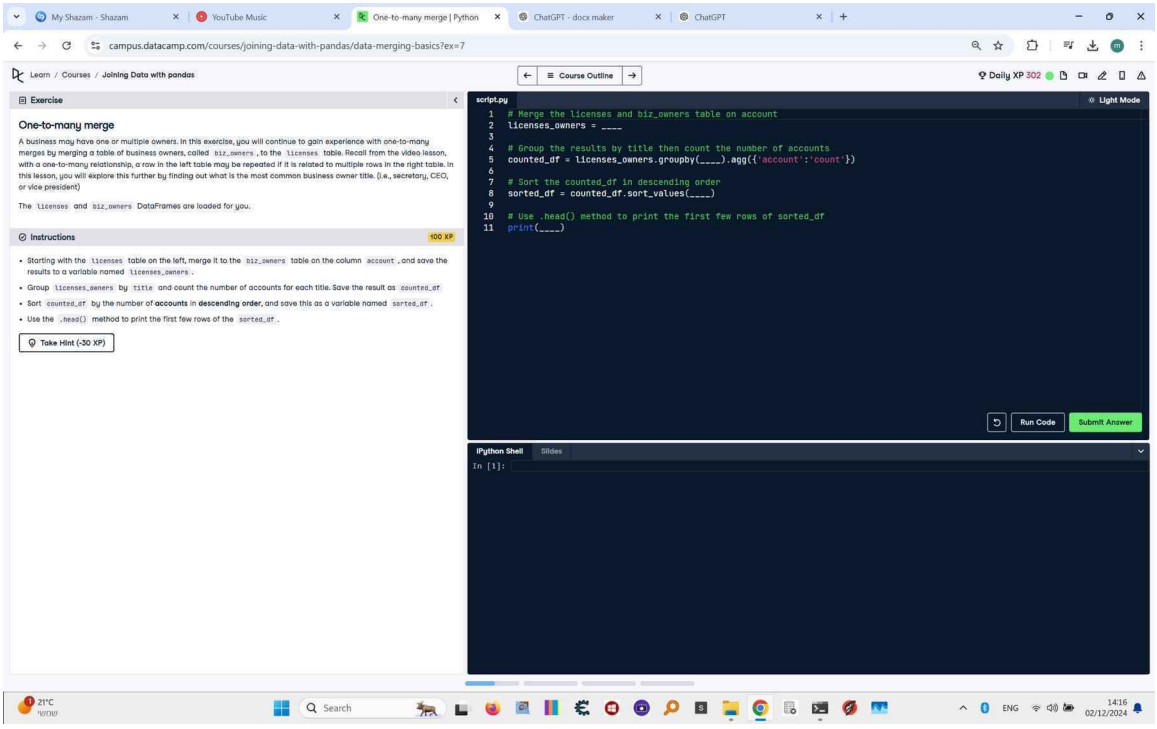
One-to-Many Merge



Question:

A business may have one or multiple owners. In this exercise, you will continue to gain experience with one-to-many merges by merging a table of business owners, called `biz_owners`, to the `licenses` table. Recall from the video lesson, with a one-to-many relationship, a row in the left table may be repeated if it is related to multiple rows in the right table. In this lesson, you will explore this further by finding out what is the most common business owner title (i.e., secretary, CEO, or vice president).

Instructions:

1. Starting with the `licenses` table on the left, merge it to the `biz_owners` table on the column `account`, and save the result to a variable named `licenses_owners`.

2. Group `licenses_owners` by `title` and count the number of accounts for each title. Save the result as `counted_df`.

3. Sort `counted_df` by the number of accounts in descending order, and save this as a variable named `sorted_df`.

4. Use the `.head()` method to print the first few rows of `sorted_df`.

Answer:

```
# Step 1: Merge the `licenses` and `biz_owners` tables on `account`
licenses_owners = licenses.merge(biz_owners, on='account')


# Step 2: Group `licenses_owners` by `title` and count the accounts
counted_df = licenses_owners.groupby('title').agg({'account': 'count'})


# Step 3: Sort `counted_df` by the number of accounts in descending order
sorted_df = counted_df.sort_values('account', ascending=False)


# Step 4: Use `.head()` to print the first few rows of `sorted_df`
print(sorted_df.head())
```

Explanation of the Code:

1. `licenses_owners = licenses.merge(biz_owners, on='account')`: This merges the `licenses` table with the `biz_owners` table on the `account` column, combining the data from both tables.

2. `counted_df = licenses_owners.groupby('title').agg({'account': 'count'})`: This groups the merged table by the `title` column and counts the occurrences of each title using the `account` column.

3. `sorted_df = counted_df.sort_values('account', ascending=False)`: This sorts the grouped data in descending order based on the count of accounts, so the most common titles appear at the top.

4. `print(sorted_df.head())`: This prints the first few rows of the sorted DataFrame, displaying the most common titles.