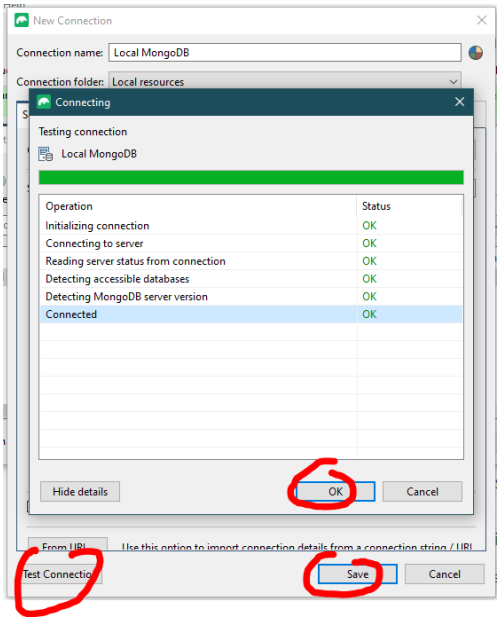
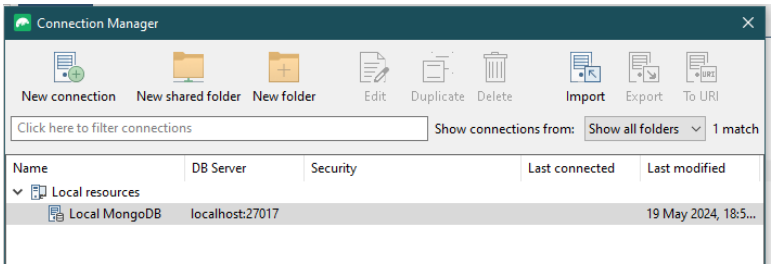
Advanced Databases : Mongodb

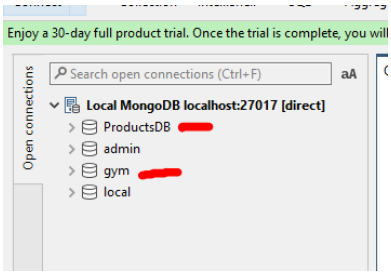
Lab 9 Solutions

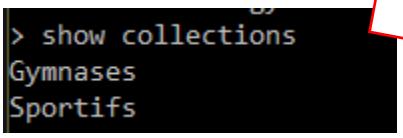
1- Open Mongod & Mongo like the last lab
2- Download Gym extract & restore as the instructions in the lab
3- Download and install Studio 3T: Run & Connect to the server: Here's how to connect to a local server in Studio 3T:

– *Open Studio 3T.*
– *Click the **Connect** button in the top-left corner of the toolbar.*
– *In the **Connection Manager dialog**, click the **New Connection** button. Alternatively, on the Quickstart tab, you can click **"Create a new connection**".*
– *In the New Connection dialog, choose **Manually configure my connection settings** and click **Next.***
– *Name the connection for easy identification ("**Local MongoDB**").*
– *In the Connection Type list, select **Standalone** (assuming you're connecting to a single MongoDB instance).*
– *Set the Server field **to localhost**. This indicates you are connecting to a **local server** on the same machine.*
– *The default **Port for MongoDB is 27017**. Leave this field unchanged unless your local MongoDB instance uses a different port.*
– *You can test the connection by clicking **Test Connection**. This verifies if Studio 3T can successfully connect to your local MongoDB server. 1 0. Click **Save** to finalize the connection setup.*

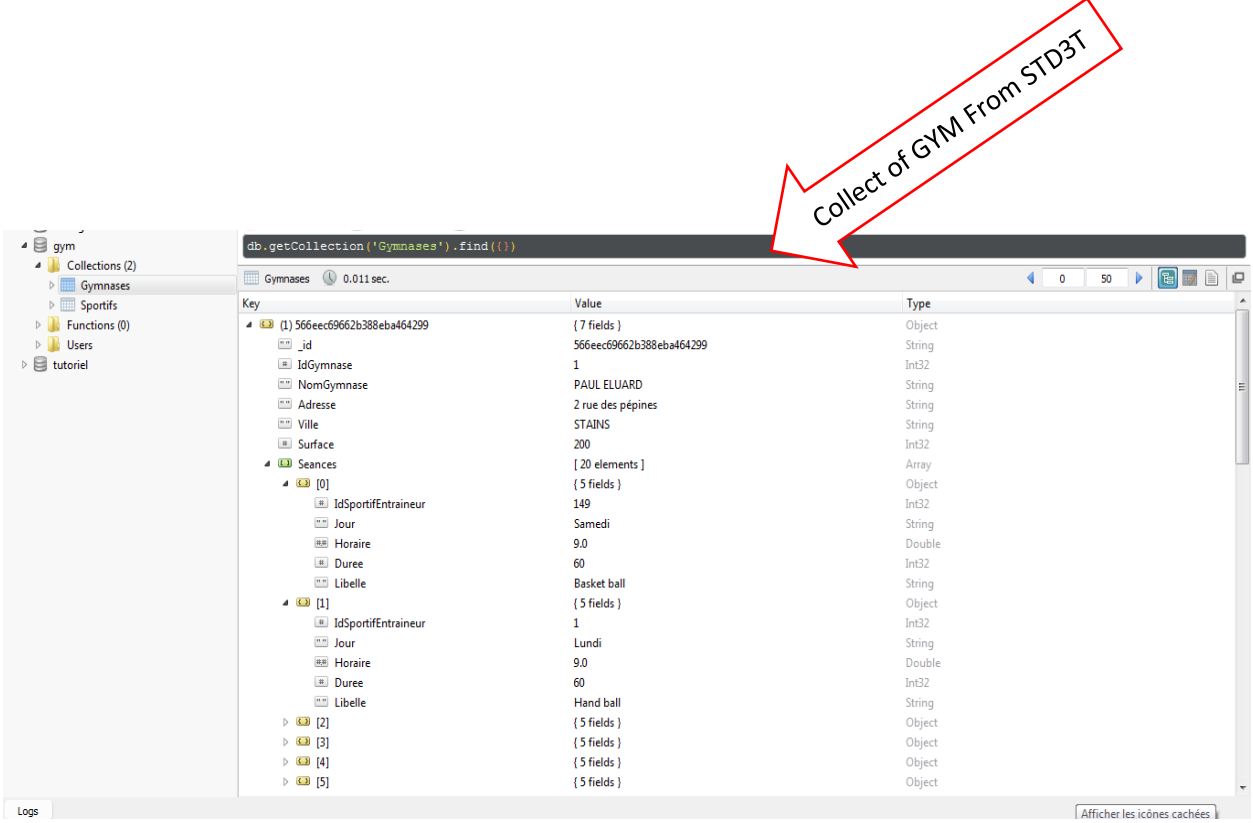




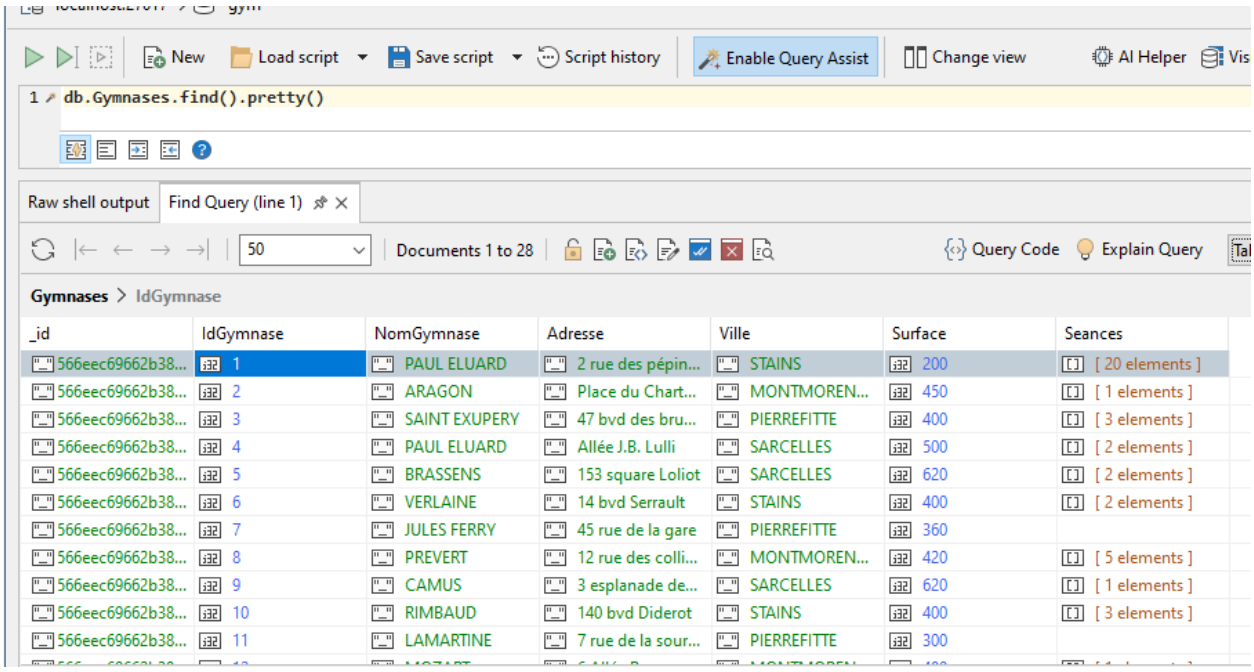


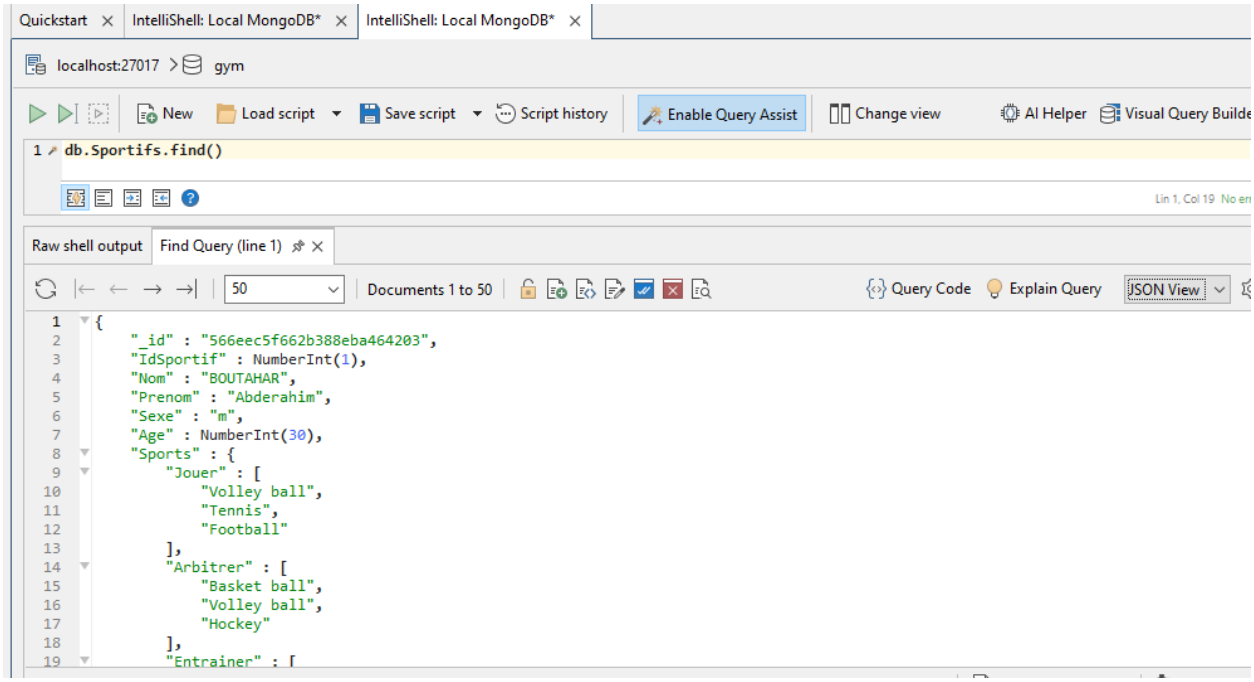


4- Answer some queries Using Robomongo:

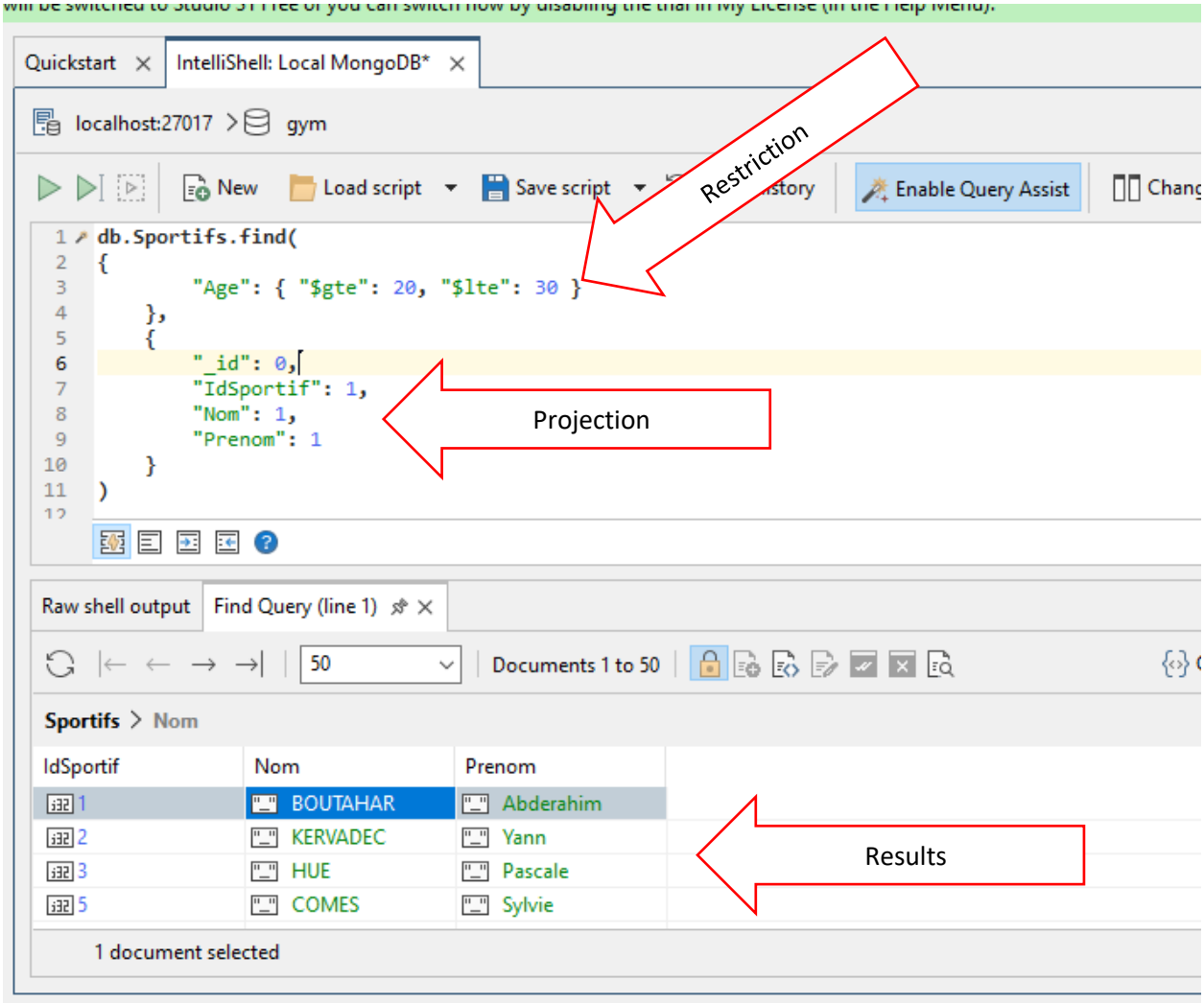


And we can use **(Names of the Columns are in French)**

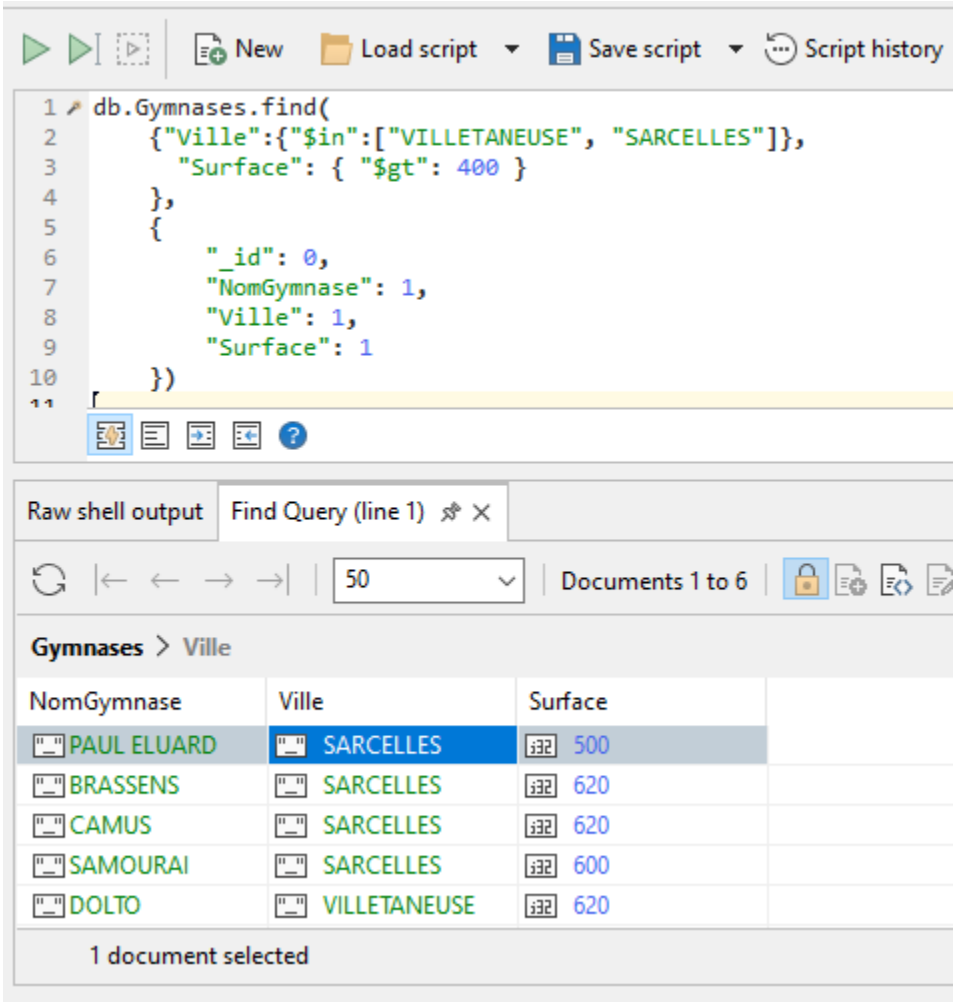




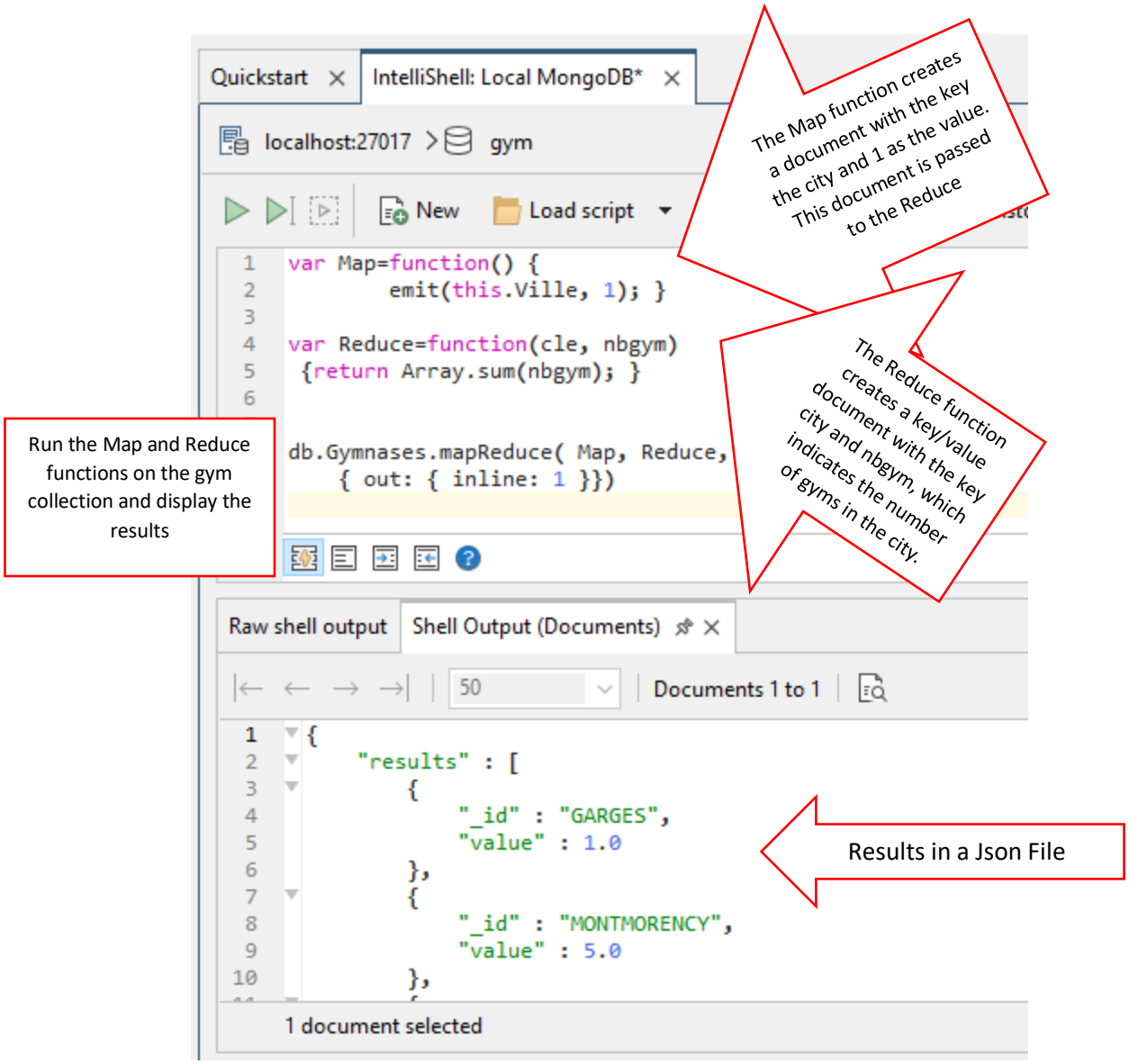
Which athletes (Id Sportifs, surname and first name) are aged between 20 and 30?

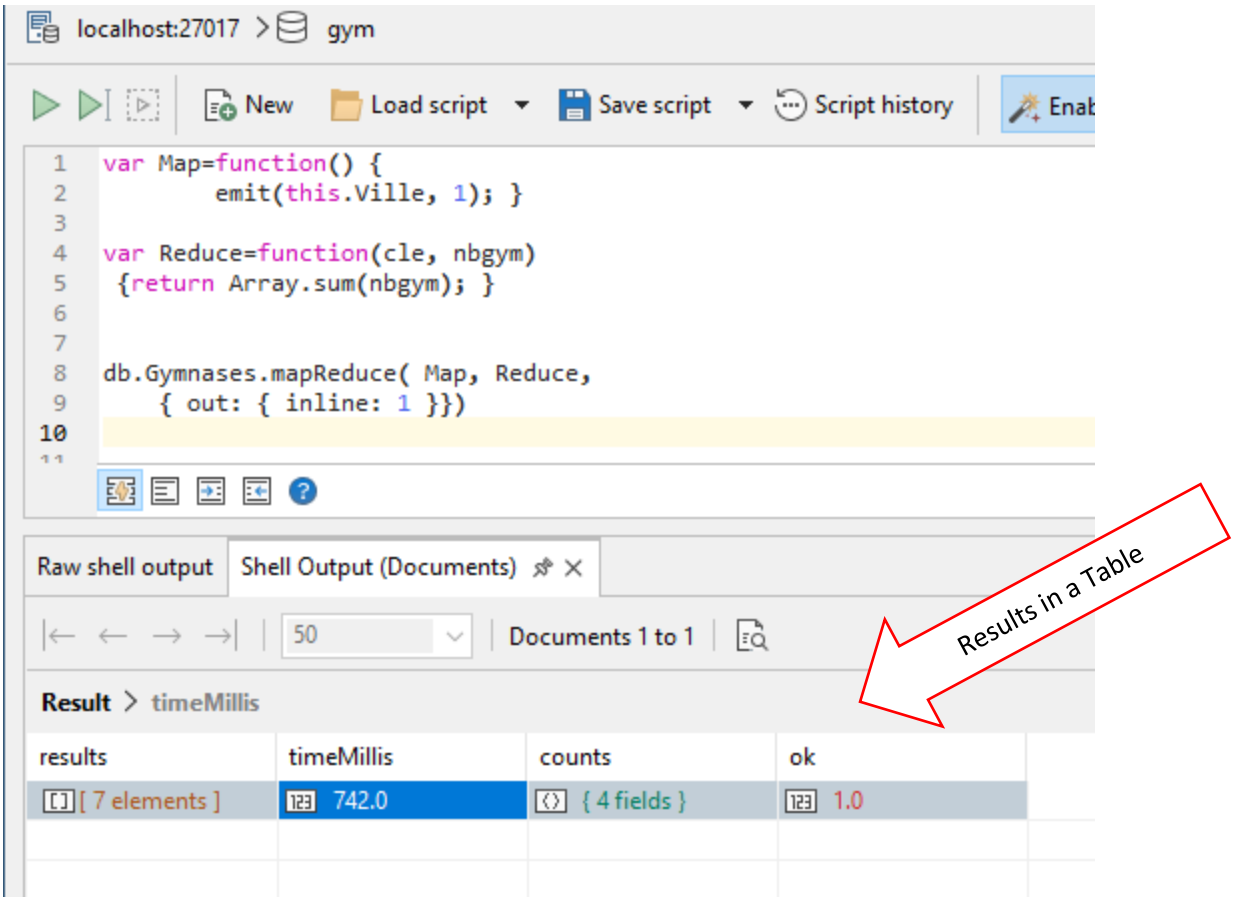


Which gyms in Villetaneuse or Sarcelles have a surface area of over 400 m^2?



5- Answer using Map-Reduce Paradigm
Calculate the number of gyms for each city


Quickstart
IntelliShell: Local MongoDB*
localhost:27017
gym
New
Load script
1 var Map=function() {
2 emit(this.Ville, 1); }
3
4 var Reduce=function(cle, nbgym)
5 {return Array.sum(nbgym); }
6
db.Gymnases.mapReduce(Map, Reduce,
{ out: { inline: 1 }})
The Map function creates a document with the key the city and 1 as the value. This document is passed to the Reduce
The Reduce function creates a key/value document with the key city and nbgym, which indicates the number of gyms in the city.
Run the Map and Reduce functions on the gym collection and display the results
Raw shell output
Shell Output (Documents)
50
Documents 1 to 1
{
"results" : [
{
"_id" : "GARGES",
"value" : 1.0
},
{
"_id" : "MONTMORENCY",
"value" : 5.0
},
1 document selected
Results in a Json File


localhost:27017
gym
New
Load script
Save script
Script history
Enab
1 var Map=function() {
2 emit(this.Ville, 1); }
3
4 var Reduce=function(cle, nbgym)
5 {return Array.sum(nbgym); }
6
7
8 db.Gymnases.mapReduce(Map, Reduce,
9 { out: { inline: 1 }})
10
Raw shell output
Shell Output (Documents)
50
Documents 1 to 1
Result
timeMillis
results
timeMillis
counts
ok
[7 elements]
742.0
{ 4 fields }
1.0
Results in a Table