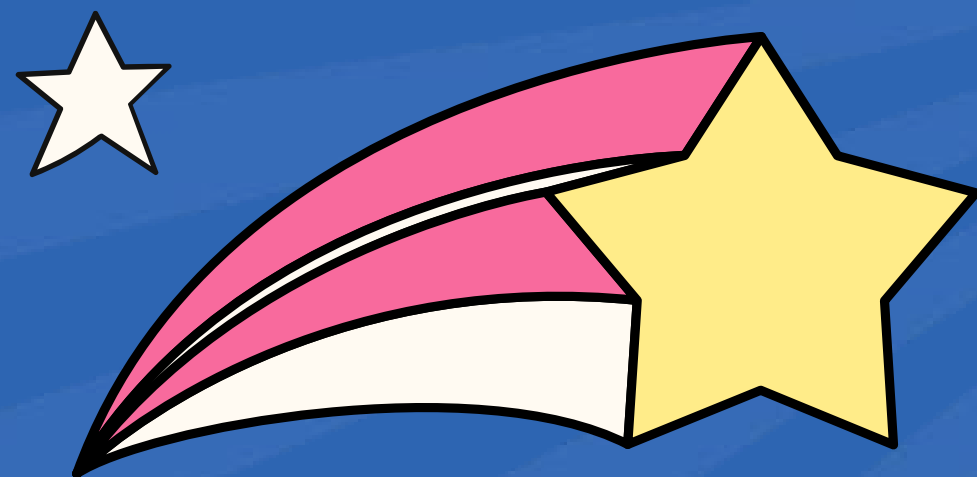
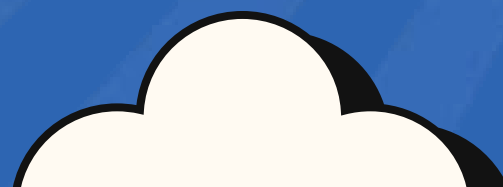


× × × ×

SQL PROJECT



DIGITAL MEDIA STORE ANALYSIS



OBJECTIVE

The purpose of the analysis is to gain insights into the operations and performance of the digital media store modeled by the Chinook database. By analyzing sales data, customer behavior, and product performance, we aim to identify trends, patterns, and opportunities for optimization.

This analysis will help stakeholders make informed decisions to improve sales, enhance customer satisfaction, and drive business growth.

LEVEL OF QUERIES



EASY-

Includes: Select,
Group By, Order
By , Limit,
Desc/Asc

MODERATE-

Includes: Joins,
Order By, Group
By, Limits.

ADVANCE-

Includes:
CTE(Common
Table
Expression)

× × × ×

EASY

```
SELECT * FROM
customer;
```

Select all columns from the customers table:

Output

	CustomerId	FirstName	LastName	Company	Address	City
▶	1	Luís	Gonçalves	Embraer - Empresa Brasileira de Aeronáutica S.A.	Av. Brigadeiro Faria Lima, 2170	São José do
	2	Leonie	Köhler	NULL	Theodor-Heuss-Straße 34	Stuttgart
	3	François	Tremblay	NULL	1498 rue Bélanger	Montréal
	4	Björn	Hansen	NULL	Ullevålsveien 14	Oslo

EASY

Select the first 5 rows from the tracks table, ordered by track name in descending order:

```
SELECT * FROM track
ORDER BY Name DESC
LIMIT 5;
```

Output

TrackId	Name	AlbumId	MediaTypeId	GenreId	Composer	Milliseconds	Bytes	UnitPrice
2505	[Untitled]	202	1	4	Billy Corgan	231784	7689713	0.99
3273	[Just Like] Starting Over	255	2	9	NULL	215549	3506308	0.99
3028	Zooropa	240	1	1	U2; Bono	392359	12807979	0.99
2926	Zoo Station	232	1	1	U2	276349	9056902	0.99
968	Zombie Eaters	77	1	4	Faith No More	360881	11835367	0.99

EASY

Count the number of tracks in each genre, ordered by genre name in ascending order:

```
SELECT GenreId, COUNT(*)  
AS track_count FROM track  
GROUP BY GenreId  
ORDER BY GenreId ASC;
```

Output

	GenreId	track_count
▶	1	1297
	2	130
	3	374
	4	332
	5	12

MODERATE

List customers who have made purchases, along with their total purchases amount, ordered by total amount in descending order

```
SELECT c.CustomerId, c.FirstName, c.LastName,  
       SUM(i.Total) AS total_purchases  
FROM customer c  
INNER JOIN invoice i ON c.CustomerId = i.CustomerId  
GROUP BY c.CustomerId, c.FirstName, c.LastName  
ORDER BY total_purchases DESC;
```

Output

	CustomerId	FirstName	LastName	total_purchases
▶	6	Helena	Holý	49.62
	26	Richard	Cunningham	47.62
	57	Luis	Rojas	46.62
	45	Ladislav	Kovács	45.62
	46	Hugh	O'Reilly	45.62

MODERATE

Show the top 10 tracks by the number of times they have been purchased, along with their genre, ordered by number of purchases in descending order

```
SELECT t.Name AS track_name, g.Name AS genre, COUNT(*)
      AS purchase_count
      FROM track t
      INNER JOIN invoiceline i ON t.TrackId = i.TrackId
      INNER JOIN genre g ON t.GenreId = g.GenreId
      GROUP BY t.TrackId, t.Name, g.Name
      ORDER BY purchase_count DESC
      LIMIT 10;
```

Output

Result Grid				Filter Rows:	Export:	Wrap Cell Content:
	track_name	genre	purchase_count			
▶	String Quartet No. 12 in C Minor, D. 703 "Quart...	Classical	2			
	Balls to the Wall	Rock	2			
	Welcome Home (Sanitarium)	Metal	2			
	Not The Doctor	Rock	2			
	Cornucopia	Metal	2			

MODERATE

Find the customer from the USA who has spent the most on purchases

```
SELECT c.FirstName, c.LastName, c.Country,
       SUM(i.total) AS total_spent
FROM customer c
INNER JOIN invoice i ON c.CustomerId = i.CustomerId
WHERE c.Country = 'USA'
GROUP BY c.FirstName, c.LastName, c.Country
ORDER BY total_spent DESC
LIMIT 1;
```

Output

Result Grid				Filter Rows:	Export:	Wrap Cell Content:
	track_name	genre	purchase_count			
▶	String Quartet No. 12 in C Minor, D. 703 "Quart...	Classical	2			
	Balls to the Wall	Rock	2			
	Welcome Home (Sanitarium)	Metal	2			
	Not The Doctor	Rock	2			
	Cornucopia	Metal	2			

ADVANCE

Calculate the total revenue generated by each employee in the Sales Support Agent role, ordered by total revenue in descending order

```
WITH agent_sales AS (  
  SELECT e.EmployeeId, e.FirstName || ' ' || e.LastName AS  
         agent_name, SUM(i.Total) AS total_sales  
         FROM employee e  
         INNER JOIN customer c ON e.EmployeeId =  
                                c.SupportRepId  
         INNER JOIN invoice i ON c.CustomerId = i.CustomerId  
         GROUP BY e.employeeId, agent_name  
)  
SELECT agent_name, total_sales  
FROM agent_sales  
ORDER BY total_sales DESC;
```



Output



	agent_name	total_sales
▶	0	833.04
	0	775.40
	0	720.16

ADVANCE

Find the number of customers who made their first purchase in each year, ordered by year in ascending order

Output

```
WITH first_purchase_year AS (  
  SELECT customer_id, MIN(invoice_date) AS  
         first_purchase  
  FROM invoices  
  GROUP BY customer_id  
)  
SELECT YEAR(first_purchase) AS purchase_year,  
       COUNT(*) AS new_customers  
FROM first_purchase_year  
GROUP BY purchase_year  
ORDER BY purchase_year ASC;
```



	purchase_year	new_customers
▶	2021	46
	2022	13

THANK YOU!