CSV to SQL Data Pipeline with Reporting Using Python & SQL

Objective:

To develop an end-to-end ETL data pipeline using Python and MySQL to clean a Netflix CSV dataset, load it into a relational database, and generate insightful reports through SQL queries.

Tools & Technologies:

- Programming Language: Python 3.11
- Libraries: pandas, MySQL-connector-python
- Database: MySQL
- IDE: Visual Studio Code
- Version Control: Git & GitHub.

Dataset Description:

- **Source:** netflix_titles.csv
- **Fields Include:** showid, type (Movie or TV Show), title, director, country, date added, release year, rating, duration, listed in (genre), description.
- **Size:** 6,000+ records.

ETL Process:

- **Extract:** Read the Netflix dataset from a CSV file using Python and the pandas library.
- **Transform:** Cleaned the data by removing incomplete rows, filling in missing values, and organizing the columns to fit the database structure.
- **Load:** Connected to a MySQL database, created a table with the right format, and inserted all the cleaned data for easy querying and analysis.

SQL Reporting & Analysis

1. **Directors who created both Movies and TV Shows**

```
SELECT director

FROM netflix_titles

WHERE director != ''

GROUP BY director

HAVING COUNT(DISTINCT type) > 1;
```

2. **Country with the most Comedy Movies**

```
SELECT country, COUNT(*) AS comedy_count

FROM netflix_titles

WHERE listed_in LIKE '%Comedy%' AND type = 'Movie'

GROUP BY country

ORDER BY comedy_count DESC

LIMIT 1;
```

3. **Top Director Each Year**

```
SELECT release_year, director, COUNT(*) AS total

FROM netflix_titles

WHERE director != ''

GROUP BY release_year, director

ORDER BY release_year, total DESC;
```

4. **Average Movie Duration by Genre**

```
SELECT listed_in,

    AVG(CAST(SUBSTRING_INDEX(duration, ' ', 1) AS UNSIGNED)) AS avg_duration

FROM netflix_titles

WHERE type = 'Movie' AND duration LIKE '%min%'

GROUP BY listed_in;
```

5. **Directors who made both Comedy & Horror**

```
SELECT director

FROM netflix_titles

WHERE director != '' AND (listed_in LIKE '%Comedy%' OR listed_in LIKE '%Horror%')

GROUP BY director

HAVING SUM(listed_in LIKE '%Comedy%') > 0

  AND SUM(listed_in LIKE '%Horror%') > 0;
```