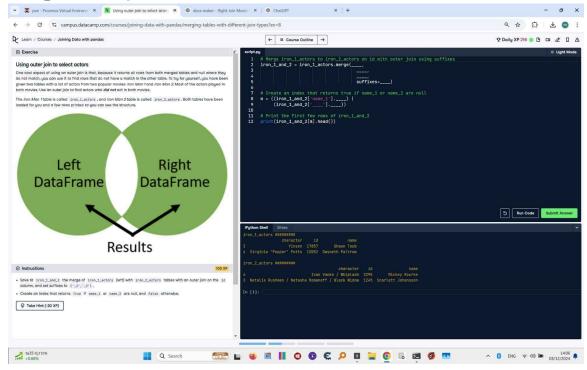
## **Using Outer Join to Select Actors - Full Solution**



Screenshot showing the exercise context for using an outer join to select actors.

## **Code Answer:**

- # Merge iron\_1\_actors to iron\_2\_actors on id with outer join using suffixes iron\_1\_and\_2 = iron\_1\_actors.merge(iron\_2\_actors, on='id', how='outer', suffixes=(' 1', ' 2'))
- # Create an index that returns True if name\_1 or name\_2 are null
  m = iron\_1\_and\_2['name\_1'].isnull() | iron\_1\_and\_2['name\_2'].isnull()
- # Print the first few rows of iron\_1\_and\_2 where the condition is True print(iron 1 and 2[m].head())

## **Explanation:**

1. The `merge` function performs an outer join on the 'id' column between the 'iron\_1\_actors' and 'iron\_2\_actors' DataFrames. The `how='outer'` parameter ensures that all rows from both DataFrames are included, with missing values filled with NaN where matches are not found. The

- `suffixes= $('_1', '_2')$ ` parameter adds suffixes to distinguish columns from the two DataFrames.
- 2. The condition `m` identifies rows where either 'name\_1' or 'name\_2' is null, using the `isnull()` method combined with the logical OR operator (`|`). This step isolates rows corresponding to actors who appear in only one of the two movies.
- 3. The final step filters the 'iron\_1\_and\_2' DataFrame using the condition `m` and displays the first few rows with `print(iron\_1\_and\_2[m].head())`. This allows verification of actors who are exclusive to one movie.