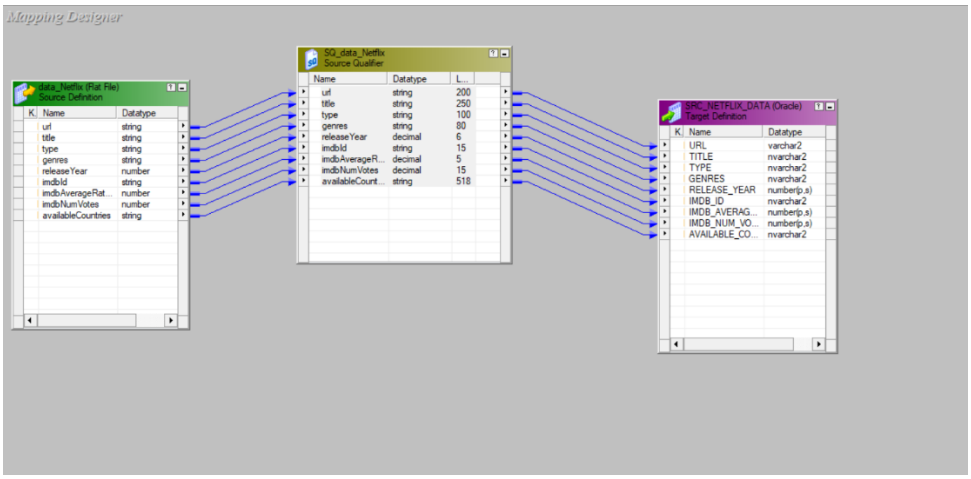
Project Documentation: Netflix Data Processing Pipeline

Project Overview

This project implements an ETL (Extract, Transform, Load) pipeline for cleaning and managing Netflix data using Informatica PowerCenter, Oracle SQL, and Python. The process loads raw data from a CSV file, applies necessary transformations, and stores the cleaned data in a target table while supporting SCD Type 1 updates for incremental changes.

1. Data Extraction and Loading (Informatica PowerCenter)

- **Source Data**: The raw dataset was downloaded from Kaggle in CSV format, containing information on Netflix movies and shows.
- **Data Transfer**: Using Informatica PowerCenter, we successfully loaded all 18,860 records from the CSV file into a staging table, `src_netflix_data`, in an Oracle database.



- Sql query to check count of records in the `src_netflix_data` after loading:

```
SELECT COUNT(*) as count_src_netflix FROM src_netflix_data
```

	COUNT_SRC_NETFLIX
1	15860

2. Data Transformation (Python)

After loading the raw data into the staging table, a Python script was executed to perform the following transformations:

- **Null Removal**: Removed any rows with null values in critical columns.
- **Data Type Conversion**: Converted data types to align with Oracle table specifications
- **String Formatting**: Standardized string fields for consistent casing and formatting.
- **Data Ordering**: Ordered the dataset based on specified columns to prepare it for analysis and reporting.
- **Resulting Data**:
    - Post-transformation, the dataset was reduced to 15,134 records, ready for loading into the target table.

3. Data Loading to Target Table

- **Target Table**: The cleaned data was loaded into the `tgt_netflix_cleaned_data` table in Oracle, ensuring it was ready for insert and update operations.
- **Success Check**: The data transfer to `tgt_netflix_cleaned_data` was successful, with all records appearing as expected.

- Sql query to show count of records in the `tgt_netflix_cleaned_data` after executing python code:

```
select count(*) as count_tgt_netflix from tgt_netflix_cleaned_data
```

	COUNT_TGT_NETFLIX
1	15134

4. Testing the ETL Pipeline

- **Insert Test**:
    - A test row was inserted into the `src_netflix_data` table with the following values:

+51	null)	Interstellar	movie	Science Fiction	2010	ab123456	5.6	1548796584	us

    - **Result**: After running the ETL code, the new row appeared in `tgt_netflix_cleaned_data`, confirming the insert functionality.

```
select * from tgt_netflix_cleaned_data where title =
'Interstellar' and genres = 'Science Fiction'
```

	URL	TITLE	TYPE	GENRES	RELEASE_YEAR	IMDB_ID	IMDB_AVERAGE_RATING	IMDB_NUM_VOTES	AVAILABLE_COUNTRIES
1	Unknown	Interstellar	Movie	Science Fiction	2010	ab123456	5.6	1548796584	US

`select count(*) as count_tgt_netflix from tgt_netflix_cleaned_data`, the number of records in the target table increased by one cause of inserted row

	COUNT_TGT_NETFLIX
1	15135

- **Update Test**:
  - We updated the test row in `src_netflix_data` to verify if updates were reflected in the target table:

```
UPDATE src_netflix_data
SET release_year = 2014,
    imdb_average_rating = 9.8
WHERE imdb_id = 'ab123456';
```

  - **Result**: Running the code successfully updated `tgt_netflix_cleaned_data`, confirming that SCD Type 1 functionality for updates is working as intended.

select * from tgt_netflix_cleaned_data where title = 'Interstellar' and genres = 'Science Fiction'

	URL	TITLE	TYPE	GENRES	RELEASE_YEAR	IMDB_ID	IMDB_AVERAGE_RATING	IMDB_NUM_VOTES	AVAILABLE_COUNTRIES
1	Unknown	Interstellar	Movie	Science Fiction	2014	ab123456	9.8	1548796584	US

5. Conclusion

This ETL process demonstrates a successful, reliable pipeline for loading, transforming, and maintaining Netflix data. It supports dynamic updates (SCD Type 1) in the target table, making it well-suited for data warehousing and business intelligence applications.

GitHub Repository

For further details and code, visit the GitHub repository: [https://github.com/SherifElshafeyy/Netflix-Data-Pipline]