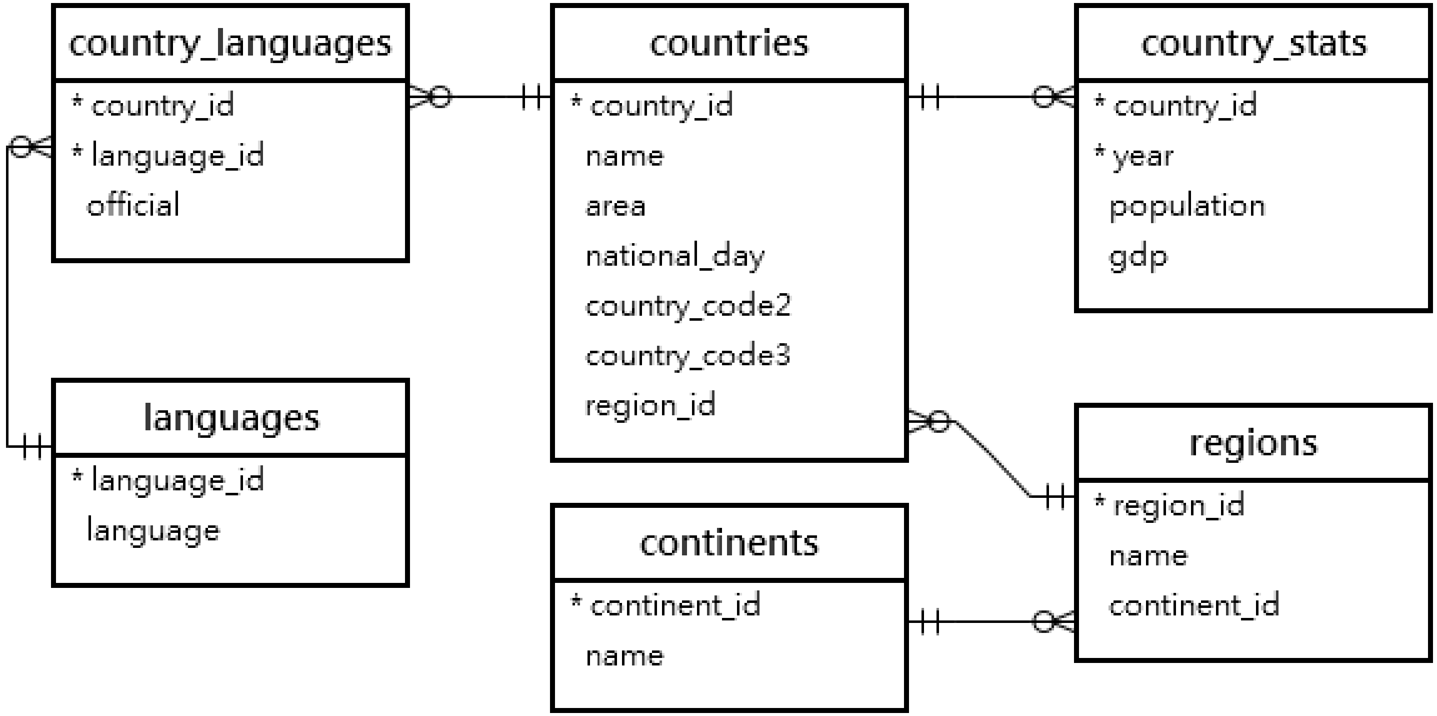
SQL SELECT Statement

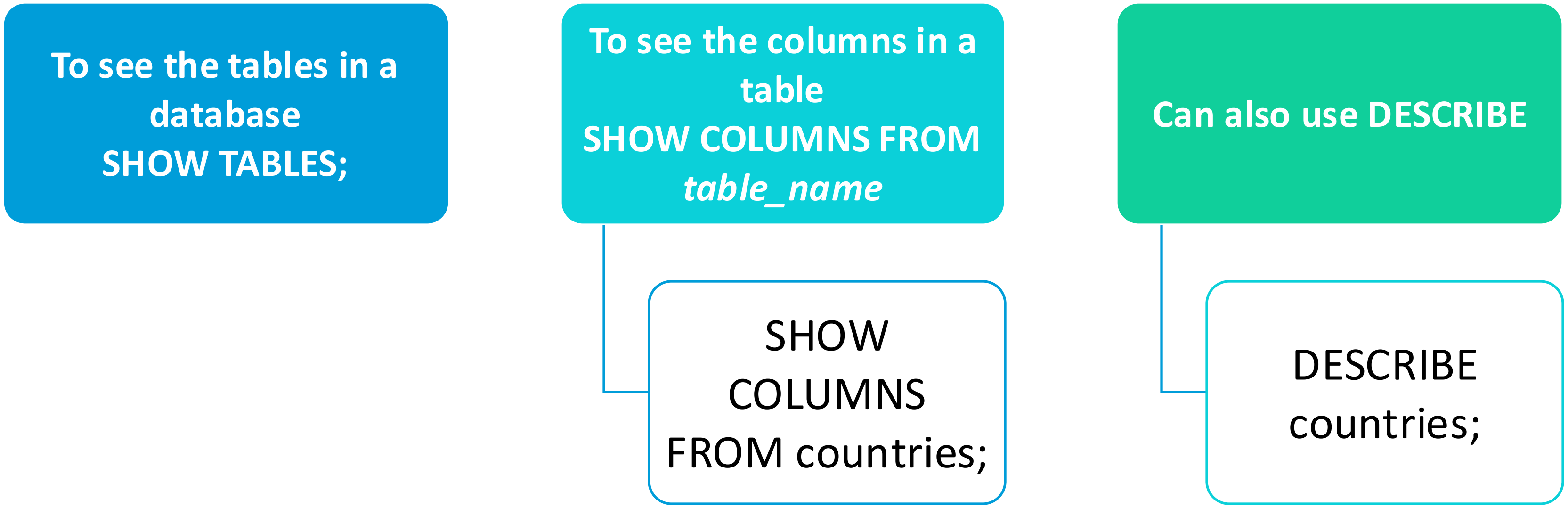
Sample Database

Sample Database

- We will be using the Nations Database for in class exercises. Follow the instructions from the MariaDB website to install the database on your computer.



Getting Information about the Database



To see the tables in a database
SHOW TABLES;

To see the columns in a table
SHOW COLUMNS FROM *table_name*

SHOW COLUMNS FROM countries;

Can also use DESCRIBE

DESCRIBE countries;

Basic SELECT Statements

Retrieving Data

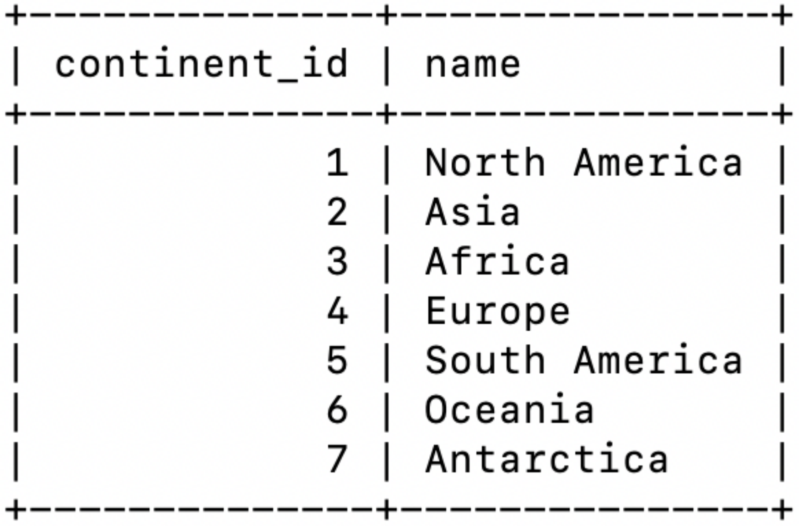
- Use the select statement
  - This is the most commonly used SQL statement
- Has 6 main clauses
  - `SELECT` Specifies which columns are to appear in output.
  - `FROM` Specifies table(s) to be used.
  - `WHERE` Filters rows.
  - `GROUP BY` Forms groups of rows with same column value.
  - `HAVING` Filters groups subject to some condition.
  - `ORDER BY` Specifies the order of the output.

SELECT Statement

- Order of the clauses cannot be changed.
- Only SELECT and FROM are mandatory.

Basic SELECT statement

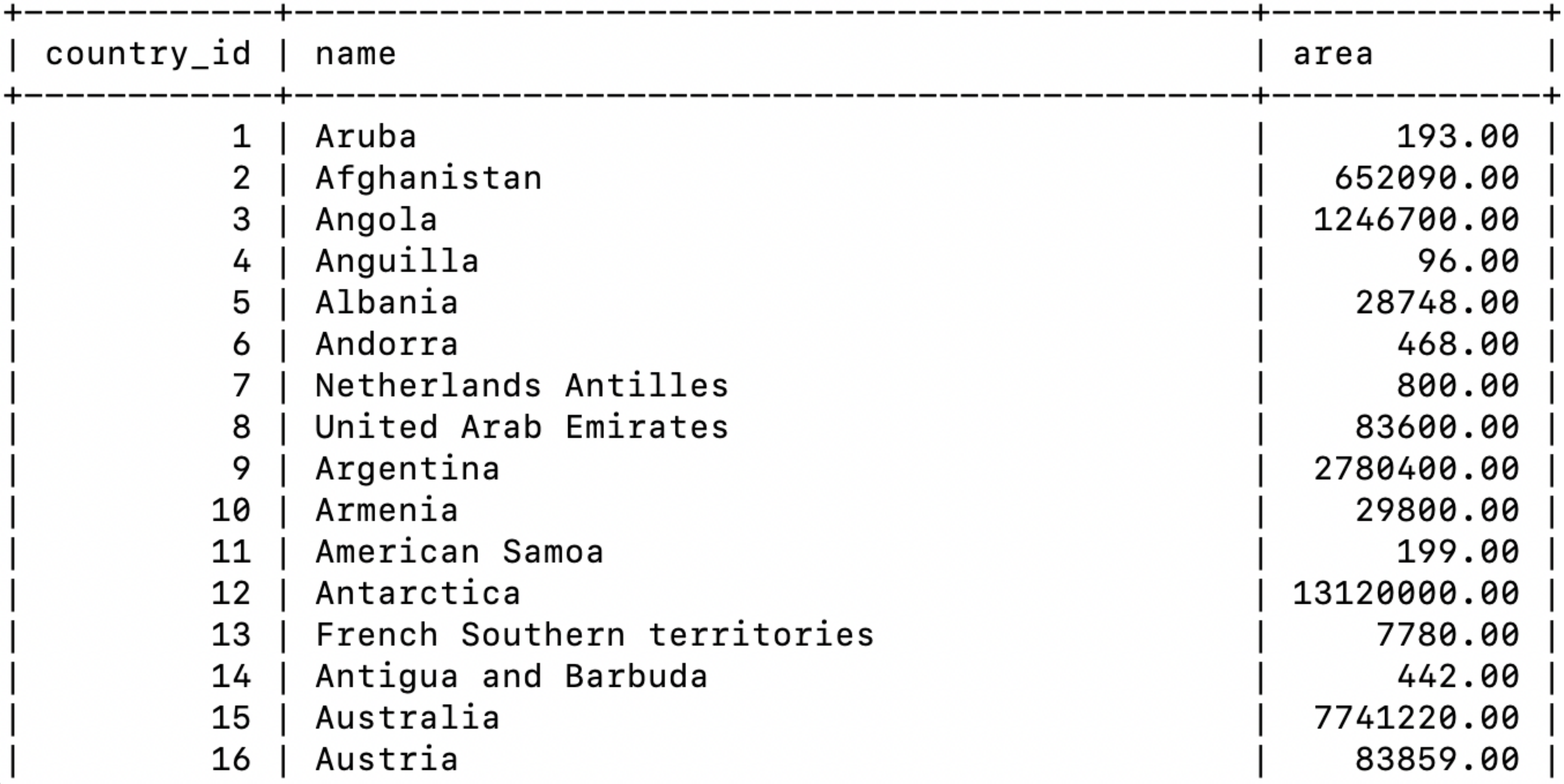
- Basic statement
  - SELECT field1, field2
    FROM table;
  - Example:
    ```
    SELECT continent_id,
    name
    FROM continents;
    ```
- Use * to select all fields
  - Example:
    ```
    SELECT * FROM
    continents;
    ```
- Only use `*` when you need to retrieve all columns from a table.



```
+--------------+---------------+
| continent_id | name          |
+--------------+---------------+
|            1 | North America |
|            2 | Asia          |
|            3 | Africa        |
|            4 | Europe        |
|            5 | South America |
|            6 | Oceania       |
|            7 | Antarctica    |
+--------------+---------------+
```

Selecting Specific Columns, All Rows

- Produce a list of all countries showing id, name and area.
- `SELECT country_id, name, area FROM countries;`



```
+------------+------------------------------------------+------------+
| country_id | name                                     | area       |
+------------+------------------------------------------+------------+
|          1 | Aruba                                    |     193.00 |
|          2 | Afghanistan                              |  652090.00 |
|          3 | Angola                                   | 1246700.00 |
|          4 | Anguilla                                 |      96.00 |
|          5 | Albania                                  |   28748.00 |
|          6 | Andorra                                  |     468.00 |
|          7 | Netherlands Antilles                     |     800.00 |
|          8 | United Arab Emirates                     |   83600.00 |
|          9 | Argentina                                | 2780400.00 |
|         10 | Armenia                                  |   29800.00 |
|         11 | American Samoa                           |     199.00 |
|         12 | Antarctica                               |13120000.00 |
|         13 | French Southern territories              |    7780.00 |
|         14 | Antigua and Barbuda                      |     442.00 |
|         15 | Australia                                | 7741220.00 |
|         16 | Austria                                  |   83859.00 |
```

SELECT – Exercises

- Select the country name, national day and region id from the countries table
- Select all fields from the regions table
- Select the name and continent id for all regions

Limit Clause

- Specifies the maximum number of rows that are returned in the result set
- Example

```
SELECT name, area
FROM countries
LIMIT 10;
```

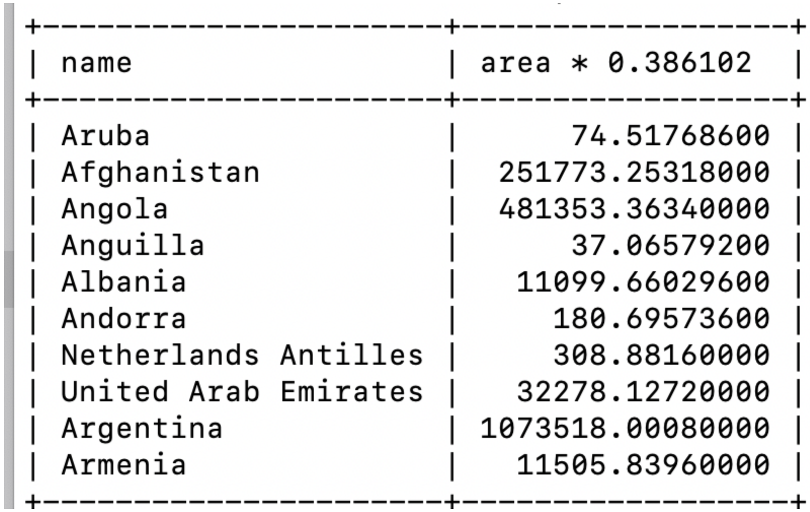
Calculated Values

- You can use arithmetic in SELECT statements
- Operators
  - Multiplication: `*`
  - Division: `/`
  - Integer division: `DIV`
  - Modulo (remainder): `%` or `MOD`
  - Addition: `+`
  - Subtraction: `–`
- Precedence
  - Multiplication, division, modulo
  - Addition, Subtraction
  - Can control using parenthesis

Example: Calculated Fields

- Produce list of countries and area in square miles. The ratio from square mile to square kilometer is 0.386102.

```
SELECT name, area * 0.386102
FROM countries;
```



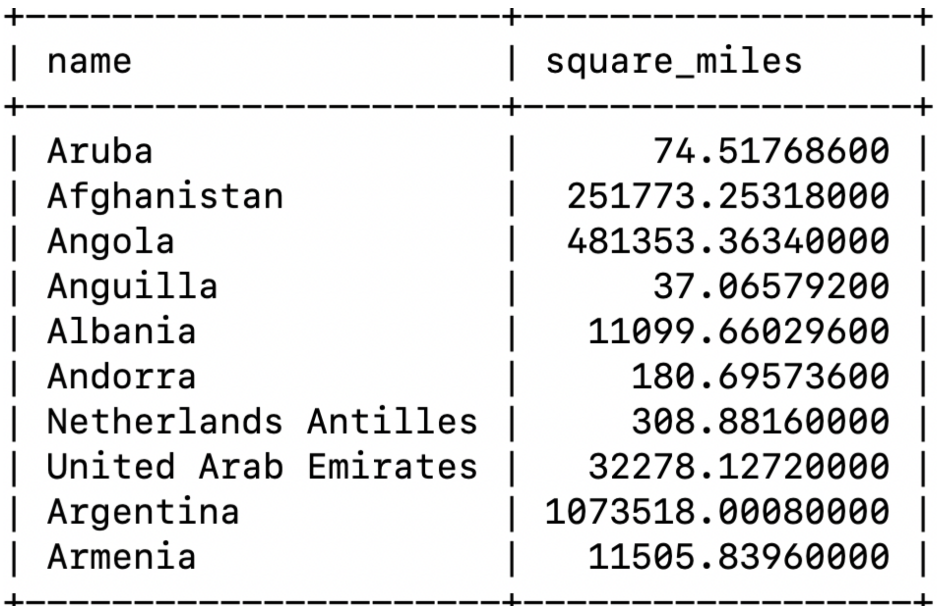
```
+----------------------+------------------+
| name                 | area * 0.386102  |
+----------------------+------------------+
| Aruba                |      74.51768600 |
| Afghanistan          |   251773.25318000 |
| Angola               |   481353.36340000 |
| Anguilla             |      37.06579200 |
| Albania              |    11099.66029600 |
| Andorra              |     180.69573600 |
| Netherlands Antilles |     308.88160000 |
| United Arab Emirates |    32278.12720000 |
| Argentina            | 1073518.00080000 |
| Armenia              |    11505.83960000 |
+----------------------+------------------+
```

Defining a Column Alias

- Renames a column heading in the result set
- Is useful with calculations
- Immediately follows the column name - there can also be the optional AS keyword between the column name and alias
- Requires double quotation marks if it contains spaces or special characters or is case sensitive

Example: Calculated Fields with Alias

```
SELECT name, area * 0.386102 as square_miles
FROM countries
LIMIT 10;
```

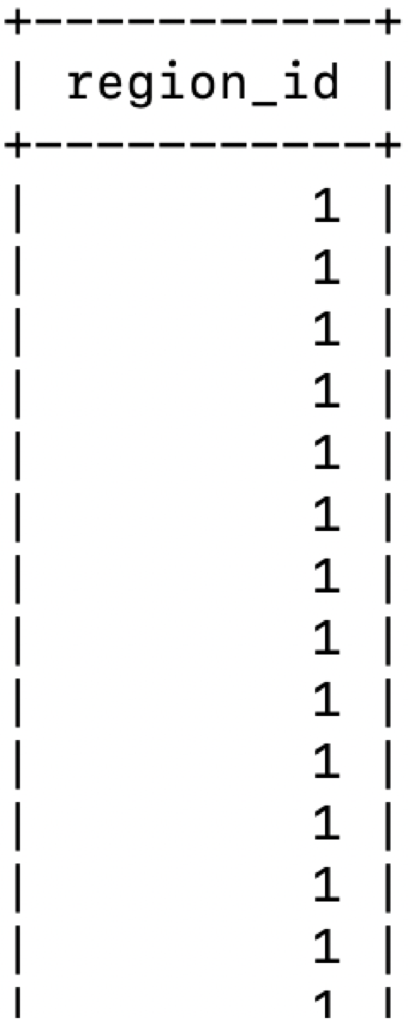


```
+----------------------+-----------------+
| name                 | square_miles    |
+----------------------+-----------------+
| Aruba                |     74.51768600 |
| Afghanistan          |  251773.25318000 |
| Angola               |  481353.36340000 |
| Anguilla             |     37.06579200 |
| Albania              |   11099.66029600 |
| Andorra              |    180.69573600 |
| Netherlands Antilles |    308.88160000 |
| United Arab Emirates |   32278.12720000 |
| Argentina            | 1073518.00080000 |
| Armenia              |   11505.83960000 |
+----------------------+-----------------+
```

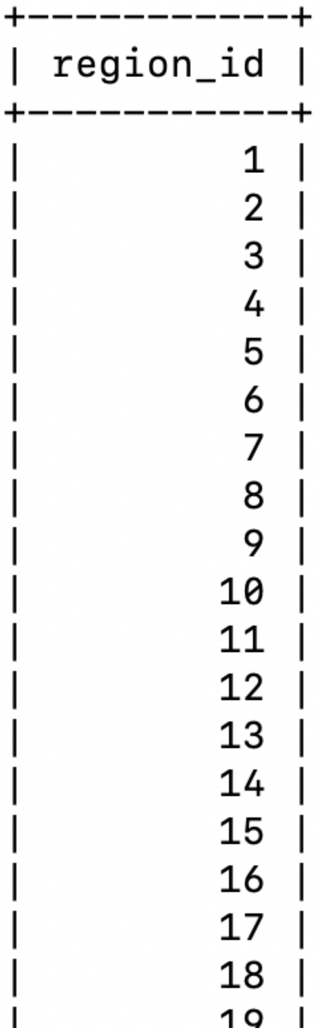
Removing Duplicate Rows

- List the region ids used in the countries table.

```
SELECT region_id
FROM countries;
```



```
+-----------+
| region_id |
+-----------+
|         1 |
|         1 |
|         1 |
|         1 |
|         1 |
|         1 |
|         1 |
|         1 |
|         1 |
|         1 |
|         1 |
|         1 |
|         1 |
|         1 |
```



```
+-----------+
| region_id |
+-----------+
|         1 |
|         2 |
|         3 |
|         4 |
|         5 |
|         6 |
|         7 |
|         8 |
|         9 |
|        10 |
|        11 |
|        12 |
|        13 |
|        14 |
|        15 |
|        16 |
|        17 |
|        18 |
|        19 |
```

Use of DISTINCT

- Use DISTINCT to eliminate duplicates:

```
SELECT DISTINCT region_id
FROM countries;
```

WHERE Clause

Retrieving data with specific criteria

- Use a WHERE clause
- Example:

```
SELECT name
FROM countries
WHERE region_id = 15;
```

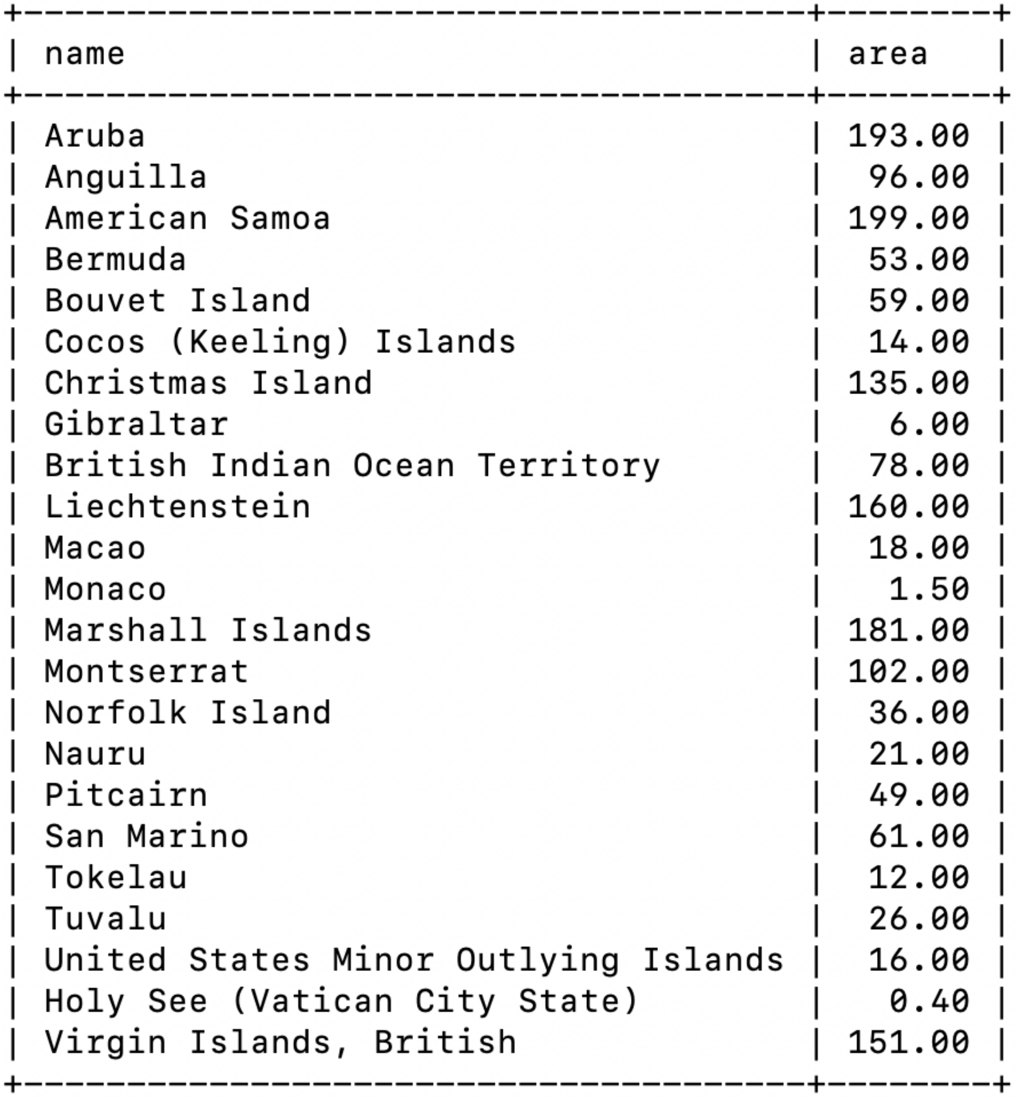
Comparison Operators

- Equality: =
- Less than: <
- Greater than: >
- Less than or equal to: <=
- Greater than or equal to: >=

Example: Comparison Search Condition

- List all countries with an area less than 200 square kilometers.

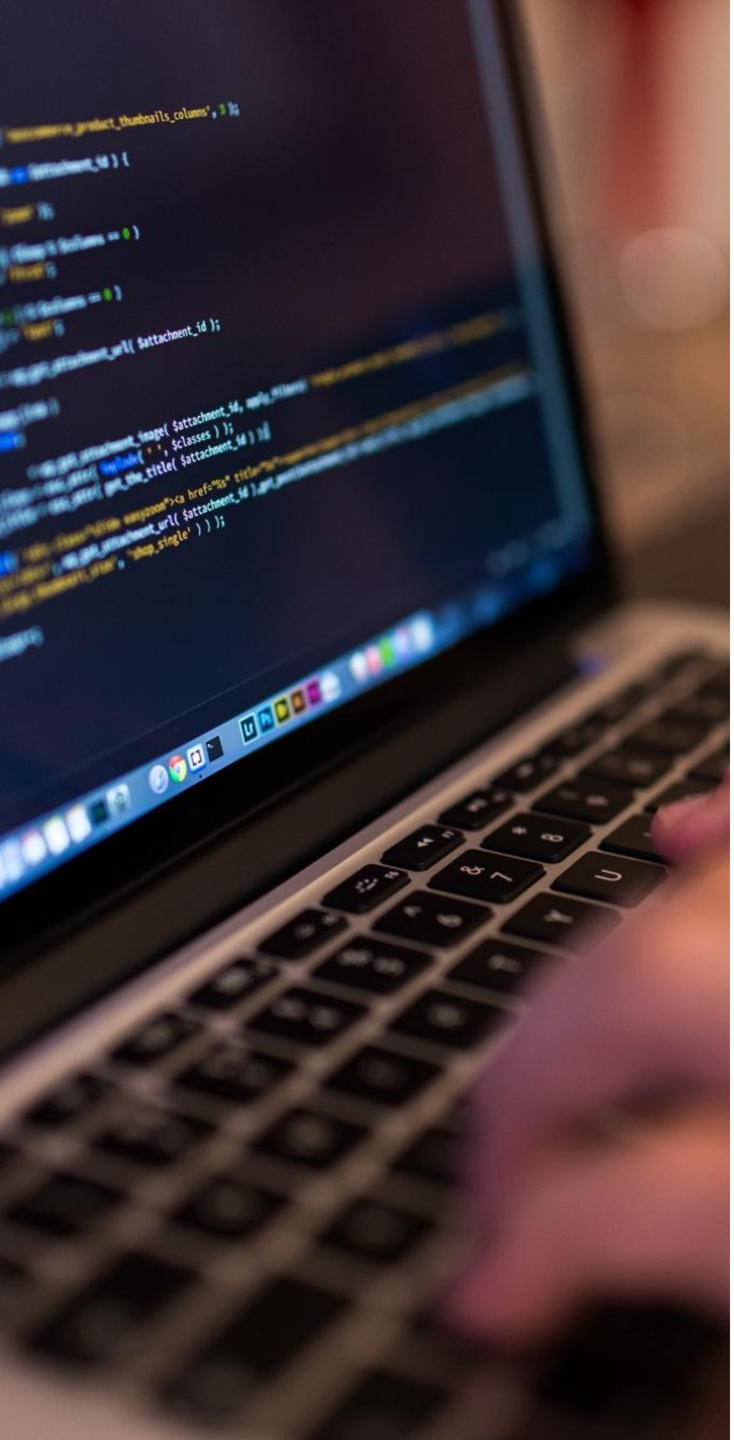
```
SELECT name, area
FROM countries
WHERE area < 200;
```



```
+--------------------------------------+--------+
| name                                 | area   |
+--------------------------------------+--------+
| Aruba                                | 193.00 |
| Anguilla                             |  96.00 |
| American Samoa                       | 199.00 |
| Bermuda                              |  53.00 |
| Bouvet Island                        |  59.00 |
| Cocos (Keeling) Islands              |  14.00 |
| Christmas Island                     | 135.00 |
| Gibraltar                            |   6.00 |
| British Indian Ocean Territory       |  78.00 |
| Liechtenstein                        | 160.00 |
| Macao                                |  18.00 |
| Monaco                               |   1.50 |
| Marshall Islands                     | 181.00 |
| Montserrat                           | 102.00 |
| Norfolk Island                       |  36.00 |
| Nauru                                |  21.00 |
| Pitcairn                             |  49.00 |
| San Marino                           |  61.00 |
| Tokelau                              |  12.00 |
| Tuvalu                               |  26.00 |
| United States Minor Outlying Islands |  16.00 |
| Holy See (Vatican City State)        |   0.40 |
| Virgin Islands, British              | 151.00 |
+--------------------------------------+--------+
```

Comparing Data

- You can compare data of unlike types, but it may produce unexpected results
- When comparing to a string literal or a date literal, you must enclose the literal in quotes
- Character comparisons are not case sensitive



Comparison – Exercises

- Find all countries with a region id of 5.
- Find all regions with a continent id of 5.
- Find all countries with a national day later than January 1, 1990.

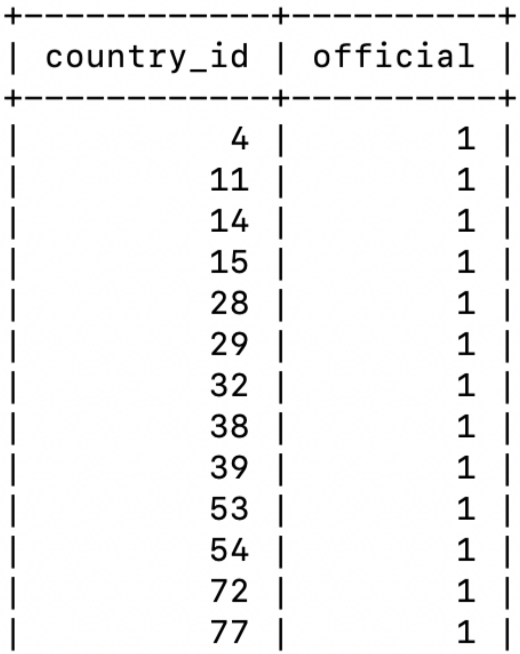
Compound Search Conditions

- Use logical operators
- AND
- OR
- NOT

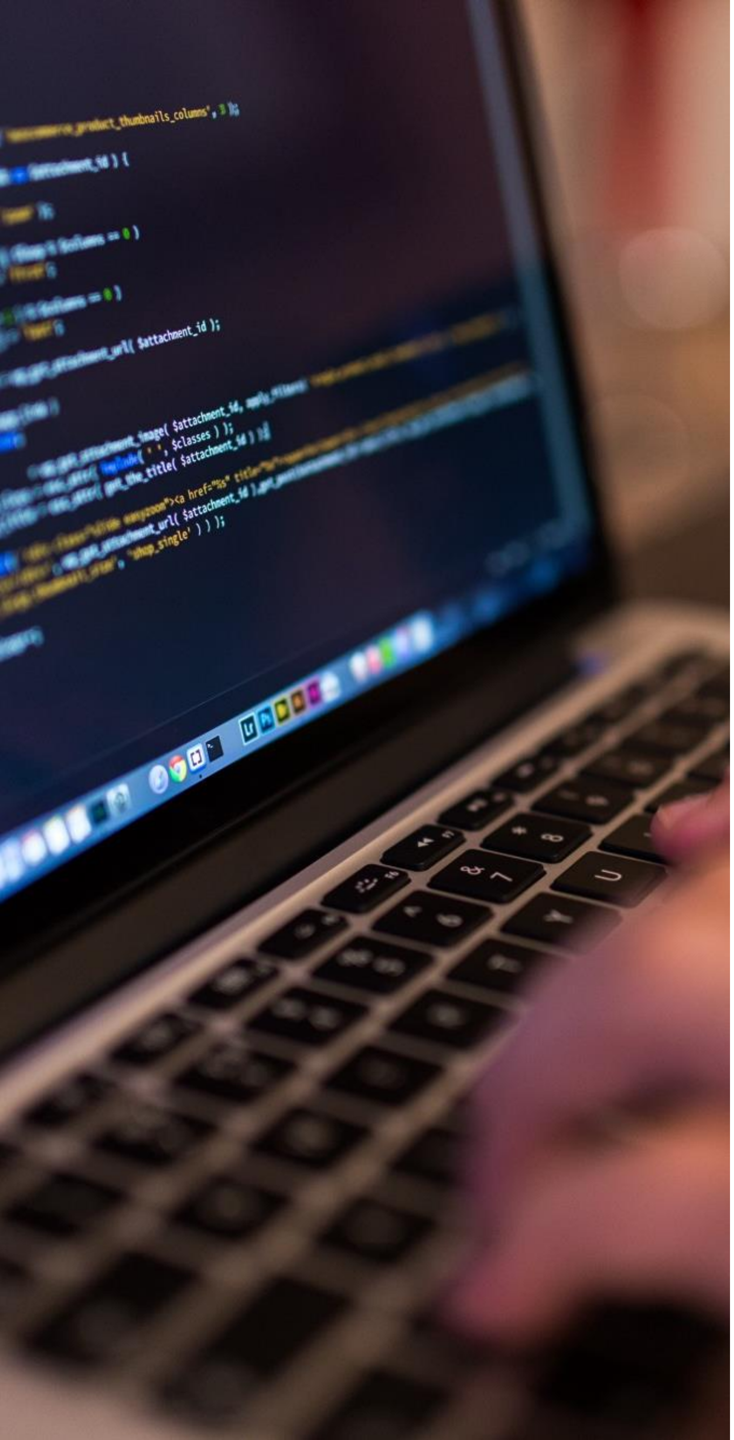
Example: Compound Comparison Search Condition

- Find all countries where the official language is English (language_id 2)

```
SELECT country_id, official
FROM country_languages
WHERE language_id = 2 and official = 1;
```



```
+------------+----------+
| country_id | official |
+------------+----------+
|          4 |        1 |
|         11 |        1 |
|         14 |        1 |
|         15 |        1 |
|         28 |        1 |
|         29 |        1 |
|         32 |        1 |
|         38 |        1 |
|         39 |        1 |
|         53 |        1 |
|         54 |        1 |
|         72 |        1 |
|         77 |        1 |
```



Compound Search Conditions – Exercises

- Find all countries in region 5 with an area of more than 500,000 square km
- Find all country_ids for countries with a population greater 50,000,000 in 1970
- Find all country_ids for countries with an unofficial language of French (language id 23)

Range Search

- Use the BETWEEN operator
  - Compares the value of a test expression to see if it falls within the range of the given values
- BETWEEN test includes the endpoints of range.
- Example:

```
SELECT name
FROM countries
WHERE national_day BETWEEN "1990-
01-01" AND "1990-12-31";
```

- Also a negated version NOT BETWEEN.
- BETWEEN does not add much to SQL's expressive power.
  - Can use relational operators

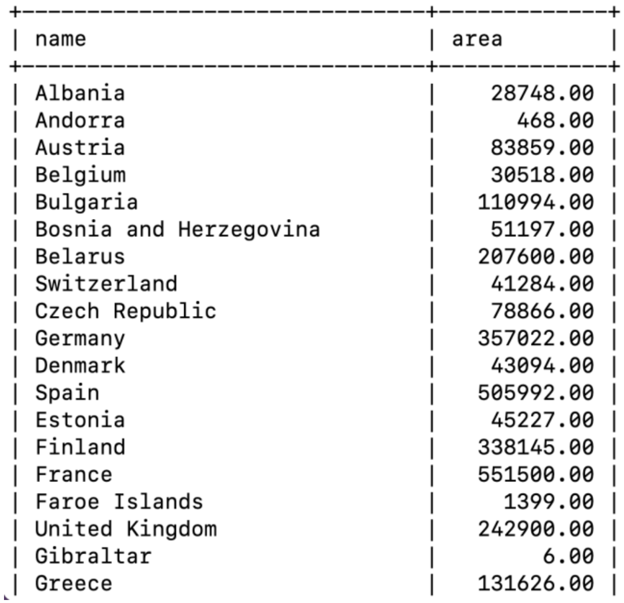
Set Membership

- `IN` operator
  - Used to test whether a value is in a specified list of items
- There is a negated version (NOT IN).
- IN does not add much to SQL's expressive power
  - Use OR operator instead
- IN is more efficient when set contains many values.

Example: Set Membership

- List all countries in Europe.

```
SELECT name, area
FROM countries
WHERE region_id IN (4, 10, 13, 19, 21, 24);
```



```
+--------------------------------+-------------+
| name                           | area        |
+--------------------------------+-------------+
| Albania                        |    28748.00 |
| Andorra                        |      468.00 |
| Austria                        |    83859.00 |
| Belgium                        |    30518.00 |
| Bulgaria                       |   110994.00 |
| Bosnia and Herzegovina         |    51197.00 |
| Belarus                        |   207600.00 |
| Switzerland                    |    41284.00 |
| Czech Republic                 |    78866.00 |
| Germany                        |   357022.00 |
| Denmark                        |    43094.00 |
| Spain                          |   505992.00 |
| Estonia                        |    45227.00 |
| Finland                        |   338145.00 |
| France                         |   551500.00 |
| Faroe Islands                  |     1399.00 |
| United Kingdom                 |   242900.00 |
| Gibraltar                      |        6.00 |
| Greece                         |   131626.00 |
```

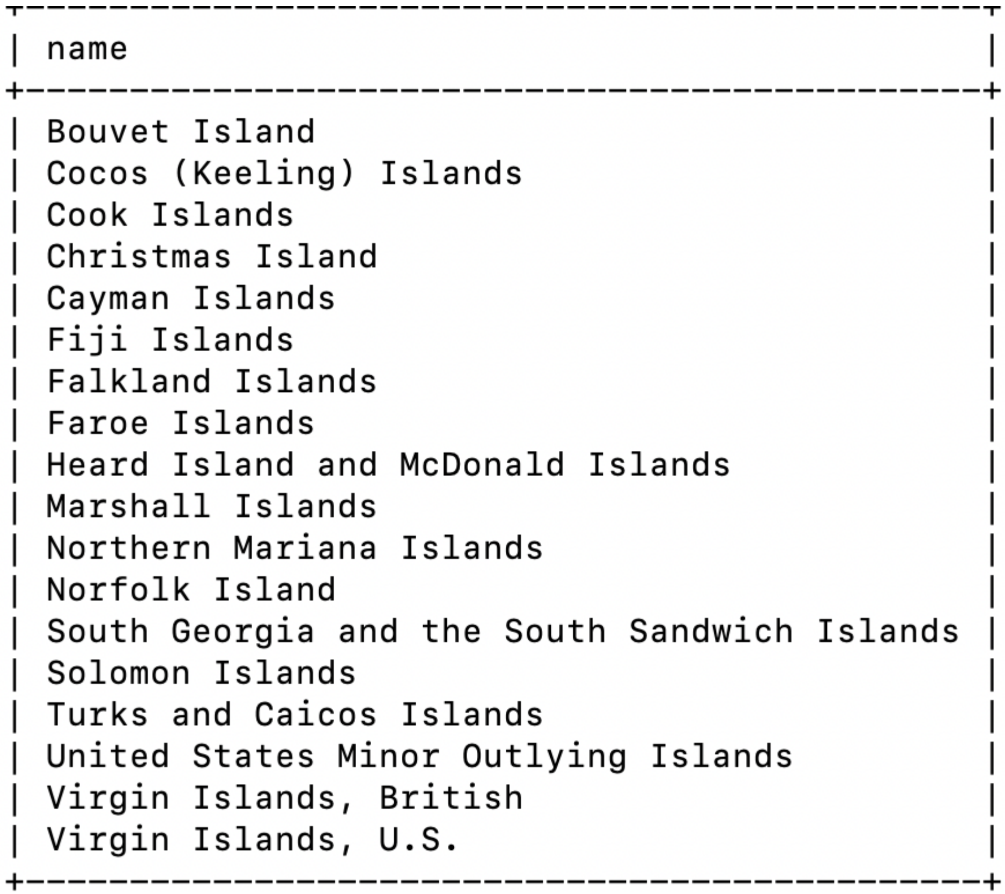
Pattern Matching

- Uses LIKE and NOT LIKE and regular expressions
  - %: sequence of zero or more characters;
  - _ (underscore): any single character.
- Can degrade performance so use sparingly
- Example : `LIKE '%Staff%'` means a sequence of characters of any length containing Staff.

Example: Pattern Matching

- Find all countries who with "island" in their name.

```
Select name
FROM countries
WHERE name LIKE
"%island%";
```



```
+-----------------------------------------------+
| name                                          |
+-----------------------------------------------+
| Bouvet Island                                 |
| Cocos (Keeling) Islands                       |
| Cook Islands                                  |
| Christmas Island                              |
| Cayman Islands                                |
| Fiji Islands                                  |
| Falkland Islands                              |
| Faroe Islands                                 |
| Heard Island and McDonald Islands             |
| Marshall Islands                              |
| Northern Mariana Islands                      |
| Norfolk Island                                |
| South Georgia and the South Sandwich Islands |
| Solomon Islands                               |
| Turks and Caicos Islands                      |
| United States Minor Outlying Islands          |
| Virgin Islands, British                       |
| Virgin Islands, U.S.                          |
+-----------------------------------------------+
```

Null Values

Represents value for an attribute that is currently unknown or not applicable for row.

Deals with incomplete or exceptional data.

Represents the absence of a value and is not the same as zero or spaces, which are values.

Null Values

- Can test for null values using
  - `IS NOT NULL`
  - `IS NULL`
- Example:

```
SELECT name
FROM countries
WHERE national_day is NULL;
```

NOT NULL constraint

- Columns may contain NULL values
- The ***NOT NULL*** constraint prevents a column from having a NULL value

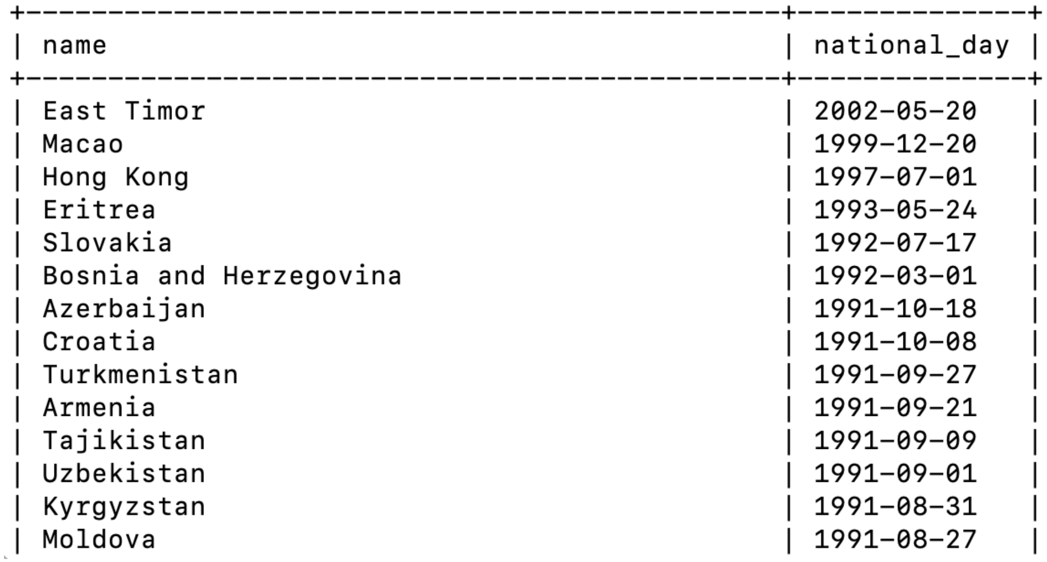
```
CREATE TABLE Coach
(
 id    SMALLINT UNSIGNED,
 name  VARCHAR(50)  NOT NULL,
 email VARCHAR(100) NOT NULL
);
```

ORDER BY Clause

Retrieving Data in a particular order

- Use the `ORDER BY` clause
  - Sorts the rows on one or more columns
- Default order is ascending
  - Can specify using the keyword `ASC`
  - Example:
    `ORDER BY name`
    is the same as
    `ORDER BY name ASC`
- To sort in descending order, use the keyword `DESC`
  - Example:
    `ORDER BY name DESC`

Example: Single Column Ordering

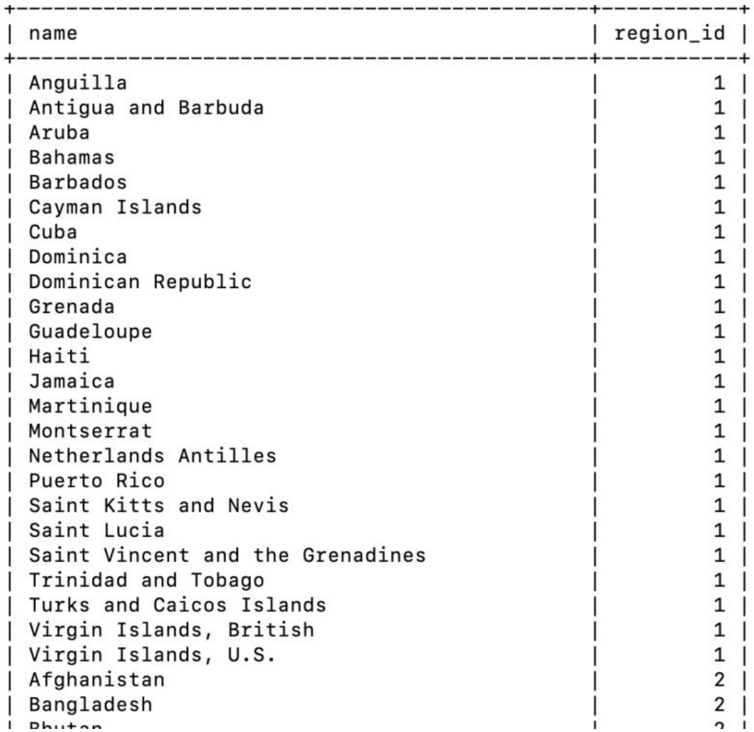


```
+------------------------------------------------+--------------+
| name                                           | national_day |
+------------------------------------------------+--------------+
| East Timor                                     | 2002-05-20   |
| Macao                                          | 1999-12-20   |
| Hong Kong                                      | 1997-07-01   |
| Eritrea                                        | 1993-05-24   |
| Slovakia                                       | 1992-07-17   |
| Bosnia and Herzegovina                         | 1992-03-01   |
| Azerbaijan                                     | 1991-10-18   |
| Croatia                                        | 1991-10-08   |
| Turkmenistan                                   | 1991-09-27   |
| Armenia                                        | 1991-09-21   |
| Tajikistan                                     | 1991-09-09   |
| Uzbekistan                                     | 1991-09-01   |
| Kyrgyzstan                                     | 1991-08-31   |
| Moldova                                        | 1991-08-27   |
```

- List national day for all countries, arranged in descending order of national day.

```
SELECT name,
national_day
FROM countries
ORDER BY
national_day DESC;
```

Example: Multiple Column Ordering



```
+-----------------------------------------------+-----------+
| name                                          | region_id |
+-----------------------------------------------+-----------+
| Anguilla                                      |         1 |
| Antigua and Barbuda                           |         1 |
| Aruba                                         |         1 |
| Bahamas                                       |         1 |
| Barbados                                      |         1 |
| Cayman Islands                                |         1 |
| Cuba                                          |         1 |
| Dominica                                      |         1 |
| Dominican Republic                            |         1 |
| Grenada                                       |         1 |
| Guadeloupe                                    |         1 |
| Haiti                                         |         1 |
| Jamaica                                       |         1 |
| Martinique                                    |         1 |
| Montserrat                                    |         1 |
| Netherlands Antilles                          |         1 |
| Puerto Rico                                   |         1 |
| Saint Kitts and Nevis                         |         1 |
| Saint Lucia                                   |         1 |
| Saint Vincent and the Grenadines              |         1 |
| Trinidad and Tobago                           |         1 |
| Turks and Caicos Islands                      |         1 |
| Virgin Islands, British                       |         1 |
| Virgin Islands, U.S.                          |         1 |
| Afghanistan                                   |         2 |
| Bangladesh                                    |         2 |
| Bhutan                                        |         2 |
```

- Produce a list of countries in each region ordered by region and country name.

```
SELECT name,
region_id
FROM countries
ORDER by region_id,
name;
```

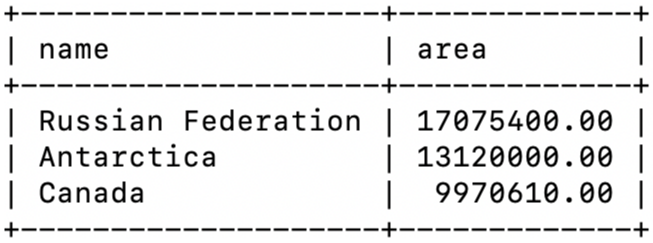
Sort by other criteria

- Can also sort by
  - A column that uses an alias
  - A calculation

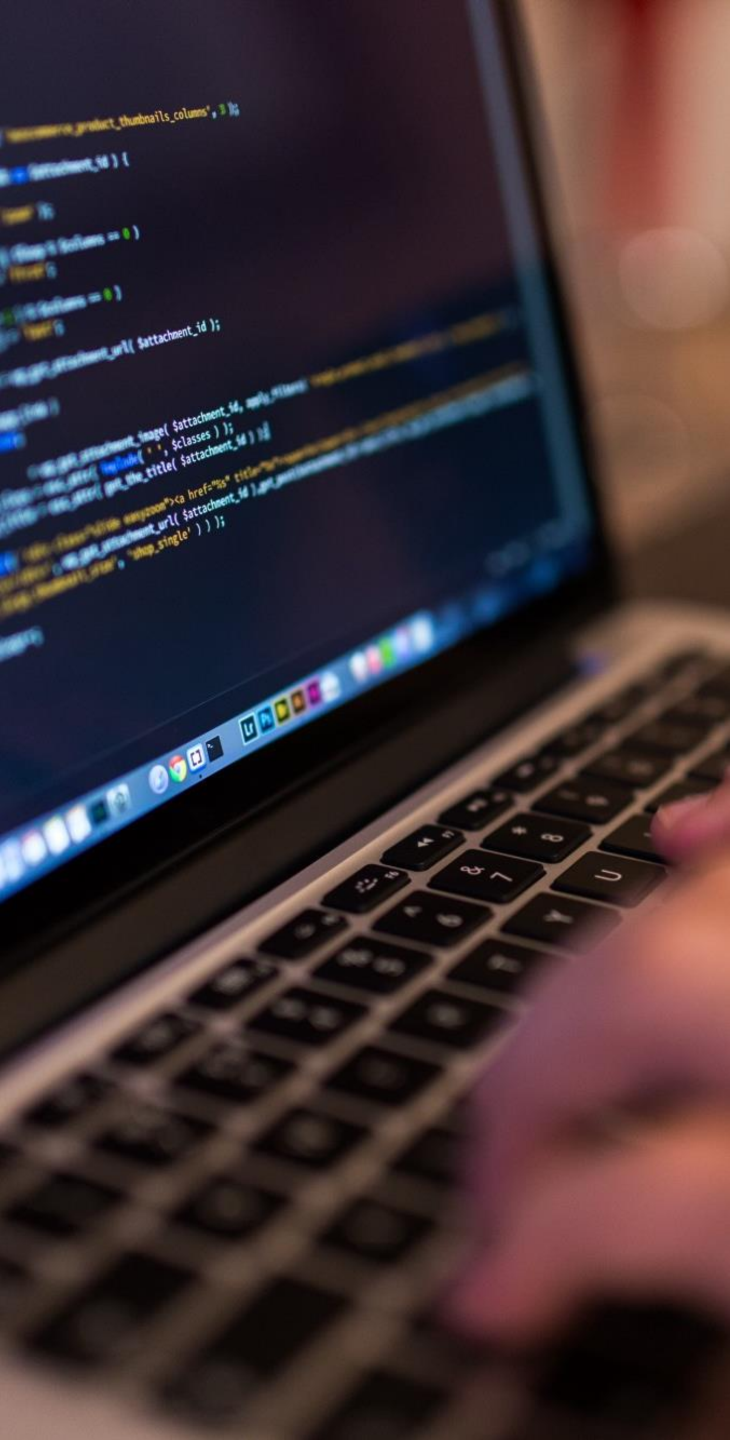
Limit Clause and Sorting

- When combining `LIMIT` and `ORDER BY` you can select the n largest or n smallest values
- Example

```
SELECT name, area
FROM countries
ORDER BY area DESC
LIMIT 3;
```



```
+--------------------+-------------+
| name               | area        |
+--------------------+-------------+
| Russian Federation | 17075400.00 |
| Antarctica         | 13120000.00 |
| Canada             |  9970610.00 |
+--------------------+-------------+
```



Sorting Exercises

- Select regions in ascending order alphabetically by region name.
- Repeat the above query but display in descending order.
- Select countries and area by region id and within each region order by area with the largest area first.