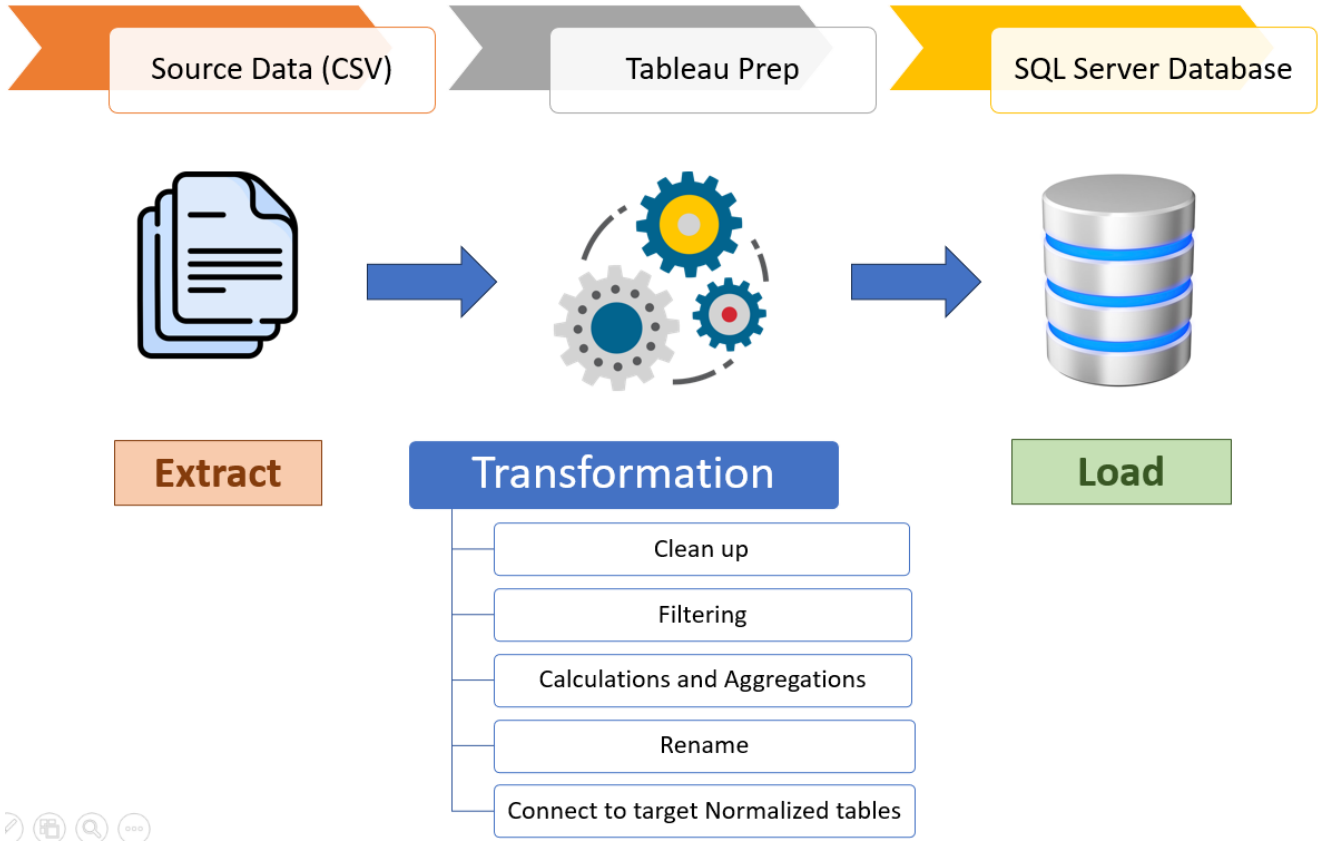
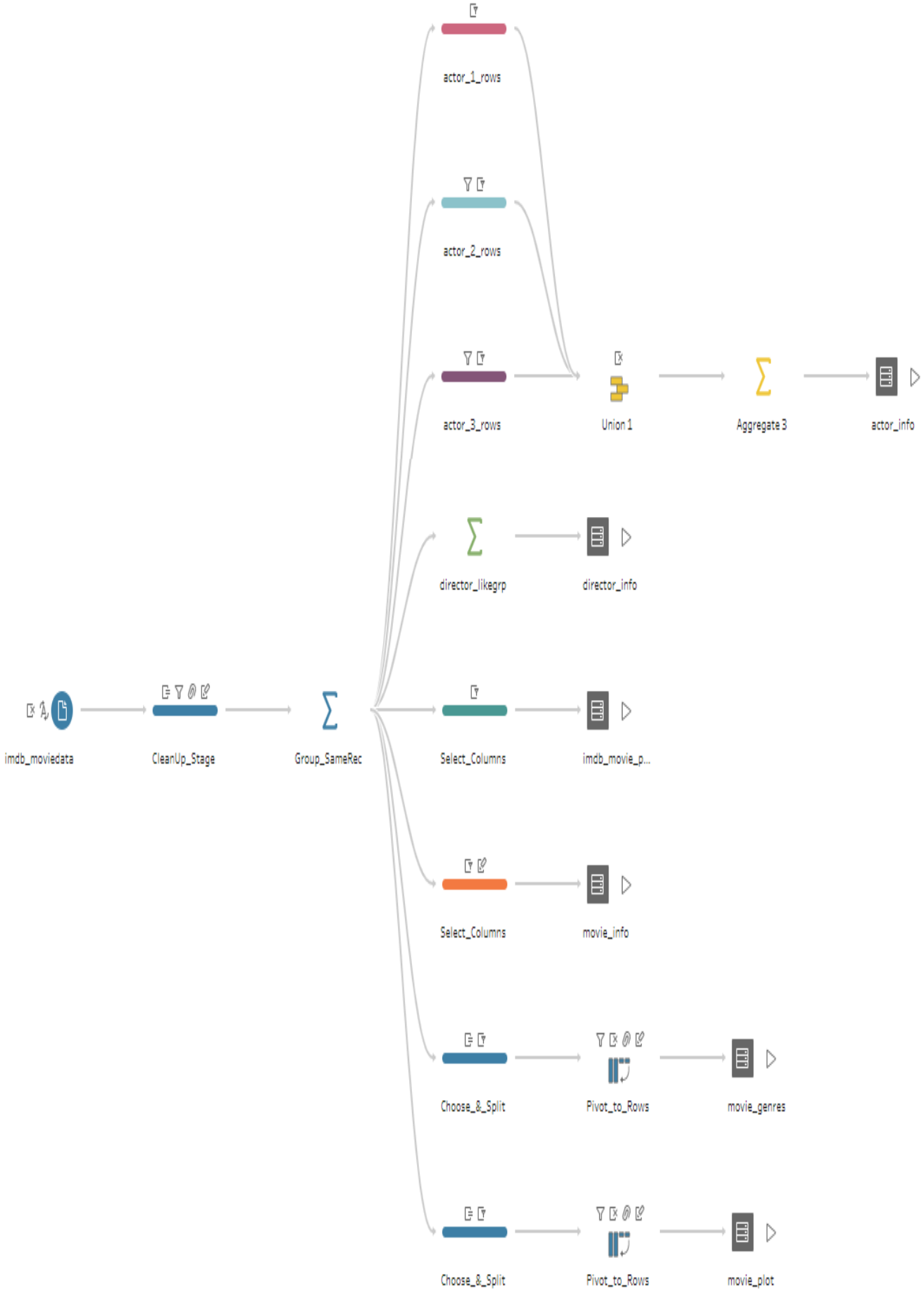
Data Engineering Project
End-to-End Data Modelling ETL and Visualization
Phase-2

Phase-2 for project is performing transformation of source data to normalized data schema that was decided during phase-1. Approach decided for phase-2 is used for project is **ETL (extract, transform, and load)**. Overall architecture used for the project is represented as:



Transformation Tableau Prep Flow:

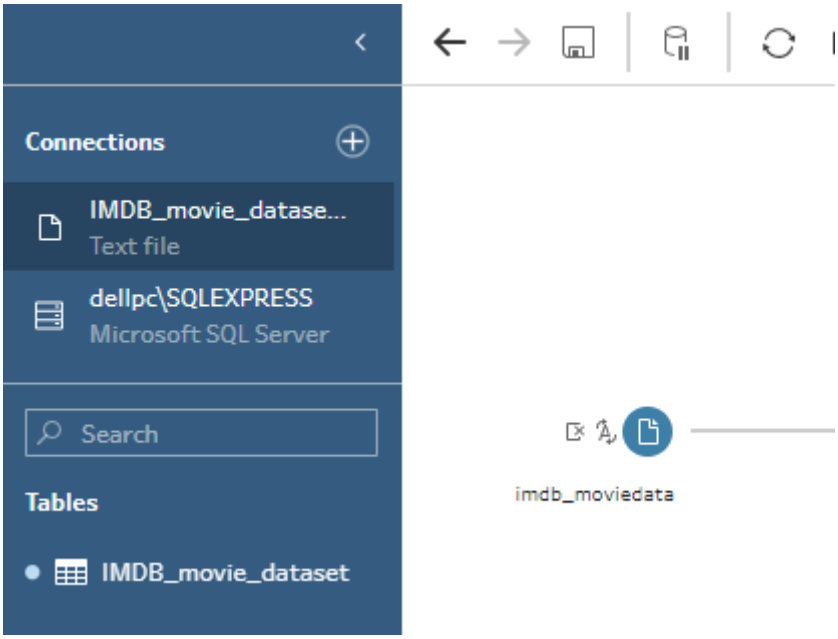


Three stages of process in sequence includes:

1. Extract

Extraction includes connecting to connecting to source of data. In this case, it is csv file that contains the records for movies and characteristics like rating, budget, earning etc. The file is in Zero-NF, and includes dirty or uncleaned data.

Screenshot of source connection:



2. Transform

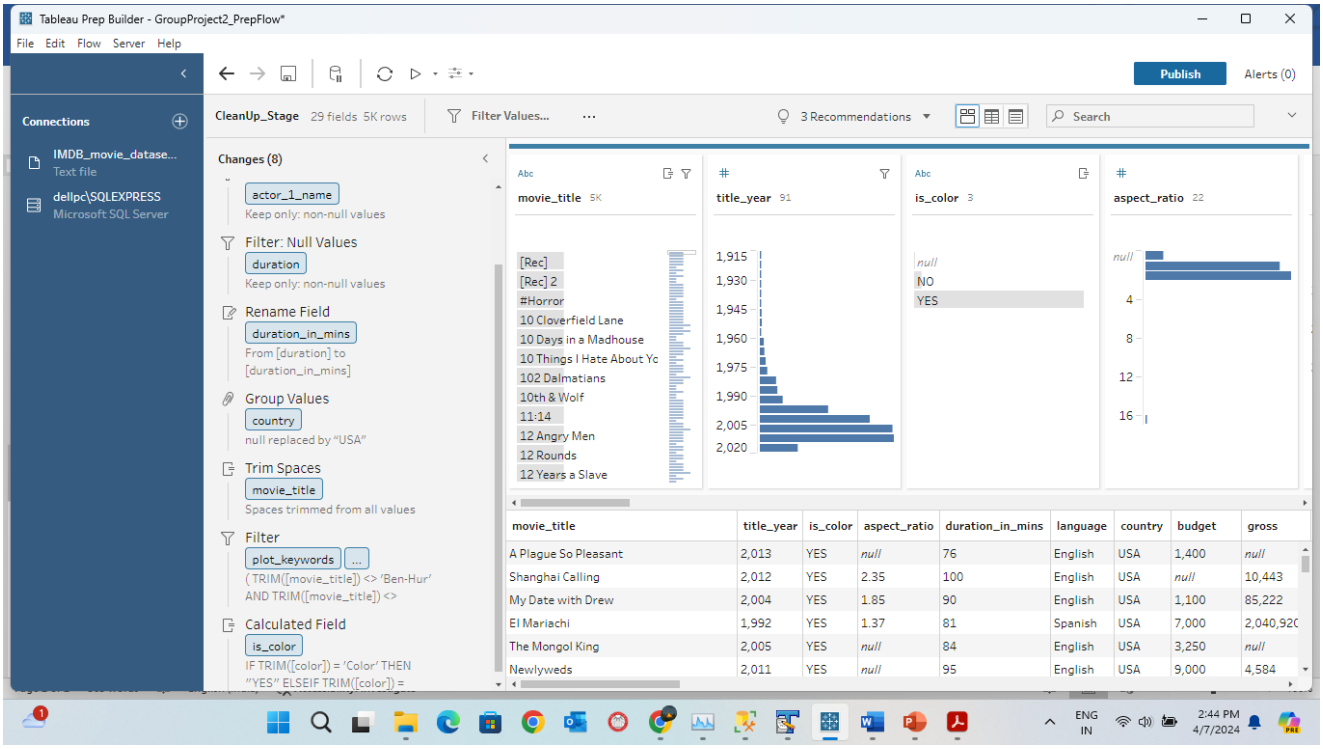
All transformation of data from source to target are performed at this stage. This process will populate the required tables in third normalisation form (3NF) that reside in target database on SQL server. List of all activities performed during transformation process are as follows:

2.1. Cleaning of dataset (CleanUp_Stage) [deliverable 2.2.1 (all parts)]:

Following cleanup activities are performed in CleanUp_Stage:

2.1.1. Filtering null values for title_year, actor_1_name, and duration.
2.1.2. Renaming field name from “duration” to “duration_in_mins”.
2.1.3. Group null value in “country” column to “USA”.
2.1.4. Trim spaces (cleanup) for column “movie_title”.
2.1.5. Filtering duplicate movie records for “Ben-Hur” and “Brothers”.
2.1.6. Calculated field “is_color” from existing column “color”.

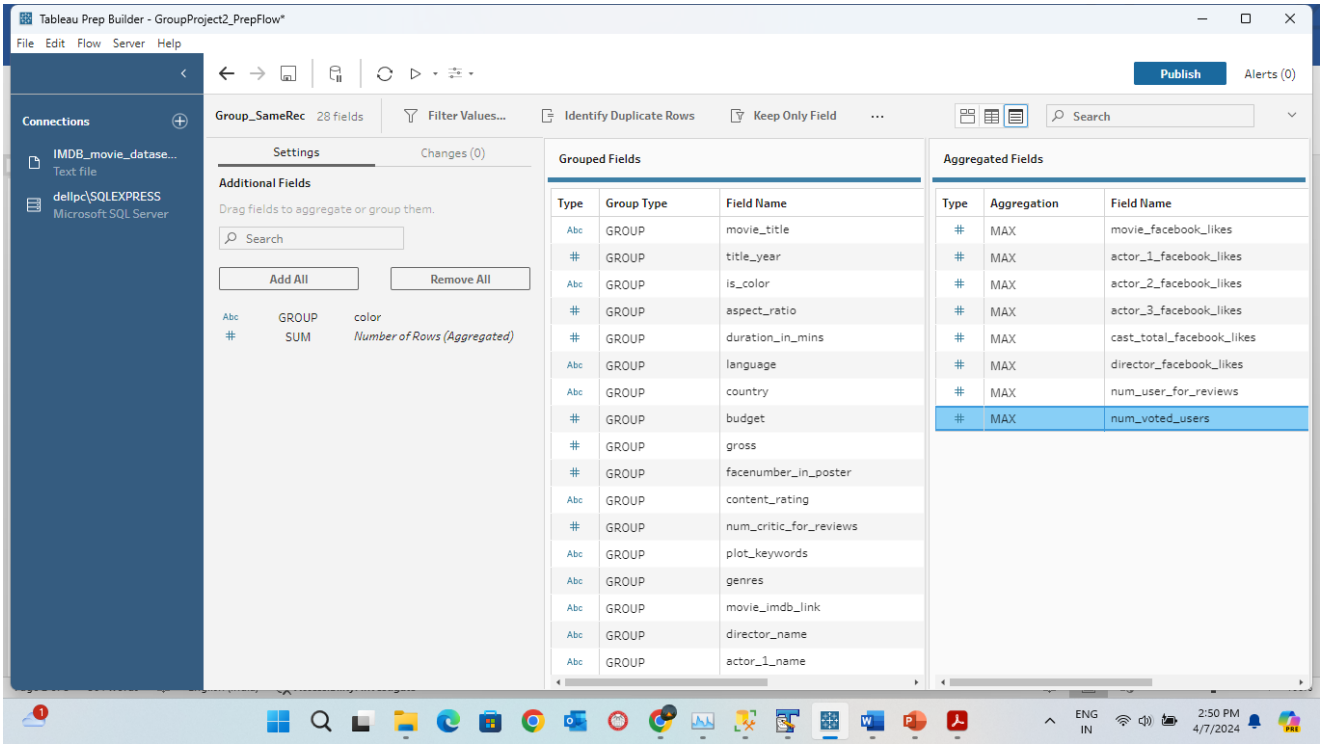
Screenshot:



2.2. Grouping of similar records (Group_SameRec):

In given uncleaned dataset, it is found that there exist almost similar records. Such records have same value for categorical variables like movie, year, actors, country ect. and very minor difference in numerical fields like facebook likes etc. Such records are considered as dirty, and have been removed from dataset. Removing these records will not loss any information.

Screenshot:



2.3. Population of target tables:

2.3.1. Populating table actor_info:

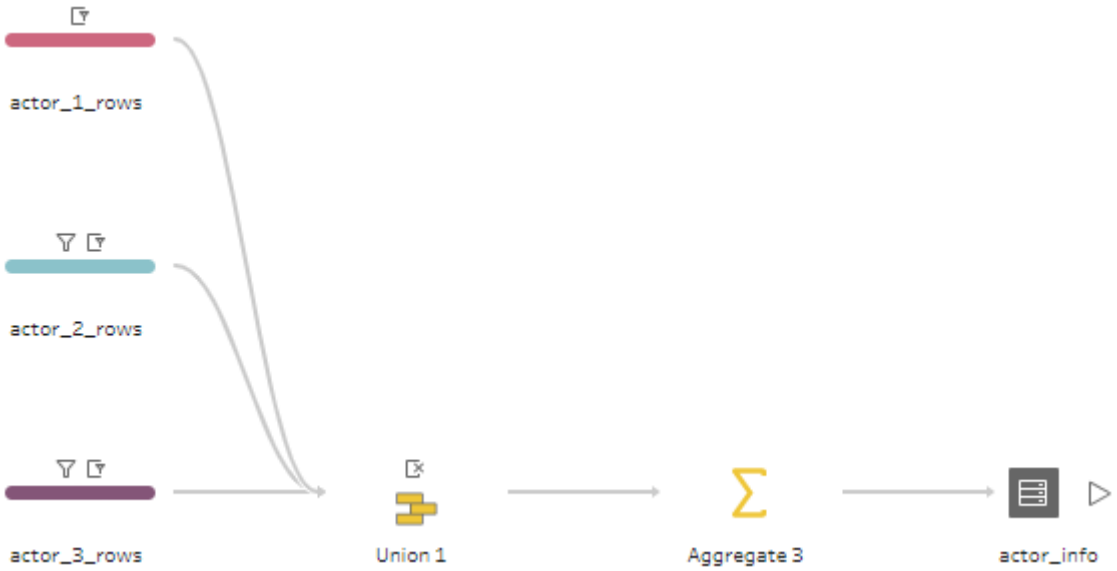
2.3.1.1. Union of actor records from columns associated to actor 1, 2, and 3:

As there are three fields in datasets that are associated with actors, so values in all three fields need to be considered while populating actor_info table. This is because a same person can be actor_1 in one movie, and same can be actor_2 or actor_3 for some another movie. So, made union of three set of actor_info with ignoring null records.

2.3.1.2. Aggregation to match duplicate actors:

This set is required to ensure to deduplicate the actor_info. In event of two records with same name and little difference in facebook likes, maximum value is considered assuming latest one.

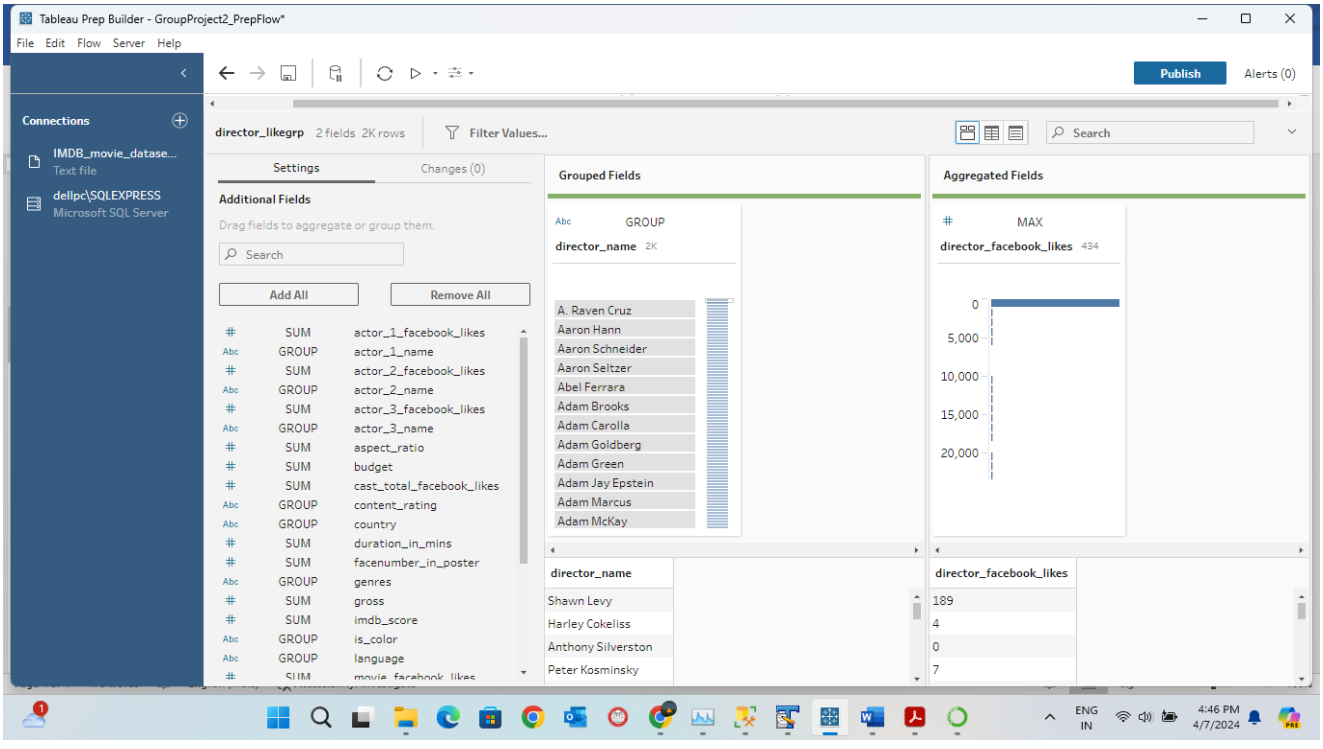
Screenshot of union and merge:



2.3.2. <u>Populating table director_info:</u>

Deduplication is performed by grouping director records in dataset.

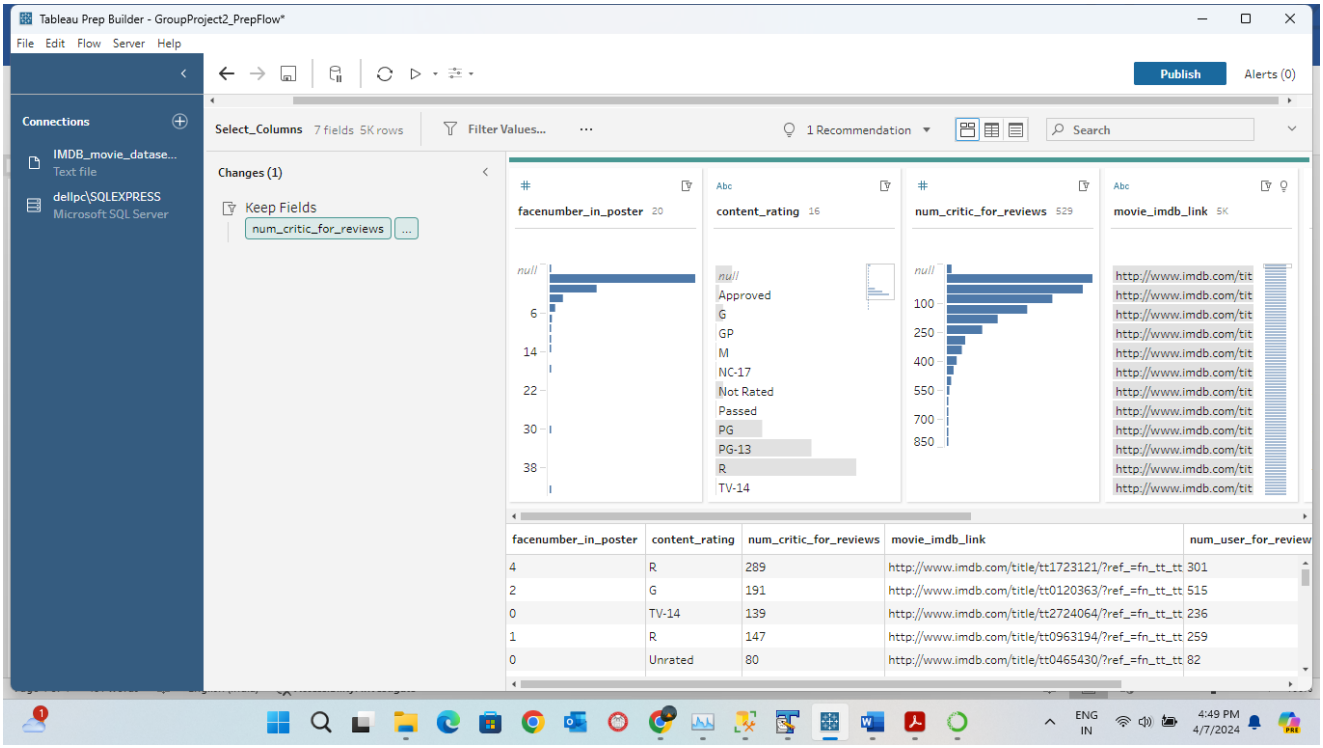
Screenshot:



2.3.3. Populating table imdb_movie_page:

Filter the columns that are required for table imdb_movie_page.

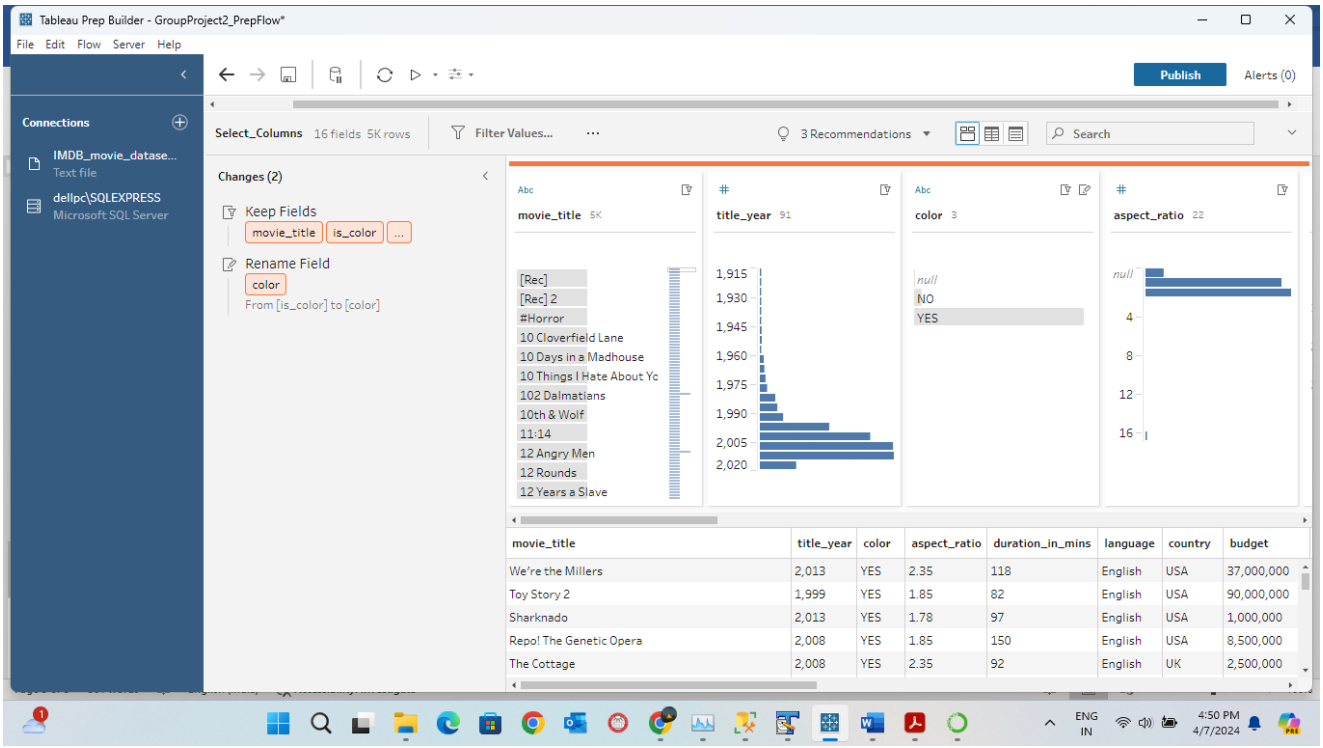
Screenshot:



2.3.4. Populating table movie_info:

Filter the columns that are required for table movie_info. At this stage, we have also, renamed the column “is_color” to “color”.

Screenshot:

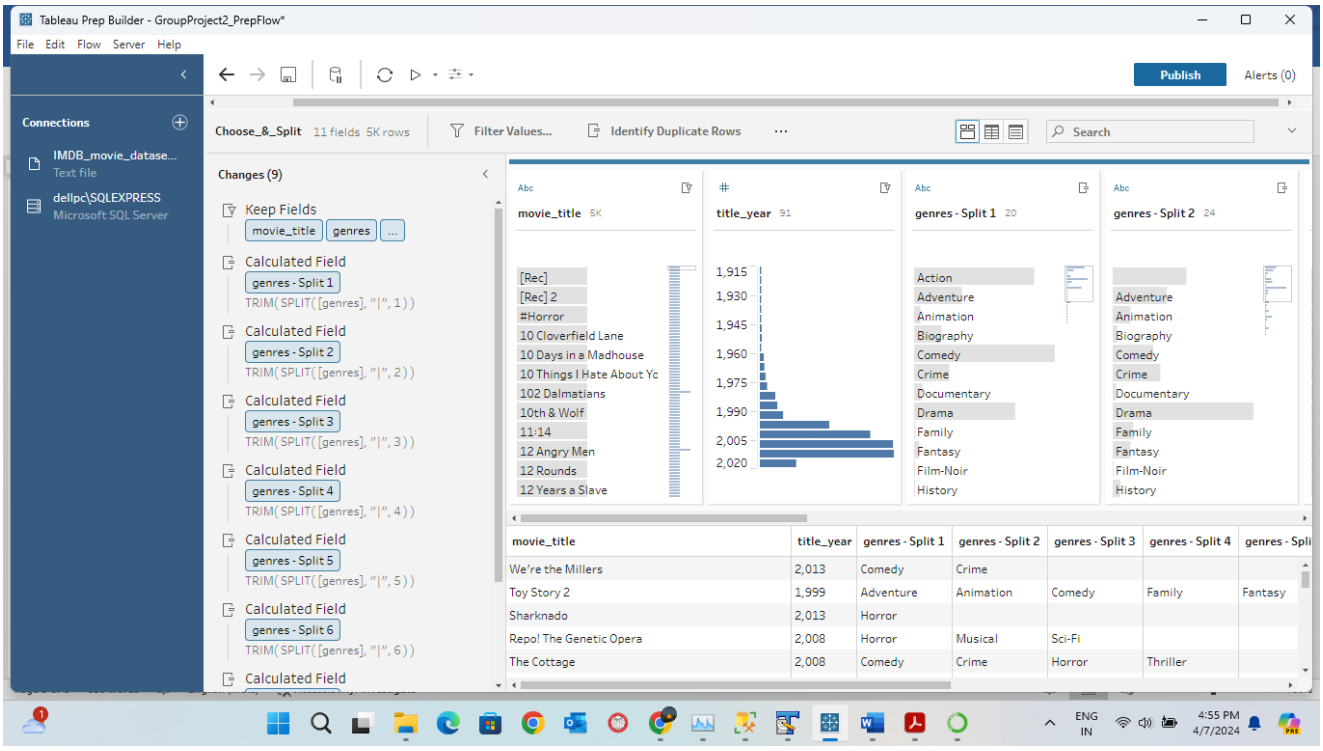


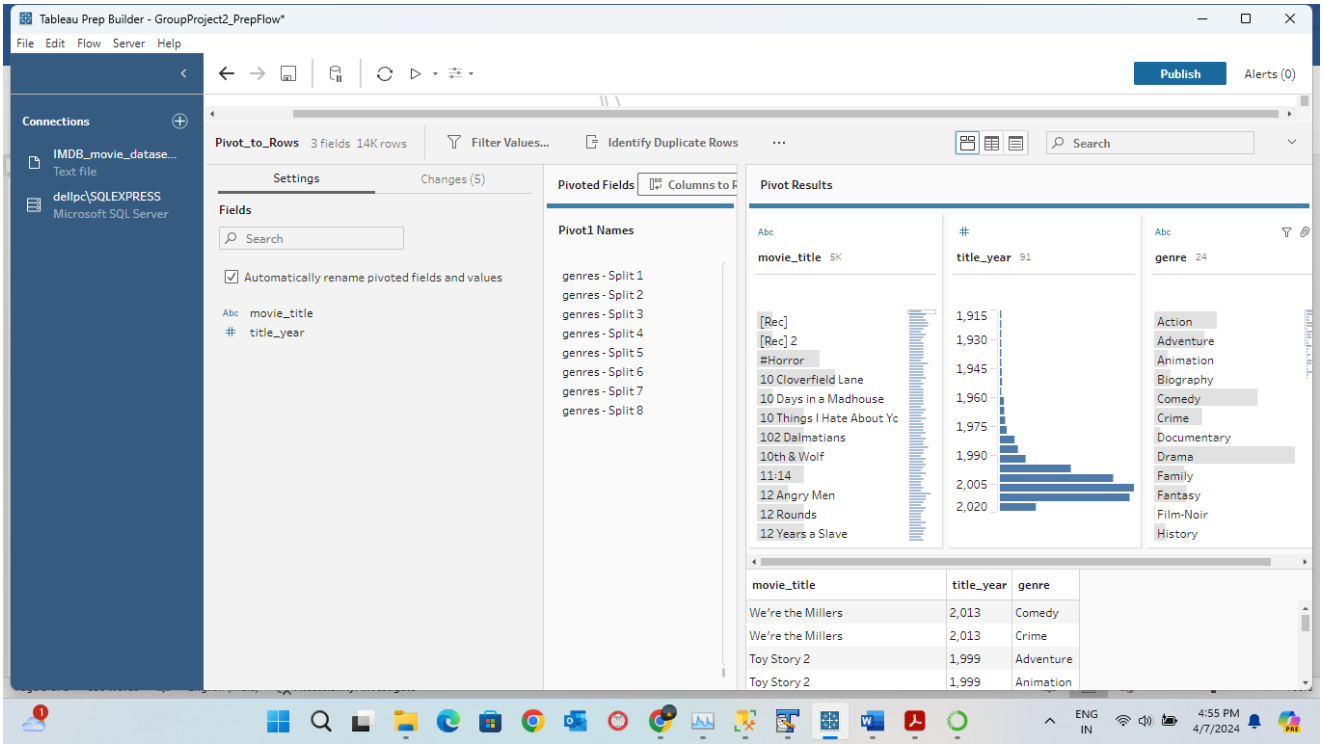
2.3.5. Populating table movie_genres:

As, single movie can have more than one genre, therefore following activities are performed to make dataset suitable for database table population:

- Filtering of required columns only, i.e. movie_title, title_year and genres.
- Split column, genre to multiple columns based on delimiter (|).
- Pivoting the split columns to form rows.

Screenshot:



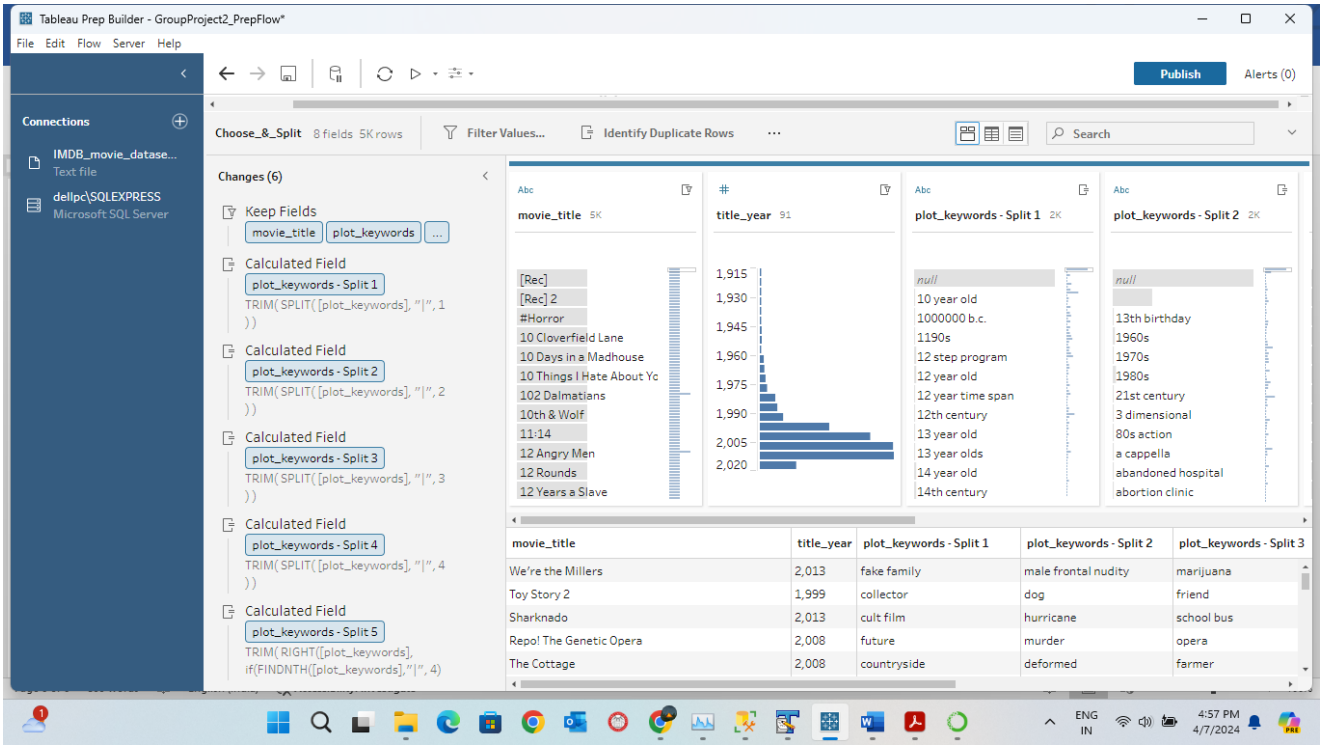


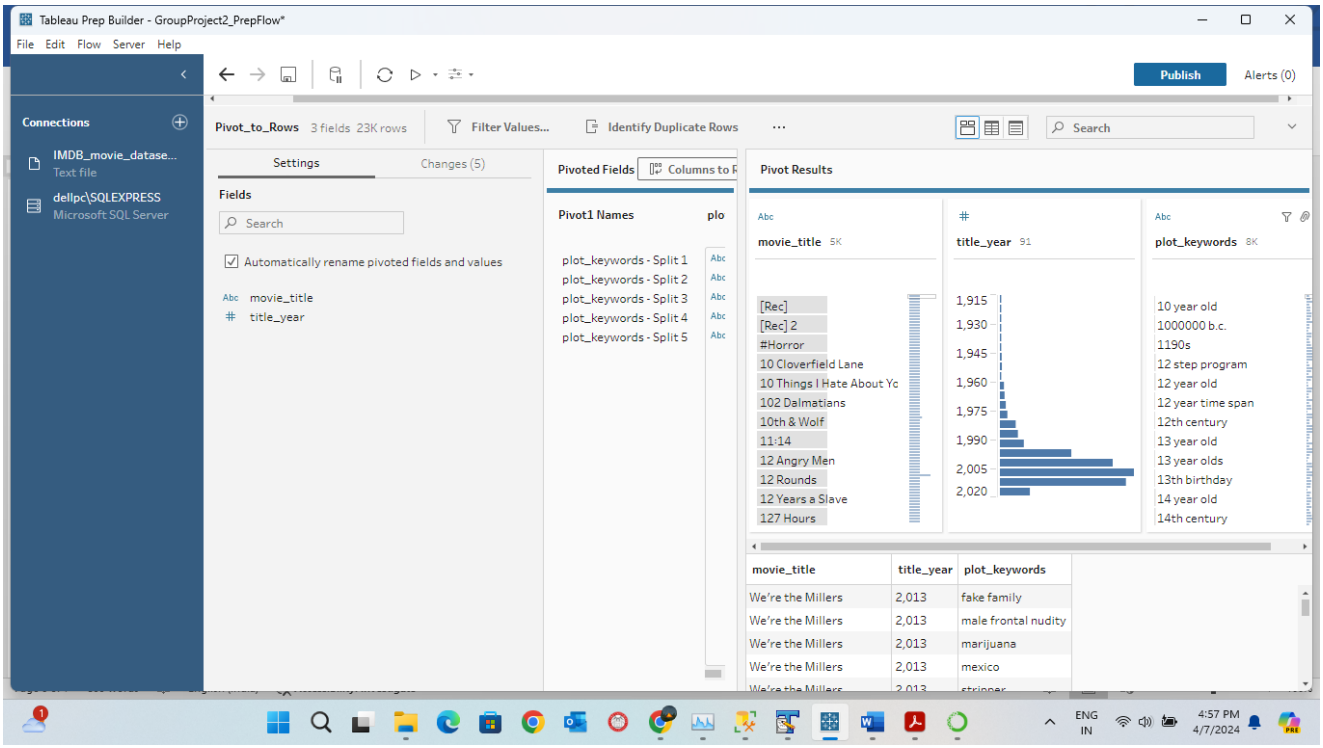
2.3.6. Populating table movie_plot:

As, single movie can have more than one plot, therefore following activities are performed to make dataset suitable for database table population:

- Filtering of required columns only, i.e. movie_title, title_year and plot_keywords.
- Excluding any records with null plot_keywords.
- Split column, genre to multiple columns based on delimiter (|).
- Pivoting the split columns to form rows.

Screenshot:

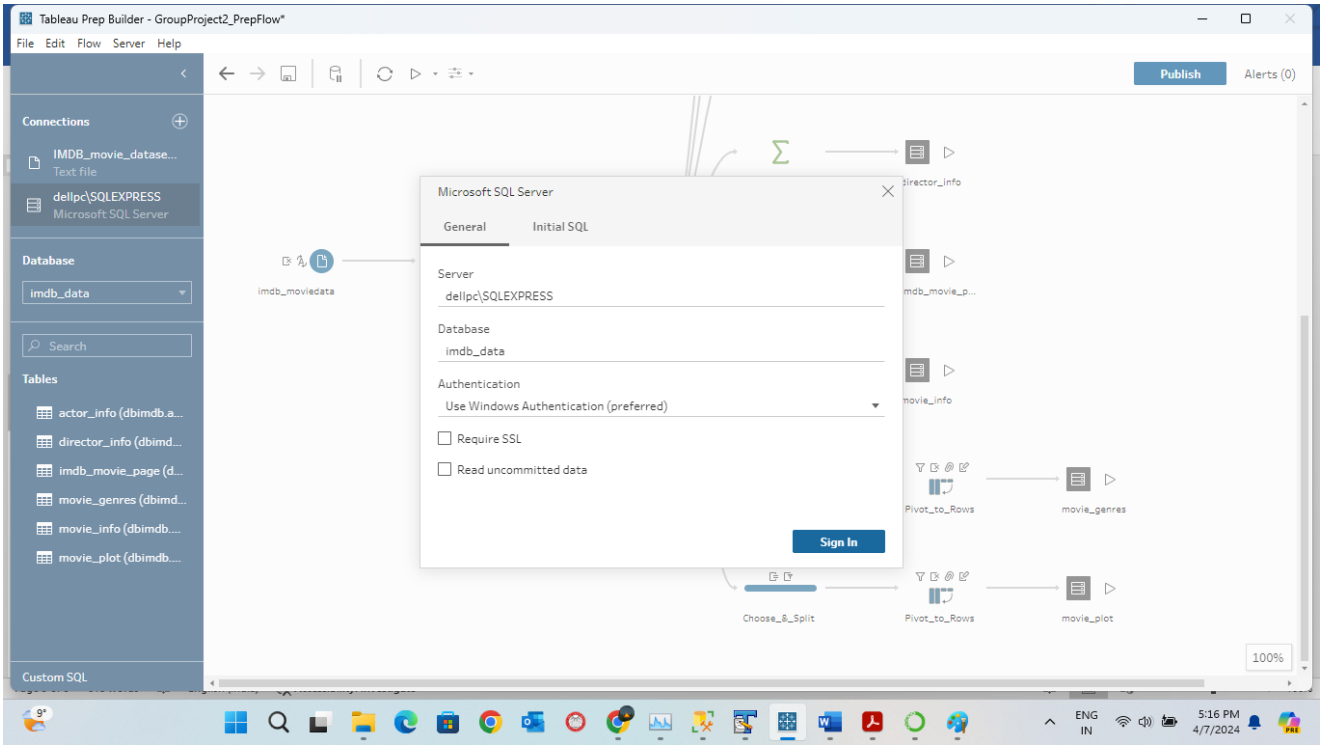




3. Load

Load step involve populating the target tables directly. No changes or transformation will be performed at target system.

Screenshot of target connection:



Important Information:

- No change in DDLs or any table structure is required as submitted in Phase-1.
- **Execution of Tableau Prep Flow:**
  All the tables are required to be populated in below order to maintain referential integrity constraints in database design:
  - actor_info
  - director_info
  - imdb_movie_page
  - movie_info
  - movie_genres
  - movie_plot
- **Clearing dataset in SQL Server before re-execution of Tableau Prep Flow:**
  This step is important in case there is need to re-populate the tables through Tableau Prep. This is because of integrity constraints (foreign key) are applicable on database. Following script consisting of commands need to be executed on SQL server database in sequence:

```
USE imdb_data;
DELETE      FROM  dbimdb.movie_plot;
DELETE      FROM  dbimdb.movie_genres;
DELETE      FROM  dbimdb.movie_info;
DELETE      FROM  dbimdb.imdb_movie_page;
DELETE      FROM  dbimdb.director_info;
DELETE      FROM  dbimdb.actor_info;
```

Information of number of records [Deliverable 2.2.2]

Source	After cleanup (through Tableau Prep.)	Target Normalized Tables		
		SN.	Table Name	Records
5,043 records [IMDB_movie_dataset.csv]	4,794 records [Filtering null records, deduplicate records]	1	actor_info	6,105
		2	director_info	2,381
		3	imdb_movie_page	4,794
		4	movie_info	4,794
		5	movie_genres	13,783
		6	movie_plot	22,998

Screenshot of target connection: