```sql
-- ##################################################################
-- ################### Netflix Data Preprocessing ###################
-- ##################################################################
-- ####################################################
-- # 1. Drop and Create the Database
-- ####################################################
-- Drop the database if it already exists to start fresh
DROP DATABASE IF EXISTS Netflix_pre_processing;
-- Create a new database
CREATE DATABASE Netflix_pre_processing;
-- Use the newly created database
USE Netflix_pre_processing;
-- ? Explanation: This ensures that we start with a fresh database by dropping any existing version and creating a new Netflix analysis database.
-- ####################################################
-- # 2. Rename the Table
-- ####################################################
-- Rename the raw dataset table for consistency
ALTER TABLE `netflix dataset (raw)` RENAME TO Netflix2021;
-- ? Explanation: Renames the raw dataset to Netflix2021 for better readability and easier reference.
-- ####################################################
-- # 3. Check for Duplicate `show_id`
-- ####################################################
-- Identify duplicate show_id values
SELECT show_id, COUNT(*)
FROM Netflix2021
GROUP BY show_id
HAVING COUNT(*) > 1; -- Shows duplicates only
-- ? Explanation: This checks for duplicate entries in the show_id column to avoid redundant data.
-- ####################################################
-- # 4. Remove Duplicates Using Window Function
-- ####################################################
-- Drop the temporary table if it exists
DROP TABLE IF EXISTS Netflix2021_temp;
-- Create a temporary table with unique rows using ROW_NUMBER()
CREATE TABLE Netflix2021_temp AS
SELECT * FROM (
SELECT *, ROW_NUMBER() OVER (PARTITION BY title, director, country, release_date ORDER BY show_id) AS row_num
FROM Netflix2021
) AS temp
WHERE row_num = 1;
-- Replace original table with the cleaned table
DROP TABLE Netflix2021;
ALTER TABLE Netflix2021_temp RENAME TO Netflix2021;
-- ? Explanation:
-- Uses ROW_NUMBER() to assign a unique number to each duplicate group based on (title, director, country, release_date).
-- Keeps only the first occurrence (row_num = 1).
-- Removes all duplicate rows and replaces the old dataset with the cleaned version.
-- ####################################################
-- # 5. Check for NULL Values
-- ####################################################
-- Count NULL values in each column
SELECT
SUM(CASE WHEN show_id IS NULL THEN 1 ELSE 0 END) AS showid_nulls,
SUM(CASE WHEN `type` IS NULL THEN 1 ELSE 0 END) AS type_nulls,
SUM(CASE WHEN title IS NULL THEN 1 ELSE 0 END) AS title_nulls,
SUM(CASE WHEN director IS NULL THEN 1 ELSE 0 END) AS director_nulls,
SUM(CASE WHEN cast_members IS NULL THEN 1 ELSE 0 END) AS cast_members_nulls,
SUM(CASE WHEN country IS NULL THEN 1 ELSE 0 END) AS country_nulls,
SUM(CASE WHEN Release_Date IS NULL THEN 1 ELSE 0 END) AS Release_Date_nulls,
SUM(CASE WHEN rating IS NULL THEN 1 ELSE 0 END) AS rating_nulls,
SUM(CASE WHEN duration IS NULL THEN 1 ELSE 0 END) AS duration_nulls,
SUM(CASE WHEN `Description` IS NULL THEN 1 ELSE 0 END) AS Description_nulls
FROM Netflix2021;
-- ? Explanation: Counts NULL values in each column to identify missing data that needs to be handled.
-- ####################################################
-- # 6. Populate NULL `director` Values Using `cast_members`
-- ####################################################
-- Update missing director values based on cast members
UPDATE Netflix2021 AS n1
JOIN (
SELECT cast_members, MAX(director) AS director
FROM Netflix2021
WHERE director IS NOT NULL
GROUP BY cast_members
) AS n2
ON n1.cast_members = n2.cast_members
SET n1.director = n2.director
WHERE n1.director IS NULL;
-- ? Explanation: Finds existing director information for movies/shows with the same cast_members and fills missing values.
-- ####################################################
-- # 7. Fill Remaining NULL `director` Values
-- ####################################################
UPDATE Netflix2021
SET director = 'Not Given'
WHERE director IS NULL;
-- ? Explanation: Any remaining NULL values are replaced with "Not Given" to maintain data consistency.
-- ####################################################
-- # 8. Populate NULL `country` Values Using `director`
-- ####################################################
UPDATE Netflix2021 AS n1
JOIN (
SELECT director, MAX(country) AS country
FROM Netflix2021
WHERE country IS NOT NULL
GROUP BY director
) AS n2
ON n1.director = n2.director
SET n1.country = n2.country
WHERE n1.country IS NULL;
-- ? Explanation: Uses existing country data from directors to fill missing values.
-- ####################################################
-- # 9. Fill Remaining NULL `country` Values
-- ####################################################
UPDATE Netflix2021
SET country = 'Not Given'
WHERE country IS NULL;
-- ? Explanation: Ensures all country values are filled.
-- ####################################################
-- # 10. Delete Rows Where Critical Columns Have NULL Values
-- ####################################################
DELETE FROM Netflix2021 WHERE Release_Date IS NULL OR rating IS NULL OR duration IS
NULL;
-- ? Explanation: Ensures data integrity by removing incomplete records.
-- ####################################################
-- # 11. Extract `release_year` from `Release_Date`
-- ####################################################
-- Add a new column for the release year
ALTER TABLE Netflix2021 ADD COLUMN release_year INT;
-- Populate the new column with extracted year
UPDATE Netflix2021
SET release_year = YEAR(STR_TO_DATE(Release_Date, '%M %d, %Y'))
WHERE Release_Date IS NOT NULL;
-- ? Explanation: Extracts the year from Release_Date and stores it in a new column.
-- ####################################################
-- # 12. Standardize Text Formatting
-- ####################################################
UPDATE Netflix2021
SET title = TRIM(title),
director = TRIM(director),
country = TRIM(country);
-- ? Explanation: Ensures consistent formatting by removing unnecessary spaces.
-- ####################################################
-- # 13. Drop Unnecessary Columns
-- ####################################################
ALTER TABLE Netflix2021
DROP COLUMN cast_members,
DROP COLUMN description;
-- ? Explanation: Optimizes the dataset by removing unnecessary columns.
-- ####################################################
-- # 14. Final Data Validation
-- ####################################################
-- Confirm all NULL values are handled
SELECT
COUNT(*) - COUNT(show_id) AS showid_nulls,
COUNT(*) - COUNT(type) AS type_nulls,
COUNT(*) - COUNT(title) AS title_nulls,
COUNT(*) - COUNT(director) AS director_nulls,
COUNT(*) - COUNT(country) AS country_nulls,
COUNT(*) - COUNT(Release_date) AS date_added_nulls,
COUNT(*) - COUNT(release_year) AS release_year_nulls,
COUNT(*) - COUNT(rating) AS rating_nulls,
COUNT(*) - COUNT(duration) AS duration_nulls
FROM Netflix2021;
-- ? Explanation:Ensures there are no remaining NULL values.
-- ####################################################
-- # 15. Business Queries
-- ####################################################
-- View the cleaned dataset
SELECT * FROM Netflix2021;
-- 1. Country with Most Content
SELECT country, COUNT(*) AS content_count
FROM Netflix2021
GROUP BY country
ORDER BY content_count DESC;
-- 2. Movie vs TV Show Distribution
SELECT type, COUNT(*) AS count
FROM Netflix2021
GROUP BY type;
-- 3. Top Rated Content
SELECT title, rating
FROM Netflix2021
ORDER BY rating DESC
LIMIT 10;
-- 4. Most Frequent Directors
SELECT director, COUNT(*) AS count
FROM Netflix2021
GROUP BY director
ORDER BY count DESC
LIMIT 10;
-- 5. Content Additions Over Time
SELECT YEAR(Release_date) AS year, COUNT(*) AS content_count
FROM Netflix2021
GROUP BY year
ORDER BY year DESC;
-- 6. Age Rating Distribution
SELECT rating, COUNT(*) AS count
FROM Netflix2021
GROUP BY rating
ORDER BY count DESC;
-- 7. Most Common Duration for Movies
SELECT duration, COUNT(*) AS count
FROM Netflix2021
WHERE type IN (SELECT type FROM Netflix2021 WHERE type LIKE '%Movie%')
GROUP BY duration
ORDER BY count DESC;
-- ####################################################
-- # Final Check: Ensure Data is Cleaned Properly
-- ####################################################
SELECT * FROM Netflix2021 LIMIT 10;
```