Transformation to Data Products

- In order to address the project requirements, we built two data products: first_data_product and second_data_product.

Data Product 1

The first_data_product answers two key project questions: “What are the top 10 highest-rated movies for the last 5 years?” and “What are the 3 highest-rated movies in each genre?”

To answer those requirements, we selected essential attributes from multiple tables:

- The avg_rating column was sourced from the title_ratings_dimension table to provide the average rating for each movie.
- The primary_title column was sourced from the titles_dimension table to display the names of the movies.
- The title_start_year column was sourced from the titles_dimension table to filter movies released in the last five years.
- The genre column was sourced from the genres_dimension table to categorize movies into genres.
- Additionally, we utilized the num_votes column from the title_ratings_dimension table to filter movies based on popularity.

The first_data_product simplifies the process of ranking and categorizing movies by combining relevant attributes from multiple tables into one table. It supports efficient querying for the specific needs of the project, and effectively answers to both questions with minimal computational process.

This data product simplifies computation by joining data from multiple tables, such as titles_dimension, title_ratings_dimension, and genres_dimension, into a unified dataset.
It allows streamlined filtering and ranking operations to extract top-rated movies over specific time periods or genres. Moreover, it minimizes the need for repetitive joins during analysis, thereby reducing query complexity and execution time.

dbt model files to create and load the data into your data product(s)

Include Markdown cells with the contents of your dbt model files used to create your data products.

Code:

```
{{ config(
    materialized = 'table',
)}}
```

- This configures the dbt model to be materialized as a physical table

```
SELECT i.avg_rating,
       t.title_start_year,
       g.genre AS genre_name,
       t.primary_title,
       tr.num_votes
```

- This selects the relevant columns for the data product:
    - avg_rating from imdb_ent_facts table
    - title_start_year from the titles_dimension table
    - genre_name from the genres_dimension table
    - primary_title from the titles_dimension table
    - num_votes from the title_ratings_dimension table.

```
FROM   {{ ref('imdb_ent_facts') }} AS i
```

- The data is sourced from the imdb_ent_facts table (aliased as i). This table is the fact table, meaning it contains the primary keys (PKs) of other tables and holds the core data for movie ratings (avg_rating).

```
JOIN   {{ ref('titles_dimension') }} AS t USING (title_id)
JOIN   {{ ref('title_ratings_dimension') }} AS tr USING (title_rating_id)
JOIN   {{ ref('genres_dimension') }} AS g USING (genre_id)
```

- The JOIN clauses link the following tables together based on their shared columns:
    - titles_dimension (aliased as t): Contains information about movie titles and release years, joined on title_id.
    - title_ratings_dimension (aliased as tr): Contains rating data and the number of votes for each movie, joined on title_rating_id.
    - genres_dimension (aliased as g): Contains the genres of each movie, joined on genre_id.

```
WHERE  g.genre IN ('Family', 'Comedy', 'Animation', 'Romance', 'Musical')
```

- This filters the results to only include movies from the specified genres: 'Family', 'Comedy', 'Animation', 'Romance', and 'Musical'.

Data Product 2

The second_data_product answers three key project questions: “Who are the directors with the highest-rated movies?”, “Who are the actors and actresses with the highest-rated movies?”, and “Which movies received the highest number of votes?”

These questions require data related to movie ratings, personnel roles, and popularity metrics. To meet these requirements, we selected essential attributes from multiple tables:

- The avg_rating column was sourced from the title_ratings_dimension table to provide the average rating for each movie.

- The primary_name column was sourced from the person_title_roles_dimension table to display the names of directors, actors, and actresses associated with each movie.
- The person_role column was sourced from the person_title_roles_dimension table to differentiate between directors, actors, and actresses.
- Additionally, the num_votes column was sourced from the title_ratings_dimension table to determine the popularity of movies based on audience engagement.

The second_data_product simplifies the process of identifying top-rated individuals (directors, actors, and actresses) and their associated movies by consolidating relevant attributes from multiple tables into one dataset. It efficiently answers the questions by providing a unified structure for querying ratings, roles, and popularity.

This data product simplifies the computation process by merging data from multiple tables, including title_ratings_dimension, person_title_roles_dimension, and titles_dimension, into a single dataset. It facilitates efficient filtering and ranking to identify the highest-rated directors, actors, and actresses, along with the top-rated movies based on their ratings and number of votes. Additionally, it reduces the need for repetitive joins, which helps minimize query complexity and improve execution time.

```
{{
   config(
     materialized="table"
   )
}}
```

- This configures the dbt model to be materialized as a physical table

```
SELECT i.avg_rating,
       p.person_role,
        p.primary_name,
        tr.num_votes
```

- This selects the relevant columns for the data product:
  - avg_rating from the imdb_ent_facts table
  - person_role from the person_title_roles_dimension table
  - primary_name from the person_title_roles_dimension table
  - num_votes from the title_ratings_dimension table

```
FROM {{ ref('imdb_ent_facts') }} AS i
```

- The data is sourced from the imdb_ent_facts table (aliased as i). This table is the fact table, meaning it contains the primary keys (PKs) of other tables and holds the core data for movie ratings (avg_rating).

```
JOIN {{ ref('title_ratings_dimension') }} AS tr USING (title_rating_id)
JOIN {{ ref('person_title_roles_dimension') }} AS p USING (person_title_role_id)
```

- The JOIN clauses link the following tables together based on their shared columns:
    - person_title_roles_dimension (aliased as p): Contains information on directors, actors, and actresses, joined on person_title_role_id.
    - title_ratings_dimension (aliased as tr): Contains movie ratings and vote counts, joined on title_rating_id.

```
WHERE p.person_role IN ('director', 'actor', 'actress')
```

- This filters the results to include only the relevant roles: Directors, actors, and actresses, which are identified in the "person_role" column