

ROUND() with a Negative Parameter

Exercise

ROUND() with a negative parameter

A useful thing you can do with `ROUND()` is have a negative number as the decimal place parameter. This can come in handy if your manager only needs to know the average number of `facebook_likes` to the hundreds since granularity below one hundred likes won't impact decision making.

Social media plays a significant role in determining success. If a movie trailer is posted and barely gets any likes, the movie itself may not be successful. Remember how 2020's "Sonic the Hedgehog" movie got a revamp after the public saw the trailer?

Let's apply this to other parts of the dataset and see what the benchmark is for movie budgets so, in the future, it's clear whether the film is above or below budget.

Instructions 100 XP

- Calculate the average `budget` from the `films` table, aliased as `avg_budget_thousands`, and round to the nearest thousand.

[Take Hint \(-50 XP\)](#)

```
1 -- Calculate the average budget rounded to the thousands
2 ---
```

query result films

No query executed yet...

Showing 0 out of 0 rows

Question

Calculate the average budget from the `films` table, aliased as `avg_budget_thousands`, and round to the nearest thousand.

Answer

```
-- Calculate the average budget rounded to the nearest thousand
SELECT ROUND(AVG(budget), -3) AS avg_budget_thousands
FROM films;
```

Code Explanation

- `SELECT ROUND(AVG(budget), -3)`: Calculates the average of the 'budget' column and rounds the result to the nearest thousand using the `ROUND` function with a negative parameter (-3).
- `AS avg_budget_thousands`: Assigns the alias 'avg_budget_thousands' to the calculated value.
- `FROM films`: Specifies the 'films' table as the source of data.