FRAPPER
Colin

Database theory
2DV513

Assignment 4 - Tasks

Note 1 : I'm sorry about my english, I'm an Erasmus student.
Note 2 : I used MongoDB as a database.

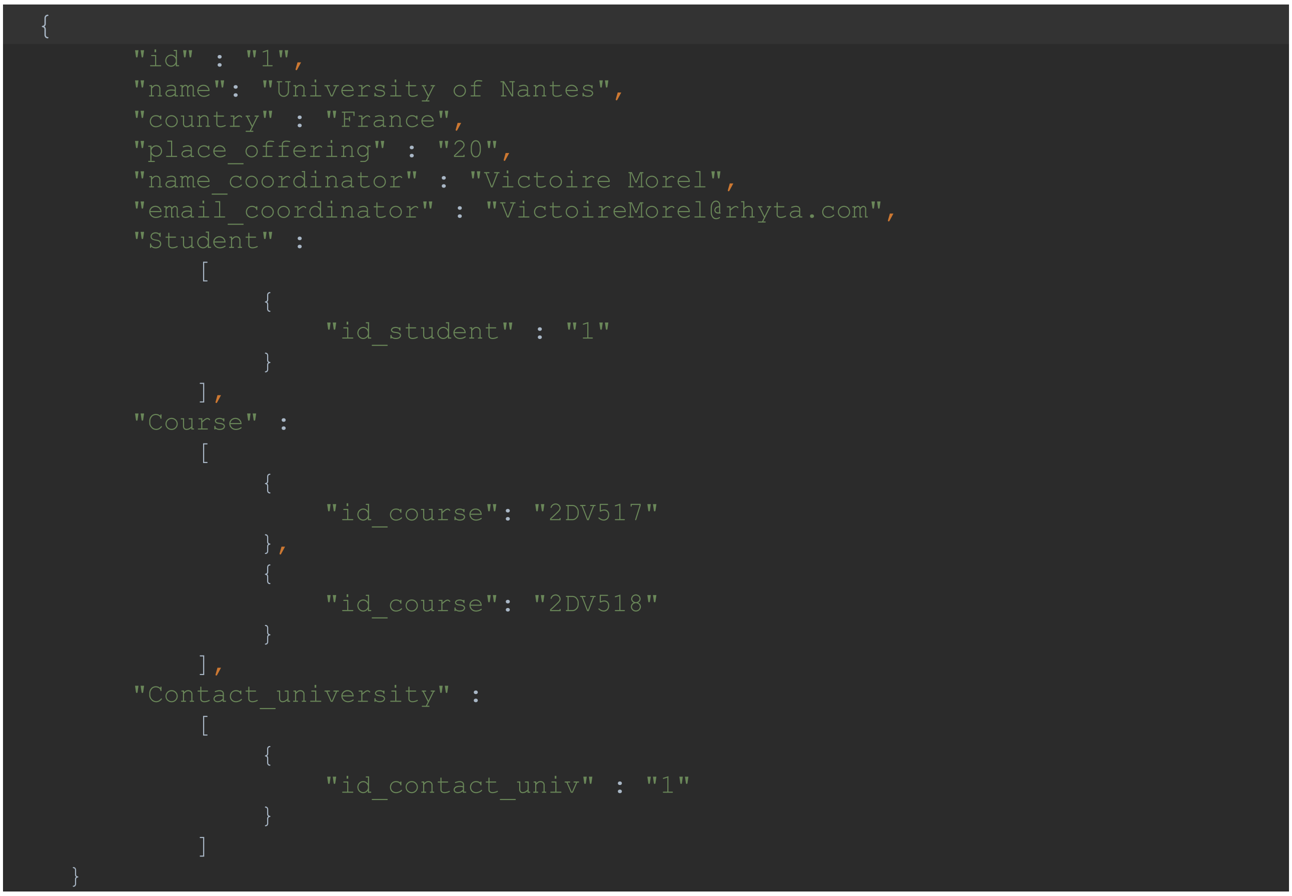
Task 1. Translate the data model

Your first step is to adjust the modelling to suit the underlying technology (key-value, document, graph, or column wide). The steps will differ depending on what technology you use, but in general you need to think about either relationships (graph) or aggregates (others).

Ok, so I choose the Assignment 3's database, the Erasmus database, I choose to have five collections, here are the design :

Collection Home_university :

The collection Home_unversity have an id, her name, country, the number of place they offering for an Erasmus exchanges, the name and the email of the coordinator who handle the erasmus exchanges so we can contact him/her if we have any question, also they have multiple Students to send to other univeristies, multiple courses, and the university which they have contact.
Here is an example :

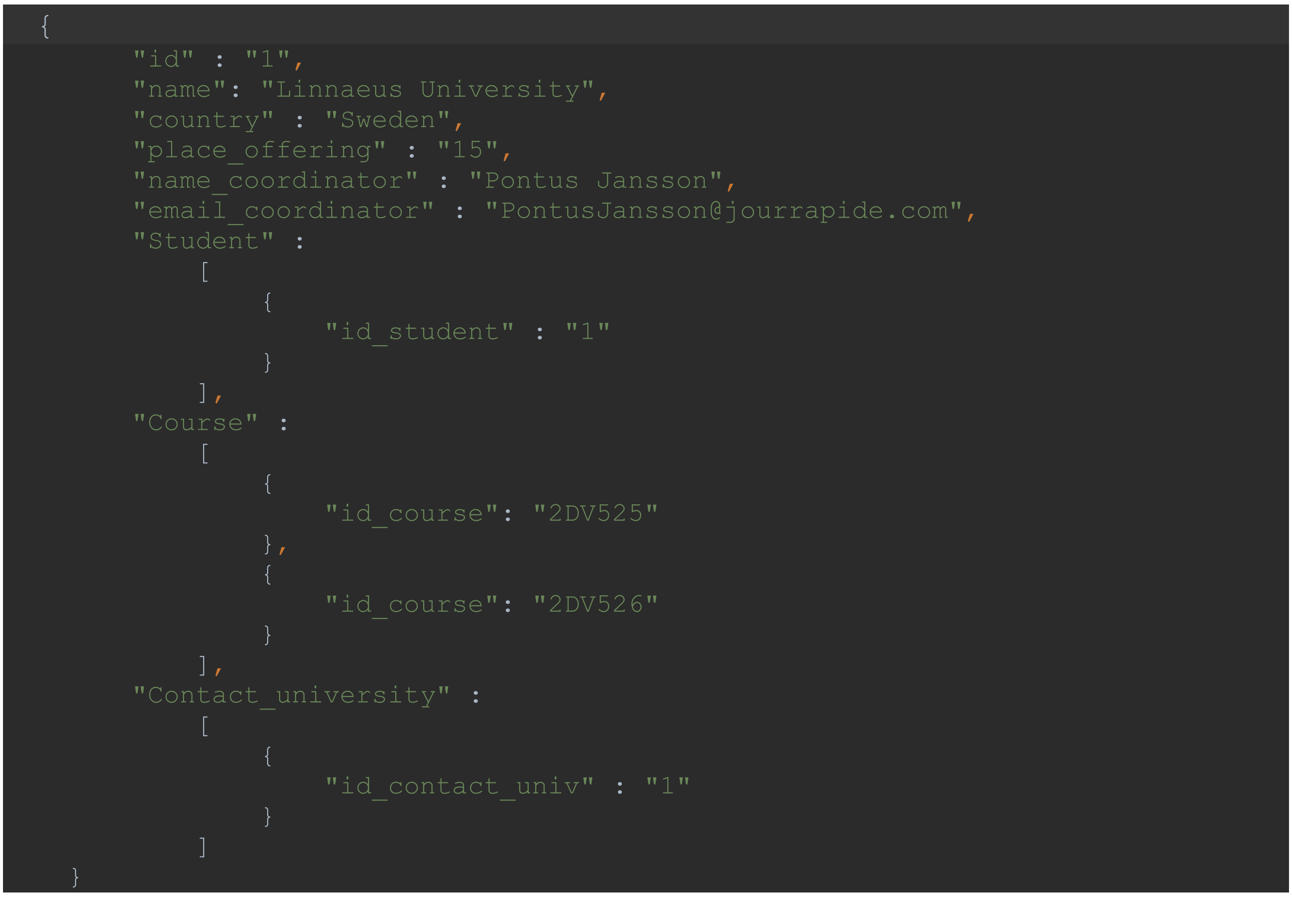


```
  {
        "id" : "1",
        "name": "University of Nantes",
        "country" : "France",
        "place_offering" : "20",
        "name_coordinator" : "Victoire Morel",
        "email_coordinator" : "VictoireMorel@rhyta.com",
        "Student" :
            [
                {
                    "id_student" : "1"
                }
            ],
        "Course" :
            [
                {
                    "id_course": "2DV517"
                },
                {
                    "id_course": "2DV518"
                }
            ],
        "Contact_university" :
            [
                {
                    "id_contact_univ" : "1"
                }
            ]
    }
```

Collection Arrival_university :

This collection is almost the same as the Home_university collection except that the field Students means the students they will welcome in their university, and the fields course is the courses available in their university.

Here is an example :

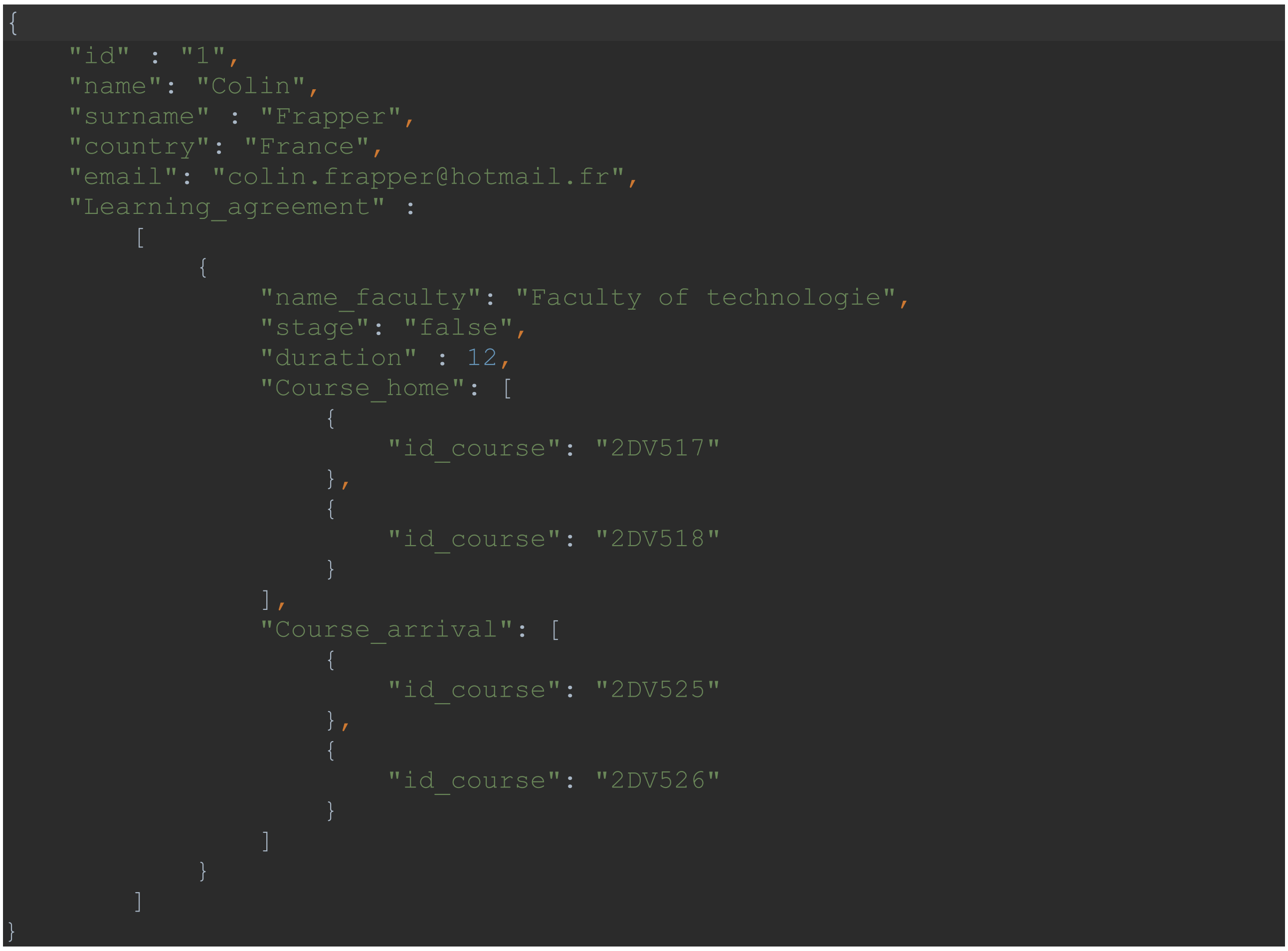


```
  {
        "id" : "1",
        "name": "Linnaeus University",
        "country" : "Sweden",
        "place_offering" : "15",
        "name_coordinator" : "Pontus Jansson",
        "email_coordinator" : "PontusJansson@jourrapide.com",
        "Student" :
            [
                {
                    "id_student" : "1"
                }
            ],
        "Course" :
            [
                {
                    "id_course": "2DV525"
                },
                {
                    "id_course": "2DV526"
                }
            ],
        "Contact_university" :
            [
                {
                    "id_contact_univ" : "1"
                }
            ]
    }
```

Collection Students :

The collection Students have the name, surname, email, country of the student, he also have a learning agreement field who resume what he will do in his erasmus program, the course they should take in their home university and the course they will take in their arrival university. Also how much longer they will stay (duration), and if they will make a stage or not, the name of the faculty they will study.

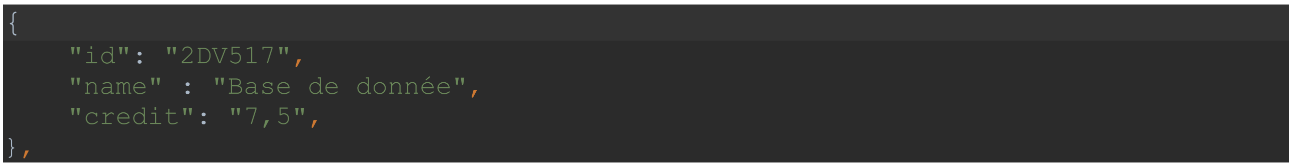
Here is an example :



```
{
    "id" : "1",
    "name": "Colin",
    "surname" : "Frapper",
    "country": "France",
    "email": "colin.frapper@hotmail.fr",
    "Learning_agreement" :
        [
            {
                "name_faculty": "Faculty of technologie",
                "stage": "false",
                "duration" : 12,
                "Course_home": [
                    {
                        "id_course": "2DV517"
                    },
                    {
                        "id_course": "2DV518"
                    }
                ],
                "Course_arrival": [
                    {
                        "id_course": "2DV525"
                    },
                    {
                        "id_course": "2DV526"
                    }
                ]
            }
        ]
}
```

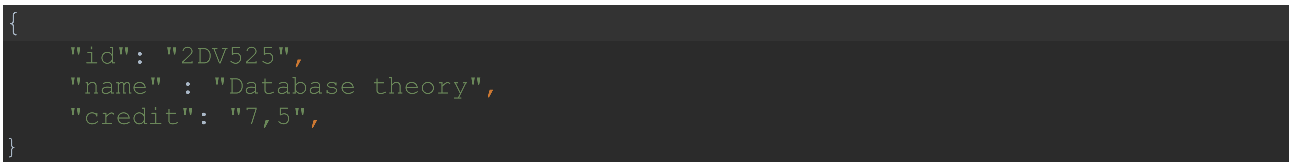
Collection Home_course :

Classic collection with the name and how much credit the course contains.
Here is an example :



```
{
    "id": "2DV517",
    "name" : "Base de donnée",
    "credit": "7,5",
},
```

Collection Arrival_course :
Same as Home_course :



```
{
    "id": "2DV525",
    "name" : "Database theory",
    "credit": "7,5",
}
```

Task 2. Implement the design

Create a program or a set or queries that implement your design. Populate the database with some values. If you use the Reddit dataset, the smaller file is sufficient.

I will add the Assignment.js in the zip file where I had methods to create collection and add data in them.

See the Assignment.js ;

Function create_collection : Function to create the five collection.

Function insert_data : Function to populate the database with values.

Function drop_collection_content : Function to drop the content of all collections.

Task 3 : Translate queries

Translate the queries to the query model of the database you used.

These five queries are almost the answer to all question that a student can ask before going to Erasmus. Even myself I asked those questions and too often without answers.

Question 1 :
I want information about a person who study in a special university and from my own country so I can contact him/her?

I created a function in the Assignment.js file ; query_1().

```
db.Arrival_university.aggregate([
    { $unwind : "$Student"},
    {
        $lookup:
        {
            from: "Students",
            localField: "Student.id_student",
            foreignField: "id",
            as: "Students_home"
        }
    },
    { $match  : { $and : [  { "name" : 'Linnaeus University' }, {
"Students_home.country" : 'France' } ] } },
    { $project : { "Students_home.name" : 1, "Students_home.surname" : 1,
"Students_home.email" : 1, "_id" : 0 }},
    {$out:"result_query_1"}
    ])
print ("\n" + "Result of query 1 ");
cursor = db.result_query_1.find();
    while ( cursor.hasNext() ) {
        printjson( cursor.next() );
    }
```

Explication : $unwind deconstructs an array field from the input documents to output a document for *each* element.

I know that MongoDB wasn't make to have join between collection but since version 3.4, it's

possible so I join the Arrival_university and the Students collection.
$match is like WHERE in SQL syntax, I search the name of the arrival university and the country of the student who will study there.
$project is like the select in SQL syntax, I want the name, surname and the email of the student.
$out is a little bit special I only used it to print the result of the query, $out will create a new collection of the previous query.
Then I print the query in the mongoShell terminal.

<u>Answer :</u>

Result of query 1

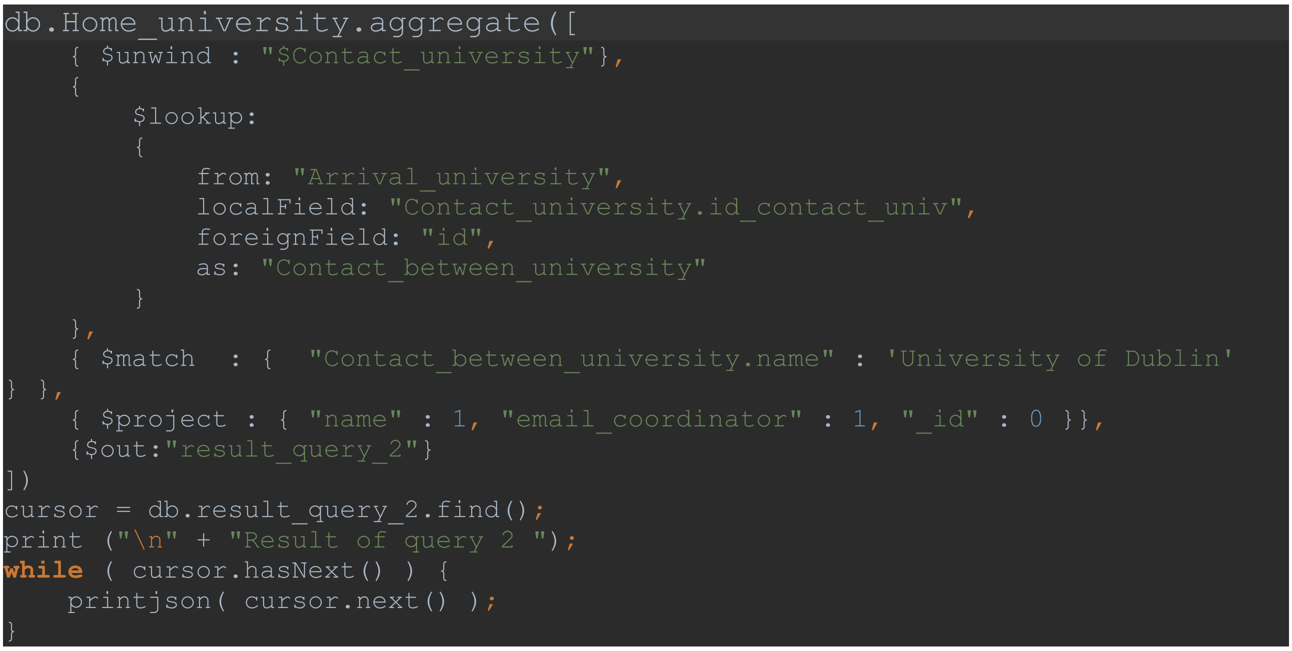
```
{
        "_id" : ObjectId("587b9e0f907b84337986a6e4"),
        "Students_home" : [
                {
                        "name" : "Colin",
                        "surname" : "Frapper",
                        "email" : "colin.frapper@hotmail.fr"
                }
        ]
}
```

Note : this query is about people who wants contact an old Erasmus student from his own country, and in a special university, this example is my own situation, a person from france and study in Linnaeus university.

<u>Question 2 :</u>

I don't know yet which university I wanna take but I'm sure that I wanna make an Erasmus exchanges in the University of Dublin, I wanna know which university offer this opportunity ?

Function query_2 :



```
db.Home_university.aggregate([
    { $unwind : "$Contact_university"},
    {
        $lookup:
        {
            from: "Arrival_university",
            localField: "Contact_university.id_contact_univ",
            foreignField: "id",
            as: "Contact_between_university"
        }
    },
    { $match  : {  "Contact_between_university.name" : 'University of Dublin'
} },
    { $project : { "name" : 1, "email_coordinator" : 1, "_id" : 0 }},
    {$out:"result_query_2"}
])
cursor = db.result_query_2.find();
print ("\n" + "Result of query 2 ");
while ( cursor.hasNext() ) {
    printjson( cursor.next() );
}
```

Explication : Similar to previous question.

Answer :

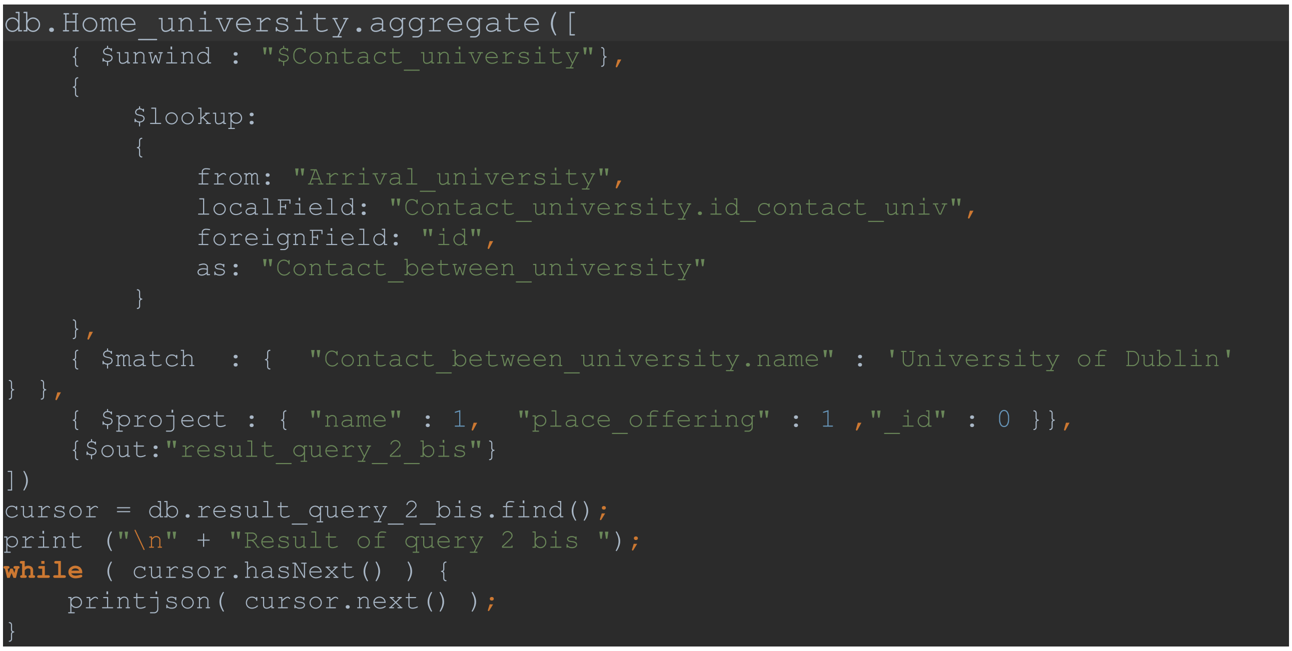
Result of query 2

```
{
        "_id" : ObjectId("587b9e0f907b84337986a6ef"),
        "name" : "University of Barcelona ",
        "email_coordinator" : "KalidFajardoJuarez@teleworm.us"
}
```

Note : This query is for person who aren't yet at university but they wanna choose the university according to what they can offer. (The name and the email are completely random it's not the real name/email)

Question 2 bis :

Same as the previous question but I wanna also know the number of the place that the university can have.



```
db.Home_university.aggregate([
    { $unwind : "$Contact_university"},
    {
        $lookup:
        {
            from: "Arrival_university",
            localField: "Contact_university.id_contact_univ",
            foreignField: "id",
            as: "Contact_between_university"
        }
    },
    { $match  : {  "Contact_between_university.name" : 'University of Dublin'
} },
    { $project : { "name" : 1,  "place_offering" : 1 ,"_id" : 0 }},
    {$out:"result_query_2_bis"}
])
cursor = db.result_query_2_bis.find();
print ("\n" + "Result of query 2 bis ");
while ( cursor.hasNext() ) {
    printjson( cursor.next() );
}
```

Answer :

Result of query 2 bis

```
{
        "_id" : ObjectId("587b9e0f907b84337986a6fa"),
        "name" : "University of Barcelona ",
        "place_offering" : "30"
}
```

Note : People also want to know how many a special university offer that's why I add the attribute number_place_offering, so they can know if they can easily make an Erasmus exchanges in this university, at contrary if they have to work hard to get an Erasmus exchanges.

Question 3 :

I wanna know which course took a special user during his Erasmus Exchanges ?

```
db.Students.aggregate([
    { $unwind : "$Learning_agreement"},
    { $unwind : "$Learning_agreement.Course_arrival"},
    {
        $lookup:
        {
            from: "Arrival_course",
            localField: "Learning_agreement.Course_arrival.id_course",
            foreignField: "id",
            as: "Course_arrival_student"
        }
    },
    { $match  : {  "name" : 'Colin' } },
    { $project : { "Course_arrival_student.name" : 1, "_id" : 0 }},
    {$out:"result_query_3"}
])
var cursor = db.result_query_3.distinct("Course_arrival_student.name");
print ("\n" + "Result of query 3 ");
printjson( cursor);
```

Explication ; This one is a bit tricky, I used the distinct method to not have duplicate name.

Answer :

Result of query 3
["Database theory", "Introduction to web programming"]

Note : This is also a important question that the Erasmus student ask to themselves, which course I should take, so now you know that a special user make his erasmus exchanges, you wanna know which course he took. This is very difficult to find exactly the same course as in his home university , I know that if I had this information that would be very useful to me.

Question 4 :

Which person make a stage more than one year and in which faculty ?

```
db.Students.aggregate([
    { $unwind : "$Learning_agreement"},
    { $match  : { $and : [  { "Learning_agreement.stage" : 'true' }, {
"Learning_agreement.duration" : { $gt: 12 } } ] } },
    { $project : { "name" : 1,  "surname" : 1 , "email" : 1,
"Learning_agreement.name_faculty" : 1, "_id" : 0 }},
    {$out:"result_query_4"}
])
cursor = db.result_query_4.find();
print ("\n" + "Result of query 4 ");
while ( cursor.hasNext() ) {
    printjson( cursor.next() );
}
```

This is the special query to see that sometimes MongoDB is way better than SQL, here are the SQL query :

SELECT name, surname , email_student,name_faculty FROM Student JOIN Learning_agreement ON id_student = student_id WHERE stage = 'true' AND duration > 12"

We can see that in mongoDB you don't have to make some join, it's more « simple » than in SQL

Answer :

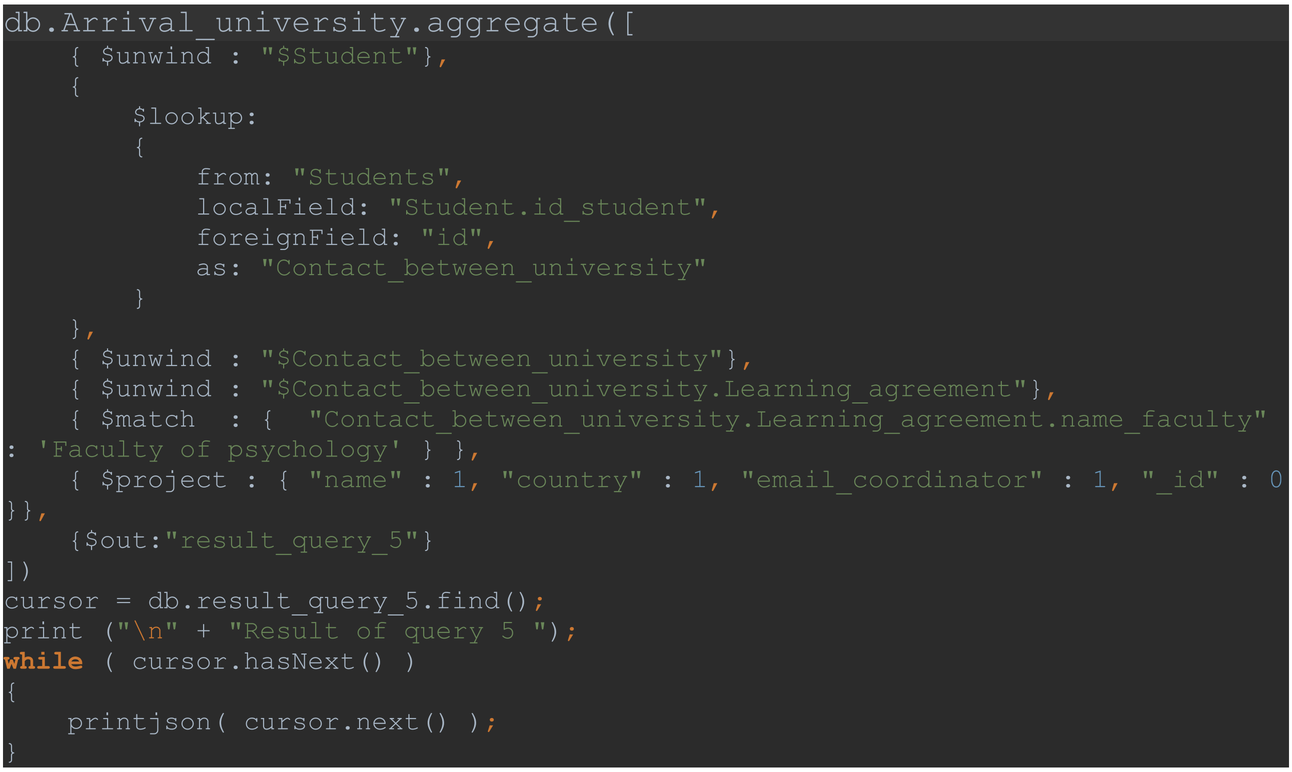
Result of query 4

```
{
	"_id" : ObjectId("587b9e10907b84337986a711"),
	"name" : "Paloma",
	"surname" : "Mejía Alejandro",
	"email" : "Paloma@rrerefe.com",
	"Learning_agreement" : {
		"name_faculty" : "Faculty of language"
	}
}
```

Note : If your plan is to make an Erasmus stage (you can also make a stage with the Erasmus program), more than one year, you wanna speak to someone who already do this and in which faculty.

Question 5 :

I wanna information about the people who are in the same faculty as mine and where they make their Erasmus ?



```
db.Arrival_university.aggregate([
    { $unwind : "$Student"},
    {
        $lookup:
        {
            from: "Students",
            localField: "Student.id_student",
            foreignField: "id",
            as: "Contact_between_university"
        }
    },
    { $unwind : "$Contact_between_university"},
    { $unwind : "$Contact_between_university.Learning_agreement"},
    { $match  : {  "Contact_between_university.Learning_agreement.name_faculty"
: 'Faculty of psychology' } },
    { $project : { "name" : 1, "country" : 1, "email_coordinator" : 1, "_id" : 0
}},
    {$out:"result_query_5"}
])
cursor = db.result_query_5.find();
print ("\n" + "Result of query 5 ");
while ( cursor.hasNext() )
{
    printjson( cursor.next() );
}
```

Anwser :

Result of query 5

```
{
	"_id" : ObjectId("587b9e10907b84337986a71c"),
	"name" : "University of Bordeaux",
	"country" : "France",
	"email_coordinator" : "MargauxRacicot@jourrapide.com"
}
```

Note : This query is for the people who want make an Erasmus according to a special faculty, with this query you have the information of the university and also the mail of the coordinator so you can contact him to ask question.

Task 4 : Reflexion

A reflection on the differences in the logical design and the queries. Do you prefer the relation or the NoSQL way?

About the design ;
Well this was the first time I used MongoDB and I must say it's pretty interresting, the thing is it's not that strict than in SQL, I mean that you have more freedom to make your design, and you can adapt your design to your future query and for this MongoDB is really cool.

About the queries ;
As I said I never used MongoDB queries and I must say that it's was kind of difficult but when I completely understand how work the pipeline aggregate it was more easy than in SQL, beause you will make the design depending on your query, so obviously the queries will be more simple.

Do you prefer the relation or the NoSQL way?
I'm a big fan of SQL, so yes I prefer the relation, but maybe this is because i'm not used to work with NoSQL, anyway it was really interesting to see another way to work with database, I learned a lot about MongoDB.

Task 5 : How to run/compile

So first you obviously need to have MongoDB installed on your computer,
You need first to create the Assignment database, I couldn't find how I can do it with javascript, open the mongoshell terminal (mongo.exe) :

Step 1 : write : use Assignment

Then write :

```
load("path/Assignment.js")
```

Example :

```
load("C:/Users/moi/Documents/Cours/Licence_3/Semester_1/Database_theory/Assignment_4/Assignment.js")
```

Now you should see the result of the query.

The code is in Assignment.js.