

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:

The screenshot shows a web application interface for learning SQL. The top navigation bar includes a logo, the text "Learn / Courses / Joining Data In SQL", a "Course Outline" button, and a "Daily XP 785" indicator. The main content area is divided into two panels. The left panel, titled "Exercise", contains the following text: "UNION vs. UNION ALL", "Nice work learning all about UNION and UNION ALL!", "Two tables, languages and currencies, are provided. Run the queries provided in the console and select the correct answer for the multiple-choice questions in this exercise.", "Instructions 3/3" (with 30 XP and two green checkmarks), "Question: What will the following SQL query produce?", a code block containing the SQL query:

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

, "Possible answers" with four radio button options: "An empty result" (selected), "A stacked list of every curr_id from currencies and every code from languages", "A SQL error, because code and curr_id are not of the same data type", and "A SQL error, because code and curr_id do not have the same name", a "Submit Answer" button, and a "Take Hint (-9 XP)" button. The right panel, titled "query.sql", shows a dark-themed editor with the number "1" at the top. Below the editor is a "query result" section with tabs for "currencies" and "languages". The "query result" section displays the message "No query executed yet..." and "Showing 0 out of 0 rows".

-- 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.