IMDB Movie Analysis Report

Description of the dataset:

The dataset contains details about the top 1000 movies listed on IMDB up to the early 2020s. It consists of 1000 rows and 16 columns. The column descriptions are as follows:

1. **Poster Link**: Contains the link to the movie's poster.
2. **Series Title**: Represents the main title of the movie.
3. **Released Year**: Indicates the year the movie was released.
4. **Certificate**: Specifies the movie's certification rating.
5. **Runtime**: Denotes the length of the movie (in minutes).
6. **Genre**: Lists the genre(s) of the movie.
7. **IMDB Rating**: Displays the IMDB rating of the movie.
8. **Overview**: Provides a brief summary of the movie.
9. **Meta Score**: Shows the Metascore rating of the movie.
10. **Director**: Names the director of the movie.
11. **Star1**: Lists the lead actor/actress of the movie.
12. **Star2**: Names the second lead actor/actress.
13. **Star3**: Names the third lead actor/actress.
14. **Star4**: Names the fourth lead actor/actress.
15. **No. of Votes**: Indicates the number of user votes on IMDb, reflecting the movie's popularity.
16. **Gross**: Represents the total earnings of the movie.

1. Database Setup:

```sql
CREATE DATABASE IMDB;

CREATE TABLE movie_details(
    Poster_Link varchar(200),
    Series_Title varchar(200),
    Released_Year int,
    Certificate varchar(15),
    Runtime varchar(10),
    Genre varchar(50),
    IMDB_Rating double precision,
    Overview varchar(100),
    Meta_score int,
    Director varchar(80),
    Star1 varchar(80),
    Star2 varchar(80),
    Star3 varchar(80),
    Star4 varchar(80),
    No_of_Votes int,
    Gross int
);

-- Loading the data
COPY movie_details FROM 'D:\DataScience\DataSets\IMBD\imdb_top_1000.csv'
DELIMITER ','
CSV HEADER;
```

A new database named **IMDB** has been set up, and within this database, a table named **movie_details** has been created. This table contains all the columns from the dataset. The data has then been successfully loaded into the table.

2. Data Cleaning and Exploration:

```sql
-- Dropping unnecessary columns from the table
ALTER TABLE movie_details
DROP COLUMN poster_link,
DROP COLUMN overview,
DROP COLUMN star3,
DROP COLUMN star4;

-- Genre should have name of only top genre
UPDATE movie_details
SET genre = SPLIT_PART(genre,',',1);

-- Changing format of data in gross column
ALTER TABLE movie_details
ALTER COLUMN gross TYPE int
USING REPLACE(gross, ',', '')::int;

-- Changing format of data in runtime column
ALTER TABLE movie_details
ALTER COLUMN runtime TYPE INT
USING LEFT(runtime, POSITION(' ' IN runtime) - 1)::INT;
```

Unnecessary columns such as **Poster Link, Overview, Star3,** and **Star4** have been removed as they are not useful for data analysis. Additionally, the **Genre** column has been modified to retain only the first listed genre, assuming it to be the primary genre of the movie. The **Gross** column has been cleaned by removing commas from the data. Similarly, the **Runtime** column has been updated to contain only numeric values.

Checking and Handling NULL values:

```sql
-- Null values in series_title column
SELECT COUNT(*) AS null_count
FROM movie_details
WHERE series_title IS NULL;
-- 0

-- Null values in released_year column
SELECT COUNT(*) AS null_count
FROM movie_details
WHERE released_year IS NULL;
-- 1

-- Dropping the data row
DELETE FROM movie_details
WHERE released_year IS NULL;

-- Null values in certificate column
SELECT COUNT(*) AS null_count
FROM movie_details
WHERE certificate IS NULL;
-- 101

-- Updating null values to 'Unknown'
UPDATE movie_details
SET certificate='Unknown'
WHERE certificate IS NULL;
```

The **Series Title** column contains no null values. In the **Released Year** column, there is one null value, which has been removed from the table. The **Certificate** column contains 101 null values, which have been replaced with **'Unknown'**.

```sql
-- Null values in runtime column
SELECT COUNT(*) AS null_count
FROM movie_details
WHERE runtime IS NULL;
-- 0

-- Null values in genre column
SELECT COUNT(*) AS null_count
FROM movie_details
WHERE genre IS NULL;
-- 0

-- Null values in IMDB_Rating column
SELECT COUNT(*) AS null_count
FROM movie_details
WHERE IMDB_Rating IS NULL;
-- 0
```

There are no null values in the **Runtime, Genre** and **IMDB Rating** columns.

```sql
-- Null values in Meta_score column
SELECT COUNT(*) AS null_count
FROM movie_details
WHERE Meta_score IS NULL;
-- 157

-- Finding no. of rows where meta_score is greater than average value
SELECT COUNT(*)
FROM movie_details
WHERE meta_score > (
    SELECT ROUND(AVG(meta_score),1) FROM movie_details
);
-- 432

-- Finding no. of rows where meta_score is lower than average value
SELECT COUNT(*)
FROM movie_details
WHERE meta_score < (
    SELECT ROUND(AVG(meta_score),1) FROM movie_details
);
-- 386

-- Finding no. of rows where meta_score is equal to average value
SELECT COUNT(*)
FROM movie_details
WHERE meta_score = (
    SELECT ROUND(AVG(meta_score),1) FROM movie_details
);
-- 24
```

The **Meta Score** column has 157 null values. Out of the total 999 rows, **432** rows have a Meta Score higher than the average, **386** rows have a lower value, and **24** rows have a Meta Score equal to the average. Therefore, the null values in this column have been replaced with the average Meta Score.

```sql
-- Null values in director column
SELECT COUNT(*) AS null_count
FROM movie_details
WHERE director IS NULL;
-- 0

-- Null values in Star1 column
SELECT COUNT(*) AS null_count
FROM movie_details
WHERE Star1 IS NULL;
-- 0

-- Null values in Star2 column
SELECT COUNT(*) AS null_count
FROM movie_details
WHERE Star2 IS NULL;
-- 0

-- Null values in No_of_votes column
SELECT COUNT(*) AS null_count
FROM movie_details
WHERE No_of_votes IS NULL;
-- 0
```

There are no null values in **Director, Star1, Star2** and **No. of Votes** columns.

```sql
-- Null values in gross column
SELECT COUNT(*) AS null_count
FROM movie_details
WHERE gross IS NULL;
-- 169

-- Replacing null values by average gross value in each genre.
UPDATE movie_details AS m
SET gross = (
    SELECT AVG(Gross) FROM movie_details
    WHERE genre = m.genre AND gross IS NOT NULL
)
WHERE gross IS NULL;

-- Since there are some rows left where gross is still null, replace it by
-- average value of gross
UPDATE movie_details
SET gross= (
    SELECT ROUND(AVG(gross))
    FROM movie_details
)
WHERE gross IS NULL;
```

The **Gross** column contains a total of **169** null values. These values have been replaced with the **average gross value for each genre**. Since some rows still had null values after this step, the remaining null values were replaced with the **overall average gross value**.

Adding new column:

```
-- Adding new column:
ALTER TABLE movie_details
ADD COLUMN decade TEXT;

UPDATE movie_details
SET decade=
    CASE
        WHEN released_year BETWEEN 1920 AND 1929 THEN '1920s'
        WHEN released_year BETWEEN 1930 AND 1939 THEN '1930s'
        WHEN released_year BETWEEN 1940 AND 1949 THEN '1940s'
        WHEN released_year BETWEEN 1950 AND 1959 THEN '1950s'
        WHEN released_year BETWEEN 1960 AND 1969 THEN '1960s'
        WHEN released_year BETWEEN 1970 AND 1979 THEN '1970s'
        WHEN released_year BETWEEN 1980 AND 1989 THEN '1980s'
        WHEN released_year BETWEEN 1990 AND 1999 THEN '1990s'
        WHEN released_year BETWEEN 2000 AND 2009 THEN '2000s'
        WHEN released_year BETWEEN 2010 AND 2019 THEN '2010s'
        WHEN released_year BETWEEN 2020 AND 2029 THEN '2020s'
        ELSE 'Unknown'
    END;
```

A new column named **Decade** has been added, indicating the decade of release based on the **Released Year** column.