



## Grade for homework: Dimensional Data Modeling Homework Submission

bootcamp@techcreator.io <bootcamp@techcreator.io>  
To:

# FINAL GRADE: A

\*\* This feedback is auto-generated from an LLM \*\*

Thank you for submitting your SQL scripts for review. Let's go through the individual tasks:

### Task 01:

1. **Composite Type film\_struct:** Correctly defines a composite type for films.
2. **Enum quality\_class\_enum:** Correctly creates an enum type for quality classes.
3. **Actors Table:** The table is well defined with a composite type array and enum. However, after Task 02, the primary key changes, which is acceptable given the requirements.

### Task 02:

1. **Adding snapshot\_year Column:** Successfully adds the required column to the actors table.
2. **Primary Key Adjustment:** Properly adjusts the primary key to include snapshot\_year.
3. **Data Insertion:** The insertion logic using aggregations and conditions for determining quality\_class handles the transformations correctly, reflecting historical actor data per snapshot year.

### Task 03:

1. **DDL for actors\_history\_scd:** Successfully creates actors\_history\_scd table with a suitable primary key.

### Task 04:

1. **Inserting into actors\_history\_scd:**
  - Proper use of LAG to identify changes in actor data across snapshots.
  - Manages grouping of data periods with change detection using a window function for summation.
  - Accurately computes start and tentative end dates using window functions.
  - Correctly prepares the final SELECT for inserting historical records by leveraging conditional window functions to predict end dates effectively.

### Task 05:

1. **Step 1: Update Changed Rows:** The update operation correctly marks the end\_date for changed historical records.
2. **Step 2: Insert New Rows:**
  - Handles insertion for new records effectively, ensuring that changes are captured.
  - The operations are repeated twice, once with TO\_DATE and once with DATE for start\_date. This is duplicative but doesn't affect the correctness. However, it signals a need for optimization by consolidating into a single operation if no additional logic distinctions were required.

Overall, your implementation of Slowly Changing Dimensions Type 2 (SCD Type 2) logic is done effectively with correct use of SQL features like window functions. The scripts align well with intended outcomes and efficiently process changes across snapshots.

### Feedback Recap:

- Redundancy in Task 05 could be optimized to avoid repetitive operations.
- Ensure clear comments accompany your scripts when duplicated logic is necessary.

### Final Grade:

```
{  
  "letter_grade": "A",
```

```
    "passes": true  
  }
```

Excellent work! Your submission demonstrates a solid understanding of SQL operations necessary for managing historical data capture in a relational database. Keep up the great work!