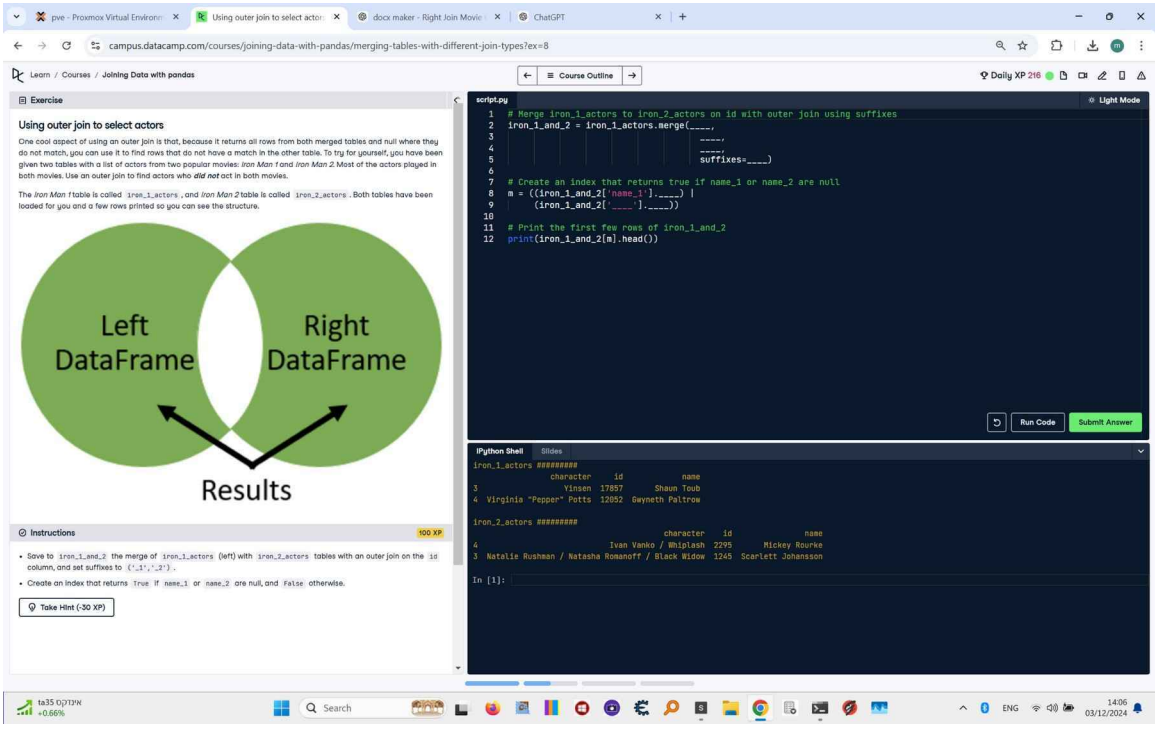
Using Outer Join to Select Actors - Full Solution



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.