Documentation Agent - Chat with Any Repo

Chat with our AI assistant to get information about your Github Repo.

how merge function do his job?

I need more context to understand which `merge` function you are referring to. Could you please specify the file or library where this function is located? This will help me provide a more accurate explanation of its functionality.

Search In Repo with...

The `merge` function in the `dataFrame` class of the `pandascpp` library combines two data frames based on a common column. Here's how it works, according to the `README.md` file:

1. **Input:** The `merge` function takes two data frames (`df1` and `df2`), a common column name (`"ID"` in the example), and a merge type (`"outer"` in the example) as input.
2. **Functionality:** It merges the two data frames based on the specified column. The "outer" merge type includes all rows from both data frames, filling in missing values with `NA` where there is no match.
3. **Output:** The result is a new data frame (`merged`) that combines columns from both input data frames, aligned by the common column.

For example:

```
dataFrame df1({
    {"ID",    column("ID",    {1,  2,  3})},
    {"Value", column("Value", {10, 20, 30})}
});

dataFrame df2({
    {"ID",    column("ID",    {1, 2,  4})},
    {"Price", column("Price", {5, 10, 15})}
});
dataFrame merged = df1.merge(df2, "ID", "outer");
cout << merged; // Output: Table with ID, Value, Price (includes NA)
```

In this example, `df1` and `df2` are merged based on the "ID" column using an outer merge. The resulting `merged` data frame will contain all IDs from both `df1` and `df2` , with corresponding "Value" and "Price" columns. If an ID exists in one data frame but not the other, the missing value will be filled with `NA` .

Sources

README.md
2025-03-13 03:23:38

give me the flow of the implementation of it

I need to inspect the source code to provide you with the implementation flow of the `merge` function. I will generate a search query to retrieve the relevant code from the `dataFrame.cpp` file.

Search In Repo with...

The `merge` function in `dataFrame.cpp` combines two data frames (`this` and `other`) based on a common column (`on`) and a specified merge type (`how`). Here's a step-by-step breakdown of its implementation:

1. **Get Merge Values:**
    - It calls `get_merg_df_values_on` for both data frames (`this` and `other`) to create unordered maps (`left_values` and `right_values`). These maps store the values of all columns *except* the `on` column, indexed by the values in the `on` column. This allows for efficient lookup during the merge process.
2. **Initialize New Data Frame:**
    - It creates a new `unordered_map` called `new_df_data` to store the data for the merged data frame. This map will hold `column` objects for each column in the merged data frame.

- It creates a `vector<string>` called `all_cols` to store the names of all columns in the merged data frame. The size of this vector is the sum of the number of columns in both input data frames (minus 1, to account for the shared `on` column).
- It iterates through the columns of the `this` data frame and copies the column data and metadata (like `mixed_type` and `name`) to `new_df_data`. It also populates the `all_cols` vector with these column names.
- It iterates through the columns of the `other` data frame. If a column name is not the `on` column and not already present in `new_df_data`, it creates a new `column` in `new_df_data` and adds the column name to `all_cols`.

3. **Iterate Through 'On' Values and Merge:**
   - The code iterates through the `left_values` map (representing the `on` column values from the `this` data frame). For each `on` value, it checks if the same `on` value exists in the `right_values` map (representing the `on` column values from the `other` data frame).
   - Based on the `how` merge type (e.g., "inner", "outer", "left", "right"), it decides whether to include the current row in the merged data frame.
   - If the `on` value exists in both `left_values` and `right_values` (or if the merge type allows including the row even if it's only in one data frame), it adds the corresponding data from both data frames to the `new_df_data` columns.
4. **Create and Return New Data Frame:**
   - After processing all `on` values, it constructs a new `dataFrame` object using the `new_df_data` map. The `columns` member of the new data frame is set to `all_cols`.
   - Finally, it returns the newly created merged `dataFrame`.

In summary, the `merge` function efficiently combines data from two data frames based on a common column, handling different merge types to include or exclude rows based on the presence of matching values in the `on` column.

Sources

thanks

You're welcome! If you have any more questions about the `merge` function or anything else related to the `pandascpp` library, feel free to ask.

Type your message here...