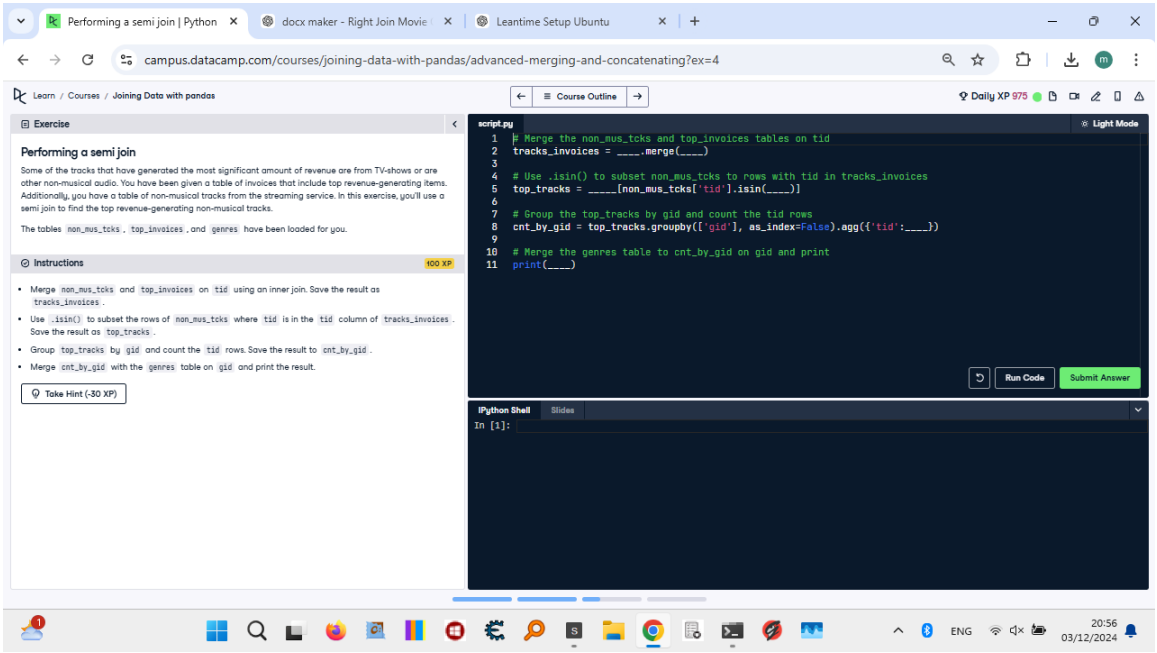
Performing a Semi Join in Pandas



Screenshot showing the exercise context for performing a semi join in pandas.

Code Answer:

```
# Merge the non_mus_tcks and top_invoices tables on tid
tracks_invoices = non_mus_tcks.merge(top_invoices, on='tid', how='inner')

# Use .isin() to subset non_mus_tcks to rows with tid in tracks_invoices
top_tracks = non_mus_tcks[non_mus_tcks['tid'].isin(tracks_invoices['tid'])]

# Group the top_tracks by gid and count the tid rows
cnt_by_gid = top_tracks.groupby(['gid'], as_index=False).agg({'tid':'count'})

# Merge the genres table to cnt_by_gid on gid and print
print(cnt_by_gid.merge(genres, on='gid'))
```

Explanation:

1. The `merge` function performs an inner join between 'non_mus_tcks' and 'top_invoices' tables on the 'tid' column, resulting in rows that exist in both tables. This step identifies the relevant non-musical tracks with matching invoices.

2. The `.isin()` method is then used to filter the rows in 'non_mus_tcks' where the 'tid' exists in the resulting 'tracks_invoices' table, retaining only those rows. The filtered DataFrame is stored as 'top_tracks'.

3. The `groupby` function is used to group the 'top_tracks' DataFrame by 'gid' (genre ID) and count the number of 'tid' rows for each genre. This grouped DataFrame is stored as 'cnt_by_gid'.

4. Finally, the 'cnt_by_gid' DataFrame is merged with the 'genres' table on 'gid', enriching the grouped data with genre names, and the result is printed.