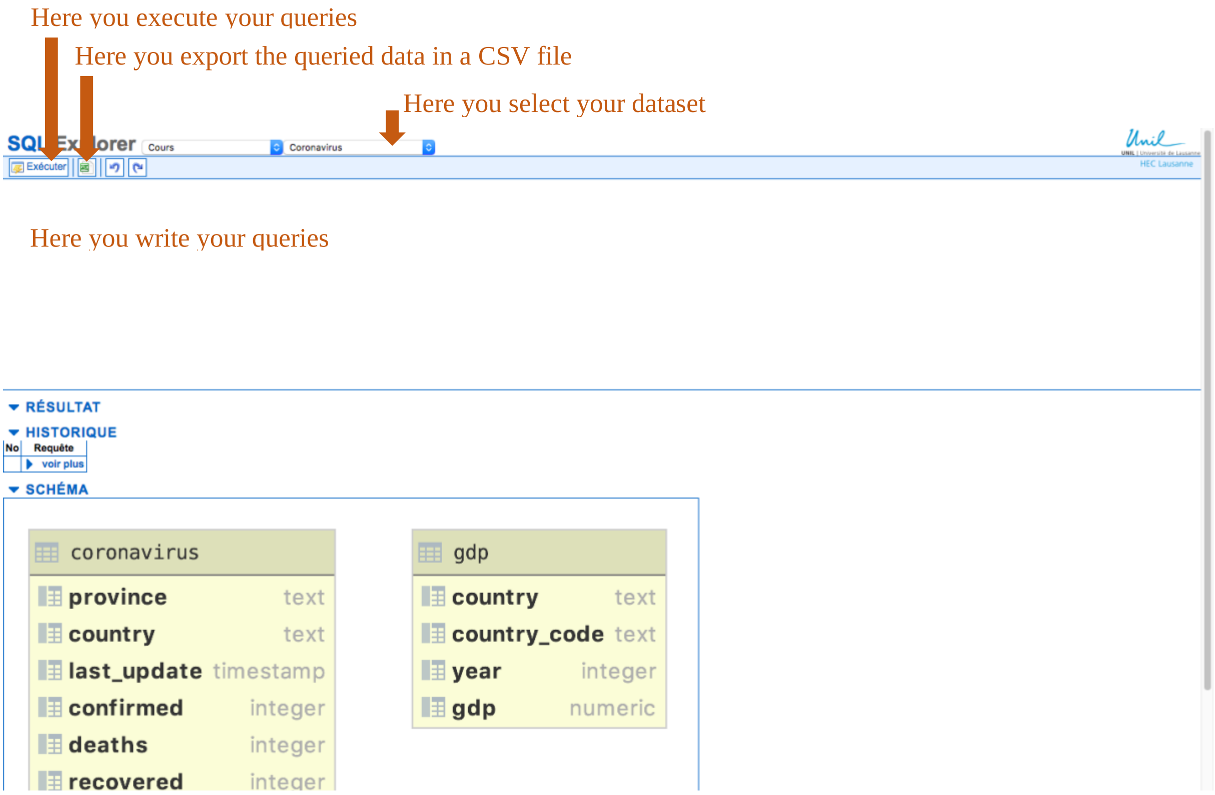
WEEK 5 - SQL Exercises

1) Walkthrough

<u>What is SQL Explorer?</u>
SQL Explorer is a reliable and effective tool used to execute and create SQL queries and manage multiple database connections and provides a simple and intuitive interface.

Access: https://hec.unil.ch/info1ere/sqlexplorer



Here you execute your queries

Here you export the queried data in a CSV file

Here you select your dataset

Here you write your queries

coronavirus	
province	text
country	text
last_update	timestamp
confirmed	integer
deaths	integer
recovered	integer

gdp	
country	text
country_code	text
year	integer
gdp	numeric

Today we will use the **coronavirus** and **temperatures** datasets.

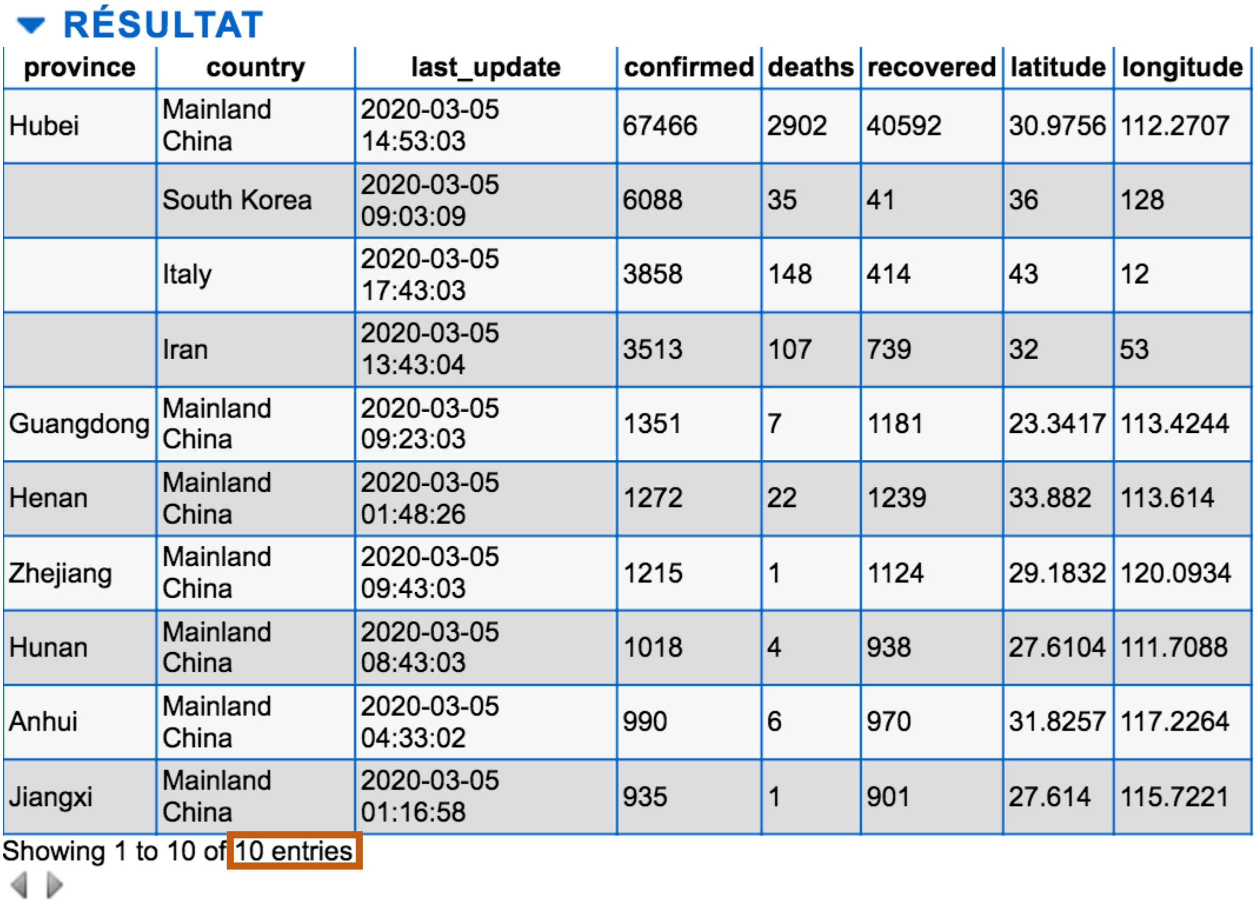
Let's start with **coronavirus**:

Have an overview of the **coronavirus** table:

```
select * from coronavirus limit 10
```

We only select the ten first ROWS the dataset

We select all COLUMNS from the dataset



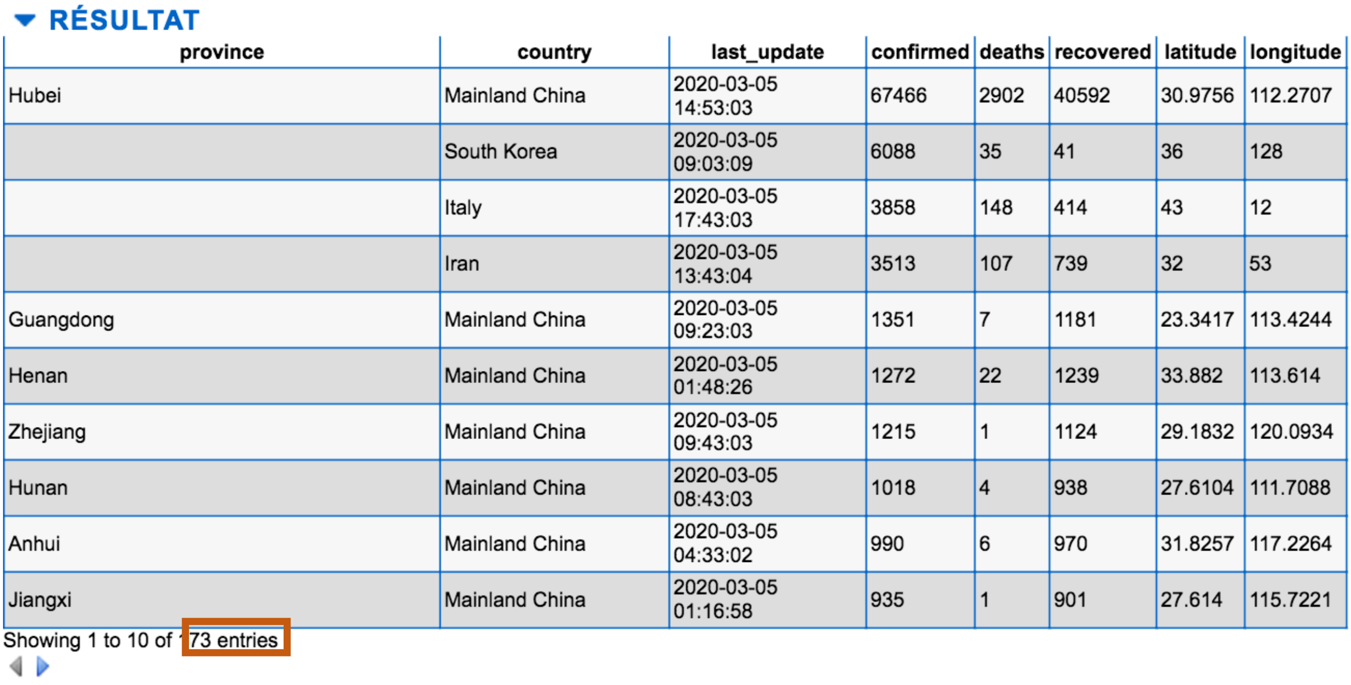
RÉSULTAT

province	country	last_update	confirmed	deaths	recovered	latitude	longitude
Hubei	Mainland China	2020-03-05 14:53:03	67466	2902	40592	30.9756	112.2707
	South Korea	2020-03-05 09:03:09	6088	35	41	36	128
	Italy	2020-03-05 17:43:03	3858	148	414	43	12
	Iran	2020-03-05 13:43:04	3513	107	739	32	53
Guangdong	Mainland China	2020-03-05 09:23:03	1351	7	1181	23.3417	113.4244
Henan	Mainland China	2020-03-05 01:48:26	1272	22	1239	33.882	113.614
Zhejiang	Mainland China	2020-03-05 09:43:03	1215	1	1124	29.1832	120.0934
Hunan	Mainland China	2020-03-05 08:43:03	1018	4	938	27.6104	111.7088
Anhui	Mainland China	2020-03-05 04:33:02	990	6	970	31.8257	117.2264
Jiangxi	Mainland China	2020-03-05 01:16:58	935	1	901	27.614	115.7221

Showing 1 to 10 of 10 entries

Select all columns and all rows:

```
select * from coronavirus
```



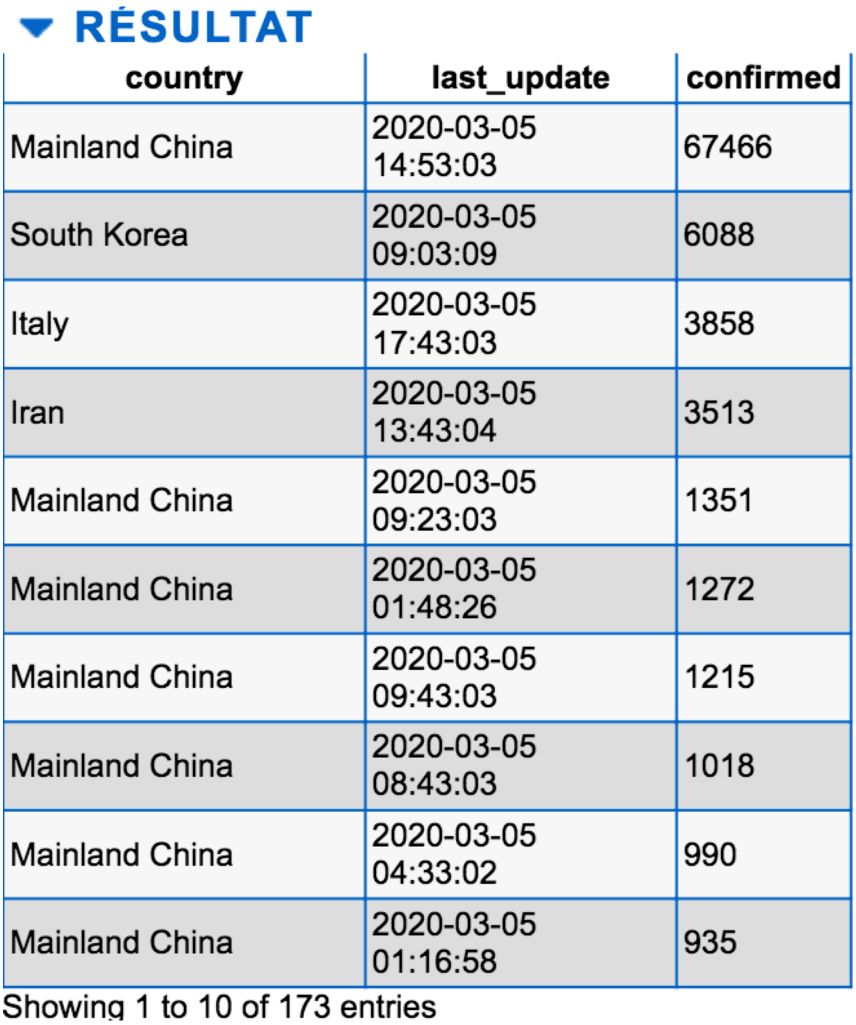
RÉSULTAT

province	country	last_update	confirmed	deaths	recovered	latitude	longitude
Hubei	Mainland China	2020-03-05 14:53:03	67466	2902	40592	30.9756	112.2707
	South Korea	2020-03-05 09:03:09	6088	35	41	36	128
	Italy	2020-03-05 17:43:03	3858	148	414	43	12
	Iran	2020-03-05 13:43:04	3513	107	739	32	53
Guangdong	Mainland China	2020-03-05 09:23:03	1351	7	1181	23.3417	113.4244
Henan	Mainland China	2020-03-05 01:48:26	1272	22	1239	33.882	113.614
Zhejiang	Mainland China	2020-03-05 09:43:03	1215	1	1124	29.1832	120.0934
Hunan	Mainland China	2020-03-05 08:43:03	1018	4	938	27.6104	111.7088
Anhui	Mainland China	2020-03-05 04:33:02	990	6	970	31.8257	117.2264
Jiangxi	Mainland China	2020-03-05 01:16:58	935	1	901	27.614	115.7221

Showing 1 to 10 of 173 entries

Select specific columns, for example **country, last_update, confirmed**:

```
Select country, last_update, confirmed from
coronavirus
```



RÉSULTAT

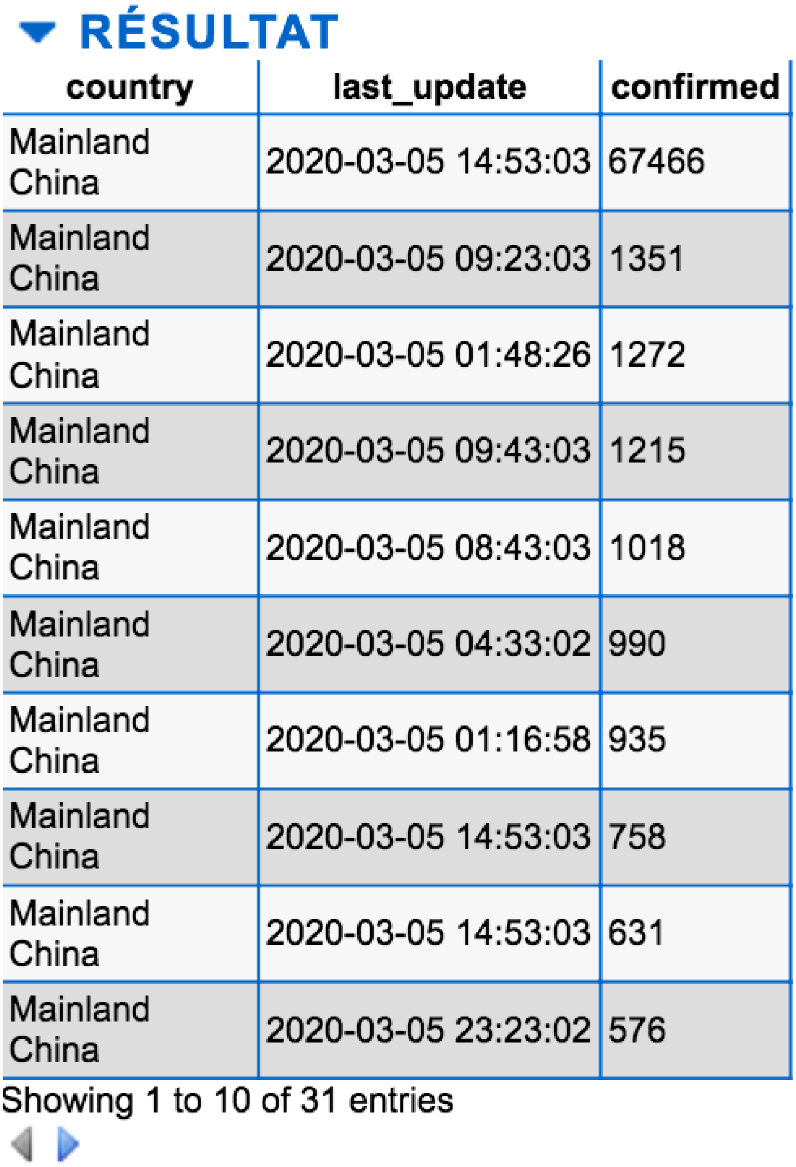
country	last_update	confirmed
Mainland China	2020-03-05 14:53:03	67466
South Korea	2020-03-05 09:03:09	6088
Italy	2020-03-05 17:43:03	3858
Iran	2020-03-05 13:43:04	3513
Mainland China	2020-03-05 09:23:03	1351
Mainland China	2020-03-05 01:48:26	1272
Mainland China	2020-03-05 09:43:03	1215
Mainland China	2020-03-05 08:43:03	1018
Mainland China	2020-03-05 04:33:02	990
Mainland China	2020-03-05 01:16:58	935

Showing 1 to 10 of 173 entries

Key word **WHERE** is used to add a condition to your request.

For example:

```
select country, last_update, confirmed
from coronavirus
where country = 'Mainland China'
```



RÉSULTAT

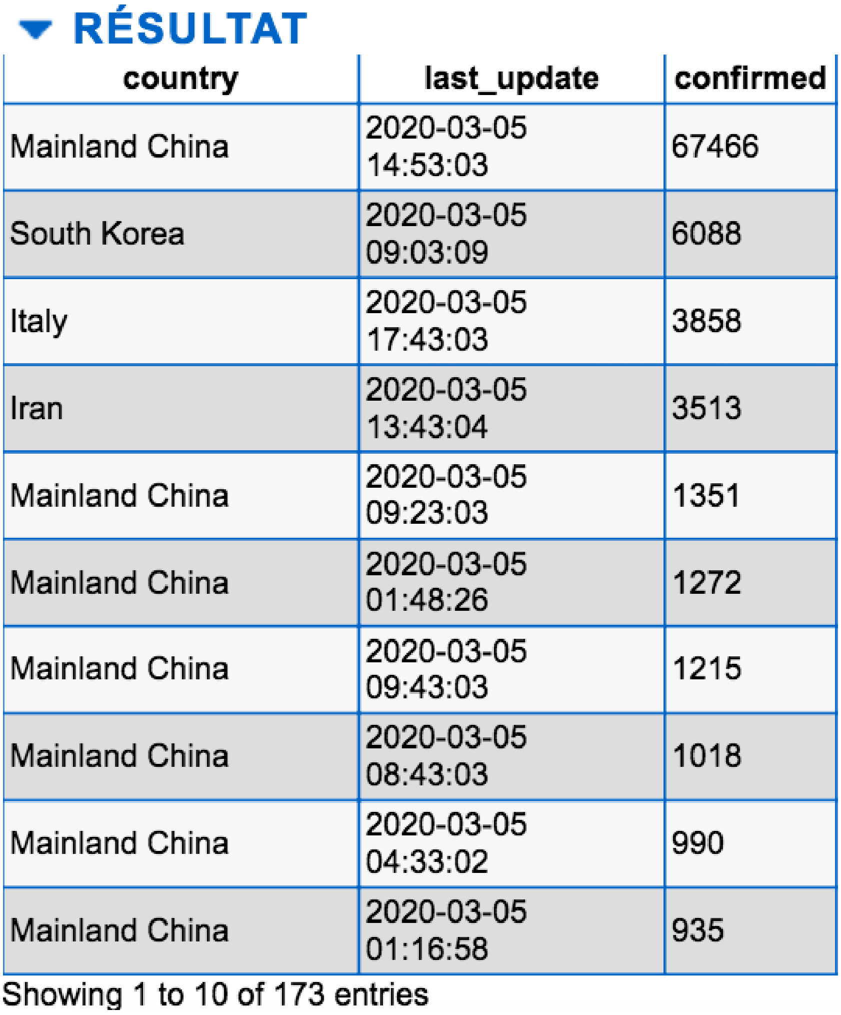
country	last_update	confirmed
Mainland China	2020-03-05 14:53:03	67466
Mainland China	2020-03-05 09:23:03	1351
Mainland China	2020-03-05 01:48:26	1272
Mainland China	2020-03-05 09:43:03	1215
Mainland China	2020-03-05 08:43:03	1018
Mainland China	2020-03-05 04:33:02	990
Mainland China	2020-03-05 01:16:58	935
Mainland China	2020-03-05 14:53:03	758
Mainland China	2020-03-05 14:53:03	631
Mainland China	2020-03-05 23:23:02	576

Showing 1 to 10 of 31 entries

Key word **ORDER BY** is used to order the results of a request by a column in an ascending (ASC) or descending (**DESC**) manner.

For example:

```
select country, last_update, confirmed
from coronavirus
order by confirmed DESC
```



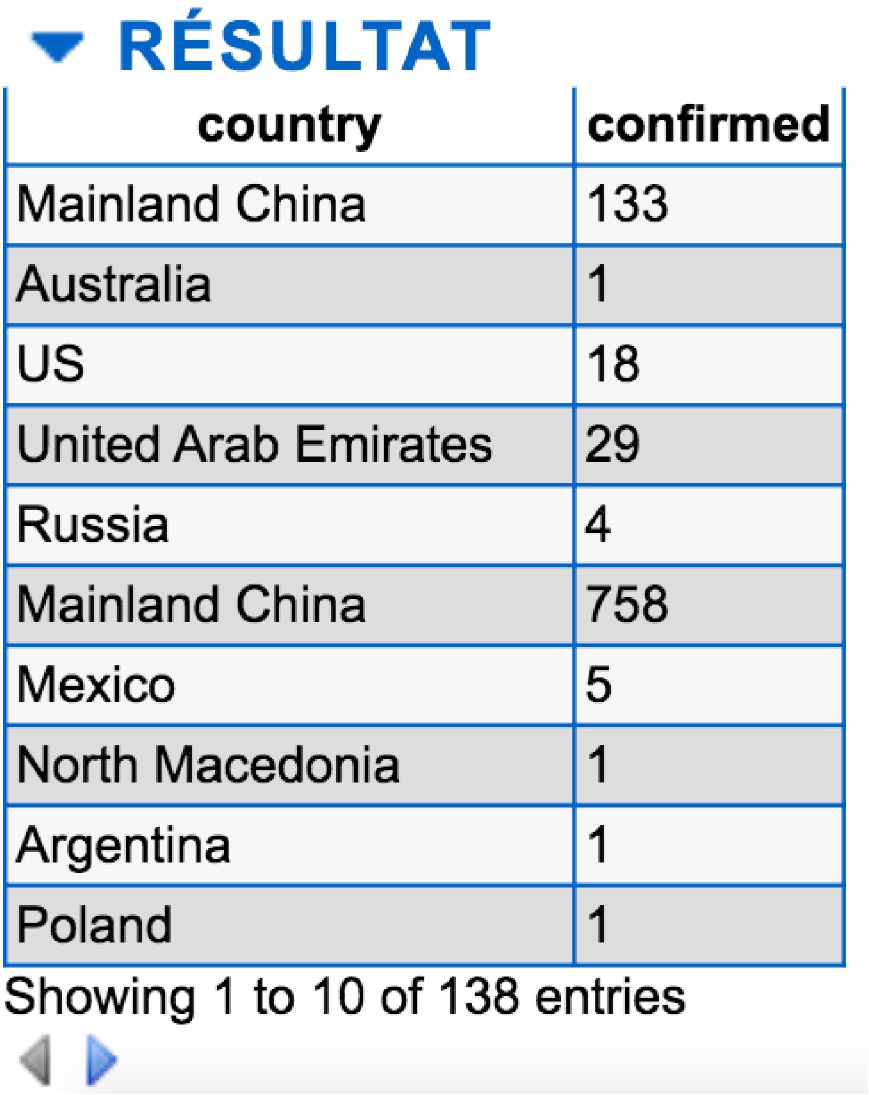
RÉSULTAT

country	last_update	confirmed
Mainland China	2020-03-05 14:53:03	67466
South Korea	2020-03-05 09:03:09	6088
Italy	2020-03-05 17:43:03	3858
Iran	2020-03-05 13:43:04	3513
Mainland China	2020-03-05 09:23:03	1351
Mainland China	2020-03-05 01:48:26	1272
Mainland China	2020-03-05 09:43:03	1215
Mainland China	2020-03-05 08:43:03	1018
Mainland China	2020-03-05 04:33:02	990
Mainland China	2020-03-05 01:16:58	935

Showing 1 to 10 of 173 entries

Key word **GROUP BY** is used to group the results of a request by a column.
For example:

```
select country, confirmed
from coronavirus
group by country, confirmed
```



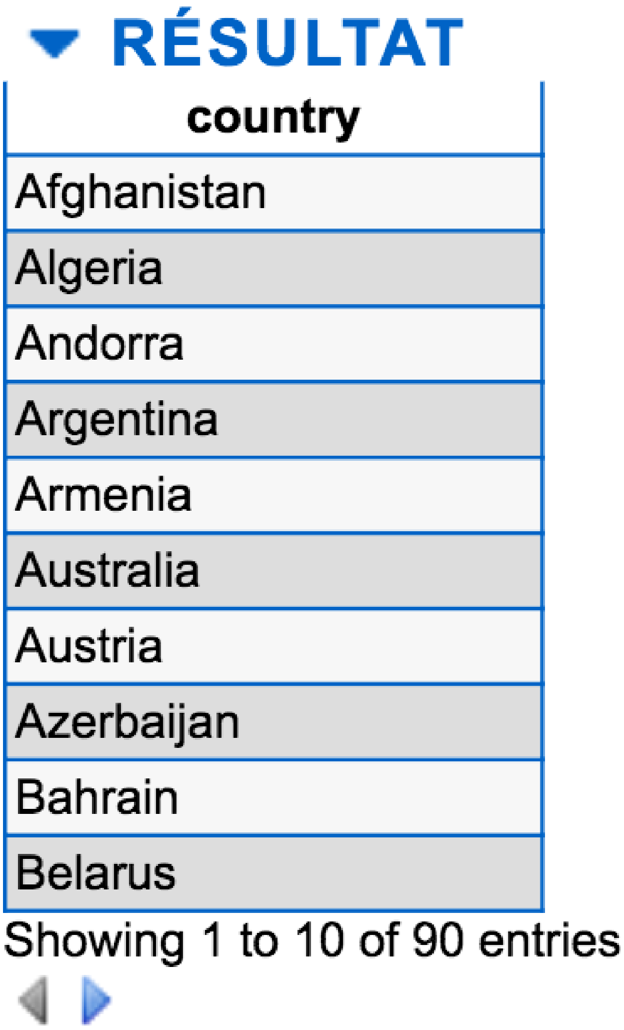
RÉSULTAT

country	confirmed
Mainland China	133
Australia	1
US	18
United Arab Emirates	29
Russia	4
Mainland China	758
Mexico	5
North Macedonia	1
Argentina	1
Poland	1

Showing 1 to 10 of 138 entries

Key word **DISTINCT** is used to display distinct (different) values.
For example:

```
select distinct country
from coronavirus
order by country ASC
```



RÉSULTAT

country
Afghanistan
Algeria
Andorra
Argentina
Armenia
Australia
Austria
Azerbaijan
Bahrain
Belarus

Showing 1 to 10 of 90 entries

Other conditions :

Comparison of values : **=, >, <, >=, <=, <>**

Ex :

```
Select country, last_update, confirmed
from coronavirus
where confirmed > 1000
```

Interval : **[NOT] BETWEEN … AND …**

EX :

```
Select country, last_update, confirmed
from coronavirus
where confirmed between 2000 and 10000
```

List of values : **[NOT] IN** (list of values)

EX :

```
Select country, last_update, confirmed
from coronavirus
where confirmed not in (1,2,3,6088,67466)
```

List of values: **[NOT] LIKE** (partial value chaine)

EX :

```
Select country, last_update, confirmed
from coronavirus
where country like 'I%'
```

/!\ ‘I%’ means all the words that start with ‘I’

Undetermination: **IS [NOT] NULL**

EX :

```
Select country, last_update, confirmed
from coronavirus
where confirmed is not null
```

/!\ NULL is different from zéro

Arithmetic expressions:

To multiply: *

To divide: /

To add: +

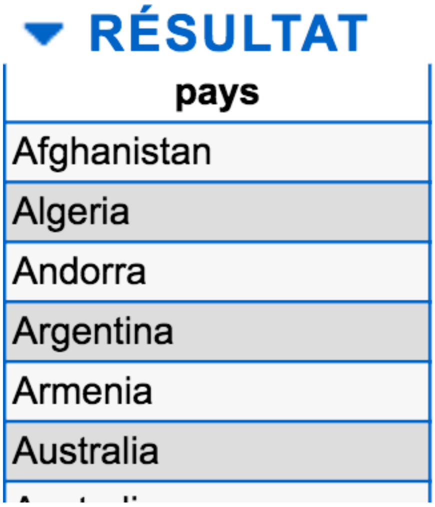
To subtract: -

/!\ only usable on variables of type date or time

Key word **AS** is used to give a new name to a column when extracting.

For example:

```
select country AS pays
from coronavirus
order by country ASC
```



RÉSULTAT

pays
Afghanistan
Algeria
Andorra
Argentina
Armenia
Australia

Logical operators:

AND: TRUE if the two conditions are true
OR: TRUE if at least one of the conditions is true
NOT: if none of the conditions are true

2) Exercises to do

A)Using Covid dataset

You can access it here:
https://hec.unil.ch/info1ere/sqlexplorer/Coronavirus

1.Find the total number of confirmed cases worldwide.

2.Find the total number of deaths in China.

3.How many confirmed cases in USA in California, Colorado and Conneticut?
[look at the data, to see how you can formulate this query]
(see how the data are)

4.Show all the cities and confirmed cases in California ordered by number of confirmed cases (descending)

5.How many will be the estimated cases next week for Switzerland if each week the cases multiply by 4?

6. Find all the records not in USA where confirmed cases are more than 100 (order in descending manner).

7. Find all the records not about USA and China that have between 10 and 20 deaths

B)Using Temperatures dataset

Now, let's use the **temperatures** datasets : https://hec.unil.ch/info1ere/sqlexplorer/Temperatures

1. Find all cities with temperature between 15 and 25, return city, country, temperature

2. Find all temperature records that contain a country whose name starts from A or a

3. What is the maximum temperature in Austria?

4. What is the average temperature in records that are from Italy or Greece?

5. Find all the countries and their population without coastline and with population more than 9 million (population column is in millions).

6. Find all cities with latitude more than 45, no coastline and population more than 9 million.

7. **How many** countries have latitude more than 45, no coastline and population more than 9 million.

8. **How many** cities have latitude more than 45, no coastline and population more than 9 million, AND what is the maximum and minimum latitude of those cities ?

9. Find the cities and countries without coastline. Sort them by descending longitude and return just 5 results.

10. What are the countries without coastline?

11. Find all pairs of cities that are close together, i.e., longitude and latitude are less than 0.5 apart (self-join: join a table with itself), do not include a city with itself!

12. What is the maximum latitude for all counties in EU? (we have maximum, which hints it may be a "group by" if it is for all entities, which it is. If it was for one entity, then we wouldn't need a group by.)