



ABOUT SQL INTERMEDIATE QUESTIONS

Structured list of SQL questions
categorized into Intermediate

MARVEL



PROJECT : MARVEL MOVIES

My job has been 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.

Data Source



01

Movies where the Audience vs Critics % Deviance is greater than 10% and Worldwide Gross exceeds 1000.

```
SELECT MovieTitle, AudienceCriticsDeviance, WorldwideGross  
FROM marvel.marvelmovies  
WHERE ABS(AudienceCriticsDeviance) > 10  
AND WorldwideGross > 1000;
```



02

Find pairs of movies released in the same year, where the first movie (Movie1**) has a higher Worldwide Gross than the second movie (**Movie2**).**

```
SELECT m1.MovieTitle AS Movie1, m2.MovieTitle AS Movie2,  
       m1.Year, m1.WorldwideGross AS Movie1Gross, m2.WorldwideGross AS Movie2Gross  
FROM marvel.marvelmovies m1  
JOIN marvel.marvelmovies m2  
ON m1.Year = m2.Year  
WHERE m1.WorldwideGross > m2.WorldwideGross  
AND m1.MovieTitle != m2.MovieTitle;
```



03

Calculate the percentage of the Worldwide Gross that comes from the Domestic Gross for each movie.

```
SELECT MovieTitle,  
       (DomesticGross / WorldwideGross) * 100 AS DomesticGrossPercentage  
FROM marvel.marvelmovies  
WHERE WorldwideGross > 0;
```



04

Find the average Budget for movies with a Critics % Score below 80%:

```
SELECT AVG(Budget) AS AverageBudget  
FROM marvel.marvelmovies  
WHERE CriticsScore < 80;
```



05

Find movies whose Worldwide Gross is higher than the average gross of all movies.

```
SELECT MovieTitle, WorldwideGross  
FROM marvel.marvelmovies  
WHERE WorldwideGross > (SELECT AVG(WorldwideGross) FROM marvel.marvelmovies);
```



06

Find movies released after 2015 with an opening weekend revenue of more than \$100 million or a second weekend drop-off less than 50%.

```
SELECT MovieTitle, Year, OpeningWeekendGross, SecondWeekendGross,  
WeekendDropOffPercentage  
FROM marvel.marvelmovies  
WHERE Year > 2015  
AND (OpeningWeekendGross > 100 OR  
WeekendDropOffPercentage < 50);
```



07

Create a column labeled "Performance" categorizing movies.

```
SELECT MovieTitle, WorldwideGross,  
       CASE  
           WHEN WorldwideGross > 1000 THEN 'Blockbuster'  
           WHEN WorldwideGross BETWEEN 500 AND 1000 THEN 'Hit'  
           ELSE 'Average'  
       END AS Performance  
FROM marvel.marvelmovies;
```



08

.Group movies by Category and find the total Worldwide Gross.

```
SELECT Category, SUM(WorldwideGross) AS TotalWorldwideGross  
FROM marvel.marvelmovies  
GROUP BY Category;
```



OVERALL LEARNING

- **JOINS:** Combining rows from the same table based on a shared column (like year). This allows you to compare different records within the same dataset, such as comparing two movies released in the same year.
- **Subqueries:** Using subqueries to calculate aggregates (like the average worldwide gross) and comparing individual records against these aggregates.
- **Conditional Logic:** Using CASE statements to categorize data (e.g., categorizing movies into "Blockbuster," "Hit," or "Average" based on their worldwide gross).





Thank For Your Attention

[LinkedIn](#) | [Github](#) | [Email](#)