Objective Questions

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

Solution:

To replace NULL values with specific values, there are 2 ways.

Firstly, we make use of the COALESCE() function used to handle null values. The syntax for COALESCE is

COALESCE(column_name, 'default_value')

we make use of UPDATE statement, which is used to update the table with a specific condition. The syntax is

UPDATE table_name
SET column1 = value1
WHERE condition;

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

1. Customer table - company,state,phone,fax
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'
Track	Composer	'Unknown'

S.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(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.

Solution:

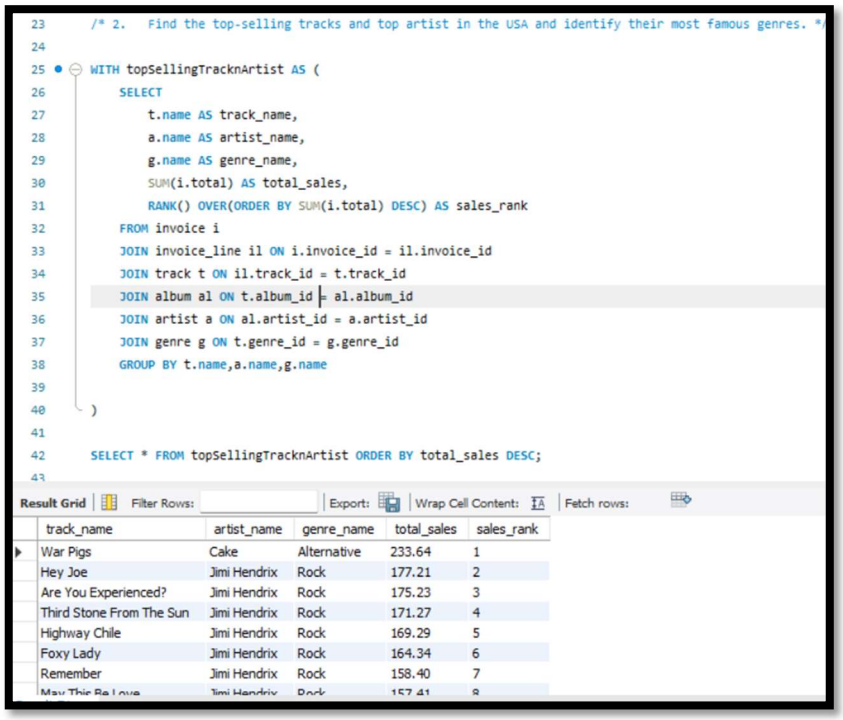
- **Concepts Used:** CTE, Aggregate Functions, GROUP BY, Joins, Sorting (ORDER BY)
- **Tables used:** invoice, invoice_line, track, album, artist, genre

- **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

---

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

Solution:

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

- **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**)**

```
/* 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

---

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

Solution:

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

- **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**)**

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

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 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	Brasília	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

---

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

Solution:

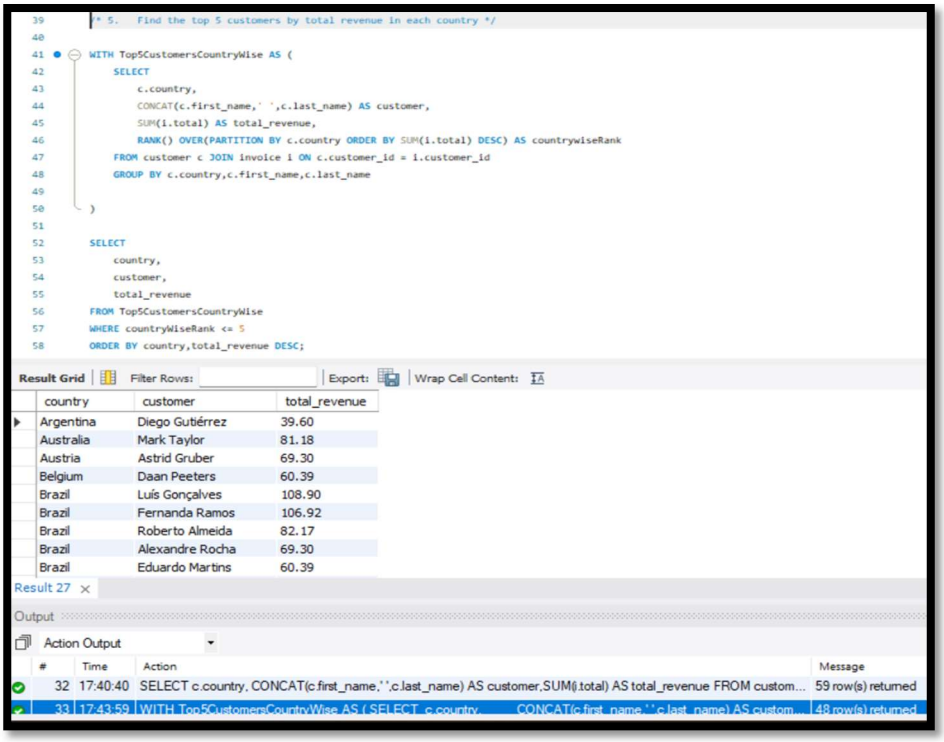
- **Concepts used:** CTE, Joins, GROUP BY, Aggregate Functions, Sorting (ORDER BY)
- **Tables used:** customer, invoice

- **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**)**



---

6. Identify the top-selling track for each customer

Solution:

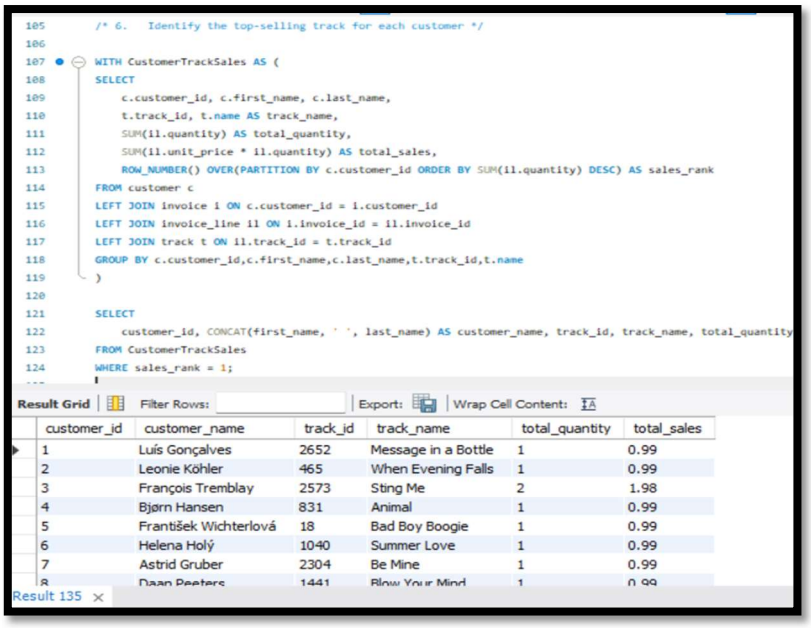
- **Concepts used:** CTE, Joins, GROUP BY, Aggregate Functions (SUM)
- **Tables used:** customer, invoice, invoice_line, track

- **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)



---

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

Solution:

7.1 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_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 1,2,3
)

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

- **Result: (59 rows** returned)

```
/* 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

7.2 Average Order Value:

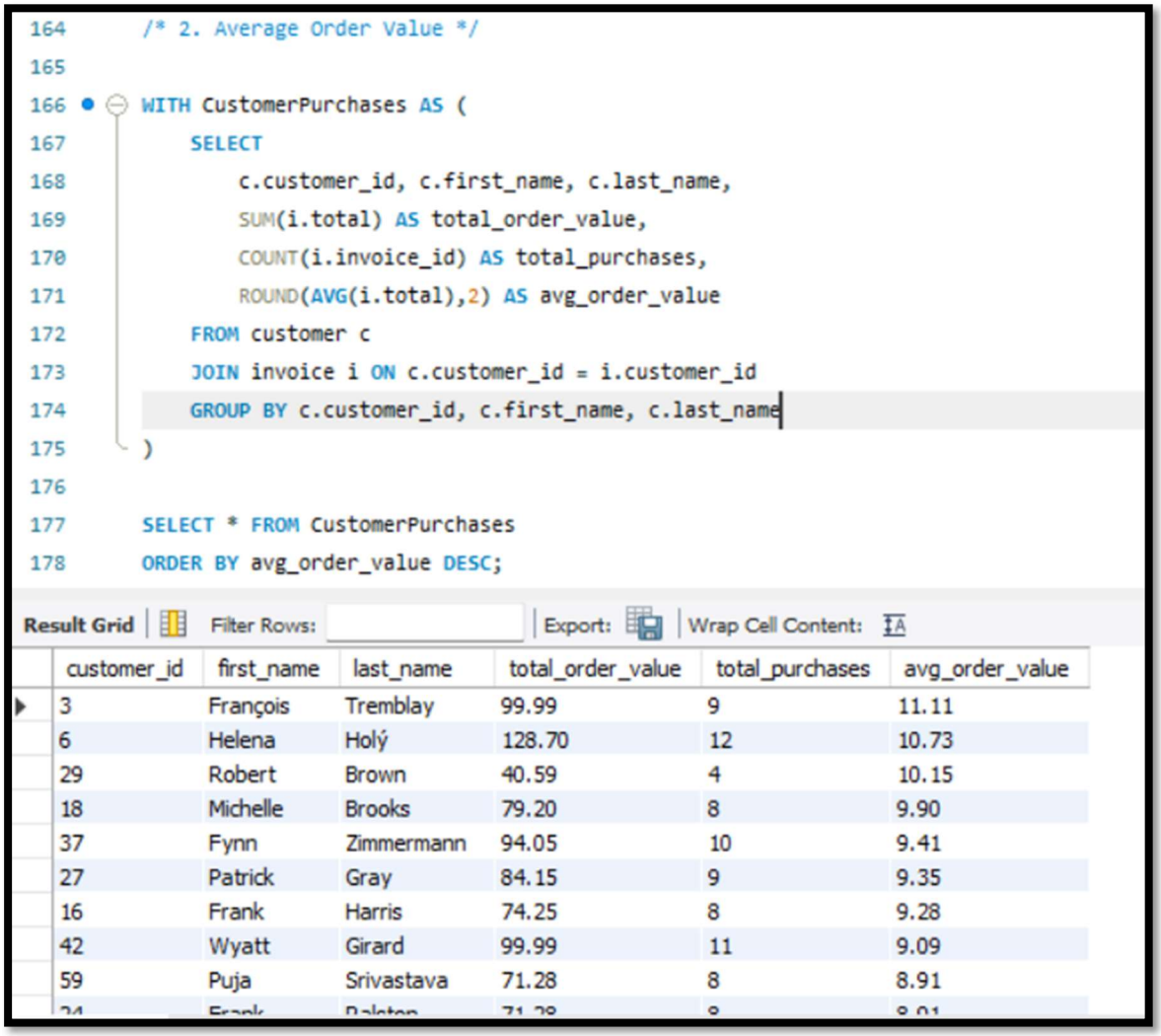
- **Concepts Used:** CTE, Aggregate Functions, GROUP BY, Sorting (ORDER BY)
- **Tables used:** customer, invoice

- **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)



```
/* 2. Average Order Value */

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 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

---

8. What is the customer churn rate?

Solution:

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

- **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;
```

- **Result:**

```sql
/* 8.   What is the customer churn rate? */
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;
```

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

churned_customers	total_customers	churn_rate
17	59	28.81

---

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

Solution:

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

- **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;
```

- **Result:**

```sql
/* 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

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:

```sql
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 LIMIT 1;
```

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

---

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

Solution:

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

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

SELECT
    c.customer_id,
    CONCAT(c.first_name,' ',c.last_name) AS customer,
    COUNT(DISTINCT t.genre_id) AS genre_count,
    COUNT(t.track_id) AS track_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 COUNT(DISTINCT g.genre_id) >=3
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
46	Hugh O'Reilly	12	116

---

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

Solution:

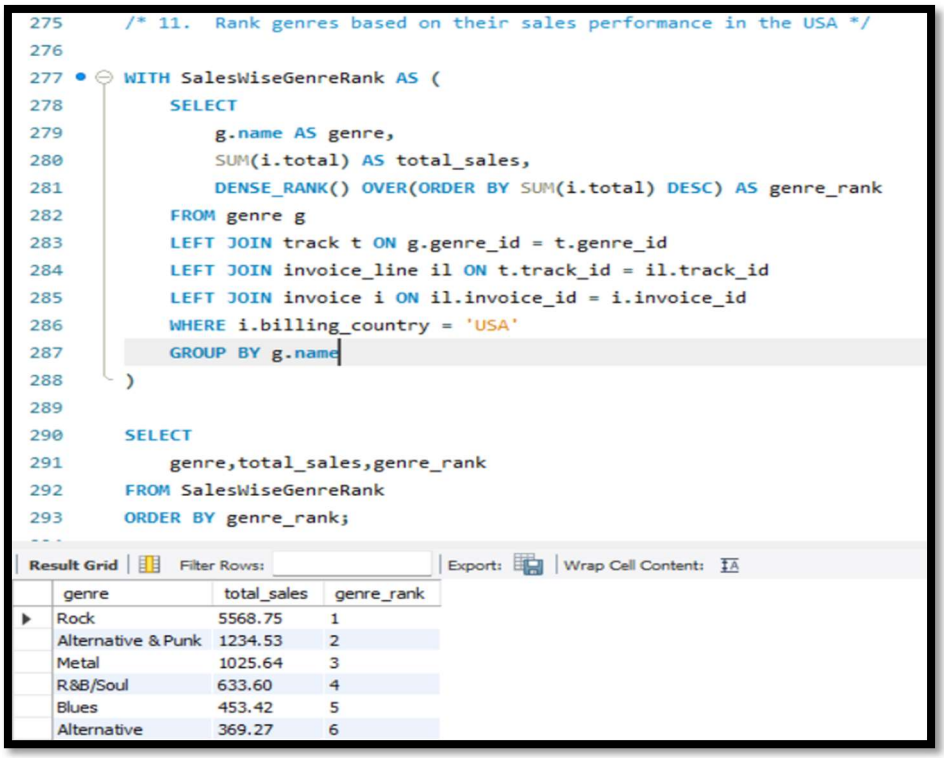
- **Concepts used:** CTE, Joins, GROUP BY, Window Functions (DENSE_RANK)
- **Tables used:** genre, track, invoice_line, invoice

- **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**)**



```
/* 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

---

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

Solution:

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

- **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

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.

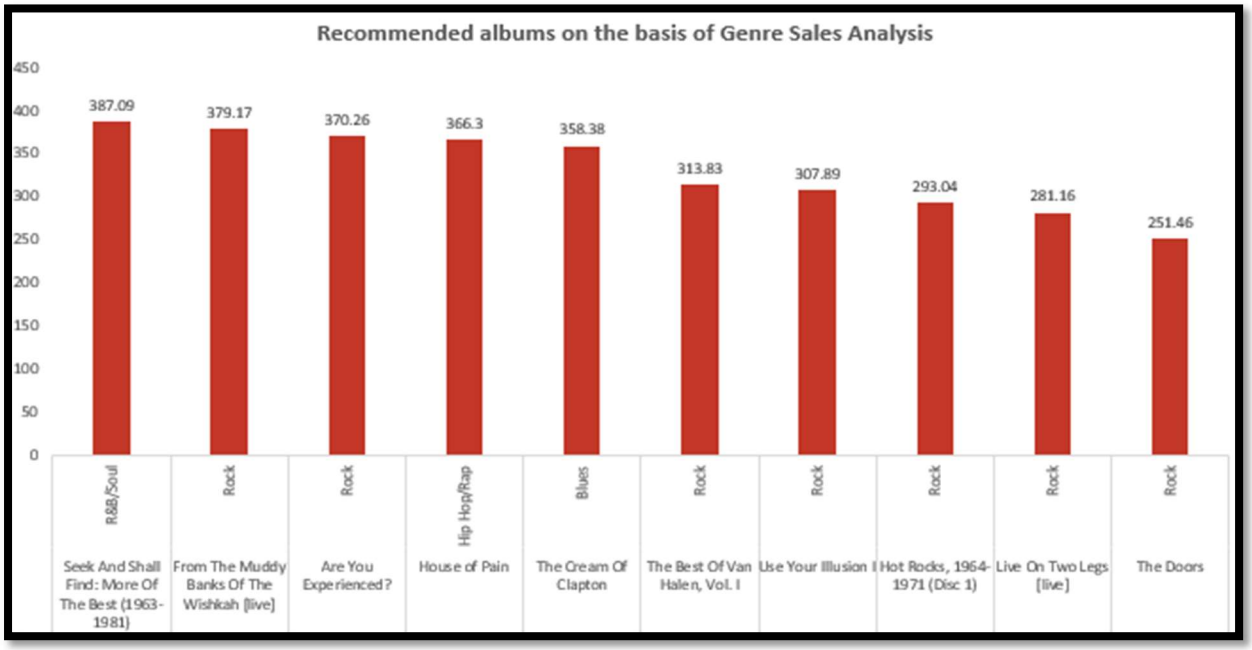
   Solution:

   Based on the Genre Sales Analysis, the following 3 albums can be prioritised for advertising and promotion in USA:

   Album 1: Genre - **Rock** (Top-selling genre)
   - Artist: Leading artist with multiple high-performing tracks in the genre.
   - Reason: High popularity in the USA, strong artist following.

   Album 2: Genre – **R&B/Soul** (Second best-selling genre)
   - Artist: Top-performing artist with tracks that show cross-genre appeal.
   - Reason: Trending genre with consistent sales growth.

   Album 3: Genre – **Hip Hop/Rap** (Third best-selling genre)
   - Artist: Rising artist with one of the top 10 highest-grossing tracks.
   - Reason: Reaches a distinct demographic with high engagement potential.

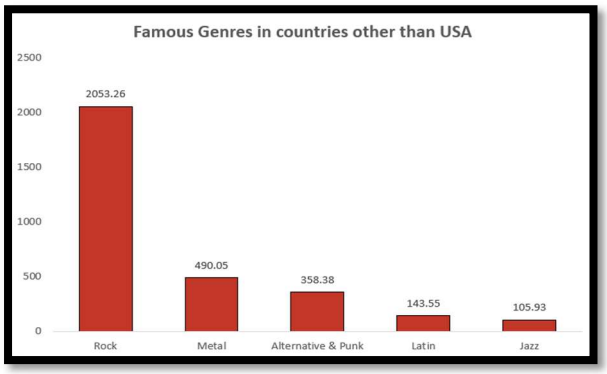


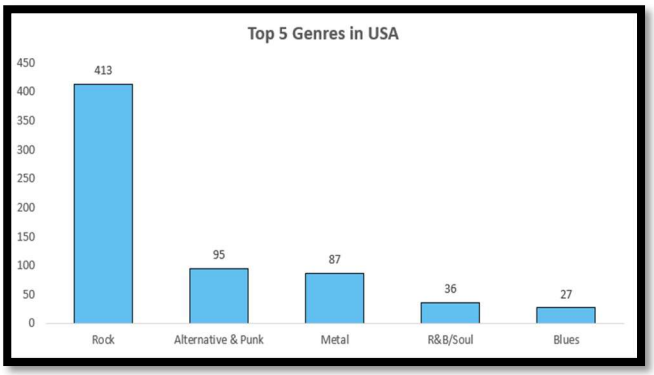
---

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

Solution:

- We can observe some similarities in terms of famous genres in countries other than USA vs USA.
- In both the cases, we can observe that Rock is the most famous Genre. The 2nd and 3rd places are interchanged in case of countries other than USA. Metal is the second popular genre in countries other than USA, followed by Alternative & Punk.



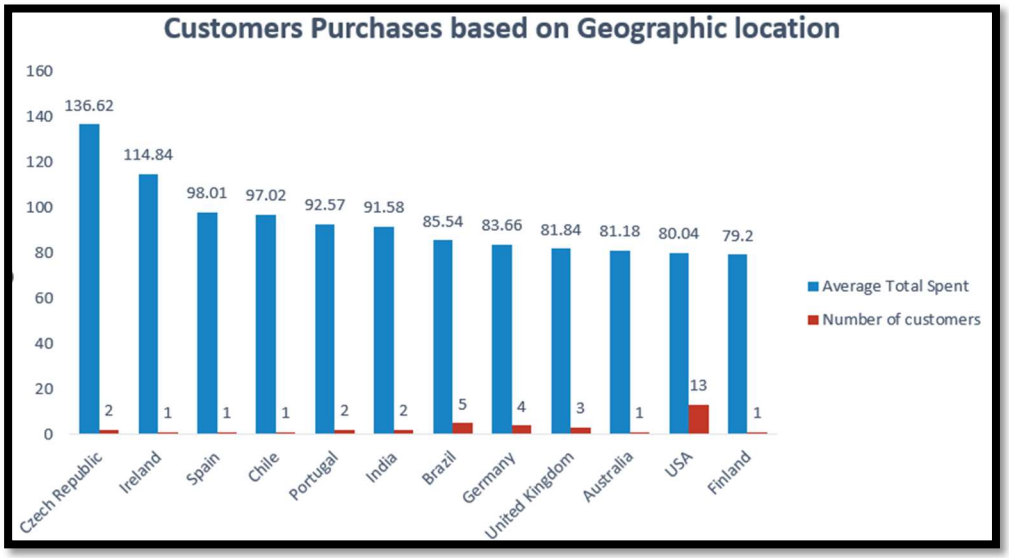


---

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?

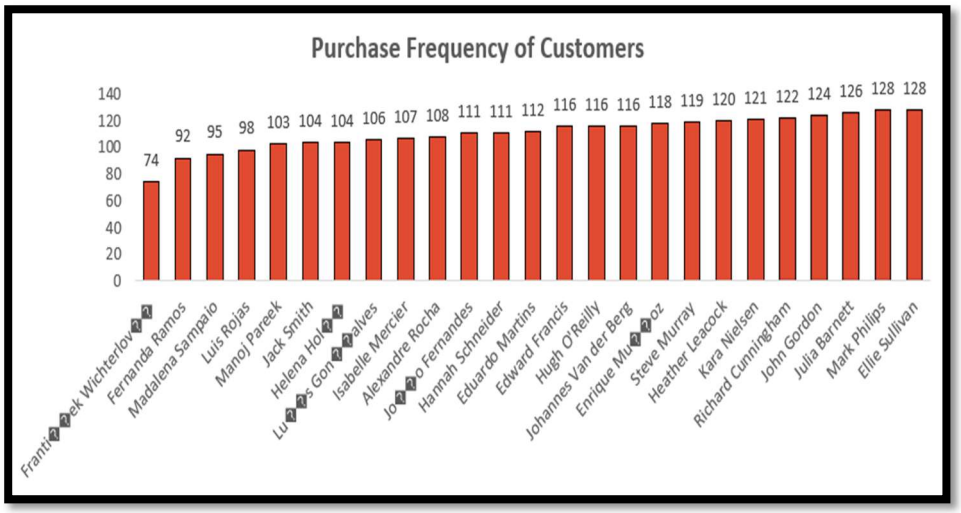
Solution:

- In most cases, frequent customers make more purchases, particularly if they are very brand loyal. This might be demonstrated by comparing the average frequency of purchases over a given time period. Regular purchases might provide insights for loyalty programs or special deals that promote ongoing participation.



- Long-term consumers may have greater basket sizes, which suggests that they have more trust in the brand and a willingness to explore more products. Finding product combinations that are often purchased might help inform tailored cross-selling and upselling suggestions.

- For new customers, offering introductory discounts could encourage initial spending, while targeted promotions for long-term customers could maintain or increase their average spend.



---

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?

Solution:

Common Genres:

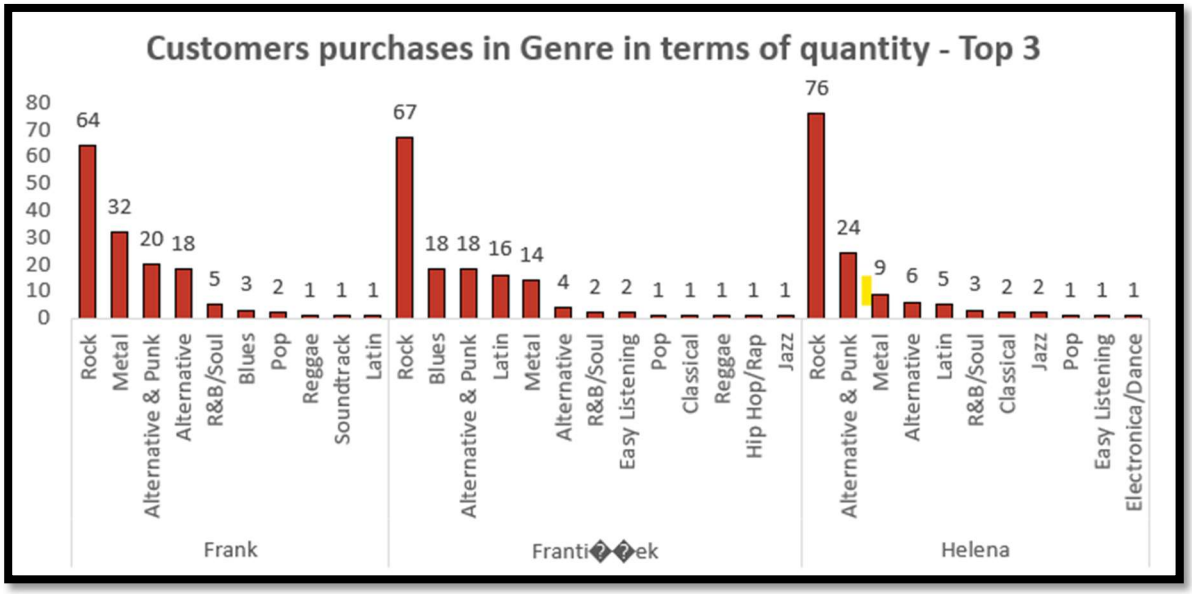
- All three considered customers favor Rock and Metal genres, indicating a shared preference for these types of music.

Product Recommendations:

- Suggesting related genres (such as Blues or Alternative) or introducing musicians inside Rock and Metal may work well for clients like Frank who buy a lot of this genre.

Cross-Selling Initiatives:

- By using their preexisting interests to promote wider musical discovery, expose clients who exhibit a high level of commitment to particular genres to related genres through carefully chosen recommendations.
- Since rock, metal, and alternative and punk music are always in style, make customized playlists for every client that feature their best songs.



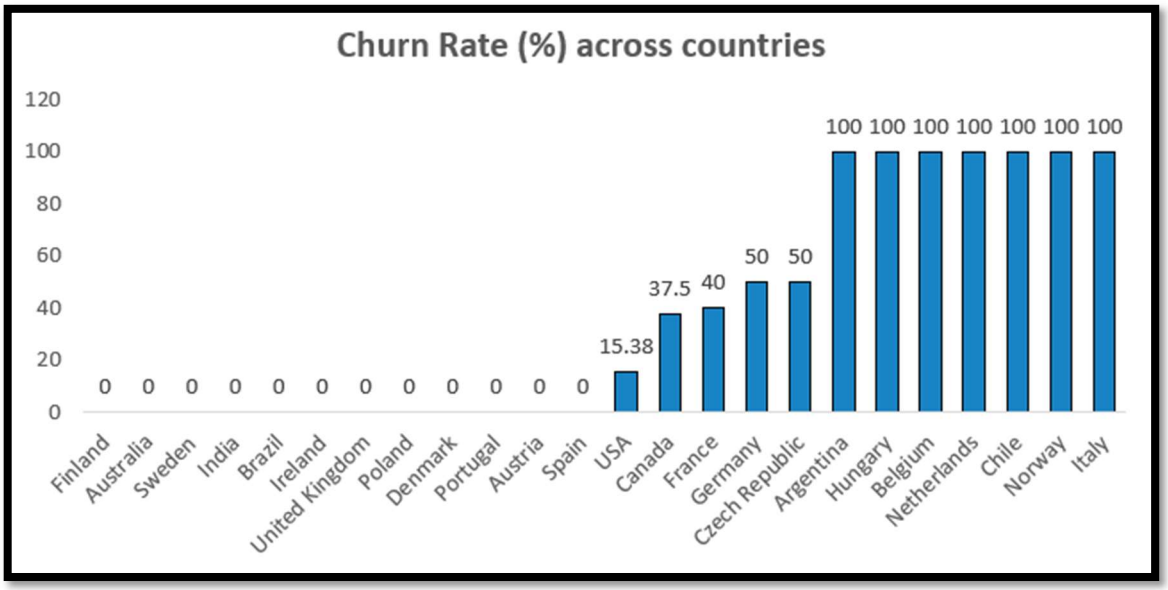
---

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?

Solution:

Based on the customer churn, the regional market analysis is performed which gives the following insights:

- A customer is considered to be churned if they have not made any purchases in the last 6 months.
- We can see that some countries like Finland, Australia, India, Spain etc. have 0 churn rate which indicates that the customers in these regions are active and make frequent purchases.



---

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?

Solution:

By calculating churn rates by country, we can determine if certain geographical segments show consistently high churn. These rates could indicate potential service, product, or market fit issues within specific locations.

A long history of infrequent purchases suggests a low-engagement customer who is more likely to churn. Additionally, high-volume spenders who show purchase frequency drops could signify declining interest, indicating an at-risk, high-value customer.

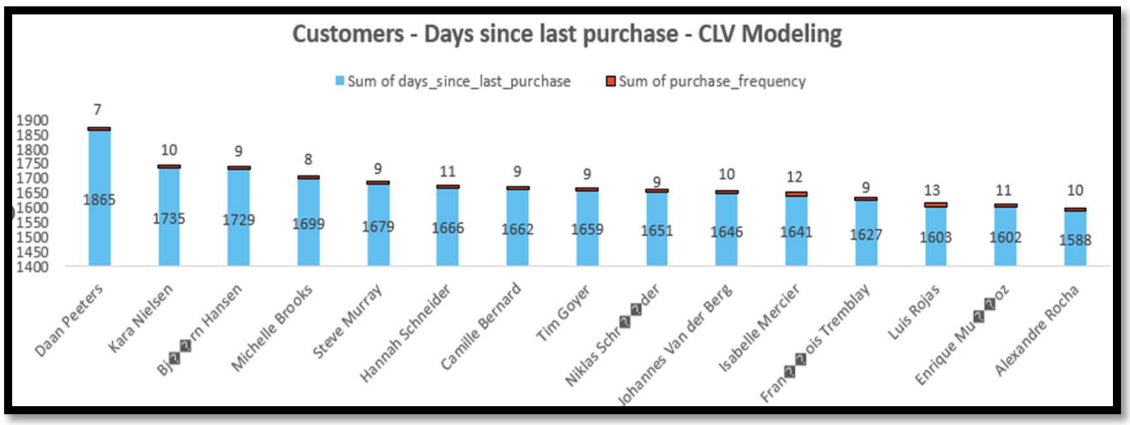
Customers with low purchase frequency or extended gaps between purchases (like the 180-day interval in this analysis) are at higher risk of churn. These customers may only respond to specific promotional periods or are less invested in the brand's products.

For high-churn countries, consider deploying localized marketing campaigns and enhancing service offerings to improve customer satisfaction.

---

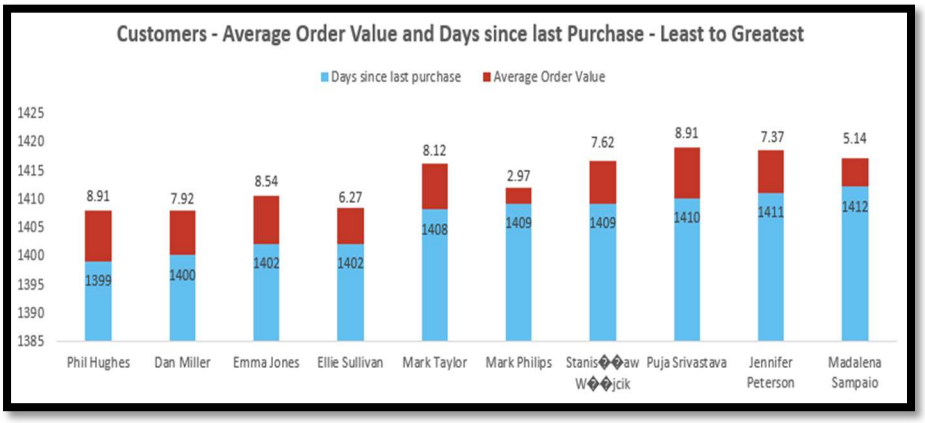
7. Customer Lifetime Value Modeling: How can you leverage customer data (tenure, purchase history, engagement) to predict the lifetime value of different customer segments? This could inform targeted marketing and loyalty program strategies. Can you observe any common characteristics or purchase patterns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

Solution:



- We can calculate the total spend and purchase frequency for each customer by summing up total from invoice and analyzing purchase dates.

  - Customers with a high value in days_since_last_purchase are at greater risk of being lost if they haven't made purchases recently.
  - Customers with long tenures but recent inactivity could be sent targeted re-engagement emails with discounts or exclusive offers.



  - Customers who buy infrequently but have a high AOV may be responsive to limited-time offers or exclusive items.
  - Identify customers with a high average order value and frequent purchases. These are prime targets for loyalty programs.

---

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?

Solution:

To measure the impact of promotional campaigns, compare sales and acquisition metrics before, during, and after each campaign.

1. For **acquisition**, analyze new customer counts linked to each campaign period.
2. To assess **retention**, track repeat purchases and churn rates among customers who engaged with the promotion.
3. **Sales impact** can be measured by comparing the total sales, average order value, and purchase frequency during campaigns versus baseline periods.

We can make use of customer segmentation to see which groups respond best, helping to refine targeting.

---

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

Solution:

If no specific questions were given, I would start by exploring the dataset broadly to uncover patterns and key insights relevant to customer behavior, sales performance, and promotional effectiveness. Here's how I would approach it:

1. **Understand Business Objectives**: First, I'd clarify business goals like increasing customer retention, boosting sales, or identifying high-value customer segments. This would help me target meaningful insights.

2. **Data Exploration and Cleaning**: I would perform data cleaning to handle any missing or inconsistent entries, then conduct exploratory analysis to understand the data's structure, distribution, and trends.

3. **Identify Key Metrics and Segments**: I'd establish KPIs such as customer acquisition rate, churn rate, lifetime value (LTV), average order value, and campaign ROI. Grouping by customer demographics, geographical location, and purchase behavior would reveal patterns within each segment.

4. **Perform Analysis and Modeling**:
   - **Churn Analysis**: I'd analyze factors that increase the likelihood of churn, such as recent purchase frequency or discount engagement.
   - **Campaign Effectiveness**: I would examine promotional data by comparing pre- and post-campaign metrics to assess changes in acquisition, retention, and sales.

5. **Interpret Results for Strategy Recommendations**: Based on findings, I'd make data-driven recommendations to improve customer retention, target high-value customers, and refine marketing strategies for optimal ROI.

---

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?

Solution:

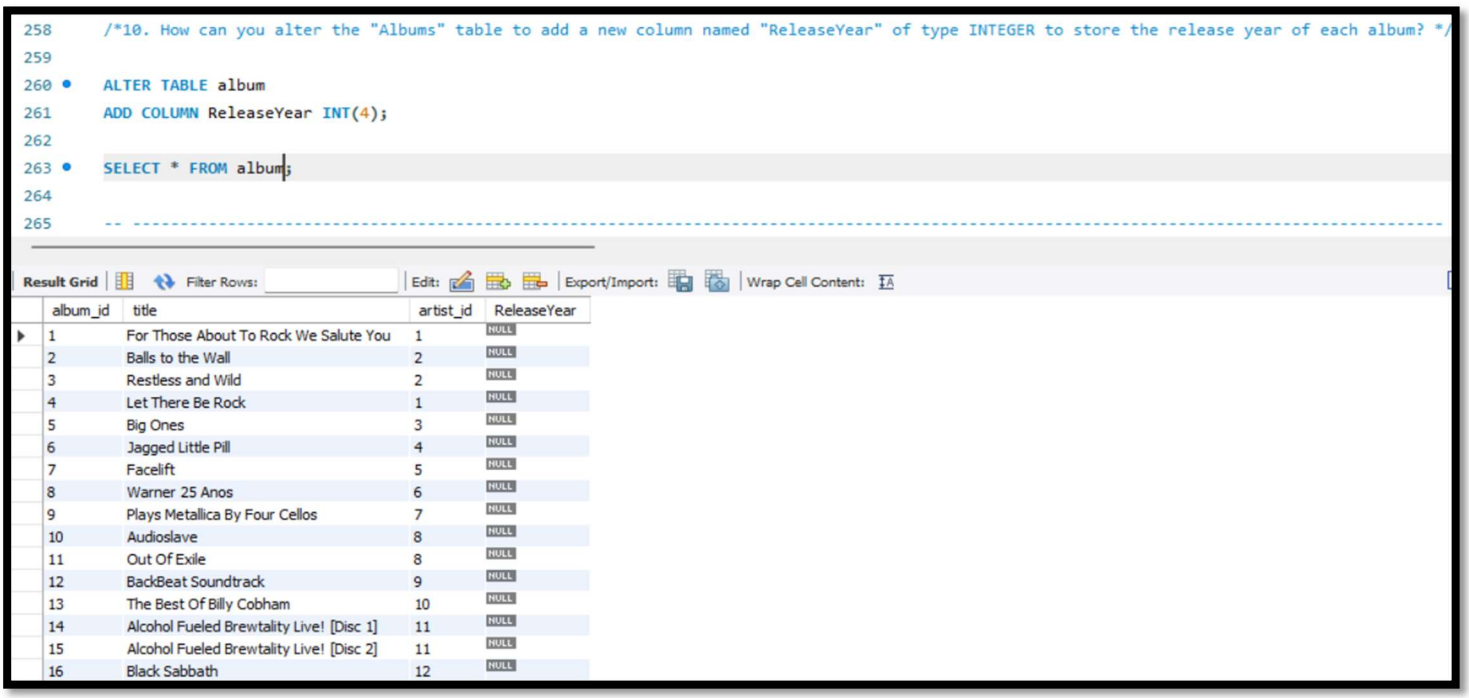
We can make use of the ALTER statement to add a new column to a table. The syntax is as follows:

ALTER TABLE table_name
ADD COLUMN column_name datatype;

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

ALTER TABLE album
ADD COLUMN ReleaseYear INT(4);

SELECT * FROM album;



```
/*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

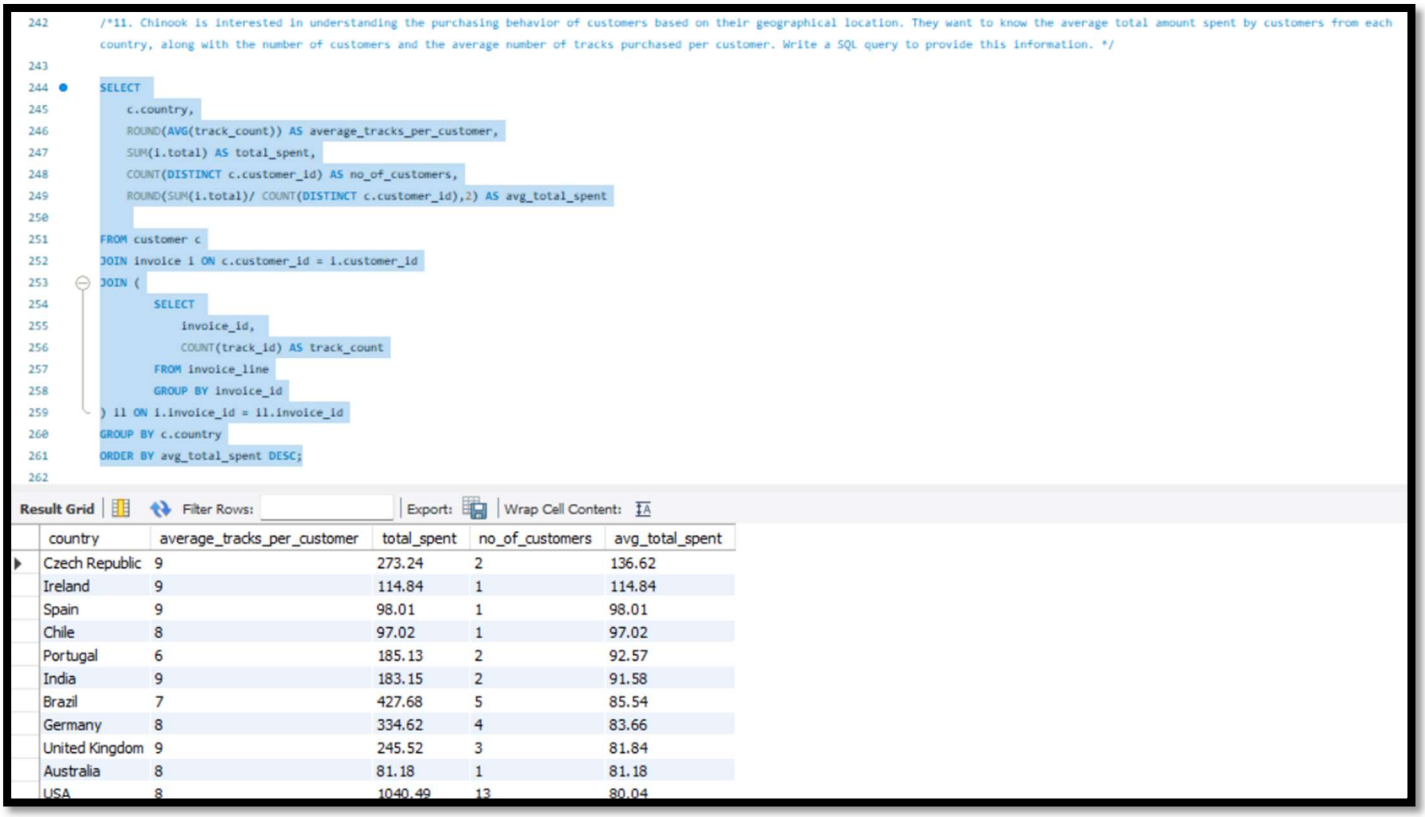
---

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.

Solution:

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;
```



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
India	9	183.15	2	91.58
Brazil	7	427.68	5	85.54
Germany	8	334.62	4	83.66
United Kingdom	9	245.52	3	81.84
Australia	8	81.18	1	81.18
USA	8	1040.49	13	80.04
