MARVEL DATASET USING SQL

Objective:

To analyze a dataset containing movie performance metrics, including financial data, audience and critic ratings, and release details. The goal is to write SQL queries to retrieve, manipulate, and analyze the data to extract meaningful insights and answer specific business or analytical questions.

About Dataset:

1. **Category**: Movie title.
2. **Year**: Release year.
3. **Worldwide Gross ($m)**: Global revenue in millions.
4. **% Budget Recovered**: Budget recouped as a percentage of gross revenue.
5. **Critics % Score**: Critics' approval rating.
6. **Audience % Score**: Audience approval rating.
7. **Audience vs Critics % Deviance**: Difference between audience and critics' scores.
8. **Budget ($m)**: Production budget in millions.
9. **Domestic Gross ($m)**: Revenue from domestic markets.
10. **International Gross ($m)**: Revenue from international markets.
11. **Opening Weekend ($m)**: Revenue during the opening weekend.
12. **Second Weekend ($m)**: Revenue during the second weekend.
13. **1st vs 2nd Weekend Drop Off**: Percentage drop in revenue from the first to second weekend.
14. **% Gross from Opening Weekend**: Proportion of the total gross earned during the opening weekend.
15. **% Gross from Domestic**: Proportion of total gross from domestic revenue.
16. **% Gross from International**: Proportion of total gross from international revenue.
17. **% Budget Opening Weekend**: Proportion of the budget recouped in the opening weekend.
18. **% Budget from Domestic**: Percentage of the budget recovered from domestic revenue (inferred column).

Here’s a structured list of SQL questions categorized into Basic, Intermediate, and Advanced levels based on the dataset you provided:

Basic Questions

1. **Retrieve Specific Columns**
   - Write a query to display `Category`, `Year`, and `Worldwide Gross ($m)` for all movies.
2. **Filter Records**
   - Write a query to find movies released after 2018.
3. **Sort Data**
   - Write a query to sort the movies by their `Worldwide Gross ($m)` in descending order.
4. **Count Records**
   - How many movies are in the dataset?

5. **Aggregate Functions**
    - Write a query to calculate the total `Worldwide Gross ($m)` for all movies.
6. **Filter with `WHERE`**
    - Write a query to display movies with a `Critics % Score` greater than 90%.
7. **Group Data**
    - Group movies by `Year` and find the total `Worldwide Gross ($m)` for each year.

Intermediate Questions

1. Advanced Filtering with Conditions
    - Write a query to find movies where the `Audience vs Critics % Deviance` is greater than 10% and the `Worldwide Gross ($m)` exceeds 1000.
2. Join Concepts (Hypothetical Additional Tables)
    - Suppose you have a table named `Genres` with columns `Category` and `Genre`. Write a query to join the `Movies` and `Genres` tables and display `Category`, `Year`, `Worldwide Gross ($m)`, and `Genre`.
3. Calculate Percentages
    - Write a query to calculate the percentage of the `Worldwide Gross ($m)` that comes from the `Domestic Gross ($m)` for each movie.
4. Use Aggregate Functions with Conditions
    - Write a query to find the average `Budget ($m)` for movies with a `Critics % Score` below 80%.
5. Subqueries
    - Write a query to find movies whose `Worldwide Gross ($m)` is higher than the average gross of all movies.
6. Multiple Filters with `AND`/`OR`
    - Find movies released after 2015 with an opening weekend revenue of more than $100 million or a second weekend drop-off less than 50%.
7. Using `CASE` Statements
    - Write a query to create a column labeled `Performance` that categorizes movies as:
        - `"Blockbuster"` if `Worldwide Gross ($m)` > 1000.
        - `"Hit"` if between 500 and 1000.
        - `"Average"` otherwise.

Advanced Questions

1. Window Functions
    - Write a query to calculate the rank of movies based on `Worldwide Gross ($m)` partitioned by `Year`.
2. Correlated Subqueries

    - Find movies whose `Worldwide Gross ($m)` is greater than the average gross of movies released in the same year.
3. Complex Joins (Hypothetical)
    - Suppose you have another table `Directors` with columns `Category` and `Director`. Write a query to find the total `Worldwide Gross ($m)` by each director.
4. CTEs (Common Table Expressions)
    - Write a CTE to calculate the percentage of the budget recovered (`% Budget Recovered`) for each movie and use it to display movies that recovered more than 400% of their budget.
5. Dynamic Aggregation
    - Write a query to calculate the yearly total, average, minimum, and maximum `Worldwide Gross ($m)`.
6. Nested Queries in `FROM` Clause
    - Write a query to find movies with a `Worldwide Gross ($m)` greater than the median gross of all movies.
7. JSON or Array Handling (if applicable)
    - Suppose the column `Audience vs Critics % Deviance` is stored as JSON. Write a query to extract its value and filter movies where it is less than 0.
8. Recursive CTEs
    - Write a recursive CTE to simulate year-over-year cumulative gross revenue for movies released after 2010.