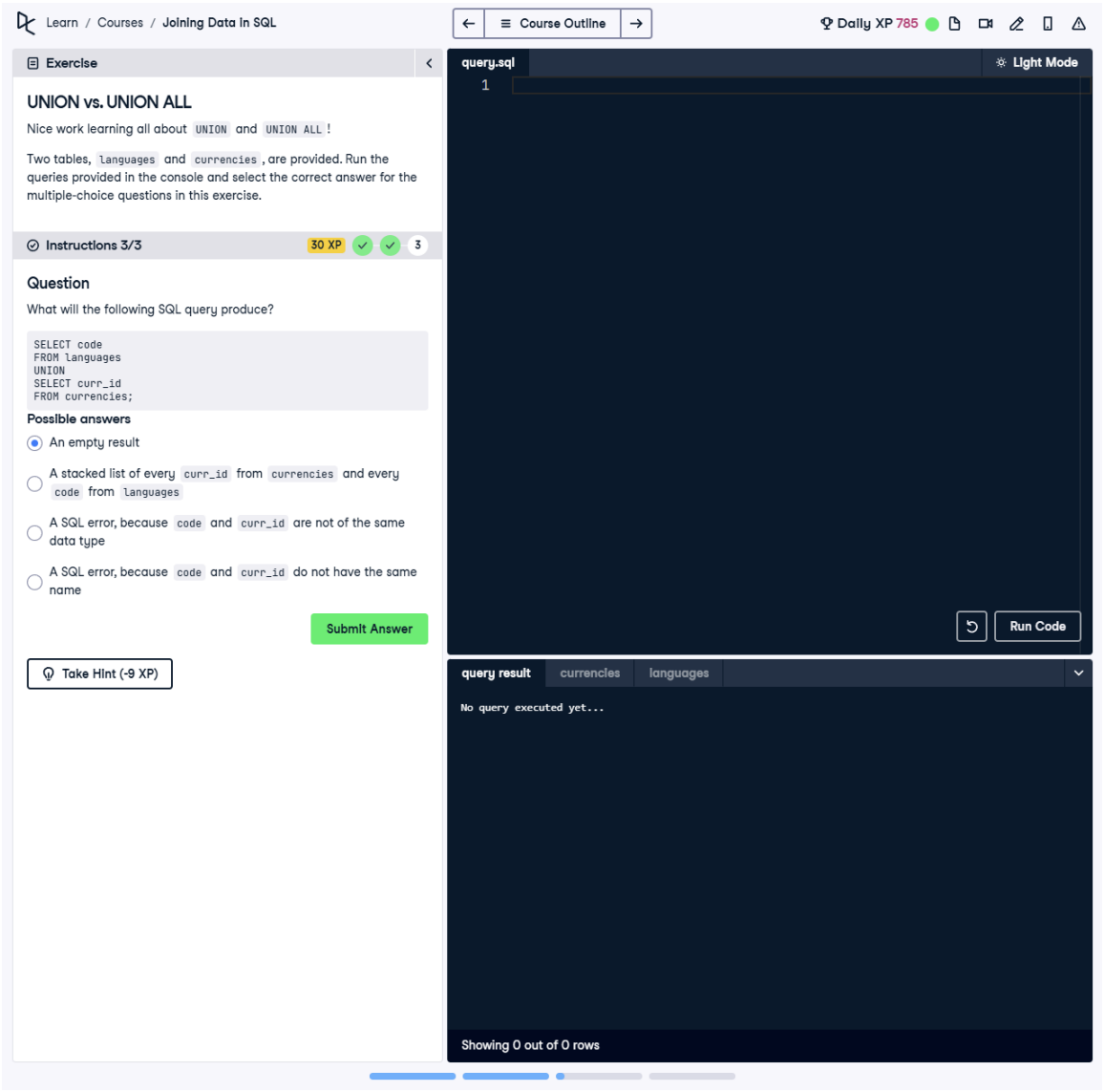
UNION vs. UNION ALL in SQL

This document explains the behavior of UNION and UNION ALL in SQL with an example query and the reasons for SQL errors when column names or types differ between tables.

Below is the task instruction and query setup:



-- SQL Query Demonstration

SQL Query:

SELECT code

FROM languages

```
UNION
SELECT curr_id
FROM currencies;
```

-- Correct Answer

The query will result in a SQL error because 'code' and 'curr_id' do not have the same name.

Explanation:

1. **UNION**:
   - Requires the same number of columns in both tables.
   - The column names must match exactly, or aliasing must be used to align them.
   - Both columns must have compatible data types.
2. In this query:
   - 'code' (from the `languages` table) and 'curr_id' (from the `currencies` table) do not have the same name.
   - The query fails with a SQL error because UNION checks for matching column names and aligned data types.
   - To fix this issue, you could use aliases to rename the columns in the SELECT clauses so that they match.

Example fix:

```
SELECT code AS id
FROM languages
UNION
SELECT curr_id AS id
FROM currencies;
```

This ensures the column names are aligned.