

Practical Course Data Science: Database Management Systems (DBMS)

MongoDB Assignment

Dr. Ahmed Al-Ghezi

Database Technologies und Data Analytics Goethe-Universität Frankfurt

MongoDB



- NoSQL Document-oriented database system
- JSON-like documents with optional schemas
- Distributed
- It uses MongoDB Query Language (MQL)

JSON Documents

- Natural and productive way to work with data.
- Supports arrays and nested objects as values.
- Allows for flexible and dynamic schemas.

```
{
  "_id": "5cf0029caff5056591b0ce7d",
  "firstname": "Jane",
  "lastname": "Wu",
  "address": {
    "street": "1 Circle Rd",
    "city": "Los Angeles",
    "state": "CA",
    "zip": "90404"
  },
  "hobbies": ["surfing", "coding"]
}
```

MQL

```
> db.users.find({ "address.zip" : "90404" })
{ "_id": "5cf0029caff5056591b0ce7d", "firstname": "Jane", "lastname": "W"
{ "_id": "507f1f77bcf86cd799439011", "firstname": "Jon", "lastname": "W"
{ "_id": "5349b4ddd2781d08c09890f3", "firstname": "Jim", "lastname": "W"
{ "_id": "5bf142459b72e12b2b1b2cd", "firstname": "Jeff", "lastname": "W"
{ "_id": "5cf003283b23d04a40d5f88a", "firstname": "Jerry", "lastname": "W"
{ "_id": "5bf142459b72e12b2b1b2cd", "firstname": "Jai", "lastname": "W"
{ "_id": "5cf0036deaa1742dd225ea35", "firstname": "Jess", "lastname": "W"
{ "_id": "54495ad94c934721ede76d90", "firstname": "Jill", "lastname": "W"
{ "_id": "566eb3c704c7b31facbb0007", "firstname": "Janet", "lastname": "W"
{ "_id": "5a999cc461d36489a27f2563", "firstname": "Jan", "lastname": "W"
```

- Expressive query language that supports: filter and sort by any field, no matter how nested it may be within a document.
- Support for aggregations.
- Queries are themselves JSON, and thus relatively easily composable.

MQL

- Support relational join.
- Supports iteration and scripting

```
session.start_transaction()
order = { line_items : [ { item : 5, quantity: 6 } ] }
db.orders.insertOne( order, session=session );
for x in order.line_items:
    db.inventory.update(
        { _id : x.item } ,
        { $inc : { number : -1 * x.quantity } },
        session=session
    )
session.commit_transaction()
```

Used Keywords and Functions

- `db.collection.find(..,)`
- `db.collection.aggregate{..}`
 - `$lookup, $unwind, $project, $group, $match`
- `db.collection.update(..set), db.collection.drop({})`

Find

MongoDB Shell ▼



```
db.inventory.insertMany([
  { item: "journal", qty: 25, size: { h: 14, w: 21, uom: "cm" }, status: "A" },
  { item: "notebook", qty: 50, size: { h: 8.5, w: 11, uom: "in" }, status: "A" },
  { item: "paper", qty: 100, size: { h: 8.5, w: 11, uom: "in" }, status: "D" },
  { item: "planner", qty: 75, size: { h: 22.85, w: 30, uom: "cm" }, status: "D" },
  { item: "postcard", qty: 45, size: { h: 10, w: 15.25, uom: "cm" }, status: "A" }
]);
```

MongoDB Shell ▼



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

MongoDB Shell ▼



```
db.inventory.find( { status: "D" } )
```

Find

Operators

```
db.movies.find( { "released": { $lt: ISODate("2000-01-01") } } );
```



```
db.movies.find( { "awards.wins": { $gt: 100 } } );
```



```
db.movies.find( { "languages": { $in: [ "Japanese", "Mandarin" ] } } )
```



Projections

```
db.movies.find( { }, { "title": 1, "directors": 1, "year": 1 } );
```



```
db.movies.find( { }, { "_id": 0, "title": 1, "genres": 1 } );
```



Join

```
{  
  $lookup:  
  {  
    from: <collection to join>,  
    let: { <var_1>: <expression>, ..., <var_n>: <expression> },  
    pipeline: [ <pipeline to execute on the joined collection> ],  
    as: <output array field>  
  }  
}
```

Join

```
db.orders.insertMany( [  
  { "_id" : 1, "item" : "almonds", "price" : 12, "quantity" : 2 },  
  { "_id" : 2, "item" : "pecans", "price" : 20, "quantity" : 1 },  
  { "_id" : 3 }  
] )
```

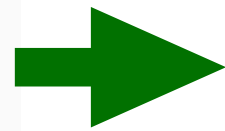


```
db.inventory.insertMany( [  
  { "_id" : 1, "sku" : "almonds", "description": "product 1", "instock" : 120 },  
  { "_id" : 2, "sku" : "bread", "description": "product 2", "instock" : 80 },  
  { "_id" : 3, "sku" : "cashews", "description": "product 3", "instock" : 60 },  
  { "_id" : 4, "sku" : "pecans", "description": "product 4", "instock" : 70 },  
  { "_id" : 5, "sku": null, "description": "Incomplete" },  
  { "_id" : 6 }  
] )
```




Join

```
db.orders.aggregate( [  
  {  
    $lookup:  
    {  
      from: "inventory",  
      localField: "item",  
      foreignField: "sku",  
      as: "inventory_docs"  
    }  
  }  
] )
```



```
{  
  "_id" : 1,  
  "item" : "almonds",  
  "price" : 12,  
  "quantity" : 2,  
  "inventory_docs" : [  
    { "_id" : 1, "sku" : "almonds", "description" : "product 1", "instock" : 1 }  
  ]  
}  
{  
  "_id" : 2,  
  "item" : "pecans",  
  "price" : 20,  
  "quantity" : 1,  
  "inventory_docs" : [  
    { "_id" : 4, "sku" : "pecans", "description" : "product 4", "instock" : 70 }  
  ]  
}  
{  
  "_id" : 3,  
  "inventory_docs" : [  
    { "_id" : 5, "sku" : null, "description" : "Incomplete" },  
    { "_id" : 6 }  
  ]  
}
```

Department Collection

Value 		Type
{ id : "SS", name : "Social Services", location : "Unter den Linden 14, Berlin, 10117" } (4 fields)		Document
{ id : "PMC", name : "Prime Minister and Cabinet", location : "2590 Pearl Street, Boulder, CO 80501" } (4 fields)		Document
{ id : "IIS", name : "Industry, Innovation and Science", location : "114 Willits Street, Birmingham, AL 35203" } (4 fields)		Document
{ id : "HS", name : "Human Services", location : "320 N. Morgan, Chicago, IL 60607" } (4 fields)		Document
{ id : "H", name : "Health", location : "355 Main Street, Cambridge, MA 02142" } (4 fields)		Document
{ id : "F", name : "Finance", location : "8 Rue de Londres, Paris, 75009" } (4 fields)		Document
{ id : "ET", name : "Education and Training", location : "7 Goldschmidtstraße, Göttingen, 37077" } (4 fields)		Document
{ id : "CA", name : "Communications and Arts", location : "1253 McGill College, Montreal, Quebec H3G 1M6" } (4 fields)		Document
{ id : "AWR", name : "Agriculture and Water Resources" } (4 fields)		Document
{ id : "DEF", name : "Defence", location : "1 Infinite Loop, Cupertino, CA 95014" } (4 fields)		Document

Enron Collections

<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@t</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@t	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("5b0d26b89587ce1de41dce14")</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("5b0d26b89587ce1de41dce14")	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@t																												
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("5b0d26b89587ce1de41dce14")																												
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																												

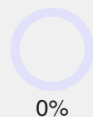
NosqlConcepts Tool

<http://162.55.237.41:8765/>

DASHBOARD

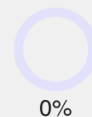
Welcome to the DBMS practical course! Please make sure that you are connected to the DB with your given username and password. If you are not connected yet, click on SIGN IN in the top right. After connecting you should be redirected to this page. You can then START your tasks. Feel free to ask your tutors for help. Have fun!

PostgreSQL



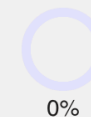
START

Cassandra



START

Neo4J



START

MongoDB

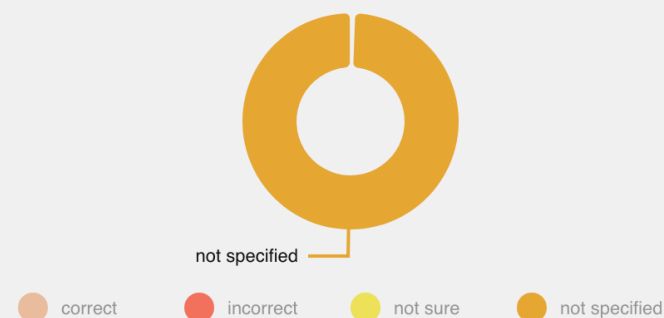


START

Statistics (Implementation In Progress...)

Countdown until Christmas
20
days left

Self-Determined Correctness of Cypher Queries



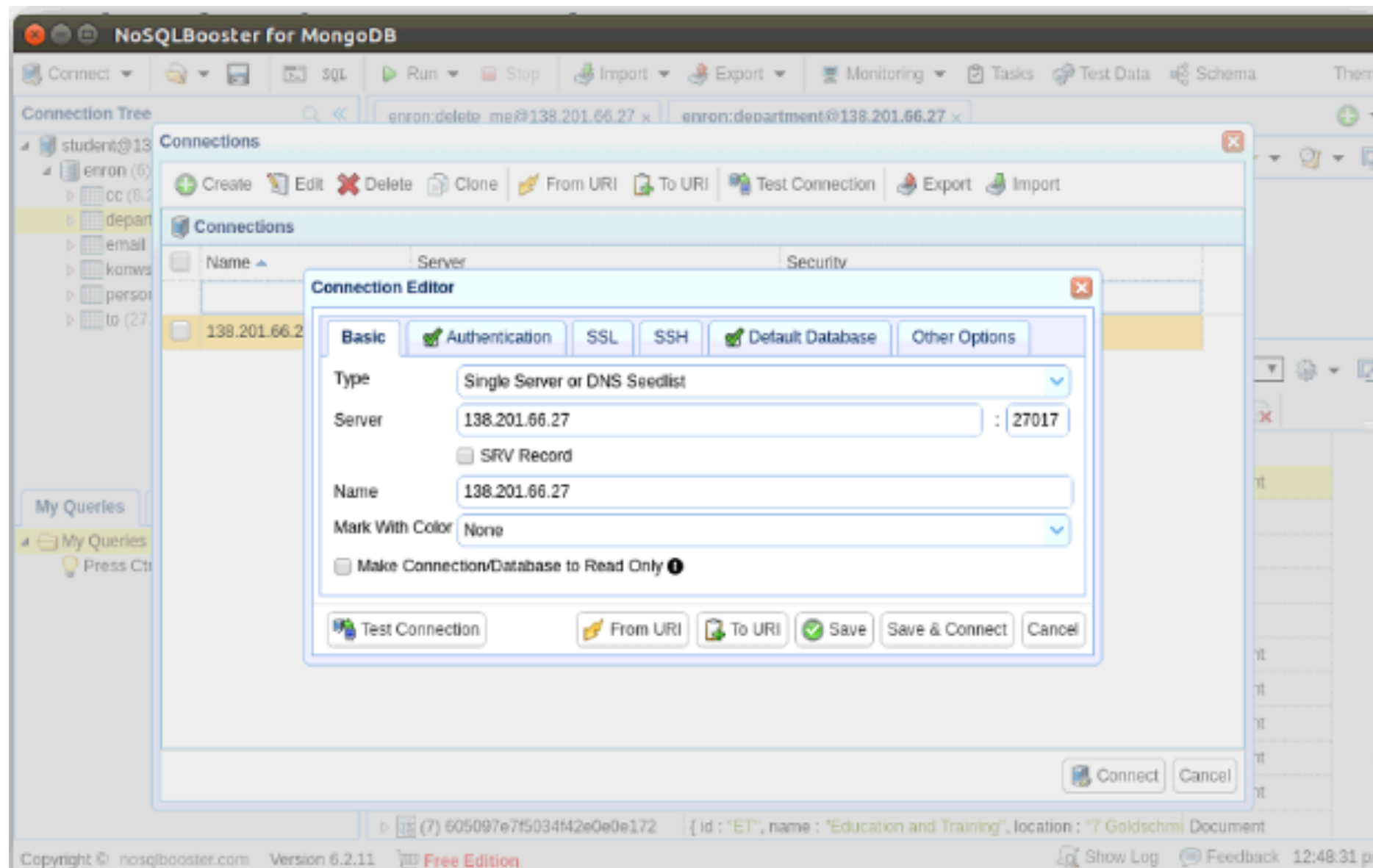
Evaluation

Follow the link to help us improving this webapplication.

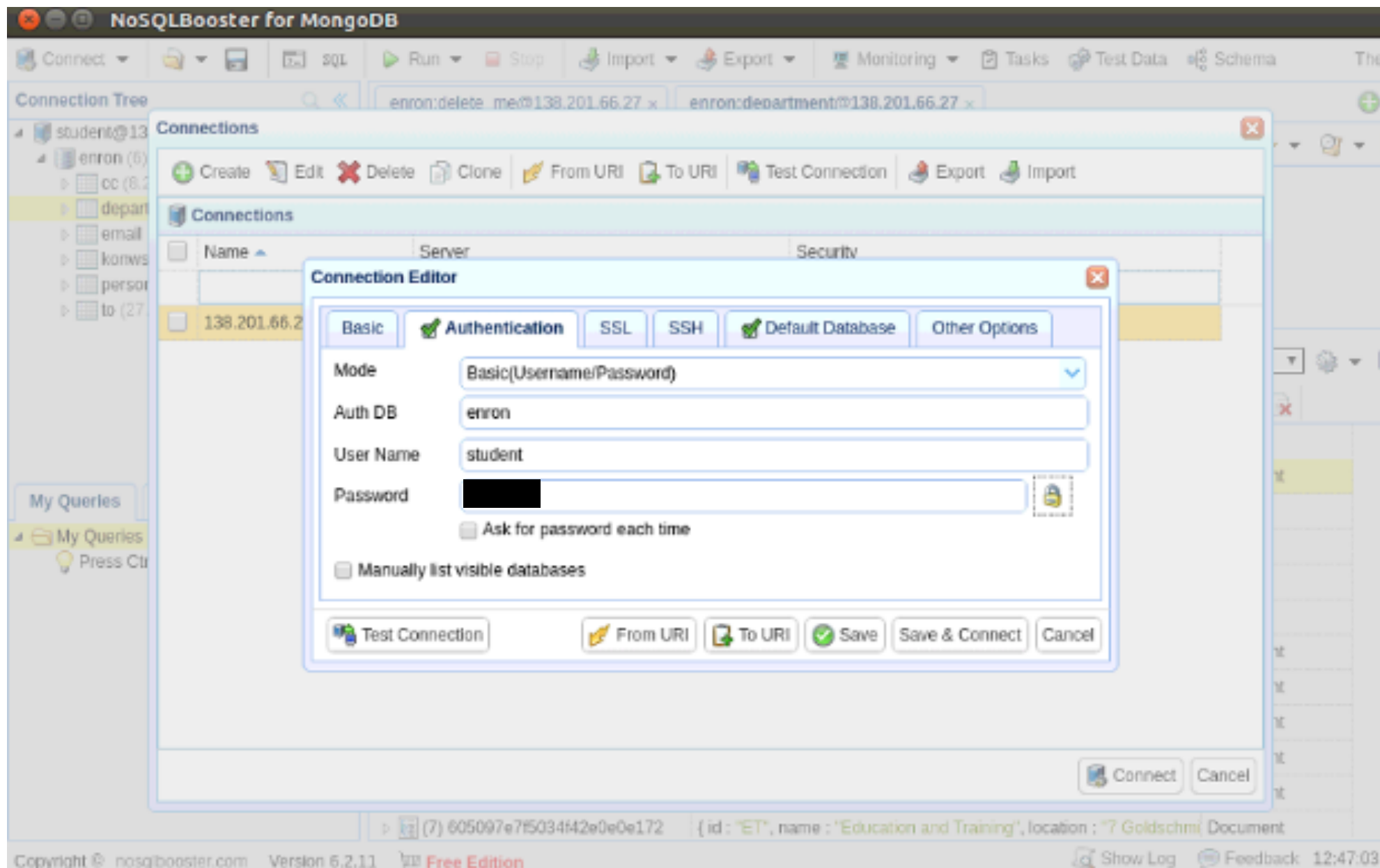
[SoSci Survey Link](#)

MongoDB Client

- To connect to our MongoDB server you can use “NoSQLBooster”.
- <https://nosqlbooster.com/downloads>



MongoDB Client



“Questions”