# SELECT within SELECT Tutorial

| Language: | English • 日本語 • 中文 |
|---|---|

This tutorial looks at how we can use SELECT statements within SELECT statements to perform more complex queries.

| name | continent | area | population | gdp |
|---|---|---|---|---|
| Afghanistan | Asia | 652230 | 25500100 | 20343000000 |
| Albania | Europe | 28748 | 2831741 | 12960000000 |
| Algeria | Africa | 2381741 | 37100000 | 188681000000 |
| Andorra | Europe | 468 | 78115 | 3712000000 |
| Angola | Africa | 1246700 | 20609294 | 100990000000 |
| ... | | | | |

Using nested SELECT

Summary

## Contents



## Exercises

## Bigger than Russia

# 1.

**List each country name where the population is larger than that of 'Russia'.**

```
world(name, continent, area, population, gdp)
```

```
SELECT name FROM world
WHERE population >
  (SELECT population FROM world
   WHERE name='Russia')
```

Submit SQL Restore default

result

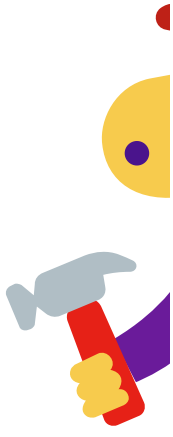
# 500 Internal Server Error

Sorry, something went wrong.

A team of highly trained monkeys has been dispatched to deal with this situation.

If you see them, send them this information as text (screenshots frighten them):


```
APkpgMW5e7R96dmZ17TdxwqLaAsCyiWQMoPdUxEUN4uFtP__tDFqaKUx
kbFKZZNFH7akpOtWi0OPF91CEKwW-6UpgrBSUwJU1wC7xvbxH1Rm-bPs
J60VDLINgOQS8Dm8vdQKB7ohIG48vidb7IQ_fhSnVDjHi6_7DxDUmar8
CLs-0XEAJg2VJ61gJSE4TA0G7Za9AHm2RfD9yMNQi7efI4zTOiq3Kq6r
xmYpzrl8iKzfP31jKcnDhoeblfJ9a6jrt4QooQW7gnlvFqoD25IQyKVS
i5o3jzBxjrofstUzvY3cnRwm8MjSb5pPFiqHjTNr8qqcUnzsmmmHGuxG
7atO9yxiiG9mYLjTlkDEn4JQMwdtDCbtOgHvEpb-LaIzV3txDXhrOC8K
iXoHARWVej49DP5WfeZwR5WDw8aLjHV6m7lcEhpKdLu4cnivaRLqEnuk
1pcGBI2FLb-s6naBQvK4aGTa7uNCUy1Esn54JOuxDbYVqzHrhyH_8RKn
5wAx-EIMdjRkCrAsPx6VvrZtnl5UY86L3_R-ugZFA1Tcz20u_a_EaKTd
RZS_tNlKlcn5nzjsHq7uugX1gbDQKijZvTzH2Ypgq8niJn7eJ3we8avY
9gQUoDpJSjQlWw8HPDN1WiM7bm5j0kjruahIBOjsylXx0toCBVfjZq72
5S1J-QL VKsvPXoXtFGwW8vcCaaVNQlI P1ElE5TuSQDS4kQF1mLEbF9
```

# Richer than UK

# 2.

**Show the countries in Europe with a per capita GDP greater than 'United Kingdom'.**

*Per Capita GDP*

```
SELECT name FROM world
WHERE gdp/population > (SELECT gdp/population FROM world
               WHERE name = 'United Kingdom')
      AND continent = 'Europe'
```

Submit SQL

Restore default

result


# Neighbours of Argentina and Australia

# 3.

**List the name and continent of countries in the continents containing either Argentina or Australia. Order by name of the country.**

```
SELECT name, continent
FROM world
WHERE (continent = (SELECT continent FROM world
                      WHERE name = 'Argentina') )
        OR
      (continent = (SELECT continent FROM world
                      WHERE name = 'Australia') )
ORDER BY name
```

Submit SQL Restore default

result

# Between Canada and Poland

# 4.

**Which country has a population that is more than Canada but less than Poland? Show the name and the population.**

```
SELECT name, population
FROM world
WHERE (population > (SELECT population FROM world
                     WHERE name = 'Canada') )
        AND
      (population < (SELECT population FROM world
                     WHERE name = 'Poland') )
```

Submit SQL Restore default

result

# Percentages of Germany

# 5.

Germany (population 80 million) has the largest population of the countries in Europe. Austria (population 8.5 million) has 11% of the population of Germany.

**Show the name and the population of each country in Europe. Show the population as a percentage of the population of Germany.**

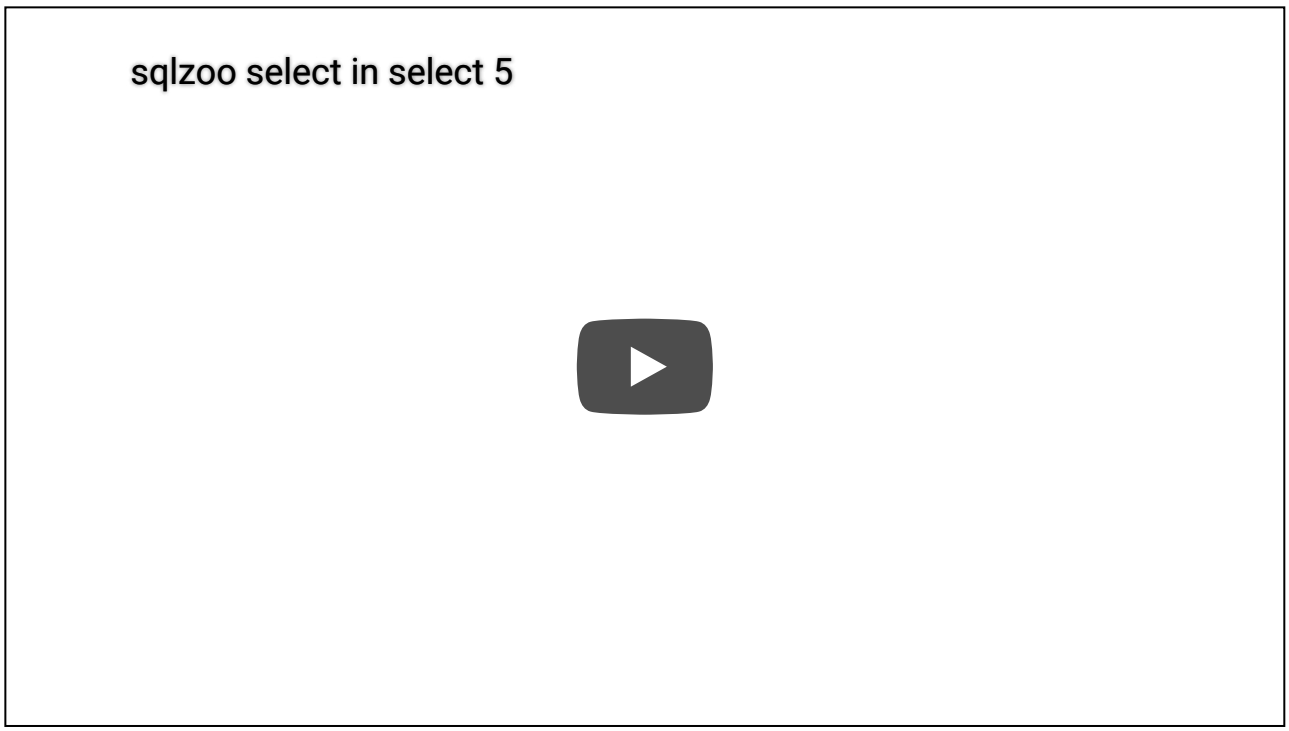
*Decimal places*
*Percent symbol %*

```
SELECT name,
       CONCAT(ROUND(100*population/(SELECT population FROM world
                                    WHERE name = 'Germany')), '%')
FROM world
WHERE continent = 'Europe'
```

Submit SQL

Restore default

result




To get a well rounded view of the important features of SQL you should move on to the next tutorial concerning aggregates.

To gain an absurdly detailed view of one insignificant feature of the language, read on.

We can use the word ALL to allow >= or > or < or <=to act over a list. For example, you can find the largest country in the world, by population with this query:

```
SELECT name
  FROM world
 WHERE population >= ALL(SELECT population
                           FROM world
                          WHERE population>0)
```

You need the condition **population>0** in the sub-query as some countries have **null** for population.

# Bigger than every country in Europe

# 6.

**Which countries have a GDP greater than every country in Europe? [Give the name only.] (Some countries may have NULL gdp values)**

```
SELECT name
FROM world
WHERE gdp > ALL(SELECT gdp
                FROM world
                WHERE gdp > 0 AND continent = 'Europe')
```

Submit SQL Restore default

result

We can refer to values in the outer SELECT within the inner SELECT. We can name the tables so that we can tell the difference between the inner and outer versions.

# Largest in each continent

# 7.

**Find the largest country (by area) in each continent, show the continent, the name and the area:**

```
SELECT continent, name, area
FROM world x
  WHERE area >= ALL
    (SELECT area FROM world y
        WHERE y.continent=x.continent
          AND area>0)
```

Submit SQL Restore default

The above example is known as a **correlated** or **synchronized** sub-query.

### *Using correlated subqueries*

result

# First country of each continent (alphabetically)

# 8.

**List each continent and the name of the country that comes first alphabetically.**

```
SELECT continent, name
FROM world x
WHERE name <= ALL
   (SELECT name FROM world y
    WHERE y.continent = x.continent)
```

Submit SQL

Restore default

result

# Difficult Questions That Utilize Techniques Not Covered In Prior Sections

# 9.

**Find the continents where all countries have a population <= 25000000. Then find the names of the countries associated with these continents. Show name, continent and population.**

```
SELECT name, continent, population
FROM world
WHERE continent = ANY (SELECT DISTINCT continent
                   FROM world x
                   WHERE 25000000 >= ALL(SELECT population FROM world y
                                         WHERE x.continent=y.continent) )
```

Submit SQL

Restore default

result

# 10.

**Some countries have populations more than three times that of any of their neighbours (in the same continent). Give the countries and continents.**

```
SELECT name, continent
FROM world x
WHERE population/3 >= ALL(SELECT population FROM world y
                          WHERE x.continent=y.continent AND x.name<>y.name)
```

Submit SQL Restore default

result

Nested SELECT Quiz