

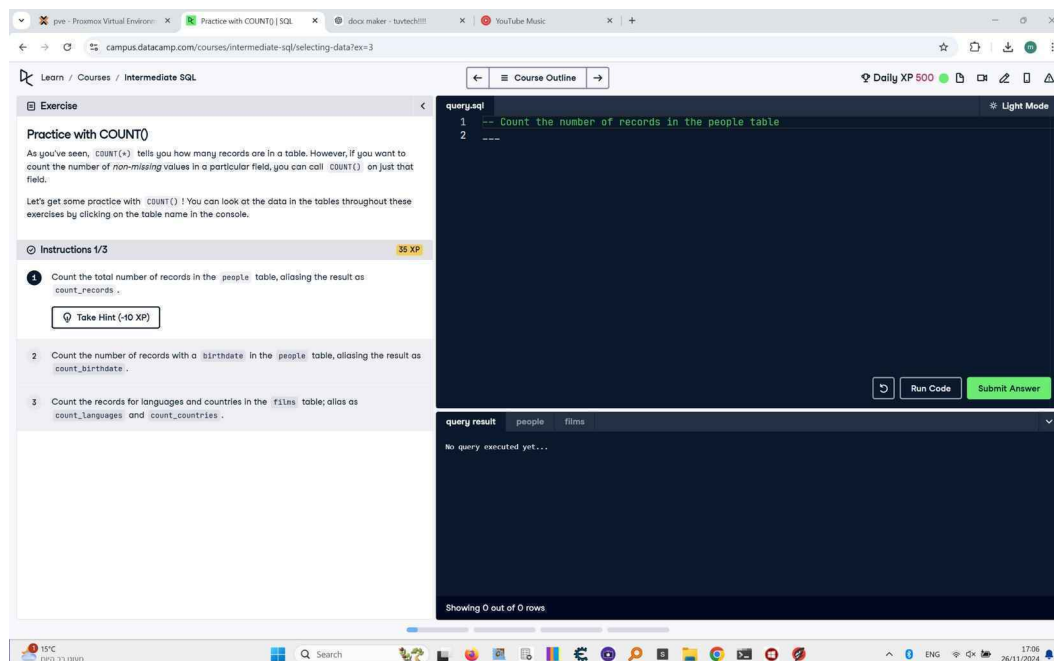
Practice with COUNT() (Corrected)

In this task, you'll get some practice with COUNT() in SQL. You will count the total number of records in the `people` table, count non-NULL records for the `birthdate` column, and count distinct records for `languages` and `countries` in the `films` table.

Instructions:

1. Count the total number of records in the `people` table, aliasing the result as `count_records`.
2. Count the number of records with a `birthdate` in the `people` table, aliasing the result as `count_birthdate`.
3. Count the records for `languages` and `countries` in the `films` table; alias as `count_languages` and `count_countries`.

Original Uploaded Image:



Corrected SQL Query Implementation:

```
-- Count the total number of records in the people table
SELECT COUNT(*) AS count_records
FROM people;
```

```
-- Count the number of records with a birthdate in the people table
SELECT COUNT(birthdate) AS count_birthdate
```

FROM people;

```
-- Count records for languages and countries in the films table
SELECT COUNT(language) AS count_languages, COUNT(country) AS
count_countries
FROM films;
```

Explanation of Corrected Queries:

1. **Alias columns correctly**: Ensure that the output columns are correctly aliased as `count_records`, `count_birthdate`, `count_languages`, and `count_countries` so that they match what is expected by the system.
2. **Total records in `people` table**: The `COUNT(*)` function counts all rows in the `people` table, including rows with NULL values, and aliases the result as `count_records`.
3. **Records with `birthdate` in `people` table**: The `COUNT(birthdate)` function counts only the rows where the `birthdate` column is not NULL and aliases the result as `count_birthdate`.
4. **Counts in `films` table**: Use `COUNT(language)` and `COUNT(country)` to count non-NULL values in the `language` and `country` columns of the `films` table, aliasing the results as `count_languages` and `count_countries`.