**Full join**

In this exercise, you'll examine how your results differ when using a full join versus using a left join versus using an inner join with the countries and currencies tables.

You will focus on the North American region and also where the name of the country is missing. Dig in to see what we mean!

Begin with a full join with countries on the left and currencies on the right. The fields of interest have been SELECTed for you throughout this exercise.

Then complete a similar left join and conclude with an inner join.

Choose records in which region corresponds to North America or is NULL.

SELECT name AS country, code, region, basic\_unit

-- From countries

FROM countries

  -- Join to currencies

  FULL JOIN currencies

    -- Match on code

    USING (code)

-- Where region is North America or null

WHERE region = 'North America' OR region IS null

-- Order by region

ORDER BY region;

# Full join (2)

You'll now investigate a similar exercise to the last one, but this time focused on using a table with more records on the left than the right. You'll work with the languages and countries tables.

Begin with a full join with languages on the left and countries on the right. Appropriate fields have been selected for you again here.

##### Instructions 1/3

**35 XP**

* [1](javascript:void(0))
  + Choose records in which countries.name starts with the capital letter 'V' or is NULL.
  + Arrange by countries.name in ascending order to more clearly see the results.
* SELECT countries.name, code, languages.name AS language
* -- From languages
* FROM languages
* -- Join to countries
* full JOIN countries
* -- Match on code
* USING (code)
* -- Where countries.name starts with V or is null
* WHERE countries.name LIKE 'V%' OR countries.name IS null
* -- Order by ascending countries.name
* ORDER BY countries.name;

# Full join (3)

You'll now explore using two consecutive full joins on the three tables you worked with in the previous two exercises.

##### Instructions

**100 XP**

* Complete a full join with countries on the left and languages on the right.
* Next, full join this result with currencies on the right.
* Use LIKE to choose the Melanesia and Micronesia regions (Hint: 'M%esia').
* Select the fields corresponding to the country name AS country, region, language name AS language, and basic and fractional units of currency.
* -- Select fields (with aliases)
* SELECT c1.name AS country, region, l.name AS language,
* basic\_unit, frac\_unit
* -- From countries (alias as c1)
* FROM countries AS c1
* -- Join with languages (alias as l)
* FULL JOIN languages AS l
* -- Match on code
* USING (code)
* -- Join with currencies (alias as c2)
* FULL JOIN currencies AS c2
* -- Match on code
* USING (code)
* -- Where region like Melanesia and Micronesia
* WHERE region LIKE 'M%esia';

# A table of two cities

This exercise looks to explore languages potentially and most frequently spoken in the cities of Hyderabad, India and Hyderabad, Pakistan.

##### Instructions 1/2

**50 XP**

* [1](javascript:void(0))
* [2](javascript:void(0))
* Create a CROSS JOIN with cities AS c on the left and languages AS l on the right.
* Make use of LIKE and Hyder% to choose Hyderabad in both countries.
* Select only the city name AS city and language name AS language.
* -- Select fields
* SELECT c.name AS city, l.name AS language
* -- From cities (alias as c)
* FROM cities AS c
* -- Join to languages (alias as l)
* CROSS JOIN languages AS l
* -- Where c.name like Hyderabad
* WHERE c.name LIKE 'Hyder%';
* Use an INNER JOIN instead of a CROSS JOIN. Think about what the difference will be in the results for this INNER JOIN result and the one for the CROSS JOIN.
* SELECT c.name AS city, l.name AS language
* FROM cities AS c
* -- Join to languages (alias as l)
* INNER JOIN languages AS l
* on l.code =c.country\_code
* -- Where c.name like Hyderabad
* WHERE c.name LIKE 'Hyder%';

# Outer challenge

Now that you're fully equipped to use OUTER JOINs, try a challenge problem to test your knowledge!

In terms of life expectancy for 2010, determine the names of the lowest five countries and their regions.

##### Instructions

**100 XP**

* Select country name AS country, region, and life expectancy AS life\_exp.
* Make sure to use LEFT JOIN, WHERE, ORDER BY, and LIMIT.
* Select c.name as country ,region,p.life\_expectancy as life\_exp
* from countries as c
* left JOIN populations as p
* ON p.country\_code = c.code
* -- Focus on 2010
* where p.year = 2010
* -- Order by life\_exp
* Order by life\_exp
* limit 5;