Dimensional Data Modeling - Week 1

This week's assignment involves working with the `actor_films` dataset. Your task is to construct a series of SQL queries and table definitions that will allow us to model the actor_films dataset in a way that facilitates efficient analysis. This involves creating new tables, defining data types, and writing queries to populate these tables with data from the actor_films dataset

Dataset Overview
The `actor_films` dataset contains the following fields:

- `actor`: The name of the actor.
- `actorid`: A unique identifier for each actor.
- `film`: The name of the film.
- `year`: The year the film was released.
- `votes`: The number of votes the film received.
- `rating`: The rating of the film.
- `filmid`: A unique identifier for each film.

The primary key for this dataset is (`actor_id`, `film_id`).

Assignment Tasks

1. **DDL for `actors` table:** Create a DDL for an `actors` table with the following fields:
    - `films`: An array of `struct` with the following fields:
                - film: The name of the film.
                - votes: The number of votes the film received.
                - rating: The rating of the film.
                - filmid: A unique identifier for each film.

    - `quality_class`: This field represents an actor's performance quality, determined by the average rating of movies of their most recent year. It's categorized as follows:
                - `star`: Average rating > 8.
                - `good`: Average rating > 7 and ≤ 8.
                - `average`: Average rating > 6 and ≤ 7.
                - `bad`: Average rating ≤ 6.
    - `is_active`: A BOOLEAN field that indicates whether an actor is currently active in the film industry (i.e., making films this year).

2. **Cumulative table generation query:** Write a query that populates the `actors` table one year at a time.

3. **DDL for `actors_history_scd` table:** Create a DDL for an `actors_history_scd` table with the following features:
    - Implements type 2 dimension modeling (i.e., includes `start_date` and `end_date` fields).
    - Tracks `quality_class` and `is_active` status for each actor in

the `actors` table.

4. **Backfill query for `actors_history_scd`:** Write a "backfill" query that can populate the entire `actors_history_scd` table in a single query.

5. **Incremental query for `actors_history_scd`:** Write an "incremental" query that combines the previous year's SCD data with new incoming data from the `actors` table.