

Chaining FULL JOINs Query Example with Corrected WHERE Clause

This document demonstrates how to chain multiple FULL JOINs to retrieve data from three tables: 'countries', 'languages', and 'currencies'. The focus is on filtering for regions starting with 'M' and ending with 'esia', such as 'Melanesia' and 'Micronesia'. The WHERE clause has been adjusted to ensure accurate filtering.

-- Corrected SQL Query Using Chained FULL JOINs

```
SELECT
    c1.name AS country,
    region,
    l.name AS language,
    basic_unit,
    frac_unit
FROM countries AS c1
-- Full join with languages (alias as l)
FULL JOIN languages AS l
USING (code)
-- Full join with currencies (alias as c2)
FULL JOIN currencies AS c2
USING (code)
WHERE region LIKE 'M%esia';
```

Explanation: This query combines data from three tables using chained FULL JOINs:

1. FULL JOIN between 'countries' and 'languages' tables on the 'code' column ensures all rows from both tables are included.
 2. Another FULL JOIN with 'currencies' table on the same 'code' column ensures no data is excluded.
 3. The SELECT statement retrieves the following columns:
 - 'country': The name of the country.
 - 'region': The region of the country.
 - 'language': The name of the language spoken.
 - 'basic_unit': The basic currency unit.
 - 'frac_unit': The fractional unit of the currency.
 4. The WHERE clause filters rows where the 'region' column matches the pattern 'M%esia', ensuring only regions starting with 'M' and ending with 'esia' are included, such as 'Melanesia' and 'Micronesia'.
- This approach ensures that all records from the three tables are considered, even if some rows are unmatched.