**SECTION 6: AI Tools & LLM**

**Option A: Prompt Engineering**

The tool that I have used is ChatGPT for which I have prompted the following

SELECT Artist.Name AS Artist,

Album.Title AS Album,

SUM(InvoiceLine.Quantity) AS TotalSold

FROM Artist

JOIN Album ON Artist.ArtistId = Album.ArtistId

JOIN Track ON Album.AlbumId = Track.AlbumId

JOIN InvoiceLine ON Track.TrackId = InvoiceLine.TrackId

GROUP BY Album.AlbumId

ORDER BY Artist.Name, TotalSold DESC;

I am solving an SQL problem,

This problem is from the Chinook database and I am using SQLite Online for the same.

This query is not returning only one artist for their most sold album

Please help me modify this query

**Response:**

You're close! The issue with your current query is that it returns all albums with their total sold count, grouped by album and artist — but you only want the most sold album per artist.

To fix this, you'll need to:

1. Compute total sales per album as a subquery.
2. Then use a window function (ROW\_NUMBER() or RANK()) to rank the albums per artist.
3. Finally, filter to keep only the top-ranked (most sold) album for each artist.

However, SQLite only supports window functions from version 3.25 onwards. If you're using a modern SQLite version (like on SQLite Online), the following should work:

WITH AlbumSales AS (

    SELECT

        ar.ArtistId,

        ar.Name AS Artist,

        al.Title AS Album,

        SUM(il.Quantity) AS TotalSold,

        ROW\_NUMBER() OVER (

            PARTITION BY ar.ArtistId

            ORDER BY SUM(il.Quantity) DESC

        ) AS rn

    FROM Artist ar

    JOIN Album al ON ar.ArtistId = al.ArtistId

    JOIN Track t ON al.AlbumId = t.AlbumId

    JOIN InvoiceLine il ON t.TrackId = il.TrackId

    GROUP BY ar.ArtistId, ar.Name, al.AlbumId, al.Title

)

SELECT Artist, Album, TotalSold

FROM AlbumSales

WHERE rn = 1

ORDER BY TotalSold DESC;

**🔍 What this gives:**

* Exactly **one album per artist**.
* The **top-selling album** by total units sold.
* Ordered by total sales across all artists (optional).

Let me know if you want it sorted alphabetically instead, or adapted for older SQL engines that don’t support ROW\_NUMBER().