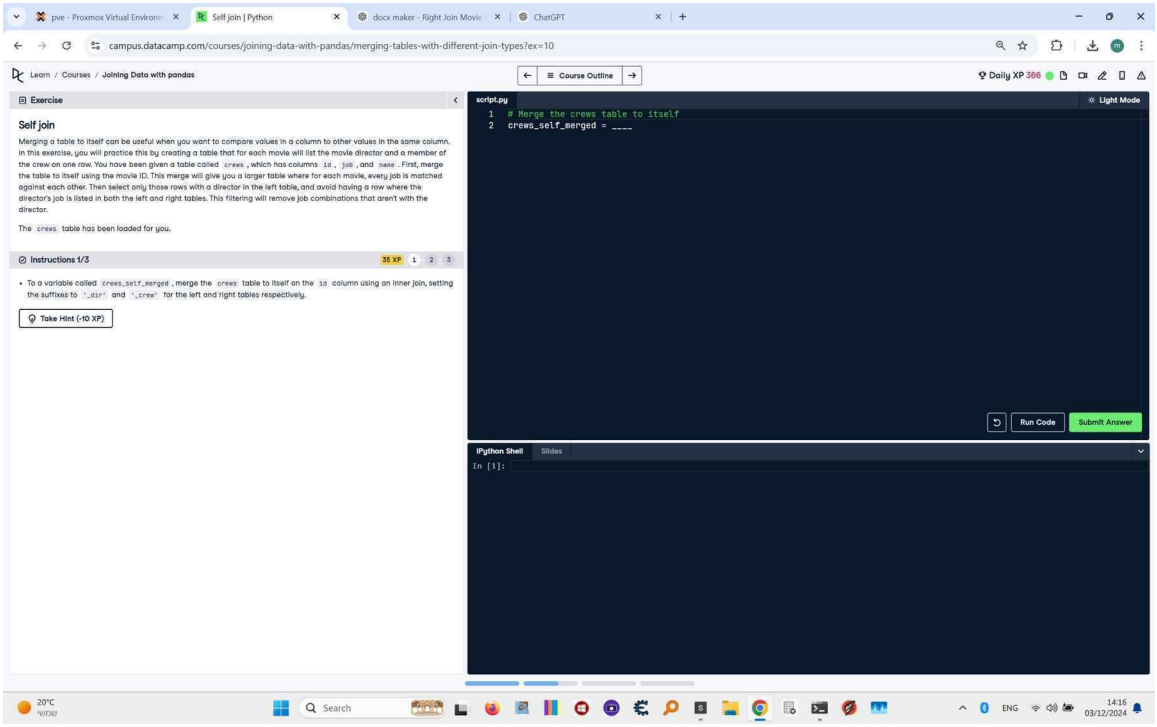
Self Join - Final Correct Solution



Screenshot showing the exercise context for performing a self join on the crews table.

Code Answer:

```
# Merge the crews table to itself
crews_self_merged = crews.merge(crews, on='id', how='inner',
                                suffixes=('_dir','_crew'))

# Create a Boolean index to select the appropriate rows
boolean_filter = ((crews_self_merged['job_dir'] == 'Director') &
                  (crews_self_merged['job_crew'] != 'Director'))
direct_crews = crews_self_merged[boolean_filter]

# Print the resulting DataFrame
print(direct_crews)
```

Explanation:

1. The `merge` function performs a self join on the 'crews' table by joining it to itself using the 'id' column. The `how='inner'` parameter ensures that only rows with matching 'id' values in both tables are included. The

`suffixes=('_dir', '_crew')` parameter is used to distinguish between columns from the left and right tables, representing directors and crew members respectively.

2. A Boolean filter is created to select rows where the 'job_dir' column is 'Director' and the 'job_crew' column is not 'Director'. This ensures that only valid combinations of directors and crew members are included.

3. The resulting DataFrame is filtered using the Boolean index to produce 'direct_crews', which contains the desired pairs of directors and their corresponding crew members.

4. Finally, the resulting DataFrame is printed to verify that only valid combinations of directors and crew members remain.