-- SUBMISSION BY: Prasang Shrivas

-- PROJECT: CHINOOK MUSIC

-- SKILL USED: MYSQL WORKBENCH

-- BATCH: DATA SCIENCE COURSE MAY 2024

select \* from employee;

select \* from customer;

select \* from invoice;

select \* from invoice\_line;

select \* from track;

select \* from genre;

select \* from playlist;

select \* from playlist\_track;

select \* from album;

select \* from artist;

select \* from media\_type;

use chinook;

-- ------------------------------------------------------OBJECTIVE QUESTIONS---------------------------------------------------------

-- Q1. Does any table have missing values or duplicates? If yes how would you handle it ?-----------------------------------

-- -- Checking for Null Values in employee table -- --

SELECT \*

FROM employee

WHERE last\_name IS NULL

OR first\_name IS NULL

OR title IS NULL

OR reports\_to IS NULL

OR birthdate IS NULL

OR hire\_date IS NULL

OR address IS NULL

OR city IS NULL

OR state IS NULL

OR country IS NULL

OR postal\_code IS NULL

OR phone IS NULL

OR fax IS NULL

OR email IS NULL;

-- -- Handling Null Values -- --

SELECT

employee\_id,

COALESCE(reports\_to, 'N/A') AS reports\_to

FROM

employee;

-- -- Checking for Null Values in customer table-- --

SELECT \*

FROM customer

WHERE first\_name IS NULL

OR last\_name IS NULL

OR company IS NULL

OR address IS NULL

OR city IS NULL

OR state IS NULL

OR country IS NULL

OR postal\_code IS NULL

OR phone IS NULL

OR fax IS NULL

OR email IS NULL

OR support\_rep\_id IS NULL;

-- -- Handling Null Values -- --

SELECT customer\_id,

COALESCE(company, 'N/A') AS company,

COALESCE(support\_rep\_id, 0) AS support\_rep\_id

FROM customer;

-- Q2. Find the top-selling tracks and top artist in the USA and identify their most famous genres?-------------------------

-- --Top Selling Track in USA-- --

SELECT

t.track\_id,

t.name AS track\_name,

SUM(il.quantity) AS total\_sold,

g.name AS genre,

a.name AS artist

FROM

invoice\_line il

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

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

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

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

INNER JOIN artist a ON al.artist\_id = a.artist\_id

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

WHERE

c.country = 'USA'

GROUP BY

t.track\_id, t.name, g.name, a.name

ORDER BY

total\_sold DESC

LIMIT 10;

-- --Top Artists in USA-- --

SELECT

a.artist\_id,

a.name AS artist\_name,

SUM(il.quantity) AS total\_sold

FROM

invoice\_line il

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

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

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

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

INNER JOIN artist a ON al.artist\_id = a.artist\_id

WHERE

c.country = 'USA'

GROUP BY

a.artist\_id, a.name

ORDER BY

total\_sold DESC

LIMIT 1;

-- --Most Famous Genres of the Top Artist -- --

SELECT

g.genre\_id,

g.name AS genre\_name,

SUM(il.quantity) AS total\_sold

FROM

invoice\_line il

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

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

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

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

INNER JOIN artist a ON al.artist\_id = a.artist\_id

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

WHERE

c.country = 'USA' AND a.artist\_id = 152

GROUP BY

g.genre\_id, g.name

ORDER BY

total\_sold DESC;

-- Q3. What is the customer demographic breakdown (age, gender, location) of Chinook's customer base?-----------------------------

WITH customer\_information\_cte as (

SELECT

customer\_id,

first\_name,

last\_name,

city,

COALESCE(state,'N.A') as state,

country

FROM

customer)

SELECT

country,

state,

city,

COUNT(customer\_id) as total\_customers

FROM

customer\_information\_cte

GROUP BY

country, state, city

ORDER BY

country, state, city;

-- Q4. Calculate the total revenue and number of invoices for each country, state, and city?--------------------------------------------

SELECT

c.country,

COALESCE(c.state,'N.A') as state,

c.city,

COUNT(i.invoice\_id) as number\_of\_invoices,

SUM(i.total) as total\_revenue

FROM

customer c

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

GROUP BY

c.country, c.state, c.city

ORDER BY

total\_revenue DESC;

-- Q5. Find the top 5 customers by total revenue in each country-------------------------------------------------------------------

select \* from customer;

select \* from invoice;

WITH customer\_wise\_revenue\_cte1 as(

SELECT

c.customer\_id,

CONCAT(c.first\_name, ' ', c.last\_name) as customers,

c.country,

SUM(i.total) as total\_revenue

FROM

customer c

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

GROUP BY

c.customer\_id, customers, c.country

ORDER BY

c.country, total\_revenue),

ranked\_customers\_cte2 as (

SELECT

customer\_id,

customers,

country,

total\_revenue,

RANK() OVER (PARTITION BY country ORDER BY total\_revenue desc) as customer\_rank

FROM

customer\_wise\_revenue\_cte1)

SELECT

customer\_id,

customers,

country,

total\_revenue,

customer\_rank

FROM

ranked\_customers\_cte2

WHERE

customer\_rank <= 5

ORDER BY

country, customer\_rank;

-- Q6. Identify the top-selling track for each customer----------------------------------------------------------------------------

WITH Customer\_track as (

SELECT

c.customer\_id,

CONCAT(c.first\_name, ' ', c.last\_name) as customers,

SUM(il.quantity) as total\_quantity

FROM

customer c

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

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

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

GROUP BY

c.customer\_id, customers),

ranked\_track as(

SELECT

Customer\_track.customer\_id,

Customer\_track.customers,

Customer\_track.total\_quantity,

t.track\_id,

t.name as track\_name,

ROW\_NUMBER() OVER (PARTITION BY Customer\_track.customer\_id ORDER BY Customer\_track.total\_quantity DESC) as track\_rank

FROM

Customer\_track

INNER JOIN invoice i ON Customer\_track.customer\_id = i.customer\_id

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

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

SELECT

customer\_id,

customers,

track\_id,

track\_name,

total\_quantity

FROM

ranked\_track

WHERE

track\_rank = 1

ORDER BY

customer\_id;

-- Q7. Are there any patterns or trends in customer purchasing behaviour...

-- (e.g., frequency of purchases, preferred payment methods, average order value)?-------------------------------------------------

-- --Frequency of Purchases -- --

SELECT

c.customer\_id,

CONCAT(c.first\_name, ' ', c.last\_name) as customers,

YEAR(i.invoice\_date) AS year,

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

FROM

customer c

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

GROUP BY

c.customer\_id, customers, YEAR(i.invoice\_date)

ORDER BY

c.customer\_id, customers, YEAR(i.invoice\_date);

-- -- Calculate the average order value for each customer -- --

SELECT

c.customer\_id,

CONCAT(c.first\_name, ' ', c.last\_name) as customers,

ROUND(AVG(i.total), 2) AS avg\_order\_value

FROM

customer c

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

GROUP BY

c.customer\_id, customers

ORDER BY

avg\_order\_value desc;

-- -- Calculate the total revenue generated by each customer -- --

SELECT

c.customer\_id,

CONCAT(c.first\_name, ' ', c.last\_name) as customers,

SUM(i.total) AS total\_revenue

FROM

customer c

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

GROUP BY

c.customer\_id, customers

ORDER BY

total\_revenue desc;

-- -- Identify the preferred purchase periods -- --

SELECT

c.customer\_id,

CONCAT(c.first\_name, ' ', c.last\_name) as customers,

DAYOFWEEK(i.invoice\_date) AS day\_of\_week,

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

FROM

customer c

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

GROUP BY

c.customer\_id, c.first\_name, c.last\_name, DAYOFWEEK(i.invoice\_date)

ORDER BY

c.customer\_id, customers, purchase\_count DESC;

-- Q8. What is the customer churn rate?---------------------------------------------------------------------------------------------

WITH MostRecentInvoice AS (

SELECT MAX(invoice\_date) AS most\_recent\_invoice\_date

FROM invoice

),

CutoffDate AS (

SELECT DATE\_SUB(most\_recent\_invoice\_date, INTERVAL 1 YEAR) AS cutoff\_date

FROM MostRecentInvoice

),

ChurnedCustomers AS (

SELECT

c.customer\_id,

COALESCE(c.first\_name, ' ',c.last\_name) as customers,

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

FROM

customer c

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

GROUP BY

c.customer\_id, customers

HAVING

MAX(i.invoice\_date) IS NULL OR MAX(i.invoice\_date) < (SELECT cutoff\_date FROM CutoffDate)

)

-- -- Calculate the churn rate -- --

SELECT

(SELECT COUNT(\*) FROM ChurnedCustomers) / (SELECT COUNT(\*) FROM customer) \* 100 AS churn\_rate;

-- Q9. Calculate the percentage of total sales contributed by each genre in the USA and identify the best-selling genres and artists.----

WITH genre\_sales\_in\_usa AS (

SELECT

g.genre\_id,

g.name AS genre\_name,

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

FROM

genre g

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

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

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

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

WHERE

c.country = 'USA'

GROUP BY

g.genre\_id, g.name),

total\_sales as(

SELECT

SUM(total\_genre\_sales) as total\_usa\_sales

FROM

genre\_sales\_in\_usa),

genre\_sales\_percentage AS(

SELECT

gs.genre\_id,

gs.genre\_name,

gs.total\_genre\_sales,

ts.total\_usa\_sales,

(gs.total\_genre\_sales/ts.total\_usa\_sales) \* 100 AS percentage\_contribution

FROM

genre\_sales\_in\_usa gs

CROSS JOIN total\_sales ts),

best\_selling\_artist AS (

SELECT

g.genre\_id,

g.name AS genre\_name,

a.artist\_id,

a.name AS artist\_name,

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

FROM

genre g

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

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

INNER JOIN artist a ON a.artist\_id = al.artist\_id

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

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

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

WHERE

c.country = 'USA'

GROUP BY

g.genre\_id, g.name, a.artist\_id, a.name)

SELECT

genre\_id,

genre\_name,

artist\_id,

artist\_name,

total\_artists\_sales,

DENSE\_RANK() OVER (PARTITION BY genre\_id ORDER BY total\_artists\_sales DESC) AS artist\_rank

FROM

best\_selling\_artist;

-- 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 customers,

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

FROM

customer c

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

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

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

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

GROUP BY

c.customer\_id, customers

HAVING

COUNT(DISTINCT g.genre\_id) >= 3

ORDER BY

genre\_count DESC;

-- Q11 Rank genres based on their sales performance in the USA --

WITH genre\_sales\_in\_usa AS (

SELECT

g.genre\_id,

g.name AS genre\_name,

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

FROM

genre g

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

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

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

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

WHERE

c.country = 'USA'

GROUP BY

g.genre\_id, g.name)

SELECT

genre\_id,

genre\_name,

total\_genre\_sales,

RANK() OVER (ORDER BY total\_genre\_sales DESC) AS genre\_rank

FROM

genre\_sales\_in\_usa

ORDER BY

genre\_rank;

-- Q12. Identify customers who have not made a purchase in the last 3 months --

WITH recent\_purchases AS (

SELECT

c.customer\_id

FROM

customer c

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

WHERE

i.invoice\_date >= CURDATE() - INTERVAL 3 MONTH)

SELECT

c.customer\_id,

CONCAT(c.first\_name, ' ', c.last\_name) as customers

FROM

customer c

LEFT JOIN recent\_purchases rp ON c.customer\_id = rp.customer\_id

WHERE

rp.customer\_id IS NULL

ORDER BY

c.customer\_id;

-- ------------------------------------------------SUBJECTIVE QUESTIONS------------------------------------------------------ --

-- Q1. Recommend the three albums from the new record label that should be

-- prioritized for advertising and promotion in the USA based on genre sales analysis ?----------------------------------------------

SELECT

g.genre\_id,

g.name AS genre\_name,

al.album\_id,

al.title AS new\_record\_label,

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

DENSE\_RANK() OVER (ORDER BY SUM(il.unit\_price \* il.quantity) DESC) AS Ranking

FROM

genre g

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

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

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

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

INNER JOIN album al on t.album\_id = al.album\_id

WHERE

c.country = 'USA'

GROUP BY

g.genre\_id, g.name, al.album\_id,

al.title

ORDER BY

total\_genre\_sales DESC;

-- Q2. Determine the top-selling genres in countries other than the USA

-- and identify any commonalities or differences ?------------------------------------------------------------------------------

-- -- Top Selling Genres in countries other than USA? -- --

SELECT

g.genre\_id,

g.name AS genre\_name,

c.country,

SUM(il.quantity) AS quantity\_sold

FROM

genre g

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

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

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

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

WHERE

country <> 'USA'

GROUP BY

g.genre\_id, genre\_name, c.country

ORDER BY

quantity\_sold DESC;

-- -- Top Selling Genres in countries in USA? -- --

SELECT

g.genre\_id,

g.name AS genre\_name,

c.country,

SUM(il.quantity) AS quantity\_sold

FROM

genre g

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

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

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

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

WHERE

country = 'USA'

GROUP BY

g.genre\_id, genre\_name, c.country

ORDER BY

quantity\_sold DESC;

-- Q3. Customer Purchasing Behaviour 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?--------------------------------------------

WITH CustomerPurchaseStats AS (

SELECT

c.customer\_id,

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

SUM(il.quantity) AS total\_items\_purchased,

SUM(i.total) AS total\_spent,

AVG(i.total) AS avg\_order\_value,

DATEDIFF(MAX(i.invoice\_date), MIN(i.invoice\_date)) AS customer\_tenure\_days

FROM

customer c

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

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

GROUP BY

c.customer\_id

),

CustomerSegments AS (

SELECT

customer\_id,

purchase\_frequency,

total\_items\_purchased,

total\_spent,

avg\_order\_value,

customer\_tenure\_days,

CASE

WHEN customer\_tenure\_days >= 365 THEN 'Long-Term'

ELSE 'New'

END AS customer\_segment

FROM

CustomerPurchaseStats

)

SELECT

customer\_segment,

ROUND(AVG(purchase\_frequency),2) AS avg\_purchase\_frequency,

ROUND(AVG(total\_items\_purchased),2) AS avg\_basket\_size,

ROUND(AVG(total\_spent),2) AS avg\_spending\_amount,

ROUND(AVG(avg\_order\_value),2) AS avg\_order\_value

FROM

CustomerSegments

GROUP BY

customer\_segment;

-- Q4. 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?

-- -- 1. Genre Affinity Analysis --

WITH track\_combinations AS (

SELECT

il1.track\_id AS track\_id\_1,

il2.track\_id AS track\_id\_2,

COUNT(\*) AS times\_purchased\_together

FROM

invoice\_line il1

JOIN

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

AND il1.track\_id < il2.track\_id

GROUP BY

il1.track\_id, il2.track\_id

),

genre\_combinations AS (

SELECT

t1.genre\_id AS genre\_id\_1,

t2.genre\_id AS genre\_id\_2,

COUNT(\*) AS times\_purchased\_together

FROM

track\_combinations tc

JOIN

track t1 ON tc.track\_id\_1 = t1.track\_id

JOIN

track t2 ON tc.track\_id\_2 = t2.track\_id

WHERE

t1.genre\_id <> t2.genre\_id

GROUP BY

t1.genre\_id, t2.genre\_id

)

SELECT

g1.name AS genre\_1,

g2.name AS genre\_2,

gc.times\_purchased\_together

FROM

genre\_combinations gc

JOIN

genre g1 ON gc.genre\_id\_1 = g1.genre\_id

JOIN

genre g2 ON gc.genre\_id\_2 = g2.genre\_id

ORDER BY

gc.times\_purchased\_together DESC;

-- -- 2. Artist Affinity Analysis --

WITH track\_combinations AS (

SELECT

il1.track\_id AS track\_id\_1,

il2.track\_id AS track\_id\_2,

COUNT(\*) AS times\_purchased\_together

FROM

invoice\_line il1

JOIN

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

AND il1.track\_id < il2.track\_id

GROUP BY

il1.track\_id, il2.track\_id

),

artist\_combinations AS (

SELECT

a1.artist\_id AS artist\_id\_1,

a2.artist\_id AS artist\_id\_2,

COUNT(\*) AS times\_purchased\_together

FROM

track\_combinations tc

JOIN

track t1 ON tc.track\_id\_1 = t1.track\_id

JOIN

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

JOIN

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

JOIN

track t2 ON tc.track\_id\_2 = t2.track\_id

JOIN

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

JOIN

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

WHERE

a1.artist\_id <> a2.artist\_id

GROUP BY

a1.artist\_id, a2.artist\_id

)

SELECT

a1.name AS artist\_1,

a2.name AS artist\_2,

ac.times\_purchased\_together

FROM

artist\_combinations ac

JOIN

artist a1 ON ac.artist\_id\_1 = a1.artist\_id

JOIN

artist a2 ON ac.artist\_id\_2 = a2.artist\_id

ORDER BY

ac.times\_purchased\_together DESC;

-- -- 3. Album Affinity Analysis --

WITH track\_combinations AS (

SELECT

il1.track\_id AS track\_id\_1,

il2.track\_id AS track\_id\_2,

COUNT(\*) AS times\_purchased\_together

FROM

invoice\_line il1

JOIN

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

AND il1.track\_id < il2.track\_id

GROUP BY

il1.track\_id, il2.track\_id

),

album\_combinations AS (

SELECT

al1.album\_id AS album\_id\_1,

al2.album\_id AS album\_id\_2,

COUNT(\*) AS times\_purchased\_together

FROM

track\_combinations tc

JOIN

track t1 ON tc.track\_id\_1 = t1.track\_id

JOIN

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

JOIN

track t2 ON tc.track\_id\_2 = t2.track\_id

JOIN

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

WHERE

al1.album\_id <> al2.album\_id

GROUP BY

al1.album\_id, al2.album\_id

)

SELECT

al1.title AS album\_1,

al2.title AS album\_2,

ac.times\_purchased\_together

FROM

album\_combinations ac

JOIN

album al1 ON ac.album\_id\_1 = al1.album\_id

JOIN

album al2 ON ac.album\_id\_2 = al2.album\_id

ORDER BY

ac.times\_purchased\_together DESC;

-- Q5. 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?--------------------------

-- -- Customer Purchasing Behaviors by Region -- --

WITH purchase\_frequency AS (

SELECT

customer\_id,

COUNT(invoice\_id) AS total\_purchase\_freq,

SUM(total) AS total\_spending,

AVG(total) AS avg\_order\_value

FROM

invoice

GROUP BY

customer\_id

),

customer\_region\_summary AS (

SELECT

c.customer\_id,

c.country,

COALESCE(c.state,'N.A') as state,

c.city,

pf.total\_purchase\_freq,

pf.total\_spending,

pf.avg\_order\_value

FROM

customer c

JOIN

purchase\_frequency pf ON c.customer\_id = pf.customer\_id

),

regional\_summary AS (

SELECT

country,

state,

city,

ROUND(COUNT(DISTINCT customer\_id),2) AS total\_customers,

ROUND(SUM(total\_purchase\_freq),2) AS total\_purchases,

ROUND(SUM(total\_spending),2) AS total\_spending,

ROUND(AVG(avg\_order\_value),2) AS avg\_order\_value,

ROUND(AVG(total\_purchase\_freq),2) AS avg\_purchase\_frequency

FROM

customer\_region\_summary

GROUP BY

country, state, city

)

SELECT

country,

state,

city,

total\_customers,

total\_purchases,

total\_spending,

avg\_order\_value,

avg\_purchase\_frequency

FROM

regional\_summary

ORDER BY

total\_spending DESC;

-- -- Churn Rate by Region -- --

WITH last\_purchase AS (

SELECT

c.customer\_id,

c.country,

COALESCE(c.state,'N.A') as state,

c.city,

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

FROM

customer c

JOIN

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

GROUP BY

c.customer\_id, c.country, c.state, c.city

),

churned\_customers AS (

SELECT

country,

state,

city,

COUNT(customer\_id) AS churned\_customers

FROM

last\_purchase

WHERE

last\_purchase\_date < DATE\_SUB(CURDATE(), INTERVAL 1 YEAR)

GROUP BY

country, state, city

)

SELECT

lc.country,

lc.state,

lc.city,

lc.churned\_customers,

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

(lc.churned\_customers / COUNT(c.customer\_id)) \* 100 AS churn\_rate

FROM

churned\_customers lc

JOIN

customer c ON lc.country = c.country AND lc.state = c.state AND lc.city = c.city

GROUP BY

lc.country, lc.state, lc.city

ORDER BY

churn\_rate DESC;

-- Q6. 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?------------------------------------------------------------------------------------------

WITH customer\_profile AS (

SELECT

c.customer\_id,

c.country,

COALESCE(c.state,'N.A') as state,

c.city,

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

SUM(i.total) AS total\_spending,

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

AVG(i.total) AS avg\_order\_value

FROM

customer c

LEFT JOIN

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

GROUP BY

c.customer\_id

),

churn\_risk AS (

SELECT

cp.customer\_id,

cp.country,

cp.state,

cp.city,

cp.total\_spending,

cp.purchase\_frequency,

cp.avg\_order\_value,

CASE

WHEN cp.last\_purchase\_date < DATE\_SUB(CURDATE(), INTERVAL 1 YEAR) THEN 'High Risk'

WHEN cp.total\_spending < 100 THEN 'Medium Risk'

ELSE 'Low Risk'

END AS risk\_profile

FROM

customer\_profile cp

),

risk\_summary AS (

SELECT

country,

state,

city,

risk\_profile,

ROUND(COUNT(customer\_id),2) AS num\_customers,

ROUND(AVG(total\_spending),2) AS avg\_total\_spending,

ROUND(AVG(purchase\_frequency),2) AS avg\_purchase\_frequency,

ROUND(AVG(avg\_order\_value),2) AS avg\_order\_value

FROM

churn\_risk

GROUP BY

country, state, city, risk\_profile

)

SELECT

country,

state,

city,

risk\_profile,

num\_customers,

avg\_total\_spending,

avg\_purchase\_frequency,

avg\_order\_value

FROM

risk\_summary

ORDER BY

risk\_profile DESC, avg\_total\_spending DESC;

-- Q7. Customer Lifetime Value Modelling: 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?-------------------------

WITH customer\_profile AS (

SELECT

c.customer\_id,

CONCAT(c.first\_name, ' ', c.last\_name) as customers,

c.country,

COALESCE(c.state,'N.A') AS state,

c.city,

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 customer\_tenure\_days,

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

SUM(i.total) AS total\_spending,

AVG(i.total) AS avg\_order\_value

FROM

customer c

LEFT JOIN

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

GROUP BY

c.customer\_id

),

customer\_lifetime\_value AS (

SELECT

cp.customer\_id,

cp.customers,

cp.country,

cp.state,

cp.city,

cp.customer\_tenure\_days,

cp.total\_purchases,

cp.total\_spending,

cp.avg\_order\_value,

CASE

WHEN cp.customer\_tenure\_days >= 365 THEN 'Long-Term'

ELSE 'Short-Term'

END AS customer\_segment,

CASE

WHEN cp.last\_purchase\_date < DATE\_SUB(CURDATE(), INTERVAL 1 YEAR) THEN 'Churned'

ELSE 'Active'

END AS customer\_status,

(cp.total\_spending / GREATEST(cp.customer\_tenure\_days, 1)) \* 365 AS predicted\_annual\_value,

cp.total\_spending AS lifetime\_value

FROM

customer\_profile cp

),

segment\_analysis AS (

SELECT

customer\_segment,

customer\_status,

COUNT(customer\_id) AS num\_customers,

AVG(customer\_tenure\_days) AS avg\_tenure\_days,

AVG(total\_spending) AS avg\_lifetime\_value,

AVG(predicted\_annual\_value) AS avg\_predicted\_annual\_value

FROM

customer\_lifetime\_value

GROUP BY

customer\_segment, customer\_status

),

churn\_analysis AS (

SELECT

country,

state,

city,

customer\_segment,

COUNT(customer\_id) AS churned\_customers,

AVG(total\_spending) AS avg\_lifetime\_value

FROM

customer\_lifetime\_value

WHERE

customer\_status = 'Churned'

GROUP BY

country, state, city, customer\_segment

)

SELECT

\*

FROM

customer\_lifetime\_value

ORDER BY

lifetime\_value DESC;

-- Additional queries to analyze the results:

-- Segment Analysis

SELECT

\*

FROM

segment\_analysis

ORDER BY

avg\_lifetime\_value DESC;

-- Churn Analysis

SELECT

\*

FROM

churn\_analysis

ORDER BY

churned\_customers DESC;

-- Q8. If data on promotional campaigns (discounts, events, email marketing) is available,

-- how could you measure their impact on customer acquisition, retention, and overall sales?---------------------------------------------

-- -- Answered in Word File

-- Q9. How would you approach this problem, if the objective and subjective questions weren't given?-------------------------------------

-- -- Answered in Word File

-- Q10. 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;

select \* from Album;

UPDATE album

SET ReleaseYear = 2017

WHERE album\_id = 1;

UPDATE album

SET ReleaseYear = 2017

WHERE album\_id = 2;

UPDATE album

SET ReleaseYear = 2017

WHERE album\_id = 3;

UPDATE album

SET ReleaseYear = 2017

WHERE album\_id = 4;

UPDATE album

SET ReleaseYear = 2017

WHERE album\_id = 5;

UPDATE album

SET ReleaseYear = 2018

WHERE album\_id = 6;

UPDATE album

SET ReleaseYear = 2018

WHERE album\_id = 7;

UPDATE album

SET ReleaseYear = 2018

WHERE album\_id = 8;

UPDATE album

SET ReleaseYear = 2018

WHERE album\_id = 9;

UPDATE album

SET ReleaseYear = 2018

WHERE album\_id = 10;

-- Q11. Chinook is interested in understanding the purchasing behaviour 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.----------------------------------------------------------------------------------------

WITH tracks\_per\_customer AS (

SELECT

i.customer\_id,

SUM(il.quantity) AS total\_tracks

FROM

invoice i

JOIN

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

GROUP BY

i.customer\_id

),

customer\_spending AS (

SELECT

c.country,

c.customer\_id,

SUM(i.total) AS total\_spent,

tpc.total\_tracks

FROM

customer c

JOIN

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

JOIN

tracks\_per\_customer tpc ON c.customer\_id = tpc.customer\_id

GROUP BY

c.country, c.customer\_id, tpc.total\_tracks

)

SELECT

cs.country,

COUNT(DISTINCT cs.customer\_id) AS number\_of\_customers,

ROUND(AVG(cs.total\_spent),2) AS average\_amount\_spent\_per\_customer,

ROUND(AVG(cs.total\_tracks),2) AS average\_tracks\_purchased\_per\_customer

FROM

customer\_spending cs

GROUP BY

cs.country

ORDER BY

average\_amount\_spent\_per\_customer DESC;

-- ----------------------------------------------------THE END-----------------------------------------------------------