Chinook Database - SQL & Advanced SQL Practice Questions

## Basic SQL (SELECT, WHERE, ORDER BY, LIMIT)

* List all customers.
* Show all tracks with their names and unit prices.
* List all employees in the sales department.
* Retrieve all invoices from the year 2011.
* Show all albums by "AC/DC".
* List tracks with a duration longer than 5 minutes.
* Get the list of customers from Canada.
* List 10 most expensive tracks.
* List employees who report to another employee.
* Show the invoice date and total for invoice ID 5.

## SQL Joins (INNER, LEFT, RIGHT, FULL)

* List all customers with their respective support representative's name.
* Get a list of all invoices along with the customer name.
* Show all tracks along with their album title and artist name.
* List all playlists and the number of tracks in each.
* Get the name of all employees and their managers (self-join).
* Show all invoices with customer name and billing country.
* List tracks along with their genre and media type.
* Get a list of albums and the number of tracks in each.
* List all artists with no albums.
* Find all customers who have never purchased anything.

## Aggregations and Group By

* Count the number of customers in each country.
* Total invoice amount by each customer.
* Average track duration per album.
* Total number of tracks per genre.
* Revenue generated per country.
* Average invoice total per billing city.
* Number of employees per title.
* Find the top 5 selling artists.
* Number of playlists containing more than 10 tracks.
* Top 3 customers by invoice total.

## Subqueries (Scalar, Correlated, IN, EXISTS)

* Get customers who have spent more than the average.
* List tracks that are more expensive than the average price.
* Get albums that have more than 10 tracks.
* Find artists with more than 1 album.
* Get invoices that contain more than 5 line items.
* Find tracks that do not belong to any playlist.
* List customers with invoices over $15.
* Show customers who have purchased all genres.
* Find customers who haven’t bought from the 'Rock' genre.
* List tracks where unit price is greater than the average unit price of its media type.

## Advanced Joins and Set Operations

* Get tracks in both 'Rock' and 'Jazz' playlists.
* List all tracks that are in 'Pop' but not in 'Rock' playlists.
* Union customers from USA and Canada.
* Intersect customers from Canada and those who bought ‘AC/DC’ albums.
* Get artists that have albums but no tracks.
* Find employees who are not assigned any customers.
* List invoices where total is greater than the sum of any other invoice.
* Get customers who have made more than 5 purchases using a correlated subquery.
* List tracks that appear in more than 2 playlists.
* Show albums where all tracks are longer than 3 minutes.

## Window Functions

* Rank customers by total spending.
* Show top 3 selling genres per country.
* Get running total of invoice amounts by customer.
* Find the invoice with the highest amount per customer.
* Get the dense rank of employees by hire date.
* List tracks along with their rank based on unit price within each genre.
* Compute average invoice total by country using window functions.
* Show lag/lead of invoice totals per customer.
* List customers and their second highest invoice.
* Get the difference in invoice total from previous invoice for each customer.

## CTEs and Recursive Queries

* List employees and their managers using recursive CTE.
* Use CTE to get top 3 customers by total spending.
* Create a CTE to list all invoice lines for albums by 'Metallica'.
* Use a CTE to show all tracks that appear in more than one playlist.
* Recursive CTE to list employee hierarchy (if > 2 levels).
* CTE to get all albums with total track time > 30 minutes.
* Get top 5 albums by total revenue using CTE and window functions.
* Use CTE to find average track price per genre and filter only those above global average.
* CTE to find customers with the longest names.
* Create a CTE to rank all albums by number of tracks.

## Advanced Analytics

* Get month-over-month revenue change.
* Calculate customer lifetime value.
* Get retention: how many customers returned for a second purchase?
* Identify top selling track in each country.
* Show invoice trends by quarter.
* Count customers acquired per year.
* Find churned customers (no purchases in last 12 months).
* Show most played tracks per user (using playlist track if usage data is simulated).
* Simulate cohort analysis by signup month.
* Calculate total revenue per artist using joins and group by.

## Data Validation and Integrity Checks

* Find invoice lines with NULL unit price.
* Detect duplicate tracks (by name, album, duration).
* List tracks with unit price < 0.
* Find customers with missing emails.
* Check for invoices without invoice lines.
* Validate if total in invoices match the sum of invoice lines.
* Find tracks assigned to multiple genres (data anomaly).
* Check for albums without artists.
* List employees who support more than 20 customers.
* Show customers who have the same first and last names.

## Business Scenarios

* Recommend top 3 tracks for a customer based on genre preference.
* Identify slow-moving tracks (not sold in last 12 months).
* Get summary of purchases per customer per genre.
* Find the artist with highest average track duration.
* Show difference in price between highest and lowest track per genre.
* Find customers who buy only once vs those who buy multiple times.
* List countries with the most revenue per capita (assume fixed population per country).
* Recommend albums with similar genre to customer past purchases.
* Estimate revenue impact if top 10% customers churn.
* Calculate the average invoice total per support rep’s customer group.

## Bonus / Optional

* Create a view for customer invoices with aggregated total.
* Write a stored procedure to get top customers by year.
* Simulate insertion of a new invoice using INSERT + SELECT.
* Write a function to return total revenue for an artist.
* Use a trigger to prevent deleting a customer with invoices.