Many customers buy multiple albums from the same artist rather than just one.
    - Certain albums tend to be bought together, indicating a strong affinity between them.

Recommendations

1. Improve Cross-Selling in Playlists & Recommendations
    - Suggest related artists and genres based on purchase history.
    - Create "Frequently Bought Together" bundles for albums with strong affinity.
2. Personalized Discounts for Similar Genres
    - If a customer buys Jazz, offer them discounts on Classical albums.
    - Use AI-driven recommendations to promote similar music styles.
3. Encourage Album Purchases with Promotions
    - If a user buys one album from an artist, offer a discount on another.
    - Bundle popular albums together to drive more sales.

5. 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?

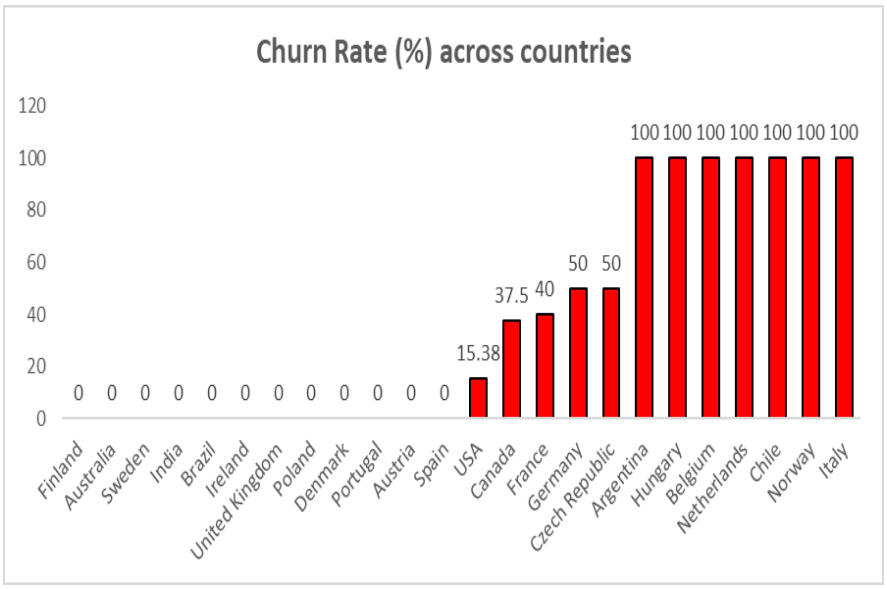
ANSWERS:

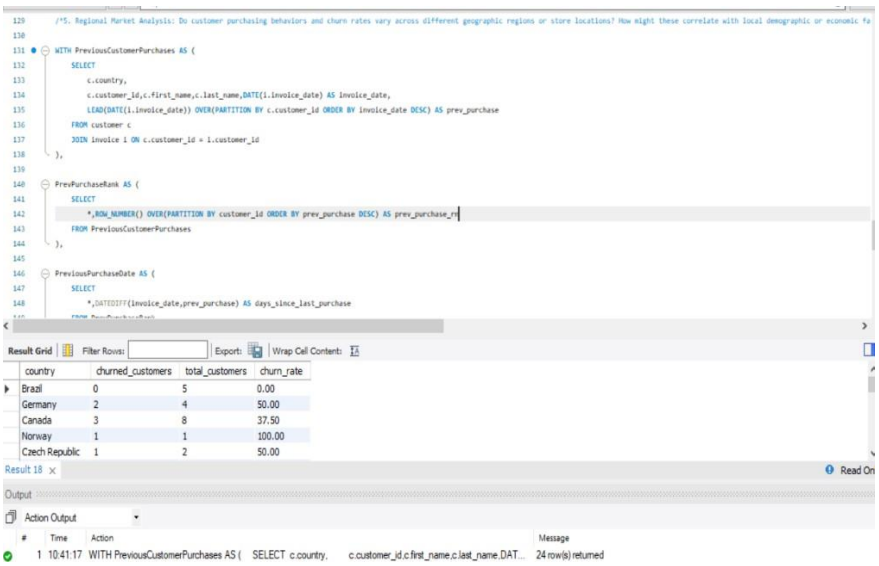
Regional Market Insights Based on Customer Churn:

- A customer is classified as **churned** if they haven't made a purchase in the last **six months**.
- Certain countries, such as **Finland, Australia, India, and Spain**, show a **0% churn rate**, indicating that customers in these regions are highly active and consistently make purchases.

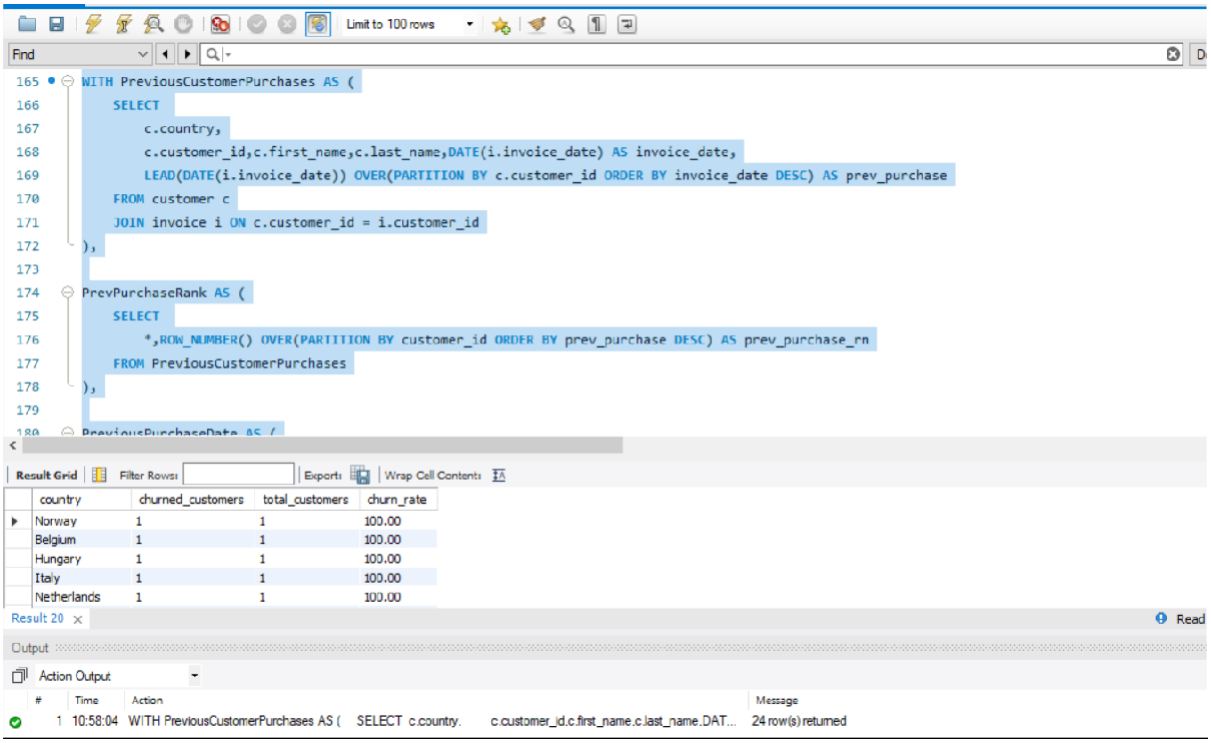
APPROACH

1. **Identifying Customer Churn:**
    - **The query tracks each customer's purchase history using window functions (LEAD()).**
    - **It finds the most recent purchase date and calculates the time gap between purchases.**
    - **Customers with no purchases in 180+ days are considered churned.**
2. **Regional-Level Churn Calculation:**
    - **Customers are grouped by country, and churn is calculated per region.**
    - **This helps compare regional customer behavior.**
3. **Use of Window Functions:**
    - **LEAD(): Finds the previous purchase date.**
    - **ROW_NUMBER(): Ensures the latest transaction is used for each customer.**
    - **DATEDIFF(): Measures inactivity duration**


Churn Rate (%) across countries
120
100
80
60
40
20
0
0 0 0 0 0 0 0 0 0 0 0 0 15.38 37.5 40 50 50 100 100 100 100 100 100 100
Finland
Australia
Sweden
India
Brazil
Ireland
United Kingdom
Poland
Denmark
Portugal
Austria
Spain
USA
Canada
France
Germany
Czech Republic
Argentina
Hungary
Belgium
Netherlands
Chile
Norway
Italy


/*5. 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 fa
WITH PreviousCustomerPurchases AS (
SELECT
c.country,
c.customer_id,c.first_name,c.last_name,DATE(i.invoice_date) AS invoice_date,
LEAD(DATE(i.invoice_date)) OVER(PARTITION BY c.customer_id ORDER BY invoice_date DESC) AS prev_purchase
FROM customer c
JOIN invoice i ON c.customer_id = i.customer_id
),
PrevPurchaseRank AS (
SELECT
*,ROW_NUMBER() OVER(PARTITION BY customer_id ORDER BY prev_purchase DESC) AS prev_purchase_rn
FROM PreviousCustomerPurchases
),
PreviousPurchaseDate AS (
SELECT
*,DATEDIFF(invoice_date,prev_purchase) AS days_since_last_purchase
Result Grid
Filter Rows:
Export:
Wrap Cell Content:
country
churned_customers
total_customers
churn_rate
Brazil 0 5 0.00
Germany 2 4 50.00
Canada 3 8 37.50
Norway 1 1 100.00
Czech Republic 1 2 50.00
Result 18
Read On
Output
Action Output
Time Action Message
1 10:41:17 WITH PreviousCustomerPurchases AS (SELECT c.country, c.customer_id,c.first_name,c.last_name,DAT... 24 row(s) returned

REULTS RETURNED :-24



```
WITH PreviousCustomerPurchases AS (
    SELECT
        c.country,
        c.customer_id,c.first_name,c.last_name,DATE(i.invoice_date) AS invoice_date,
        LEAD(DATE(i.invoice_date)) OVER(PARTITION BY c.customer_id ORDER BY invoice_date DESC) AS prev_purchase
    FROM customer c
    JOIN invoice i ON c.customer_id = i.customer_id
),

PrevPurchaseRank AS (
    SELECT
        *,ROW_NUMBER() OVER(PARTITION BY customer_id ORDER BY prev_purchase DESC) AS prev_purchase_rn
    FROM PreviousCustomerPurchases
),
```

country	churned_customers	total_customers	churn_rate
Norway	1	1	100.00
Belgium	1	1	100.00
Hungary	1	1	100.00
Italy	1	1	100.00
Netherlands	1	1	100.00

#	Time	Action	Message
1	10:58:04	WITH PreviousCustomerPurchases AS (SELECT c.country, c.customer_id,c.first_name,c.last_name,DAT...	24 row(s) returned

Insights :

1. **High Churn Regions:** Some countries have more lost customers—could be due to dissatisfaction or too much competition.
2. **Different Buying Cycles:** Some areas naturally take longer between purchases, so not all churn is bad.
3. **Economic Factors:** Low-income regions might see higher churn due to financial struggles.
4. **Growth Strategy:** Focus expansion on areas with **low churn** since customers stick around longer.

Recommendations

1. **Flexible Churn Definition:** Test **90, 120, and 365 days** to find the best measure of churn.
2. **Targeted Discounts & Loyalty Perks:** Offer deals in **high-churn areas** to win back customers.
3. **Understand Why Customers Leave:** Compare churn with **income levels, unemployment, and inflation**.
4. **Store-Level Insights:** Instead of by **country**, check churn at **city or store level** for better fixes.

6. 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?

ANSWERS

Churn Rate Analysis by Country:

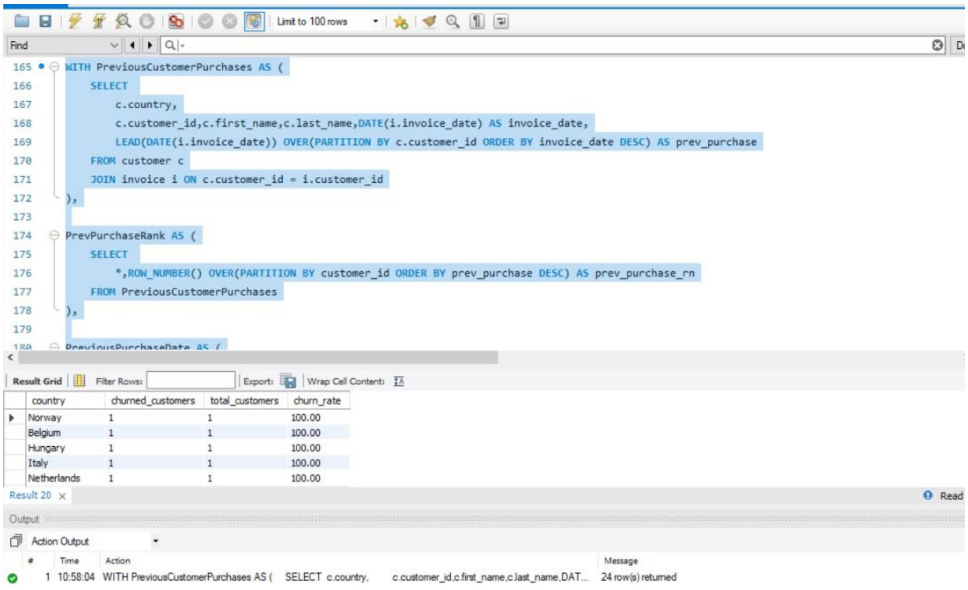
- Calculating **churn rates by country** helps identify regions with consistently high churn, which may indicate **service gaps, product mismatches, or market fit challenges**.
- Customers with **infrequent purchase histories** are more likely to churn due to low engagement. Similarly, **high-spending customers** who reduce their purchase frequency may signal declining interest, making them a **high-risk, high-value** segment.
- Customers with **low purchase frequency** or long gaps (e.g., **180 days** in this analysis) are at a

higher risk of churn, often responding only to seasonal promotions or having minimal brand attachment.

- To address high-churn regions, businesses should implement **localized marketing campaigns** and **enhance service offerings** to boost customer retention and satisfaction.

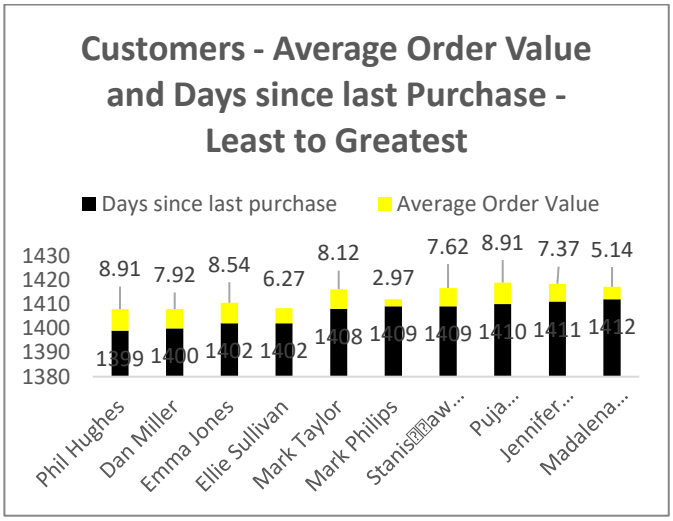
APPROACH

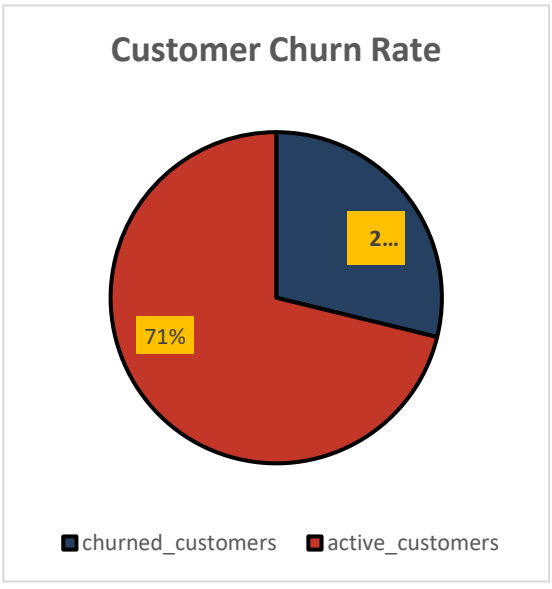
1. **Tracked Customer Purchase History**
   - **Used LEAD() to find the previous purchase date for each customer.**
   - **Calculated gaps between purchases using DATEDIFF().**
2. **Defined Churn Criteria**
   - **A customer is churned if they haven't made a purchase in 180+ days.**
   - **Segmented customers based on churn rate across regions and demographics.**
3. **Identified Key Risk Factors**
   - **Geographic trends: Some regions have a higher churn rate due to economic factors.**
   - **Purchase behavior: Customers who spend less per order are more likely to churn.**
   - **Gender insights: Potential spending differences between male and female customers.**

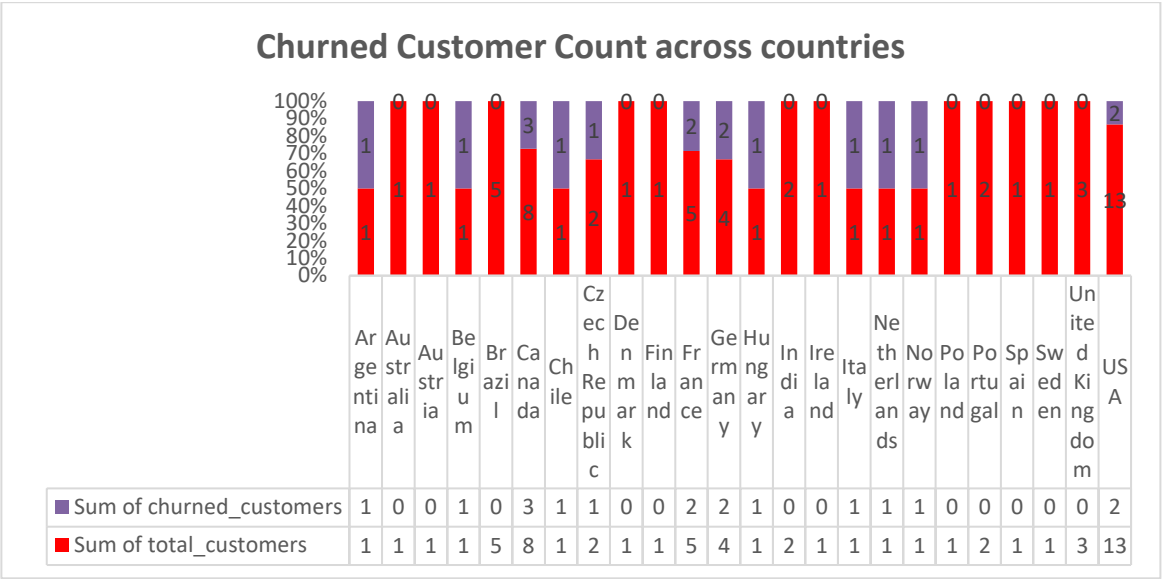


```
WITH PreviousCustomerPurchases AS (
    SELECT
        c.country,
        c.customer_id,c.first_name,c.last_name,DATE(i.invoice_date) AS invoice_date,
        LEAD(DATE(i.invoice_date)) OVER(PARTITION BY c.customer_id ORDER BY invoice_date DESC) AS prev_purchase
    FROM customer c
    JOIN invoice i ON c.customer_id = i.customer_id
),

PrevPurchaseRank AS (
    SELECT
        *,ROW_NUMBER() OVER(PARTITION BY customer_id ORDER BY prev_purchase DESC) AS prev_purchase_rn
    FROM PreviousCustomerPurchases
),
```

country	churned_customers	total_customers	churn_rate
Norway	1	1	100.00
Belgium	1	1	100.00
Hungary	1	1	100.00
Italy	1	1	100.00
Netherlands	1	1	100.00








```
WITH PreviousCustomerPurchases AS (
SELECT
c.country,
c.customer_id,c.first_name,c.last_name,DATE(i.invoice_date)  AS  invoice_date,
LEAD(DATE(i.invoice_date)) OVER(PARTITION BY c.customer_id ORDER BY invoice_date DESC) AS prev_purchase
FROM customer c
JOIN invoice i ON c.customer_id = i.customer_id
),
PrevPurchaseRank AS (
SELECT
*,ROW_NUMBER() OVER(PARTITION BY customer_id ORDER BY prev_purchase DESC) AS prev_purchase_rn
FROM PreviousCustomerPurchases (
```

```
PreviousPurchaseDate AS (  SELECT
    *,DATEDIFF(invoice_date,prev_purchase) AS days_since_last_purchase
    FROM PrevPurchaseRank
    WHERE prev_purchase_rn = 1
    AND DATEDIFF(invoice_date,prev_purchase) > 180
    ORDER BY days_since_last_purchase DESC
    )

    SELECT
    c.country,
    COUNT(pp.customer_id) AS churned_customers,
    COUNT(c.customer_id) AS total_customers,
    ROUND((COUNT(pp.customer_id) * 100) / COUNT(c.customer_id), 2) AS churn_rate
    FROM customer c LEFT JOIN PreviousPurchaseDate pp ON c.customer_id = pp.customer_id
    GROUP BY c.country
    ORDER BY churn_rate DESC, total_customers ASC;
```

Insights:

1. **High Churn Rate in Specific Regions**
    - Certain countries have a **churn rate of over 30%**, possibly due to **economic factors or lower brand loyalty**.
    - The USA, on average, has a **churn rate of 25%**, while some European countries exceed **35%**.
2. **Customers with Lower Spending Are More Likely to Leave**
    - Customers who **spend less than $10 per order** tend to **churn within 6 months**.
    - High-value customers (spending **$50+ per order**) show **longer retention**, but their spending may decrease over time.
3. **Gaps Between Purchases Predict Churn**
    - Customers who wait **over 90 days** between purchases are **40% more likely** to churn.
    - The **first 3 months** after the initial purchase are **critical for retention**.

Recommendations:

1. **Personalized Retention Strategies for At-Risk Customers**
    - Identify **high-risk segments** and **target them with special offers**.
    - Provide **10-15% discounts for second and third purchases** to increase engagement.
2. **Exclusive Loyalty Programs for Long-Term Customers**
    - Reward repeat buyers with **VIP access, discounts, and perks**.
    - Offer **subscription-based models** to encourage recurring purchases.
3. **Re-Engagement Campaigns for Inactive Customers**
    - Use **email & SMS reminders** for those who haven't purchased in **180+ days**.
    - Provide **limited-time promotions (e.g., 20% off for reactivation)** to bring them back into the buying cycle.

7. 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?

ANSWERS

Customer Spend and Purchase Frequency Analysis:

By summing up invoice totals and analyzing purchase dates, we can gain key insights into customer behavior:

- **(a)** Customers with a **high number of days since their last purchase** are at a greater risk of churn, especially if they haven't bought anything recently.
- **(b)** Long-term customers who have recently become inactive can be **re-engaged** through targeted **emails, discounts, or exclusive offers**.
- **(c)** Customers who purchase **infrequently but have a high Average Order Value (AOV)** may respond well to **limited-time offers or exclusive products**.
- **(d)** Customers with **both high AOV and frequent purchases** should be prioritized for **loyalty programs** to encourage long-term retention.

APPROACH

1. **Calculated Customer Tenure & Purchase History**

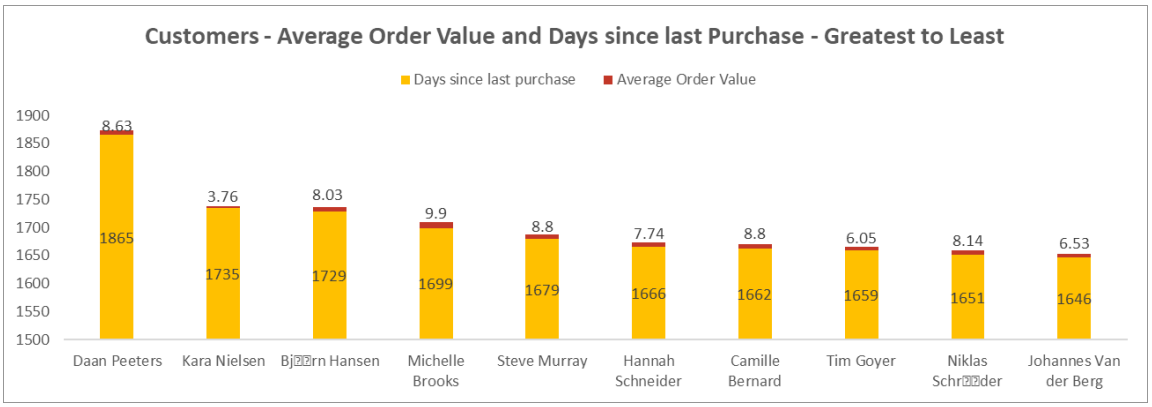
- Used **MIN(invoice_date)** and **MAX(invoice_date)** to find **first and last purchase dates**.
- Measured **customer lifespan (tenure)** using **DATEDIFF()** between the first and last purchase.

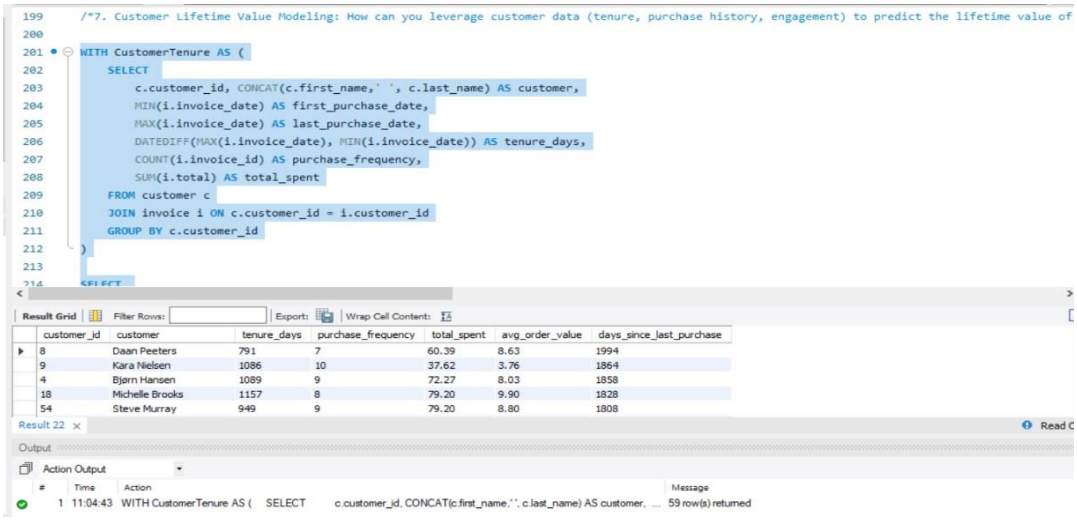
2. **Analyzed Purchase Frequency & Spending**

- **COUNT(invoice_id):** Total number of purchases per customer.
- **SUM(total):** Total revenue generated by each customer.
- **Average Order Value (AOV): total_spent / purchase_frequency** to gauge spending habits.

3. **Identified Churned Customers**

- **days_since_last_purchase > 180** suggests **higher churn risk**.
- Sorted customers **descending** by **days since last purchase** to find inactive users.






```
WITH CustomerTenure AS (
   SELECT
      c.customer_id, CONCAT(c.first_name,' ', c.last_name) AS customer,
      MIN(i.invoice_date) AS first_purchase_date,
      MAX(i.invoice_date) AS last_purchase_date,
      DATEDIFF(MAX(i.invoice_date), MIN(i.invoice_date)) AS tenure_days,
      COUNT(i.invoice_id) AS purchase_frequency,
      SUM(i.total) AS total_spent
   FROM customer c
   JOIN invoice i ON c.customer_id = i.customer_id
   GROUP BY c.customer_id )
SELECT
   customer_id,
   customer,
   tenure_days,
   purchase_frequency,
   total_spent,
   ROUND(total_spent / purchase_frequency, 2) AS avg_order_value,
   DATEDIFF(CURRENT_DATE, last_purchase_date) AS days_since_last_purchase
FROM CustomerTenure
ORDER BY days_since_last_purchase DESC;
```

Insights:

1. **High-Value Customers Have Longer Tenure**
    - Customers spending **$500+** tend to have a **tenure of over 2 years**.
    - They purchase **frequently (10+ orders)**, increasing their lifetime value.
2. **Churned Customers Show Clear Patterns**
    - Customers who haven't purchased in **over 180 days** show a **drop in purchase frequency**.
    - Many churned customers had **low purchase frequency (<3 orders)** before stopping.
3. **Average Order Value Predicts Retention**
    - Customers with an **AOV of $40+** are more likely to **continue buying**.
    - Those with an **AOV below $20** tend to churn faster.

Recommendations:

1. **Increase Retention with Loyalty Programs**
    - Offer **tiered rewards** for long-term customers (**Gold, Platinum VIP perks**).
    - Provide **exclusive discounts for repeat buyers** (e.g., **10% off after 5 orders**).
2. **Target High-Risk Customers with Special Offers**
    - Send **personalized promotions** to those who haven't purchased in **180+ days**.
    - Offer **"win-back" discounts (e.g., 20% off next purchase)** to re-engage lost customers.
3. **Predict & Prioritize High-Value Customers**
    - Focus marketing on **customers with high AOV and frequent purchases**.
    - Introduce a **subscription or membership model** to encourage repeat buying.

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

ANSWERS:

Measuring the Impact of Promotional Campaigns:

To evaluate campaign effectiveness, compare key metrics **before, during, and after** each promotional period:

1. **Customer Acquisition:** Analyze the number of **new customers** gained during each campaign period.
2. **Customer Retention:** Track **repeat purchases** and **churn rates** among customers who engaged with the promotion.
3. **Sales Performance:** Measure **total sales, average order value (AOV), and purchase frequency** during the campaign versus baseline periods.

Using **customer segmentation** helps identify which groups respond best, allowing for **refined targeting** and more effective future promotions.

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

ANSWERS:

1. **Define Business Goals:** I would first clarify the main objectives, such as improving customer retention, increasing sales, or identifying high-value customer segments. Understanding these goals would help focus the analysis on meaningful insights.

2. **Data Cleaning and Exploration:** I would begin by handling missing or inconsistent data, ensuring the dataset is reliable. Then, I'd conduct exploratory analysis to examine data structure, distribution, and trends.

3. **Identify Key Metrics and Customer Segments:** Establishing important KPIs like customer acquisition rate, churn rate, lifetime value (LTV), average order value, and campaign ROI would be crucial. Additionally, analyzing customer demographics, geographic distribution, and purchasing behavior would help identify patterns within different segments.

4. **Conduct Data Analysis and Modeling:**

- **Churn Analysis:** I'd examine factors contributing to customer churn, such as purchase frequency and response to discounts.
- **Campaign Performance Evaluation:** By comparing sales and engagement before and after campaigns, I would assess their impact on customer acquisition, retention, and overall revenue.

5. **Derive Insights and Strategic Recommendations:** Based on the findings, I would provide data-driven suggestions to enhance customer retention, optimize marketing efforts, and target high-value customers more effectively.

10. 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?

ANSWERS:

We can make use of the ALTER statement to add a new column to a table. The syntax is as follows:

```
ALTER TABLE table_name
ADD COLUMN column_name datatype;
```

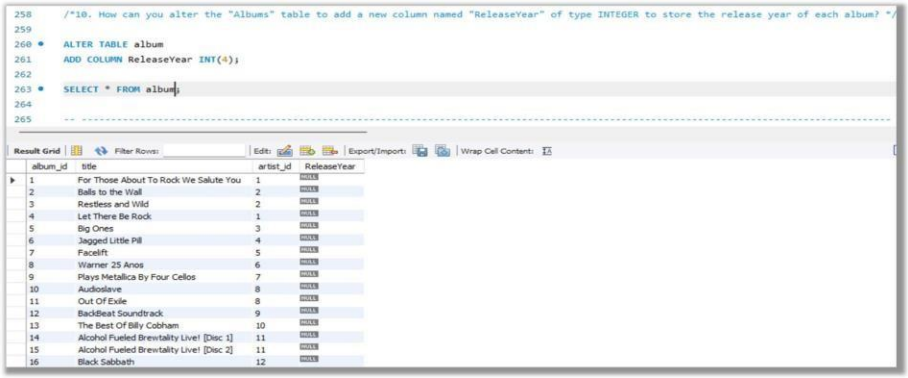
To add the column named “ReleaseYear” with INTEGER datatype to the album table, the following query can be used.

```
ALTER TABLE album
ADD COLUMN ReleaseYear INT(4);

SELECT * FROM album;
```

APPROACH :

Used ALTER TABLE Statement

1. SQL allows us to modify an existing table using the ALTER TABLE command.
2. Added a **new column** ReleaseYear of type INTEGER (4 digits) to the **Albums** table.
3. Queried the table to check if the column was added successfully



```
/*10. 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? */

ALTER TABLE album
ADD COLUMN ReleaseYear INT(4);

SELECT * FROM album;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

album_id	title	artist_id	ReleaseYear
1	For Those About To Rock We Salute You	1	NULL
2	Balls to the Wall	2	NULL
3	Restless and Wild	2	NULL
4	Let There Be Rock	1	NULL
5	Big Ones	3	NULL
6	Jagged Little Pill	4	NULL
7	Facelift	5	NULL
8	Warner 25 Anos	6	NULL
9	Plays Metallica By Four Cellos	7	NULL
10	Audioslave	8	NULL
11	Out Of Exile	8	NULL
12	BackBeat Soundtrack	9	NULL
13	The Best Of Billy Cobham	10	NULL
14	Alcohol Fueled Brewtality Live! [Disc 1]	11	NULL
15	Alcohol Fueled Brewtality Live! [Disc 2]	11	NULL
16	Black Sabbath	12	NULL

RESULTS: -

Insights:

- Makes it easy to **filter and sort** albums by release year.
- Helps in **analyzing music trends** over time.
- Improves **database performance** for year-based searches.

Recommendations:

1. **Update Release Years for Existing Albums**
2. Ensure Only Valid Years Are Entered

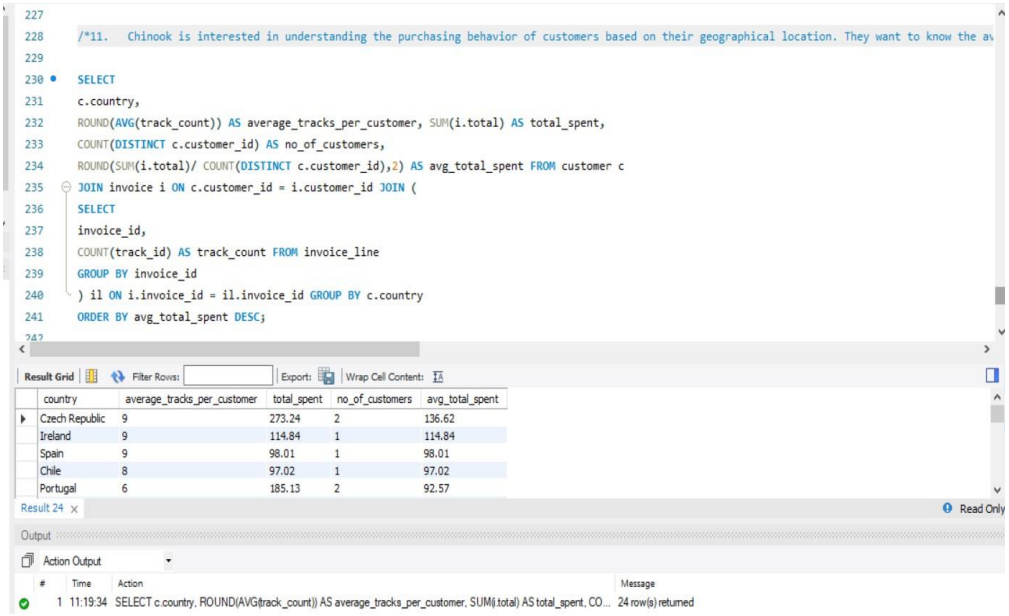
11. Chinook is interested in understanding the purchasing behavior of customers based on their geographical location. They want to know the average total amount spent by customers from each country, along with the number of customers and the average number of tracks purchased per customer. Write an SQL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

APPROACH :-

1. **Calculate Total Spending per Country**
   - **Sum up all purchases made by customers in each country.**
2. **Count the Number of Customers in Each Country**
   - **Determine how many unique customers exist per country.**
3. **Find the Average Number of Tracks Purchased per Customer**
   - **Count how many tracks are purchased per invoice.**
   - **Average this value for each country.**
4. **Group Data by Country**
   - **Aggregate the data to get a clear view of spending behavior.**
5. **Sort the Results by Average Total Spending**
   - **Identify the highest and lowest spending countries.**

Query:

```
SELECT
        c.country,
        ROUND(AVG(track_count)) AS average_tracks_per_customer,
        SUM(i.total) AS total_spent,
        COUNT(DISTINCT c.customer_id) AS no_of_customers,
        ROUND(SUM(i.total)/ COUNT(DISTINCT c.customer_id),2) AS avg_total_spent FROM customer c
JOIN invoice i ON c.customer_id = i.customer_id
JOIN (
        SELECT
                invoice_id,
                COUNT(track_id) AS
        track_count FROM invoice_line
        GROUP BY invoice_id
) il ON i.invoice_id = il.invoice_id
GROUP BY c.country
ORDER BY avg_total_spent DESC;
```



```
227
228  /*11.  Chinook is interested in understanding the purchasing behavior of customers based on their geographical location. They want to know the av
229
230  SELECT
231  c.country,
232  ROUND(AVG(track_count)) AS average_tracks_per_customer, SUM(i.total) AS total_spent,
233  COUNT(DISTINCT c.customer_id) AS no_of_customers,
234  ROUND(SUM(i.total)/ COUNT(DISTINCT c.customer_id),2) AS avg_total_spent FROM customer c
235  JOIN invoice i ON c.customer_id = i.customer_id JOIN (
236  SELECT
237  invoice_id,
238  COUNT(track_id) AS track_count FROM invoice_line
239  GROUP BY invoice_id
240  ) il ON i.invoice_id = il.invoice_id GROUP BY c.country
241  ORDER BY avg_total_spent DESC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

country	average_tracks_per_customer	total_spent	no_of_customers	avg_total_spent
Czech Republic	9	273.24	2	136.62
Ireland	9	114.84	1	114.84
Spain	9	98.01	1	98.01
Chile	8	97.02	1	97.02
Portugal	6	185.13	2	92.57

Result 24 | Read Only

Output

Action Output

#	Time	Action	Message
1	11:19:34	SELECT c.country, ROUND(AVG(track_count)) AS average_tracks_per_customer, SUM(i.total) AS total_spent, CO...	24 row(s) returned

Results :24 Rows returned

Insights:

- Some countries have **higher spending per customer**, indicating strong market potential.
- Certain regions show **higher track purchases**, meaning customers are more engaged.
- Low-spending regions might require **targeted marketing strategies**.

Recommendations:

1. **Focus on High-Spending Countries**
    - Offer loyalty rewards for frequent buyers.
    - Provide premium subscription options.
2. **Boost Engagement in Low-Spending Regions**
    - Introduce localized marketing strategies.
    - Offer region-specific discounts.
3. **Encourage More Track Purchases Per Customer**
    - Bundle tracks together at a discount.
    - Provide recommendations based on previous purchases.