Take Home Test

SQL

This document provides a SQL schema and a set of queries for a take-home test. The schema includes the following tables: `cm_artist`, `cm_track`, `l_cm_track_cm_artist`, `cm_artist_cache_history`, `cm_album`, and `l_cm_track_cm_album`. The queries include a variety of tasks, such as retrieving artist names along with their total number of tracks, finding the earliest release year of an artist's album, identifying tracks associated with multiple artists, and calculating the average and running total of follower counts for each artist from the `cm_artist_cache_history` table.

Schema

cm_artist

Column	Data Type
id	integer
name	text
created_at	timestamp

Primary Key: id

sample rows

id	name	created_at
1	Artist A	2023-01-01 12:00:00
2	Artist B	2023-01-02 12:00:00

---

cm_track

Column	Data Type
id	integer
name	text

isrc	text

Primary Key: id

sample rows

id	name	isrc
1	Track A	ISRC123456
2	Track B	ISRC789012

---

l_cm_track_cm_artist

Column	Data Type
id	integer
cm_track	integer
cm_artist	integer

Primary Key: id

Foreign Key: cm_track REFERENCES cm_track(id)

Foreign Key: cm_artist REFERENCES cm_artist(id)

sample rows

id	cm_track	cm_artist
1	1	1
2	1	2
3	2	2

---

cm_artist_cache_history

Column	Data Type
id	integer
cm_artist	integer

follower_count	integer
created_at	timestamp

Primary Key: id

Foreign Key: cm_artist REFERENCES cm_artist(id)

sample_rows

id	cm_artist	follower_count	created_at
1	1324	1000	2023-02-01 12:00:00
2	521	1500	2023-01-02 12:00:00
3	232	2000	2023-04-01 12:00:00
4	2	2200	2023-06-01 12:00:00
5	2	2250	2023-06-02 12:00:00
6	521	2000	2023-01-03 12:00:00

---

cm_album

Column	Data Type
id	integer
name	text
release_date	timestamp

Primary Key: id

sample_rows

id	name	release_date
1	Album X	2023-02-01 12:00:00
2	Album Y	2023-05-01 12:00:00

---

l_cm_track_cm_album

Column	Data Type

id	integer
cm_track	integer
cm_album	integer

Primary Key: id

Foreign Key: cm_track REFERENCES cm_track(id)

Foreign Key: cm_album REFERENCES cm_album(id)

sample rows

id	cm_track	cm_album
1	1	1
2	2	1
3	3	2

Questions

1. Write a query to retrieve the artist names along with the total number of tracks each artist has, sorted in descending order of the number of tracks.
2. Write a query to find the artist names along with the total number of tracks each artist has and the release year of their earliest album, sorted by artist name.
3. Implement a query to find the track names that have been associated with multiple artists, along with the names of those artists.

Bonus

1. Write a query to find the artist names along with the average follower count and the running total of follower counts for each artist from the `cm_artist_cache_history` table.

Python

Using the `Dataset.csv` proceed with the following analysis. It is recommended to use Jupyter notebooks and share the results as a HTML or PDF file of the notebook.

Dataset.csv

Basic Data Manipulation

1. Load the dataset into a `pandas` DataFrame.
2. Calculate the summary statistics (mean, median, min, max) for the `YOUTUBE_VIEWS` column.

Data Cleaning

In a separate DataFrame: For every distinct `CM ID` find the corresponding `SPOTIFY_PLAYS`, `SPOTIFY_POPULARITY`, `TIKTOK_TOP_VIDEOS_VIEWS`, and `AIRPLAY_STREAMS`.

1. Remove any duplicate rows from the dataset.
2. Filter out rows where the `SPOTIFY_POPULARITY` is less than 10.

Data Exploration

1. What are the top 5 tracks with the highest `SPOTIFY_PLAYS`?
2. Plot a stacked bar chart for the track with the highest increase in `AIRPLAY_STREAMS` in a months time**.** The stack would include the actual value and the monthly difference percentage value.

Data Aggregation

1. Calculate the total streams for each album. Albums have a unique identifier called `UPC`. After calculating the total streams, find the difference of total streams for each album from the average of each album.
2. Calculate the average `YOUTUBE_VIEWS` for each album.

Statistical Analysis

1. Perform a correlation analysis between `YOUTUBE_VIEWS` and `SPOTIFY_PLAYS`.
2. Create a unique ranking system for the artists to have one rank which is defined by track metrics given in the dataset. The final data should be a table having columns: `ARTIST_NAME`, `RANK` . Please explain your methodology - there is no right answer here.

Looker

Create a Tableau/Looker dashboard that visualizes the key performances of the tracks. You can create different views for each platform specific metric. One of those should have a heat map of the 5 important track metrics which include Spotify, TikTok, Shazam, Airplay/Radio, and YouTube.