

Counting Missing Rows with Left Join

The screenshot shows a web browser window with multiple tabs. The active tab is 'Counting missing rows with left join' on the DataCamp website. The page displays an exercise titled 'Counting missing rows with left join'. The instructions state: 'The Movie Database is supported by volunteers going out into the world, collecting data, and entering it into the database. This includes financial data, such as movie budget and revenue. If you wanted to know which movies are still missing data, you could use a left join to identify them. Practice using a left join by merging the `movies` table and the `financials` table. The `movies` and `financials` tables have been loaded for you.' Below the instructions, there is a 'Question' section asking: 'What column is likely the best column to merge the two tables on?'. There are three 'Possible answers' listed: 'on="budget"', 'on="popularity"', and 'on="id"'. The 'on="id"' option is selected. A 'Submit Answer' button is visible. To the right of the question, there is a code editor window titled 'script.py' with a 'Run Code' button. Below the code editor, there is a 'Python Shell' window. The browser's address bar shows the URL 'campus.datacamp.com/courses/joining-data-with-pandas/merging-tables-with-different-join-types/ex=2'. The browser's taskbar at the bottom shows various application icons and the system clock indicating 18:29 on 02/12/2024.

Question:

The Movie Database is supported by volunteers collecting and entering data into the database. This data includes financial information such as movie budgets and revenue. If you want to know which movies are still missing data, you can use a left join to identify them. Merge the `movies` and `financials` tables.

Instructions:

1. Determine the best column to merge the two tables (`movies` and `financials`) on.
2. Choose the correct option from the possible answers:
 - `on='budget'`
 - `on='popularity'`
 - `on='id'`

Answer:

The best column to merge the `movies` and `financials` tables on is `on='id'`.

This is because `id` uniquely identifies each movie across both tables, making it the most reliable key for merging the datasets.

Explanation of the Answer:

1. ``on='id'``: The ``id`` column is a unique identifier for movies in both the ``movies`` and ``financials`` tables. Using this column ensures that the data from both tables is correctly matched for each movie.
2. ``on='budget'``: This column represents the budget of a movie, which is not unique. Using it for merging could result in incorrect matches or duplicated data.
3. ``on='popularity'``: This column indicates how popular a movie is, which is also not unique. It is unsuitable for merging as it does not uniquely identify movies.