Project Title: Chinook Data Analytics - Unlocking Business Insights from a Digital Media Store

Project Overview:

This project involves an in-depth exploration and analysis of the **Chinook database**, a comprehensive sample dataset representing a digital media store. Leveraging SQL (Structured Query Language), the goal is to extract, transform, and analyze various facets of the store's operations, including sales performance, customer behaviour, employee efficiency, and media catalogue trends. This project will demonstrate proficiency in a wide range of SQL concepts, from basic data retrieval to complex analytical queries, ultimately providing actionable business insights.

Context:

As of July 2025, understanding customer purchasing habits, optimizing inventory, and evaluating sales strategies are more critical than ever for digital media businesses. This project simulates a real-world scenario where a database professional is tasked with providing data-driven recommendations to improve the Chinook store's profitability and operational efficiency.

Project Objectives:

The project will address key business questions and involve the following objectives:

Database Understanding & Schema Navigation:

- Thoroughly understand the Chinook database schema, including all tables (Artists, Albums, Tracks, Customers, Employees, Invoices, Invoice Lines, Genres, Media Types, Playlists, Playlist Track) and their relationships (Primary Key/Foreign Key constraints).
- Create an Entity-Relationship Diagram (ERD) or describe the key relationships within the database.

Core Data Retrieval & Manipulation:

- Practice basic SELECT statements with WHERE, ORDER BY, and LIMIT/TOP clauses.
- Demonstrate proficiency in INSERT, UPDATE, and DELETE operations for data management (e.g., adding a new track, updating customer details).

Advanced Querying & Data Aggregation:

- Joining Tables: Utilize various types of JOIN clauses (INNER JOIN, LEFT JOIN, RIGHT JOIN, FULL JOIN) to combine data from multiple tables to answer complex questions (e.g., "List all tracks by a specific artist and their genre").
- Aggregation and Grouping: Employ GROUP BY and aggregate functions (COUNT, SUM, AVG, MIN, MAX) with HAVING clauses to perform statistical analysis (e.g., "Calculate total sales per country," "Find the average invoice amount for each customer").
- Subqueries & Common Table Expressions (CTEs): Apply subqueries and/or
 CTEs for multi-step data processing and improved query readability (e.g.,
 "Find customers who have spent more than the average customer spending").
- Window Functions: Explore the use of window functions for more sophisticated analytical tasks (e.g., "Rank employees by sales performance within their respective departments").

Business Insights & Reporting:

Sales Analysis:

- Identify top-performing artists and genres by revenue.
- Analyze monthly/quarterly sales trends over the four-year period.
- Determine the average value of an invoice and the average number of items per invoice.

Customer Behaviour Analysis:

- Segment customers based on their total spending.
- Identify the top 10 most valuable customers.
- Determine the most popular genres or artists among specific customer groups.

Employee Performance:

- Evaluate sales performance of individual employees and their contribution to overall revenue.
- Analyze employee hierarchy and reporting structures.

Media Catalogue Analysis:

- Identify the distribution of tracks by genre and media type.
- Analyze the average duration of tracks per album or genre.

Technology Used:

• **Database:** Chinook Database

• Language: SQL (MySQL)

• Tools: MySQL Workbench

Deliverables:

• A documented SQL script file (.sql) containing all queries used to achieve the project objectives, clearly commented and organized by objective.

• A brief project report summarizing the key findings, insights derived from the SQL queries, and potential business recommendations.

Data Cleaning

Replace Null value in each table →

Customer Table --

```
-- Removing null value with "Not available"

set sql_safe_updates = 0;

UPDATE customer

SET

company = COALESCE(Company, 'Not Available'),
    State = COALESCE(State, 'Not Available'),
    Fax = COALESCE(Fax, 'Not Available'),
    Phone = COALESCE(Phone, 'Not Available'),
    PostalCode = COALESCE(PostalCode, 'N/A');
```

Invoice Table -

```
868 • UPDATE invoice
869 SET
870 BillingState = COALESCE(BillingState, 'Not Available'),
871 BillingPostalCode = COALESCE(BillingPostalCode, 'N/A');
872
```

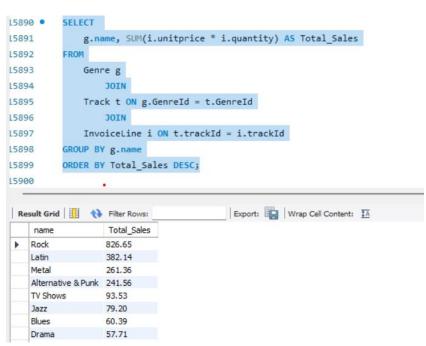
Table Track -

```
L5882 • select * from track;
L5883 • update track
L5884 set
L5885 Composer = coalesce(Composer,'Not Available');
L5886
```

Data Analysis

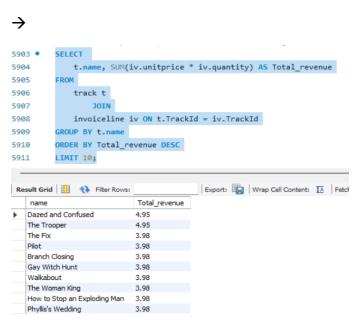
Q.1 Total Sales by Genre:





Q.2 Top 10 Selling Tracks:

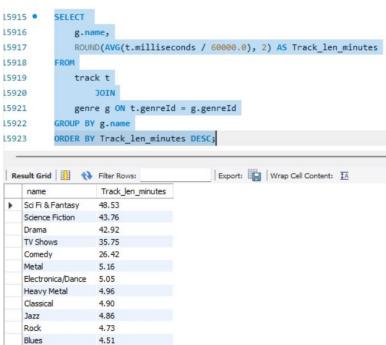
-- Identify the top 10 Track Names that have generated the highest total revenue. Include the track name and total revenue.



Q.3 Average Track Length by Genre:

-- For each Genre Name, calculate the average Milliseconds of its tracks. Convert milliseconds to minutes.





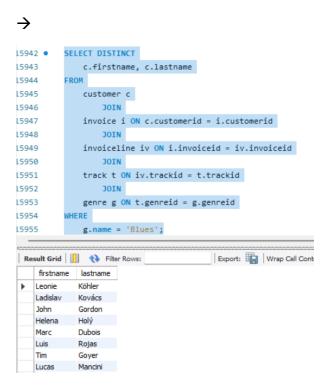
Q.4 Tracks Sold in a Specific Year:

-- Count the total number of Invoice Line items that occurred in the year 2022.

```
L5927 •
          SELECT
            YEAR(i.invoicedate) AS s year,
            COUNT(t.trackid) AS total_sold_track
15929
15930
L5931
            track t
15932
                JOIN
         invoiceline iv ON t.trackid = iv.trackid
15933
15934
15935
             invoice i ON iv.invoiceid = i.invoiceid
        GROUP BY s_year
15936
        HAVING s year = '2022'
L5937
15938
         ORDER BY total_sold_track DESC;
                                      Export: Wrap (
s_year total_sold_track
▶ 2022
          455
```

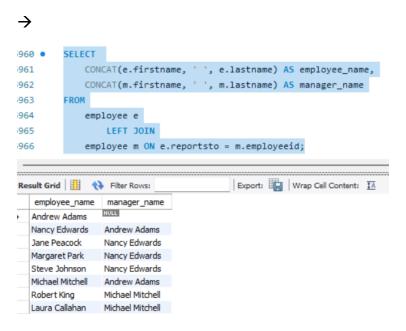
Q.5 Customers Who Purchased 'Blues' Genre:

-- List FirstName, LastName of Customers who have purchased at least one Track from the 'Blues' Genre. Do not list duplicates.



Q.6 Employees and Their Direct Manager:

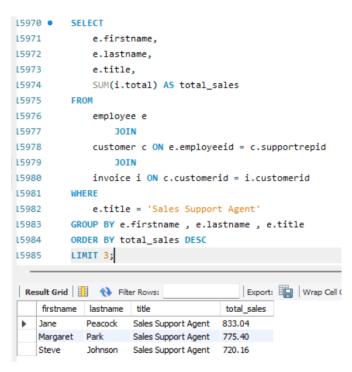
-- List Employee FirstName, LastName (as 'Employee Name') and their manager's FirstName, LastName (as 'Manager Name').



Q.7 Top 3 Support Representatives by Sales:

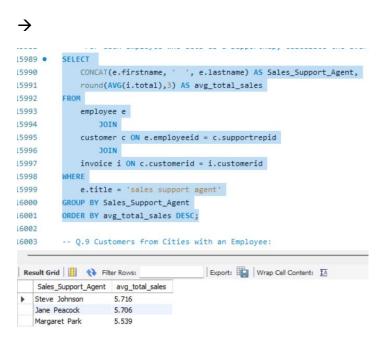
-- Identify the top 3 Employees (FirstName, LastName, Title) who are 'Sales Support Agent's and have generated the most total sales (Invoice, Total) from their supported customers.





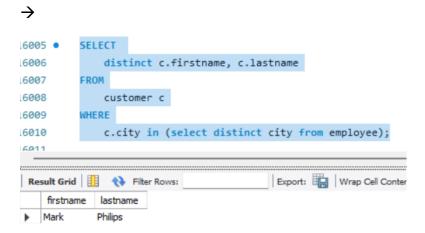
Q.8 Average Sales per Employee (as Support Rep):

-- For each Employee who acts as a SupportRep, calculate the average Total of the Invoices associated with their customers



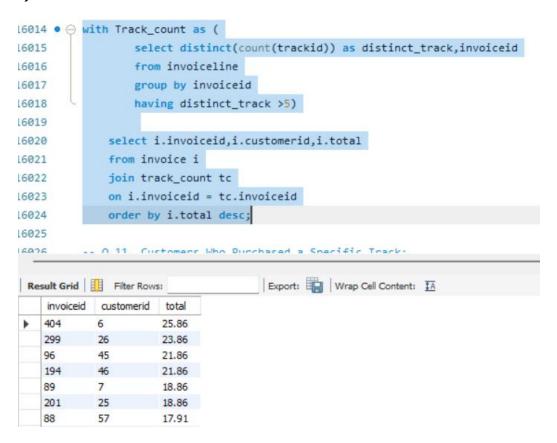
Q.9 Customers from Cities with an Employee:

-- Find the FirstName, LastName of Customers who live in the same City as at least one Employee. Do not list duplicates.



Q.10 Invoices with More Than 5 Tracks:

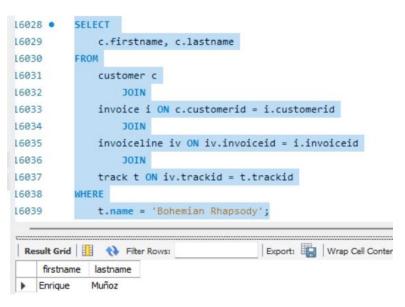
-- List InvoiceId, CustomerId, and Total for invoices that contain more than 5 distinct Tracks.



Q.11 Customers Who Purchased a Specific Track:

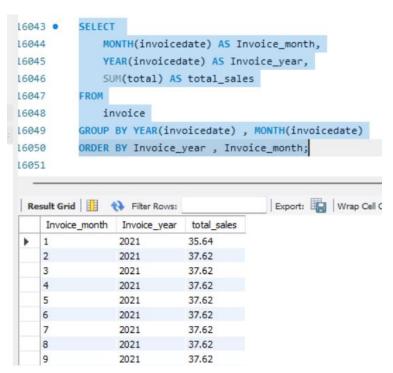
-- List FirstName, LastName of Customers who purchased the track 'Bohemian Rhapsody'.





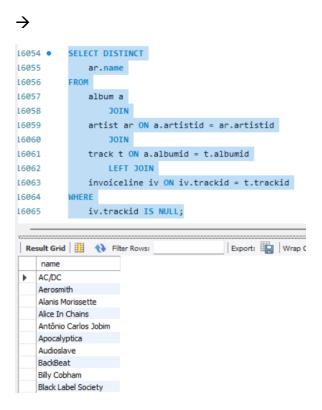
Q.12 Monthly Sales Trend:

-- Calculate the total sales (Invoice.Total) for each month of each year. Display InvoiceYear, InvoiceMonth, and TotalSales, ordered chronologically.



Q.13 Artists Whose Tracks Have Never Been Sold:

-- List Artist Names whose Albums have Tracks that have never appeared in any Invoice Line.



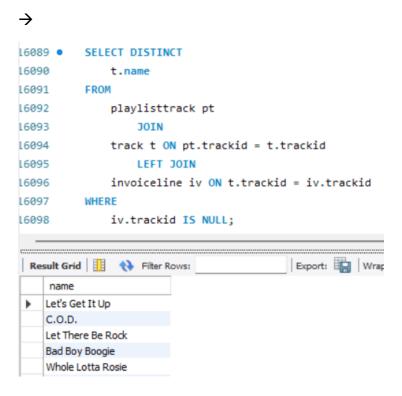
Q.14. Customers Who Purchased Every Genre:

-- Find the FirstName and LastName of Customers who have purchased at least one Track from every single Genre available in the store.

```
\rightarrow
16069 ullet \bigcirc with customer_genre as (
               select c.customerid,c.firstname,c.lastname,
               count( distinct t.genreid) as genre_count
16071
16072
               from customer c
16073
                join invoice i
                on c.customerid = i.customerid
16075
               join invoiceline iv
16076
                on i.invoiceid = iv.invoiceid
16077
               join track t
                on iv.trackid = t.trackid
16079
               group by c.customerid.c.firstname.c.lastname).
16080 \ominus total_genre as (
                   select count(distinct genreid) as total_genre_count
16082
                   from genre)
16083
               select cg.firstname,cg.lastname
16984
                from customer_genre cg,total_genre tg
16085
                where cg.genre_count = tg.total_genre_count;
 Result Grid Filter Rows:
                                                 Export: Wrap Cell Content: IA
     firstname lastname
```

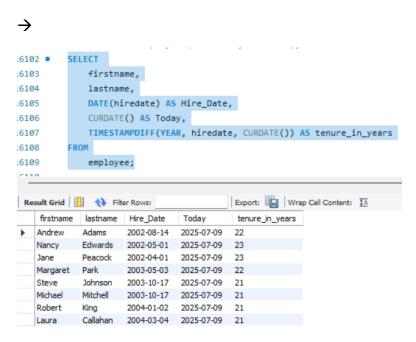
Q.15 Tracks Available in Playlists but Not Sold:

-- Find Track Names that exist in at least one Playlist but have never been sold in an Invoice Line.



Q.16 Employee Tenure (in years):

-- For each Employee (FirstName, LastName), calculate their tenure in years from their Hire Date to the current date.



Q.17 Customers Who Are Managers:

-- Identify if any Customer (FirstName, LastName) is also an Employee and holds a Title like 'Manager' or 'General Manager'.



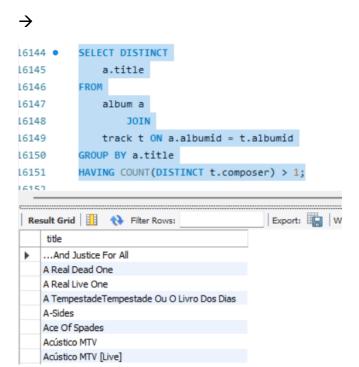
Q.18 Customer Acquisition by Support Rep:

-- For each SupportRep (FirstName, LastName), count the number of new Customers they acquired in each year based on Customer.SupportRepId and Customer.

```
\rightarrow
6128 • SELECT
6129
            e.firstname,
6130
             e.lastname,
6131
            YEAR(e.hiredate) AS aquire_year,
            COUNT(DISTINCT c.customerid) AS new_customer
6132
6133
       FROM
6134
           customer c
6135
                left JOIN
6136
            employee e ON c.supportrepid = e.employeeid
6137
        WHERE
             e.hiredate IS NOT NULL
6138
6139
         GROUP BY e.firstname , e.lastname , aquire_year
6140
         ORDER BY e.firstname , e.lastname , aquire_year;
Export: Wrap Cell Content: 1
   firstname
            lastname aquire_year new_customer
                    2002
   Jane
            Peacock
   Margaret Park
                    2003
                              20
   Steve
           Johnson
                   2003
                              18
```

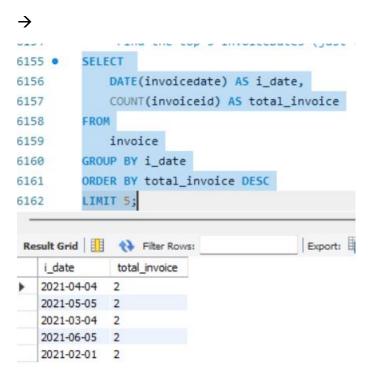
Q.19 Albums with Tracks from Multiple Composers:

-- List Album Titles that contain tracks from more than one unique Composer.



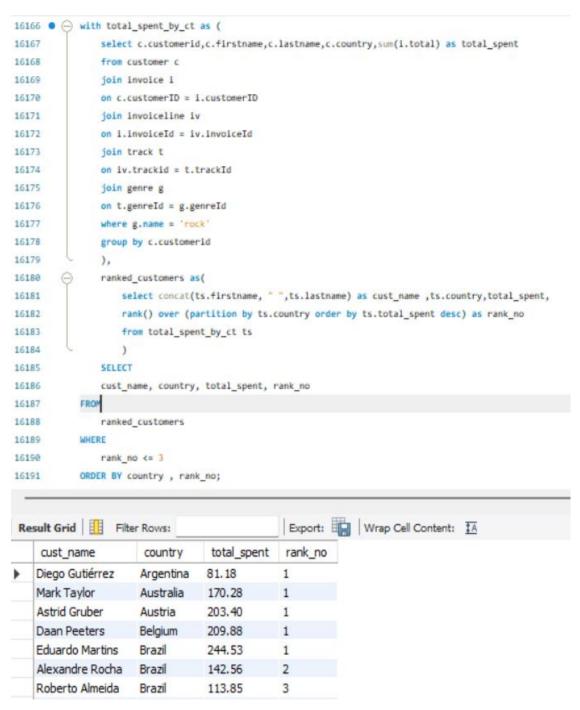
Q.20 Top 5 Busiest Invoice Dates:

-- Find the top 5 Invoice Dates (just the date part) that had the highest number of Invoices.



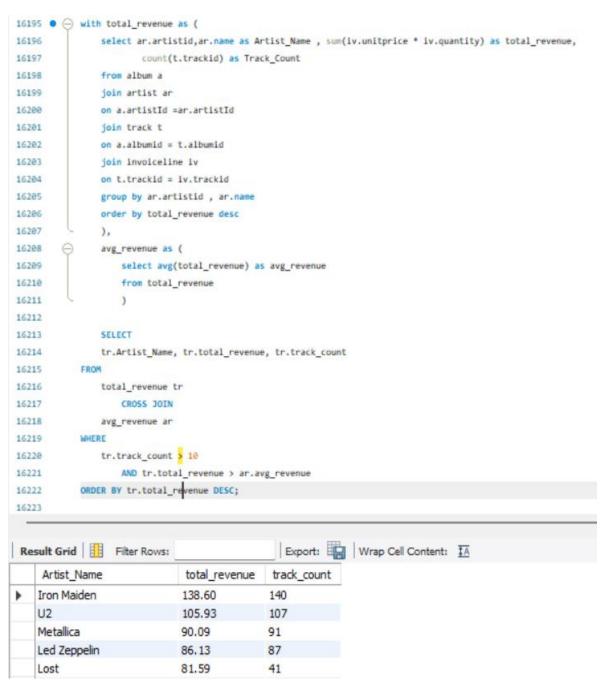
Q.21 Top Spenders by Genre:

-- Find the top 3 customers in each country who have spent the most money on "Rock" music. For each customer, list their full name, country, and total amount spent on Rock music.



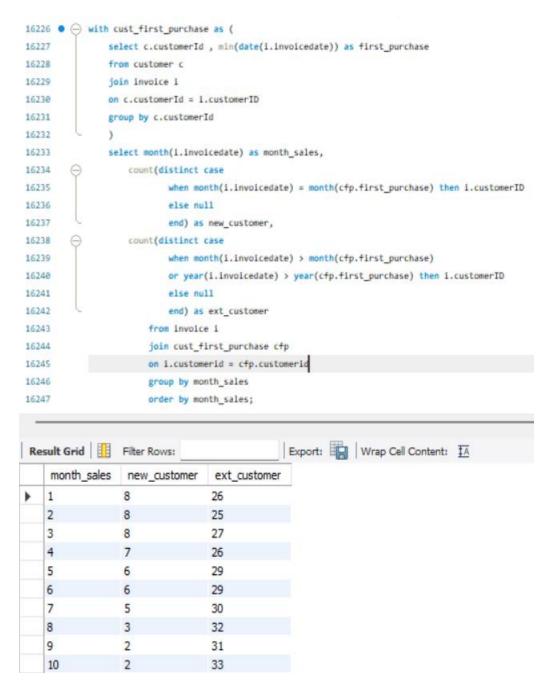
Q.22 Artist Popularity by Track Count and Revenue:

-- Identify artists who have more than 10 tracks and whose total revenue generated from sales is above the average revenue generated by all artists. Display the artist's name, the number of tracks, and their total revenue, ordered by revenue in descending order.



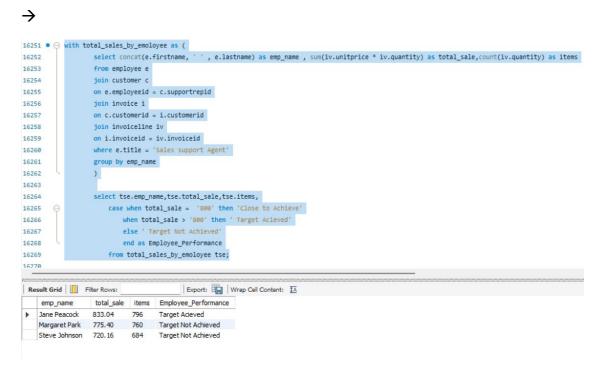
Q.23 Customer Retention Analysis (Monthly):

-- For each month, calculate the number of new customers acquired and the number of existing customers who made a purchase in that month.



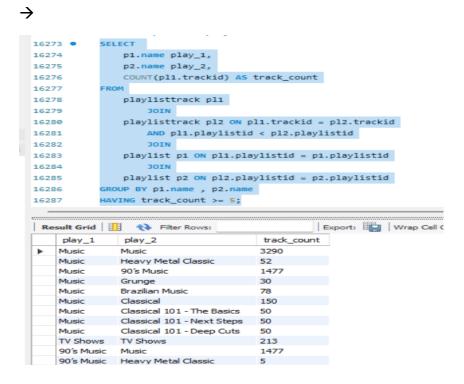
Q.24 Employee Performance vs. Sales Targets:

-- Assume a sales target of \$800 for each invoice line item. For each sales support agent, calculate their total sales, the number of invoice line items they processed, and the percentage by which they exceeded or fell short of their hypothetical sales target.



Q.25 Playlist Content Overlap:

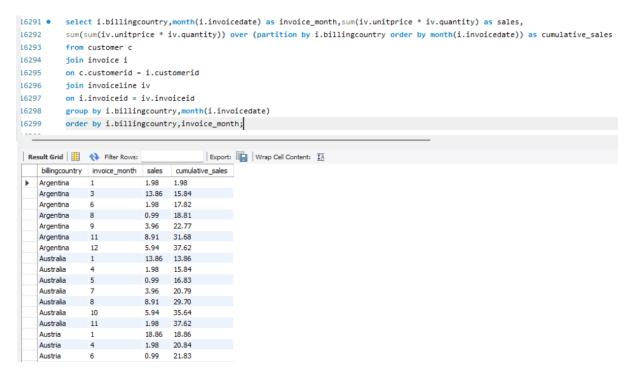
-- Find pairs of playlists that share at least 5 common tracks. For each pair, display the names of both playlists and the count of shared tracks.



Q.26 Cumulative Sales by Country and Month:

-- Calculate the cumulative total sales for each country, month by month.





Q.27 Median Track Length by Genre:

-- For each genre, find the median track length (in milliseconds).

```
select g.name as Genrename , t.milliseconds as Trk_lngth,
16304
16305
              row_number() over (partition by g.name order by t.milliseconds) as row_n,
16306
              count(t.trackId) over (partition by g.name) as total_track
16307
              from track t
16308
              join genre g
16309
               on t.genreid = g.genreid
16310
16311
              SELECT
16312
              Genrename, round(AVG(Trk_lngth),3) AS medianTrk
16313
         FROM
16314
              track_rank
16315
          WHERE
16316
              row_n IN ((total_track + 1) / 2 , (total_track + 2) / 2)
16317
          GROUP BY Genrename
          ORDER BY Genrename:
16318
Result Grid Filter Rows:
                                           Export: Wrap Cell Content: IA
     Genrename medianTrk
    Alternative
                     240255.000
    Alternative & Punk 230974.000
    Blues
                     251219.000
    Bossa Nova 204956.000
    Classical
                     274504,000
    Comedy 1302093.000
                     2610250.000
    Drama
    Easy Listening 188499.000
    Electronica/Dance 304143.000
Heavy Metal 300956.000
    Hip Hop/Rap
                     185103.000
```

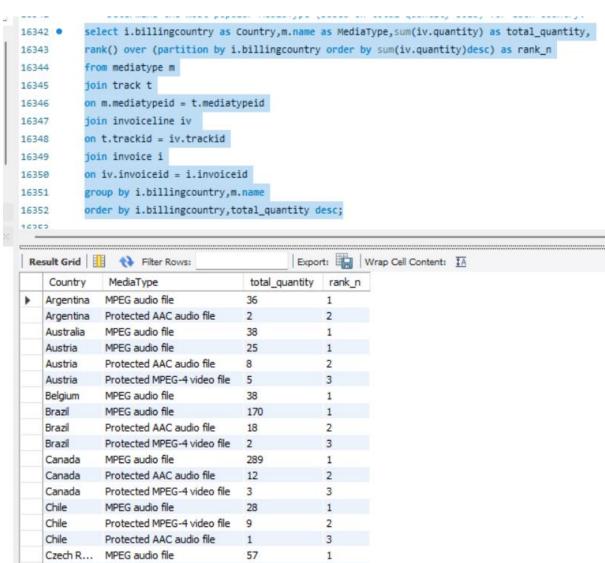
Q.28 Customers Who Haven't Purchased a Specific Genre:

-- Find all customers (full name and email) who have never purchased a "Classical" music track.

```
\rightarrow
            SELECT
16322 •
                CONCAT(c.firstname, ' ', c.lastname) AS cust_name, c.email
16323
            FROM
16324
                customer c
16325
                    LEFT JOIN
16326
16327
                (SELECT DISTINCT
16328
                    i.customerid
                FROM
16329
                    invoice i
16330
                JOIN invoiceline iv ON i.invoiceid = iv.invoiceid
16331
                JOIN track t ON iv.trackid = t.trackid
16332
                JOIN genre g ON t.genreid = g.genreid
16333
                WHERE
16334
                    g.name = 'Classical') AS classic_cust
16335
                    ON c.customerid = classic_cust.customerid
16336
16337
            WHERE
                classic_cust.customerid IS NULL;
16338
                                                   Export: Wrap Cell Content: I
 Result Grid
                Filter Rows:
     cust_name
                          email
                          leonekohler@surfeu.de
    Leonie Köhler
    František Wichterlová frantisekw@jetbrains.com
    Helena Holý
                          hholy@gmail.com
    Daan Peeters
                          daan_peeters@apple.be
    Kara Nielsen
                          kara.nielsen@jubii.dk
    Eduardo Martins
                          eduardo@woodstock.com.br
     Alexandre Rocha
                          alero@uol.com.br
    Roberto Almeida
                          roberto.almeida@riotur.gov.br
    Mark Philips
                          mphilips 12@shaw.ca
    Jennifer Peterson
                          jenniferp@rogers.ca
```

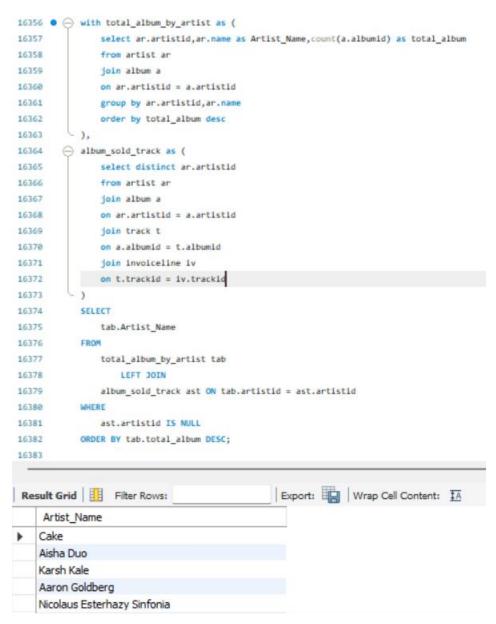
Q.29 Most Popular Media Type per Country:

-- Determine the most popular MediaType (based on total quantity sold) for each country.



Q.30 Artist with Most Albums Without Any Tracks Sold:

-- Find the artist(s) who have the most albums in the database, but none of their tracks have ever been sold.



Q.31 Churn Prediction - Customers at Risk:

Jennifer Peterson

2024-12-15

207

-- Identify customers who made their first purchase more than 6 months ago, but haven't made any purchases in the last 3 months. For these customers, list their full name, their last purchase date, and the total number of days since their last purchase.

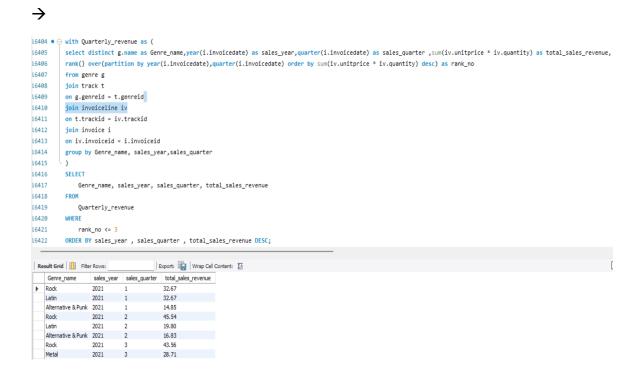
```
\rightarrow
16387
          SELECT
              CONCAT(c.firstname, ' ', c.lastname) AS cust_name,
16388
16389
              DATE(MAX(i.invoicedate)) AS last_purchase,
16390
              DATEDIFF(CURRENT_DATE(), DATE(MAX(i.invoicedate))) AS day_since_lp
          FROM
16391
16392
              customer c
                   JOIN
16393
              invoice i ON c.customerID = i.customerid
16394
16395
          GROUP BY cust_name

→ HAVING date(min(i.invoicedate)) 

✓ DATE_SUB(CURRENT_DATE(),
16396
              INTERVAL 6 MONTH)
16397
16398
              AND MONTH(last_purchase) < DATE_SUB(CURRENT_DATE(),
              INTERVAL 3 MONTH)
16399
          ORDER BY day_since_lp DESC;
L6400
 Export: Wrap Cell Content: IA
    cust_name
                                  day_since_lp
                     last_purchase
    Puja Srivastava
                     2024-05-30
                                 406
    Niklas Schröder
                     2024-06-30 375
    Leonie Köhler
                     2024-07-13
                                 362
                    2024-07-31
    Jack Smith
                                 344
    Dominique Lefebvre 2024-08-13
                                 331
    Mark Taylor
                  2024-08-31
                                 313
    Tim Goyer
                     2024-09-13
                                 300
    João Fernandes 2024-10-01
                                 282
    Luis Rojas
                     2024-10-14
                                 269
    Fernanda Ramos
                    2024-11-01
                                 251
    Hannah Schneider
                     2024-11-14
                                 238
    Joakim Johansson 2024-12-02
                                220
```

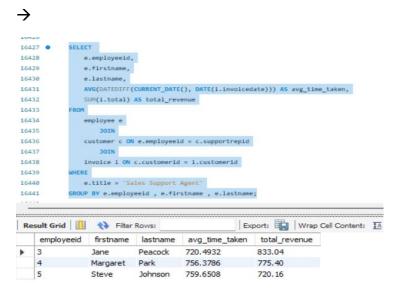
Q.32 Genre Popularity Shift Over Time (Quarterly Analysis):

-- For each quarter of each year present in the data, determine the top 3 most popular genres by total sales revenue. Display the year, quarter, genre name, and the total revenue for that genre in that quarter.



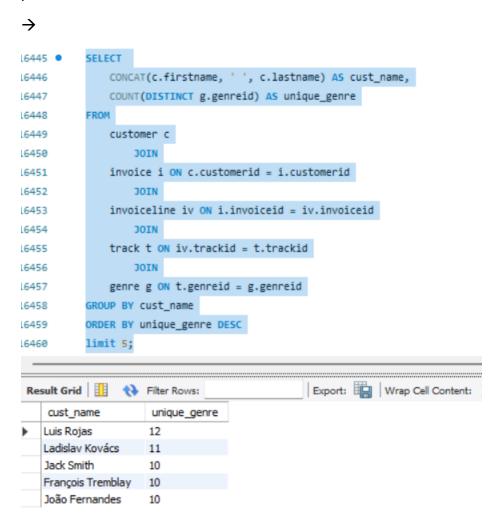
Q.33 Employee Efficiency - Average Time to Close an Invoice:

-- For each sales support agent (employee), calculate the average time (in days) between the Invoice Date and the date the invoice was paid (assume Billing Country is a proxy for "closed" if Invoice Date is the only date field). If no 'paid' date exists, assume its CURRENT_DATE. Also, show the total revenue generated by each agent.



Q.34 Most Diverse Customer Portfolios:

-- Identify the top 5 customers who have purchased tracks from the most unique genres. For each of these customers, list their full name, and the count of unique genres they've purchased from.



Q.35 Artist-to-Artist Collaboration Network (Indirect Sales Influence):

-- Identify pairs of artists where a customer who purchased a track from Artist A also purchased a track from Artist B, and these two artists are not directly related by having tracks on the same album. Quantify the "collaboration" by the number of unique customers who bought from both. Order the results by the collaboration count in descending order.

```
\rightarrow
```

```
16464 • - with cutomer_artist as (
              select distinct c.customerid,ar.artistid,ar.name as Artist_Name
16466
              from customer c
              join invoice i
16467
             on c.customerid = i.customerid
16468
16469
              join involceline iv
16470
              on i.invoiceid = iv.invoiceid
              join track t
16471
16472
              on iv.trackid = t.trackid
              join album a
16473
16474
              on t.albumid = a.albumid
16475
              join artist ar
              on a.artistid = ar.artistid
16476
16477
16478 — artist_album as (
              select distinct a.albumid,a.artistid
16479
16480
              from album a
              join track t
16481
16482
               on a.albumid = t.albumid
               where a.artistid is not null
16483
16484
              ),
16485 — artist_share_album as (
16486
              select aal.artistid as ArtistId1,aa2.artistid as ArtistId2
              from artist_album aa1
16487
16488
              join artist_album aa2
16489
              on aa1.albumid = aa2.albumid
               and aal.artistid < aa2.artistid
16498
16491
```

```
16493
         SELECT
16494
             ar1.name AS ArtistA,
16495
              ar2.name AS ArtistB,
16496
              COUNT(DISTINCT cal.customerid) AS c_count
16497
          FROM
               cutomer_artist cal
16498
16499
                  JOIN
16500
              cutomer_artist ca2 ON ca1.customerid = ca2.customerid
                 AND cal.artistid < cal.artistid
16501
16502
                   JOIN
16503
               artist ar1 ON ca1.artistid = ar1.artistid
16504
                   JOIN
               artist ar2 ON ca2.artistid = ar2.artistid
16505
16506
                  LEFT JOIN
16507
               artist_share_album asa ON (ca1.artistid = asa.artistid1
16508
                 AND ca2.artistid = asa.artistid2)
16509
                  OR (cal.artistid = asa.artistid2
16510
                  AND ca2.artistid = asa.artistid1)
16511
         WHERE
               asa.artistid1 IS NULL
16512
           GROUP BY ar1.name , ar2.name
16513
16514
           ORDER BY c_count DESC , ArtistA , ArtistB
           LIMIT 10;
16515
16516
Result Grid Filter Rows:
                                                Export: Wrap Cell Content: IA
    ArtistA
                     ArtistB
                                               c_count
   Led Zeppelin
                    U2
                                              14
    Iron Maiden
                    U2
                                              12
    Metallica
                    Iron Maiden
                                              12
    Iron Maiden
                 Lenny Kravitz
                                              11
    Iron Maiden
                    R.E.M.
                                              10
```

10

10

Iron Maiden

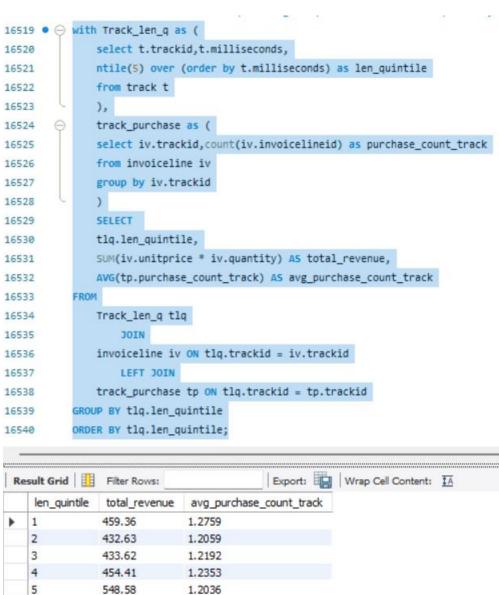
Os Paralamas Do Sucesso

Led Zeppelin Led Zeppelin

Metallica

Q.36 Impact of Track Length on Sales:

-- Divide tracks into 5 equal length 'quintiles'. For each quintile, calculate the total sales revenue and the average number of times tracks from that quintile were purchased. `



Report:

"To better understand the operational dynamics of our simulated digital media store, this report utilizes SQL to query the Chinook database. Our objective is to identify topperforming areas and potential opportunities for growth."

Insights:

Most Selling Genre: -

We can identify Popularity of Genre and most selling genre is "Rock"

Top Selling Tracks: -

Top selling Tracks are "Dazed and Confused", "The Trooper", "The Fix", etc.

> Top 3 Employee as Sales representative: -

Top 3 Employees are "Jane Peacock", "Margret Park", "Steve Johnson"

Monthly Trend Analysis: -

We can see that Sales are increasing after 1st month.

Employee Tenure: -

Most senior employee is "Nancy Edwards" who is working for 23 years.

> Artist with no Track sold: -

Artists whose track have not sold are "AC/DC", "Aerosmith", "Alice in chains" etc.

Most Popular Artists: -

Most Popular Artists are "Iron Maiden", "U2", "Lost" etc.

> Most Popular MediaType by Country: -

In Argentina "MPEG audio file", "Protected AAC audio file",

In Australia "MPEG audio file",

These are the Popular media types in each country.

Customer at Risk: -

"Pooja Srivastav", "Niclas Schröder", "Leonie Köhler", these are one of the customers who made their last purchase 350 days ago and losing interest in buying.

> Most valuable Customer: -

"Loise Rojas", "Ladislav Kovacs", "Jack Smith"

These are one of the most valuable customers as they have most diverse portfolio.