

Advanced Data Management - Assignment N°01

Assignment sheet N°04 contains one exercise. Deadline for submitting the assignments will be 19.12.2023 23:59. For each task, there is a maximum time. Please do not work longer than that time on a task! If you think that you will not be able to finish the task in the given maximum time, stop working on it 10 minutes before the end, and provide an explanation contain the following information:

- Whether you think that the task is solvable with the current system at all, and why?
- If you think that is solvable with more time: which approach would you try out next?

Note: Because of the fact that all the students use the same data that is already available on the server, please don't change the structure and the data.

Exercise 1: Document database

<p>email</p> <p>(3) ObjectId("5b0d270a" { 6 fields }</p> <table> <tr> <td>_id</td><td>ObjectId("5b0d270a9587ce1de41e5c00")</td></tr> <tr> <td>ID</td><td><39889.1075846790345.JavaMail.evans@thyme></td></tr> <tr> <td>MESSAGE_BODY</td><td>Here are a few valuable documents for Edu</td></tr> <tr> <td>MESSAGE_DATE</td><td>2000-02-01 02:38:00+0000</td></tr> <tr> <td>MESSAGE_FROM</td><td>jennifer.rudolph@enron.com</td></tr> <tr> <td>MESSAGE_SUBJECT</td><td>To better serve</td></tr> </table>	_id	ObjectId("5b0d270a9587ce1de41e5c00")	ID	<39889.1075846790345.JavaMail.evans@thyme>	MESSAGE_BODY	Here are a few valuable documents for Edu	MESSAGE_DATE	2000-02-01 02:38:00+0000	MESSAGE_FROM	jennifer.rudolph@enron.com	MESSAGE_SUBJECT	To better serve	<p>person</p> <p>(2) ObjectId("5b0d" { 8 fields }</p> <table> <tr> <td>_id</td><td>ObjectId("5b0d26b69587ce1de41dce14")</td></tr> <tr> <td>id</td><td>arnold-j</td></tr> <tr> <td>firstname</td><td>Ben</td></tr> <tr> <td>middlename</td><td>Luca</td></tr> <tr> <td>lastname</td><td>Arnold</td></tr> <tr> <td>works_in</td><td>HS</td></tr> <tr> <td>salary</td><td>65.000 (65.0K)</td></tr> <tr> <td>email_address</td><td>arnold@enron.com</td></tr> </table>	_id	ObjectId("5b0d26b69587ce1de41dce14")	id	arnold-j	firstname	Ben	middlename	Luca	lastname	Arnold	works_in	HS	salary	65.000 (65.0K)	email_address	arnold@enron.com
_id	ObjectId("5b0d270a9587ce1de41e5c00")																												
ID	<39889.1075846790345.JavaMail.evans@thyme>																												
MESSAGE_BODY	Here are a few valuable documents for Edu																												
MESSAGE_DATE	2000-02-01 02:38:00+0000																												
MESSAGE_FROM	jennifer.rudolph@enron.com																												
MESSAGE_SUBJECT	To better serve																												
_id	ObjectId("5b0d26b69587ce1de41dce14")																												
id	arnold-j																												
firstname	Ben																												
middlename	Luca																												
lastname	Arnold																												
works_in	HS																												
salary	65.000 (65.0K)																												
email_address	arnold@enron.com																												
<p>department</p> <p>(3) ObjectId { 4 fields }</p> <table> <tr> <td>_id</td><td>ObjectId("5b0d26fe9587ce1de41e5bf4")</td></tr> <tr> <td>id</td><td>CA</td></tr> <tr> <td>name</td><td>Communications and Arts</td></tr> <tr> <td>location</td><td>1253 McGill College, Montreal, Quebec, H3B 2Y5</td></tr> </table>	_id	ObjectId("5b0d26fe9587ce1de41e5bf4")	id	CA	name	Communications and Arts	location	1253 McGill College, Montreal, Quebec, H3B 2Y5	<p>knows</p> <p>(2) ObjectId { 3 fields }</p> <table> <tr> <td>_id</td><td>ObjectId("5b0d26e29587ce1de41e3826")</td></tr> <tr> <td>pid1</td><td>allen-p</td></tr> <tr> <td>pid2</td><td>arnold-j</td></tr> </table>	_id	ObjectId("5b0d26e29587ce1de41e3826")	pid1	allen-p	pid2	arnold-j														
_id	ObjectId("5b0d26fe9587ce1de41e5bf4")																												
id	CA																												
name	Communications and Arts																												
location	1253 McGill College, Montreal, Quebec, H3B 2Y5																												
_id	ObjectId("5b0d26e29587ce1de41e3826")																												
pid1	allen-p																												
pid2	arnold-j																												
<p>to</p> <p>(2) ObjectId { 3 fields }</p> <table> <tr> <td>_id</td><td>ObjectId("5b0d26cb9587ce1de41dceac")</td></tr> <tr> <td>id</td><td><11703025.1075855236918.JavaMail.evans@thyme></td></tr> <tr> <td>email</td><td>kay.mann@enron.com</td></tr> </table>	_id	ObjectId("5b0d26cb9587ce1de41dceac")	id	<11703025.1075855236918.JavaMail.evans@thyme>	email	kay.mann@enron.com	<p>cc</p> <p>(2) ObjectId { 3 fields }</p> <table> <tr> <td>_id</td><td>ObjectId("5b0d26ef9587ce1de41e3c0f")</td></tr> <tr> <td>id</td><td><11703025.1075855236918.JavaMail.evans@thyme></td></tr> <tr> <td>email</td><td>hallb@gtlaw.com</td></tr> </table>	_id	ObjectId("5b0d26ef9587ce1de41e3c0f")	id	<11703025.1075855236918.JavaMail.evans@thyme>	email	hallb@gtlaw.com																
_id	ObjectId("5b0d26cb9587ce1de41dceac")																												
id	<11703025.1075855236918.JavaMail.evans@thyme>																												
email	kay.mann@enron.com																												
_id	ObjectId("5b0d26ef9587ce1de41e3c0f")																												
id	<11703025.1075855236918.JavaMail.evans@thyme>																												
email	hallb@gtlaw.com																												

Figure1. structure of documents in each collection

DATA IN MongoDB

MongoDB is a document database with BSON documents as its storage format. In MongoDB, databases hold collections of documents. Collections are analogous to tables in relational databases. By default, a collection does not require its documents to have the same schema; i.e. the documents in a single collection do not need to have the same set of fields and the data type for a field can differ across documents within a collection. In Enron database, we have 4 collections: department, person, email, and knows. A sample document of each collection is shown in Figure 1.

SYSTEM PREPARATION

To connect to MongoDB server, there are two options:

1) The NosqlConcept tool: <http://162.55.237.41:8765/>

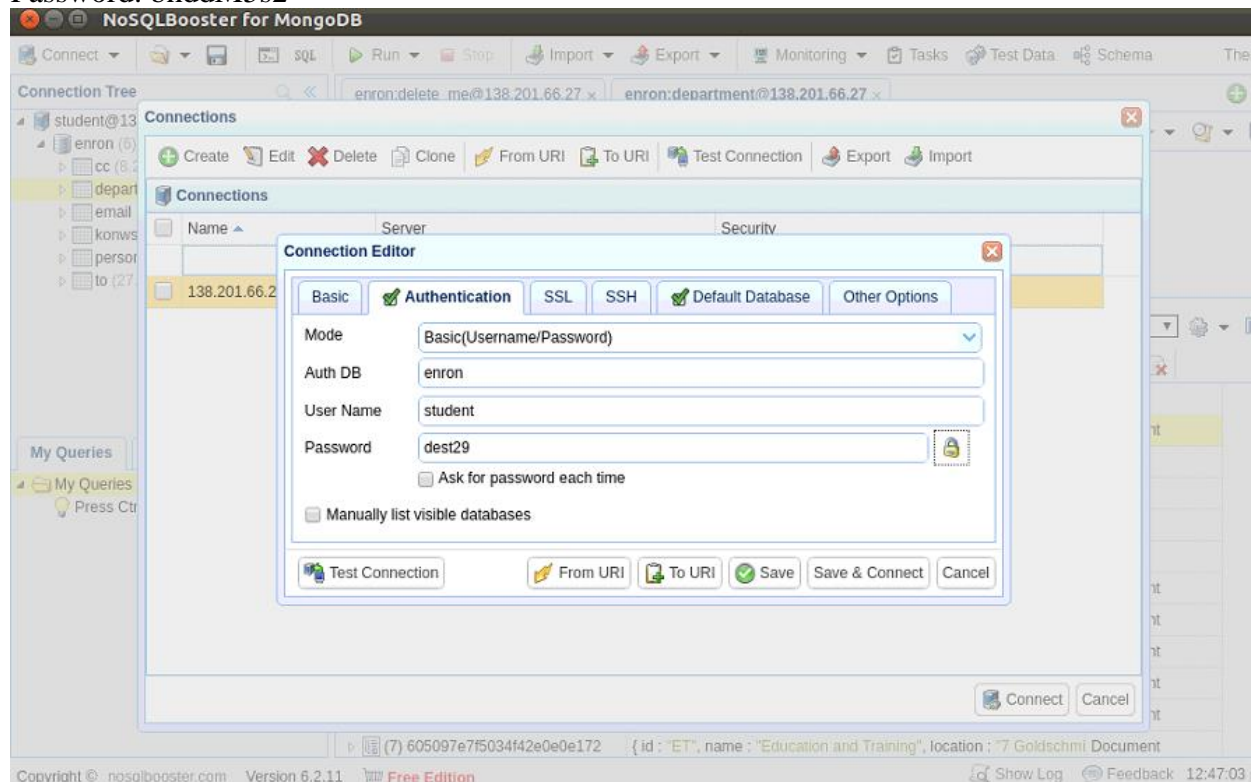
2) Use the client platform “NoSQLBooster” (download link: <https://nosqlbooster.com/downloads>). Create a new connection with this information:

Server: 162.55.237.41 port: 27017

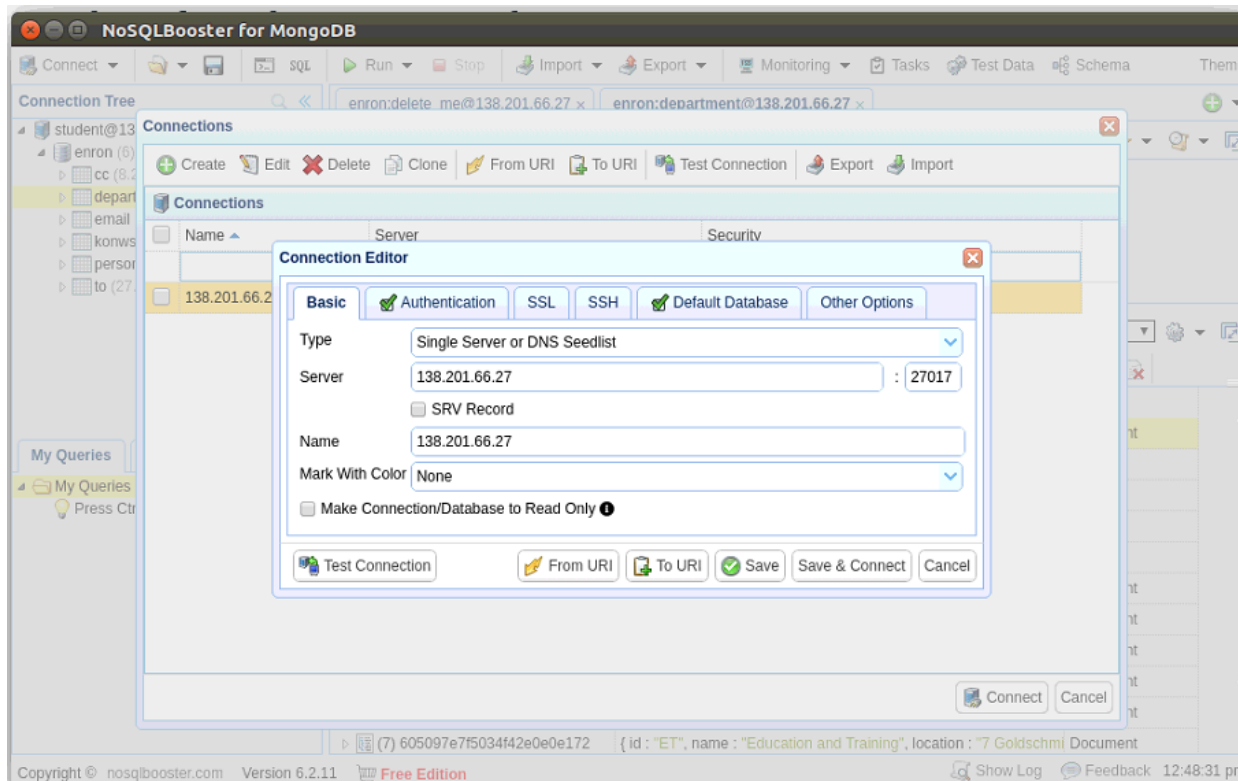
Auth DB: (leave blank)

Username: student

Password: bhddM5s2



Add the connection information as in the screen-shot below:



Now you can see the collections in Enron database.

```
db.person.find({})
```

Exercises of document database

Exercise1 TASK 1 Use Case: Equi Join

1.1 List of people with their department Max Time: 0,5h

For each person you want to know in which department she or he works. Therefore, you have to make an output that contains a person's first name and last name and the name of the department she or he is working at.

1.2 Department ordered by sent emails Max Time: 0,5h

You want to know how different departments are sending emails. Sort the departments by the average number of email sent per day. (hint: carbon copies included)

1.3 Weekly workload Max Time: 0,5h

Employees could have different workload during the week. What is the busiest week-day in term of email sending in Ernon ?

1.4 Weekly workload Max Time: 0,5h

What is the ratio of workload variation over the week?

Hint : the ratio can be found as:

$$r = \frac{\text{number of email sent in the bussiest day} - \text{average number of email sent per day}}{\text{average number of email sent per day}}$$

Exercise1 TASK 2 Use Case: Theta Join

2.1 Correlation between salary and number of emails Max Time: 2h

Do people that earn more than the average salary in their department write more emails than those who don't?

Query for people that earn more than the average salary at their department and find out whether they write more emails than the other employees that earn less than the average salary at their department (equal is not considered). Check that for each department. First compute the result for the average salary (avg. S.) per department that contains the brief-name of all the departments and the average salary for that department. Then produce the output of all the people that earn more than the avg. Salary and accordingly produce the output for all the people who earn less than the avg. Salary.

Produce a query result per department that contains the number of emails written by the people earning more and the people earning less than the average.

Exercise1 TASK 3 Use Case: Schema Evolution

3.1 Update the schema Max Time: 0,50h

Adding new information about salary in the person table would reduce its normal form. This is solved by creating new table called salary with attributes (ID, salary group, starting salary). Create the given table with name "salary_your_name" , and create a copy of person table with name "person_your_name" (create both tables in the "**public**" schema). Now connect the two created table with appropriate foreign key.

Remember: Delete the tables that you have created.

Exercise1 TASK 4 Use Case: Missing values

4.1 Find missing values Max Time: 0,5h

Find missing values for each attribute of the e-mails. Which attribute has the most missing values?

Exercise1 TASK 5 Use Case: Range queries

5.1 Salaries between two values Max Time: 0,25h

Select all employees who have salaries between 40000 and 45000.

5.2 Salaries between two values for certain department Max Time: 0.25h

Select all employees who have salaries between 40000 and 45000 and works for the defense department.

Exercise1 TASK 6 Use Case: Network analysis

Network analysis can be done to investigate social structures. In the related field of social network analysis a network is characterized by *nodes* (individual actors, people, or things within the network) and the *edges* (relationships or interactions) that connect them. To find out, how far from each other two nodes of the network are, we can count hops. To calculate the number of hops, we calculate $m+1$, where m is the number of intermediate nodes between the two nodes we're looking at. In Enron example, each employee can be seen as a node in the network and edges are represented by the emails sent from one to another employee or by the "knows" relationship between two employees. A relational Database provides us only records; therefore solutions to network analysis must be derived somehow different.

Hint: you can use a UDF (User defined function) to solve the following two tasks.

Remember: Delete any created functions before you leave the site.

6.1 E-mail network Max Time: 1h

Find the number of persons who never sent any emails to each other in all departments.

6.2 E-mail network Max Time: 1h

Do employees send email to other department? Order the department by the number of sent email to other departments.

6.3 Social network Max Time: 1h

Consider the knows relationship to find the employee in Enron who has the most number of know people in his social network up to 3 hops.

6.4 Social network Max Time: 1h

Repeat task 6.3 but consider the emails and the "from" and "to" fields to find whether two persons know each other.

Exercise1 TASK 7 Use Case: User defined functions

7.1 Unique words in a certain email Max Time: 2h

Take an email text and find how many unique words it has. You might need to use an external programming language to implement an UDF (User defined function) to split the email text into words. Apply the UDF to a particular email body.

Remember: Delete any created functions before you leave the site.

7.2 Words distance in a certain email Max Time: 2h

For all the emails, count the existence of two words: “meeting” and “time”, such that the distance between the words in text is no more than 3 words. The output should be the email and the count of occurrence. Similar to the previous task, an UDF might be necessary.

Remember: Delete any created functions before you leave the site.