Netflix Data Cleaning and Analysis

Tools Used: SQL & Python
Objective: Clean the Netflix dataset, handle inconsistencies, and extract insights for better content understanding and decision-making.

1. Project Objective

To clean and normalize the Netflix dataset, resolve inconsistencies, and perform insightful analysis to support content-based decisions.

---

2. SQL-Python Integration

- Connected SQL Server with Python to handle raw data efficiently.
- Optimized data types in the `netflix_raw` table:
    - Changed `varchar` to `nvarchar` for handling foreign characters.
    - Limited maximum length to improve performance.

---

3. Data Cleaning in SQL

Duplicate Handling

- Checked for duplicates based on `title` and `type`.
- Removed 3 exact duplicates using `ROW_NUMBER()`.
- Final count: **8807 ➔ 8804** rows.

Table Normalization

Split multi-valued columns into separate tables using `STRING_SPLIT()`:

- `Netflix_Directors`
- `Netflix_Cast`
- `Netflix_Country`
- `Netflix_Genre`

Missing Values

- **Country:** Populated missing values using existing director-country relationships.
- **Duration:** Corrected misplaced values from `rating` to `duration` column (3 records).
- Final table created as `Netflix`.

📊 4. Data Analysis & Insights

📌 Q1. Directors who created both Movies & TV Shows

NETFLIX_SQL_PROJ...UMAN\91998 (58))*

```sql
/*Q.1 For each director count the no.of.movies & Tv showa Created by thtm in seperate columns
for direcctor who have created tv shows and movies both */
SELECT ND.director, COUNT( DISTINCT N.TYPE) AS DIST_TYPE
FROM NETFLIX N
INNER JOIN Netflix_Directors ND
ON N.show_id = ND.show_id
GROUP BY ND.director
HAVING COUNT( DISTINCT N.TYPE)>1
ORDER BY DIST_TYPE DESC
```

121 %

Results | Messages

	director	DIST_TYPE
1	Abhishek Chaubey	2
2	Alastair Fothergill	2
3	Alban Teurlai	2
4	Alessandro Angulo	2
5	Andrew Tan	2
6	Anurag Kashyap	2
7	B. V. Nandini Reddy	2
8	BB Sasore	2
9	Bejoy Nambiar	2
10	Billy Corben	2
11	Brad Anderson	2
12	Bunmi Ajakaiye	2
13	Cosima Spender	2
14	Dan Forrer	2
15	Daniel Kontur	2
16	David Ayer	2
17	Elaine McMillion Sheldon	2
18	Eli Roth	2
19	Estela Renner	2

- 83 directors identified.
- Used conditional aggregation to get counts:
    - `NO_OF_MOVIES`
    - `NO_OF_TVSHOWS`

📌 Q2. Country with Highest Number of Comedy Movies

NETFLIX_SQL_PROJ...UMAN\91998 (58))*

```sql
-- Q.2 WHICH COUNTRY HAS HIGHEST NUMBER OF COMEDY MOVIES

SELECT TOP 1 NC.Country, COUNT (DISTINCT NG.SHOW_ID) AS NO_OF_MOVIES
FROM NETFLIX_GENRE NG
INNER JOIN Netflix_Country NC ON NC.show_id = NG.show_id
INNER JOIN Netflix N ON N.show_id = NG.show_id
WHERE NG.GENRE='Comedies' AND N.TYPE = 'MOVIE'
GROUP BY NC.Country
ORDER BY NO_OF_MOVIES DESC
```

121 %

Results | Messages

	Country	NO_OF_MOVIES
1	United States	685

- Filtered by `genre = 'Comedies'` and `type = 'Movie'`
- Top contributing country identified.

📌 Q3. Most Active Directors by Year

NETFLIX_SQL_PROJ...UMAN\91998 (58))*

```sql
--Q.3 FOR EACH YEAR (AS PER ADDED TO NETFLIX), WHICH DIRECTOR HAS MAX NO.OF.MOVIES RELEASED

WITH CTE AS (
SELECT ND.director,YEAR(date_added) AS YEAR,
COUNT(DISTINCT N.show_id) AS NO_OF_MOVIES
FROM Netflix_Directors ND
INNER JOIN NETFLIX N ON N.show_id=ND.show_id
WHERE TYPE = 'MOVIE'
GROUP BY ND.director, YEAR(DATE_ADDED)
),
CTE2 AS (
SELECT *,
ROW_NUMBER() OVER (PARTITION BY YEAR ORDER BY NO_OF_MOVIES DESC, DIRECTOR) AS RN
FROM CTE
--ORDER BY YEAR, NO_OF_MOVIES DESC
)
SELECT * FROM CTE2 WHERE RN = 1
```

121 %

Results | Messages

	director	YEAR	NO_OF_MOVIES	RN
1	Sorin Dan Mihalcescu	2008	1	1
2	Joe Dante	2009	1	1
3	Jim Monaco	2010	1	1
4	Arthur Allan Seidelman	2011	1	1
5	Constance Marks	2012	1	1
6	Dave Higby	2013	1	1
7	Leo Riley	2014	2	1
8	Jay Karas	2015	2	1
9	Jan Suter	2016	4	1
10	Jay Chapman	2017	7	1
11	Jan Suter	2018	12	1
12	Cathy Garcia-Molina	2019	7	1
13	Youssef Chahine	2020	11	1
14	Rajiv Chilaka	2021	17	1

- Used `YEAR(date_added)`
- Applied `ROW_NUMBER()` to rank top directors per year.

📌 Q4. Average Duration by Genre

NETFLIX_SQL_PROJ...UMAN\91998 (58))*

```sql
----Q.4 WHAT IS THE AVG DURATION OF MOVIES IN EACH GENRE

-- DURATION COL IS A VARCHAR D TYPE AND IT CONSISTS 'MIN' WE HAVE TO FIND AVG DURATION
SELECT NG.GENRE, AVG(CAST(REPLACE(duration, ' MIN','')AS INT)) AS AVG_DURATION
FROM NETFLIX N
INNER JOIN Netflix_genre NG ON NG.show_id = N.show_id
WHERE TYPE = 'MOVIE'
GROUP BY NG.genre
```

121 %

Results | Messages

	GENRE	AVG_DURATION
1	Documentaries	81
2	Sports Movies	94
3	Faith & Spirituality	105
4	Horror Movies	98
5	Classic Movies	118
6	Comedies	103
7	LGBTQ Movies	94
8	Sci-Fi & Fantasy	106
9	Thrillers	107
10	Music & Musicals	106
11	Action & Adventure	113
12	International Movies	110
13	Movies	47
14	Stand-Up Comedy	67
15	Dramas	113
16	Romantic Movies	110
17	Cult Movies	104

Query executed successfully.

- Cleaned duration field (removed 'min' and cast to INT).
- Calculated average duration grouped by genre.

📌 Q5. Directors Who Made Both Horror & Comedy Movies

NETFLIX_SQL_PROJ...UMAN\91998 (58))*

```sql
/*  Q.5 FIND THE LIST OF DIRECTORS WHO HAVE CREATED HORROR & COMEDY MOVIES
BOTH. DISPLAY DIRECTORS NAMES ALOMG WITH NO_OF_COMEDY AND HORROR MOVIES DIRECTED
BY GENRE */

SELECT ND.director,
COUNT(DISTINCT CASE WHEN NG.GENRE = 'COMEDIES' THEN N.SHOW_ID END) AS NO_OF_COMEDY,
COUNT(DISTINCT CASE WHEN NG.GENRE = 'HORROR MOVIES' THEN N.SHOW_ID END) AS NO_OF_HORROR
FROM NETFLIX N
INNER JOIN Netflix_GENRE NG ON N.show_id=NG.show_id
INNER JOIN Netflix_Directors ND ON ND.show_id=N.show_id
WHERE TYPE = 'MOVIE' AND NG.genre IN ('Comedies','Horror Movies')
GROUP BY ND.director
HAVING COUNT(DISTINCT NG.GENRE)=2
```

121 %

Results | Messages

	director	NO_OF_COMEDY	NO_OF_HORROR
1	Adam Egypt Mortimer	1	1
2	Ahmed Zein	2	2
3	Álex de la Iglesia	2	1
4	Amar Kaushik	1	1
5	Anggy Umbara	1	1
6	Anthony Scott Burns	1	2
7	Anurag Kashyap	1	1
8	B.N. Shajeer Sha	1	1
9	Banjong Pisanthanakun	1	3
10	Brad Peyton	1	1
11	Brent Maddock	1	1
12	Dennis Widmyer	1	1
13	Dibakar Banerjee	3	1
14	Don Michael Paul	3	3

Query executed successfully.

- Identified using `genre IN ('Comedies', 'Horror Movies')`
- Listed directors with counts of both genres.

📅 5. Final Outcome

- Cleaned and normalized dataset ready for advanced analytics.
- Multiple business questions answered through SQL analysis.
- Project showcased data wrangling, transformation, and insight generation.