8/11/2021

DATABASE IMPLEMENTATION

PROJECT 2

SUDHARSAN RAJAM (1001874246)
SIMRAN SINGHI (1001863958)

DATABASE PROJECT 2

ABSTRACT:

In this project, we used MongoDB as an example of a document-oriented NOSQL system, and see how data is stored and queried in such a system. We have stored data in a document (complex object) JSON format. The files are loaded to MongoDB and queries are fired using Python Program.

Designed three document collections (complex objects) corresponding to the following data and stored each as a document collection in MongoDB

1. The PROJECTS document collection will store a collection of PROJECT documents. Each PROJECT document will include the following data about each PROJECT object (document): PNAME, PNUMBER, DNAME (for the controlling DEPARTMENT), and a collection of the workers (EMPLOYEES) who work on the project. This will be nested within the PROJECT object (document) and will include for each worker: EMP_LNAME, EMP_FNAME, HOURS.
2. The EMPLOYEES document collection will store a collection of EMPLOYEE documents. Each EMPLOYEE document will include the following data about each EMPLOYEE object (document): EMP_LNAME, EMP_FNAME, DNAME (department where the employee works), and a collection of the projects that the employee works on. This will be nested within the EMPLOYEE object (document) and will include for each project: PNAME, PNUMBER, HOURS.
3. The DEPARTMENTS document collection will store a collection of DEPARTMENT documents. Each DEPARTMENT document will include the following data about each DEPARTMENT object (document): DNAME, MANAGER_LNAME (the last name of the employee who manages the department), MGR_START_DATE, and a collection of the employees who work for that department. This will be nested within the DEPARTMENT object (document) and will include for each employee: E_LNAME, E_FNAME, SALARY.

PSUEDOCODE

1. Define function department()
2. Initialize variables mydb, mycol, mycol2;
   #store collections in db named COMPANY
3. **Aggregation functions to execute**

```
'$lookup': {
   'from': 'EMPLOYEE',
   'localField': 'MANAGER_SSN',
   'foreignField': 'EMP_SSN',
   'as': 'MANAGER_LNAME'
}
```

```
},
{
  '$lookup': {
    'from': 'EMPLOYEE',
    'localField': 'DNUM',
    'foreignField': 'DNUM',
    'as': 'EMP_DETAILS'
  }
},
{
  '$project': {
    'DNAME': 1,
    'MANAGER_LNAME.E_LNAME': 1,
    'MGR_START_DATE': 1,
    'EMP_DETAILS.E_LNAME': 1,
    'EMP_DETAILS.E_FNAME': 1,
    'EMP_DETAILS.SALARY': 1
  }
}
```

4. result_array = [];
5. For item in result, store iterm in result_array;
6. Define employee();
7. Repeat Step 2
8. Aggregation function to execute employee query

```
{
  '$lookup': {
    'from': 'DEPARTMENT',
    'localField': 'DNUM',
    'foreignField': 'DNUM',
    'as': 'DEPARTMENT_DETAILS'
  }
},
{
  '$lookup': {
    'from': 'PROJECT',
    'localField': 'DNUM',
    'foreignField': 'DNUM',
    'as': 'PROJECT_DETAILS'
  }
},
{
  '$lookup': {
    'from': 'WORKS_ON',
    'localField': 'EMP_SSN',
    'foreignField': 'EMP_SSN',
    'as': 'PROJECT_HOURS'
```

```
            }
        },
        {
            '$project': {
                'E_FNAME': 1,
                'E_LNAME': 1,
                'DEPARTMENT_DETAILS.DNAME': 1,
                'PROJECT_DETAILS.PNAME': 1,
                'PROJECT_DETAILS.PNUM': 1,
                'PROJECT_HOURS.HOURS': 1
            }
        }
    ])
```

9. Repeat step 4 and 5;
10. Define Project();
11. Repeat Step 2;
12. Aggregation function to execute project query

```
    {
        '$lookup': {
            'from': 'DEPARTMENT',
            'localField': 'DNUM',
            'foreignField': 'DNUM',
            'as': 'DEPARTMENT_DETAILS'
        }
    },
    {
        '$lookup': {
            'from': 'EMPLOYEE',
            'localField': 'DNUM',
            'foreignField': 'DNUM',
            'as': 'EMPLOYEE_DETAILS'
        }
    },
    {
        '$lookup': {
            'from': 'WORKS_ON',
            'localField': 'PNUM',
            'foreignField': 'PNUMBER',
            'as': 'PROJECT_HOURS'
        }
    },
    {
        '$project': {
            'PNAME': 1,
            'PNUM': 1,
            'DEPARTMENT_DETAILS.DNAME': 1,
            'EMPLOYEE_DETAILS.E_FNAME': 1,
```

```
                'EMPLOYEE_DETAILS.E_LNAME': 1,
                'PROJECT_HOURS.HOURS': 1
            }
        }
    ])
```

13. Repeat Steps 4 and 5
14. Function call:
15. department();
16. employee();
17. project();
18. # Export Collection as JSON FILES
19. get_json(conn,'RESULT_DEPARTMENT.json','RESULT_DEPARTMENT')
20. get_json(conn,'RESULT_EMPLOYEE.json','RESULT_EMPLOYEE')
21. get_json(conn,'RESULT_PROJECT.json','RESULT_PROJECT')

22. # close Connection
23. conn.close()

24. # Convert Json File to XML Files
25. convertJson2Xml('JSON_FILES/RESULT_DEPARTMENT.json','RESULT_DEPARTMENT.xml')
26. convertJson2Xml('JSON_FILES/RESULT_EMPLOYEE.json','RESULT_EMPLOYEE.xml')
27. convertJson2Xml('JSON_FILES/RESULT_PROJECT.json','RESULT_PROJECT.xml')

PROGRAMMING LANGUAGE AND SYSTEMS USED:

- Python Programming
- DATABASE: MONGO DB
- SYSTEMS: MONGO DB COMPASS, VS CODE
- PYTHON PACKAGES: PYMONGO

APPROACH EMPLOYED:

- **OPTION 2:** Load the data into MongoDB as a relational normalized data. Then write three different queries to extract the data needed for each document collection that creates new document collections to convert the query result from the normalized format into JSON for loading the nested structures in MongoDB. But **you must create the nested structure within each object**

PROJECT CREATION STEPS:

- Downloaded VS Code and Mongo DB COMPASS
- Converted DEPARTMENT.txt, EMPLOYEE.txt, PROJECT.txt, WORKS_ON.txt to CSV files.
- Set UP connection in MONGO DB LOCAL HOST
- Create database COMPANY in MONGODB
- Create collections named EMPLOYEE, DEPARTMENT, PROJECT and WORKS_ON
- Upload the csv files in EMPLOYEE, DEPARTMENT, PROJECT and WORKS_ON collection one by one the mongo DB compass
- Set up connection to the local host:27017 in Mongo DB Compass
- In VSCODE, written a connection code and Aggregation function used to query the documents provided
- Stored the result in a Array
- Created a queries() function to see the query results in querydocument.txt
- Run the program and results will be generated in Mongo DB compass

PROJECT CREATION STEPS FOR EXTRA CREDIT(JSON TO XML):

- Created a extracredit.py python file
- Import json2xml package
- PSUEDO CODE
    - def convertJson2Xml(input_file_name,output_file_name):
    - data = readfromjson(input_file_name)
    - xml = json2xml.Json2xml(data, wrapper="all", pretty=True, attr_type=False).to_xml()

SECOND APPROACH----------

- Imported json and dicttoxml package
- Read the JSON file DEPARMENT_RESULT and load it into Data variable
- Apply xml = dicttoxml.dicttoxml(data)
- Write str(xml) to file sample.txt
- A sample.xml file will be generated

HOW TO RUN main.py (INSTRUCTIONS FOR TA)

- Download VSCODE and MongoDB
- Create database named COMPANY in Mongo DB Compass
- Create collections named EMPLOYEE, DEPARTMENT, PROJECT and WORKS_ON
- Upload the csv files in EMPLOYEE, DEPARTMENT, PROJECT and WORKS_ON collection one by one the mongo db compass
- Set up connection to the local host:27017 in Mongo DB Compass
- Import pymongo package if not installed by the command “pip3 install pymongo”
- Run main.py using the command "python main.py" or else you can click the play button above to run the program
- Go to Mongo DB compass to see the Result files by clicking refresh button
- Also, A "querydocument" will be generated. it contains the result of queries fired in the queries() function.

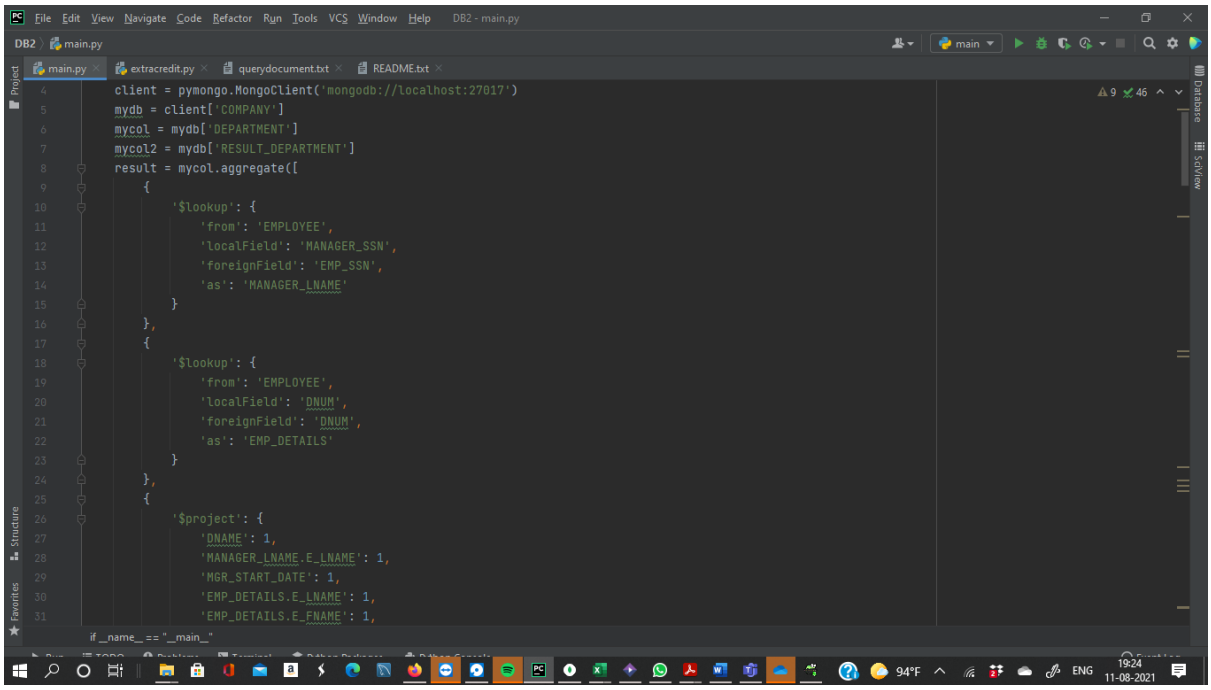
TEAM MEMBERS CONTRIBUTION:

Both team member worked together and did equal amount of work

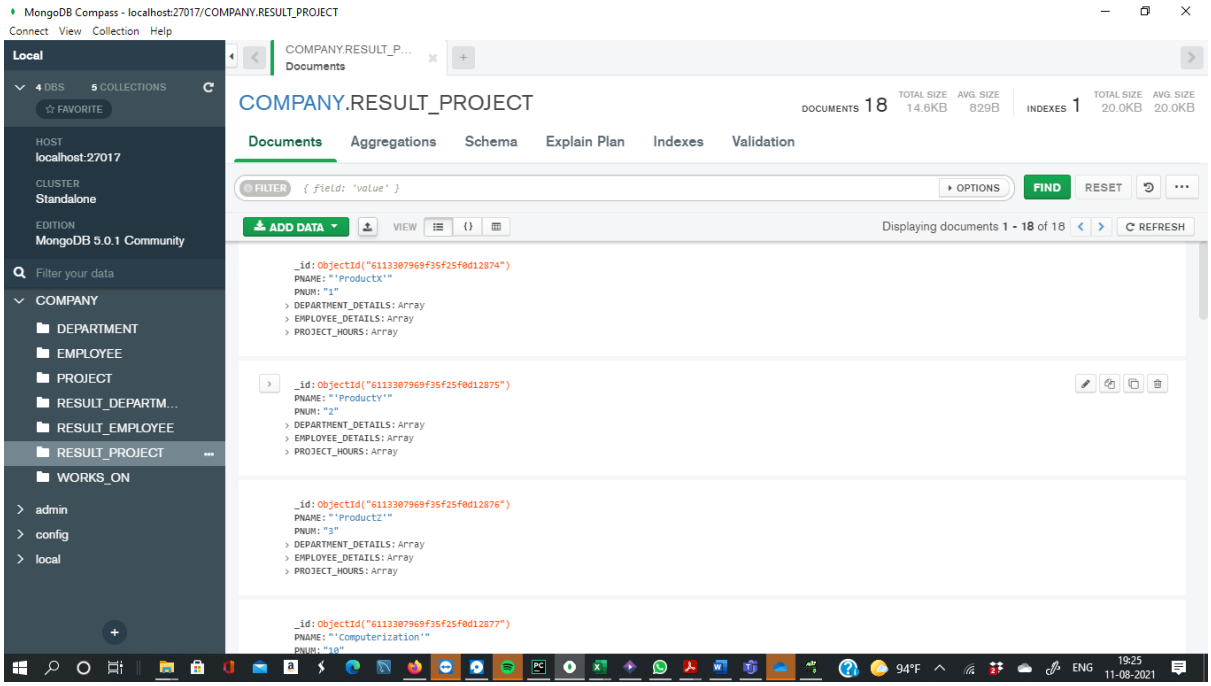
- SUDHARSAN RAJAM:
  - Read and understood aggregation functions and their use
  - Written the pseudocode for program
  - Downloaded all the resources needed for the project
  - Helped in collecting all the info need for report content
  - Converted txt files to csv
  - Set up connection to mongo DB

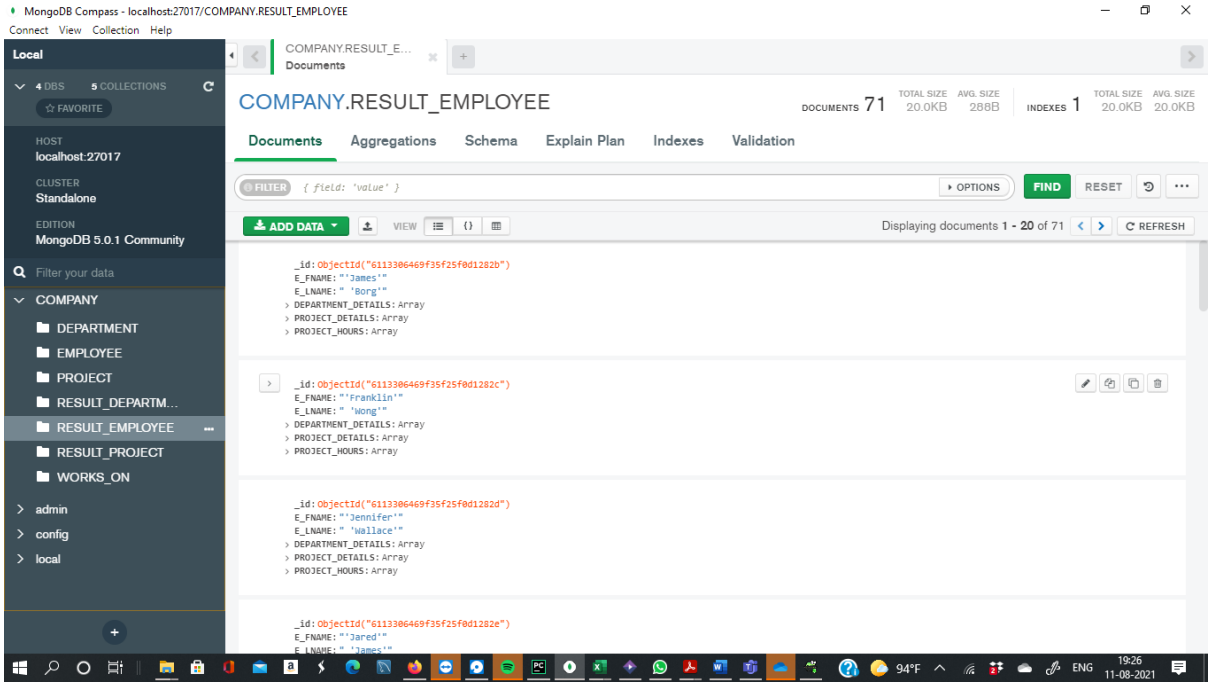
- SIMRAN SINGHI:
  - Imported the CSV files in the collection and created database
  - Read and understood aggregation functions and their use
  - Created Python program using the Aggregation function like $lookup, $project etc.
  - Read and understood the process of converting JSON to XML and executed extra credit program
  - Documented Project Report

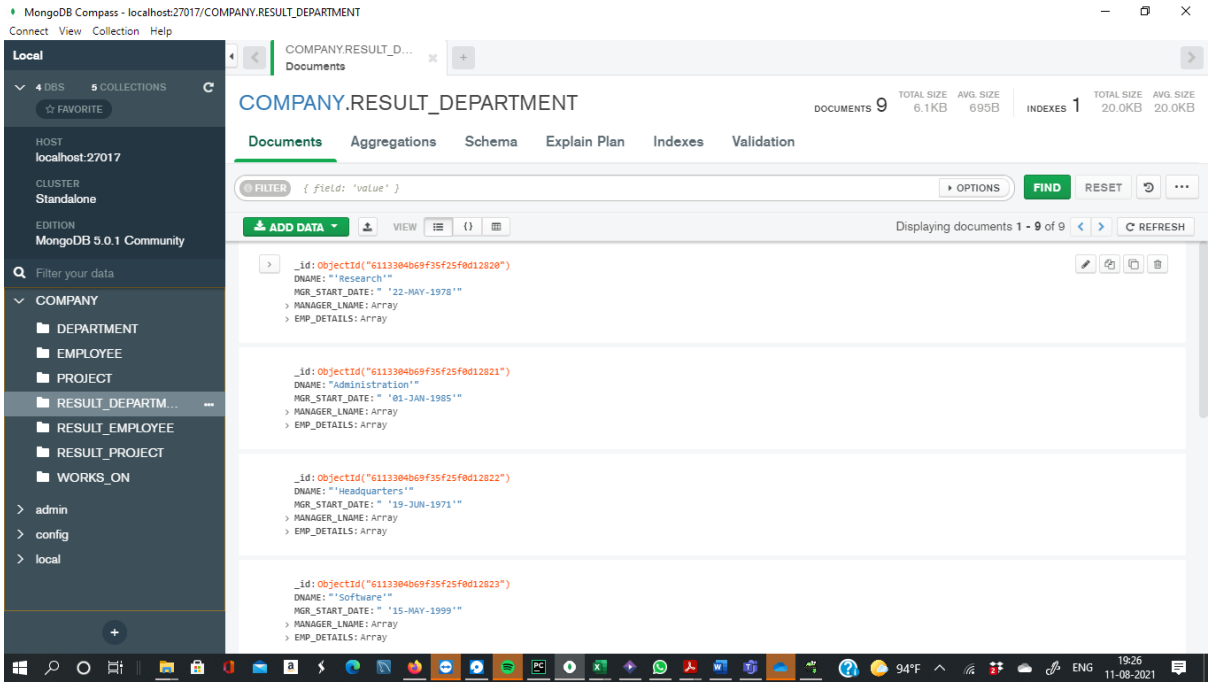
SCREENSHOTS FOR REFERENCE

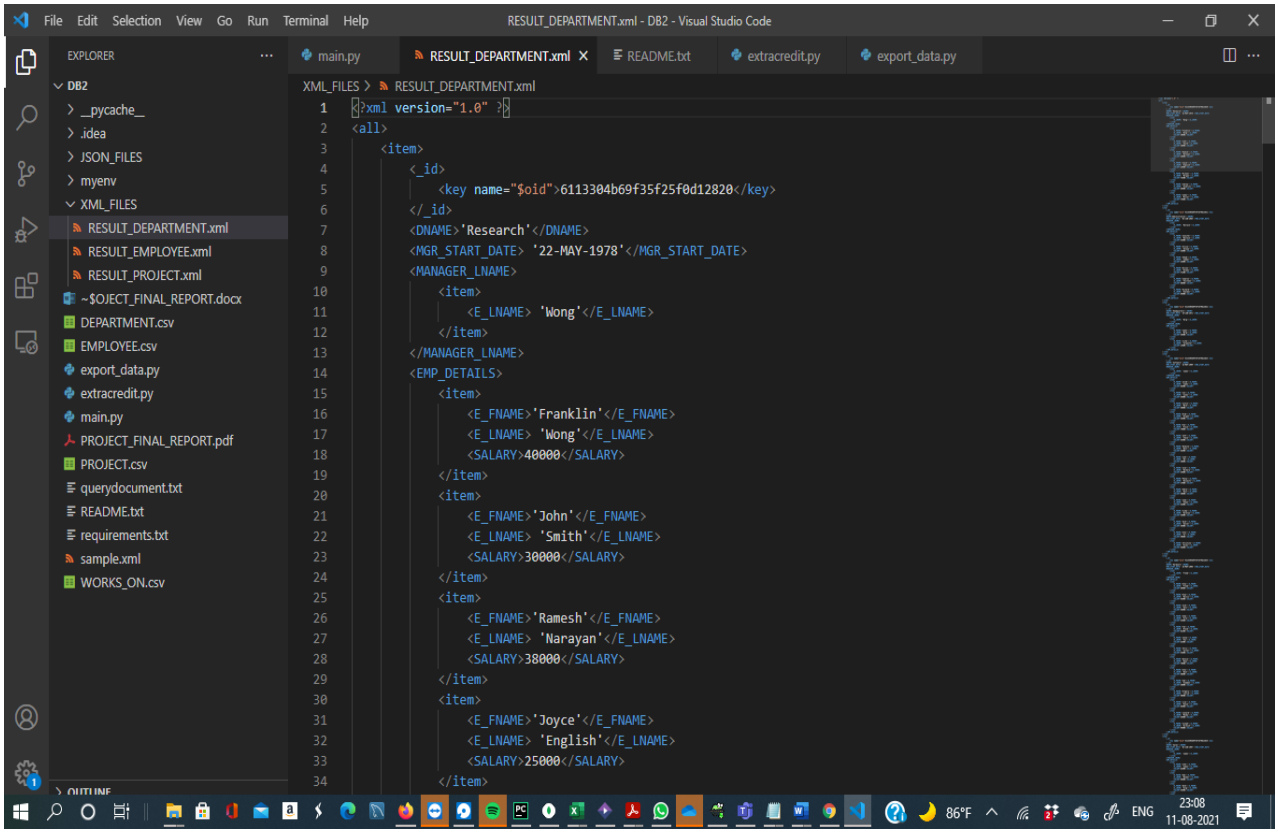


```
client = pymongo.MongoClient('mongodb://localhost:27017')
mydb = client['COMPANY']
mycol = mydb['DEPARTMENT']
mycol2 = mydb['RESULT_DEPARTMENT']
result = mycol.aggregate([
    {
        '$lookup': {
            'from': 'EMPLOYEE',
            'localField': 'MANAGER_SSN',
            'foreignField': 'EMP_SSN',
            'as': 'MANAGER_LNAME'
        }
    },
    {
        '$lookup': {
            'from': 'EMPLOYEE',
            'localField': 'DNUM',
            'foreignField': 'DNUM',
            'as': 'EMP_DETAILS'
        }
    },
    {
        '$project': {
            'DNAME': 1,
            'MANAGER_LNAME.E_LNAME': 1,
            'MGR_START_DATE': 1,
            'EMP_DETAILS.E_LNAME': 1,
            'EMP_DETAILS.E_FNAME': 1,
```


MongoDB Compass - localhost:27017/COMPANY.RESULT_PROJECT
Connect View Collection Help
Local
4 DBS 5 COLLECTIONS
FAVORITE
HOST
localhost:27017
CLUSTER
Standalone
EDITION
MongoDB 5.0.1 Community
Filter your data
COMPANY
DEPARTMENT
EMPLOYEE
PROJECT
RESULT_DEPARTM...
RESULT_EMPLOYEE
RESULT_PROJECT
WORKS_ON
admin
config
local
COMPANY.RESULT_P...
Documents
COMPANY.RESULT_PROJECT
DOCUMENTS 18 TOTAL SIZE 14.6KB AVG. SIZE 829B
INDEXES 1 TOTAL SIZE 20.0KB AVG. SIZE 20.0KB
Documents Aggregations Schema Explain Plan Indexes Validation
FILTER { field: 'value' }
OPTIONS
FIND
RESET
ADD DATA
VIEW
Displaying documents 1 - 18 of 18
REFRESH
_id: ObjectId("6113307969f35f25f0d12874")
PNAME: "'ProductX'"
PNUM: "1"
DEPARTMENT_DETAILS: Array
EMPLOYEE_DETAILS: Array
PROJECT_HOURS: Array
_id: ObjectId("6113307969f35f25f0d12875")
PNAME: "'ProductY'"
PNUM: "2"
DEPARTMENT_DETAILS: Array
EMPLOYEE_DETAILS: Array
PROJECT_HOURS: Array
_id: ObjectId("6113307969f35f25f0d12876")
PNAME: "'ProductZ'"
PNUM: "3"
DEPARTMENT_DETAILS: Array
EMPLOYEE_DETAILS: Array
PROJECT_HOURS: Array
_id: ObjectId("6113307969f35f25f0d12877")
PNAME: "'Computerization'"
94°F
ENG
19:25
11-08-2021


MongoDB Compass - localhost:27017/COMPANY.RESULT_EMPLOYEE
Connect View Collection Help
Local
4 DBS 5 COLLECTIONS
FAVORITE
HOST
localhost:27017
CLUSTER
Standalone
EDITION
MongoDB 5.0.1 Community
Filter your data
COMPANY
DEPARTMENT
EMPLOYEE
PROJECT
RESULT_DEPARTM...
RESULT_EMPLOYEE
RESULT_PROJECT
WORKS_ON
admin
config
local
COMPANY.RESULT_E...
Documents
COMPANY.RESULT_EMPLOYEE
DOCUMENTS 71 TOTAL SIZE 20.0KB AVG. SIZE 288B
INDEXES 1 TOTAL SIZE 20.0KB AVG. SIZE 20.0KB
Documents Aggregations Schema Explain Plan Indexes Validation
FILTER { field: 'value' }
OPTIONS
FIND
RESET
ADD DATA
VIEW
Displaying documents 1 - 20 of 71
REFRESH
_id: ObjectId("6113306469f35f25f0d1282b")
E_FNAME: "'James'"
E_LNAME: " 'Borg'"
DEPARTMENT_DETAILS: Array
PROJECT_DETAILS: Array
PROJECT_HOURS: Array
_id: ObjectId("6113306469f35f25f0d1282c")
E_FNAME: "'Franklin'"
E_LNAME: " 'Wong'"
DEPARTMENT_DETAILS: Array
PROJECT_DETAILS: Array
PROJECT_HOURS: Array
_id: ObjectId("6113306469f35f25f0d1282d")
E_FNAME: "'Jennifer'"
E_LNAME: " 'Wallace'"
DEPARTMENT_DETAILS: Array
PROJECT_DETAILS: Array
PROJECT_HOURS: Array
_id: ObjectId("6113306469f35f25f0d1282e")
E_FNAME: "'Jared'"
94°F
ENG
19:26
11-08-2021


MongoDB Compass - localhost:27017/COMPANY.RESULT_DEPARTMENT
Connect View Collection Help
Local
4 DBS 5 COLLECTIONS
FAVORITE
HOST
localhost:27017
CLUSTER
Standalone
EDITION
MongoDB 5.0.1 Community
Filter your data
COMPANY
DEPARTMENT
EMPLOYEE
PROJECT
RESULT_DEPARTM...
RESULT_EMPLOYEE
RESULT_PROJECT
WORKS_ON
admin
config
local
COMPANY.RESULT_D...
Documents
COMPANY.RESULT_DEPARTMENT
DOCUMENTS 9 TOTAL SIZE 6.1KB AVG. SIZE 695B
INDEXES 1 TOTAL SIZE 20.0KB AVG. SIZE 20.0KB
Documents Aggregations Schema Explain Plan Indexes Validation
FILTER { field: 'value' }
OPTIONS
FIND
RESET
ADD DATA
VIEW
Displaying documents 1 - 9 of 9
REFRESH
_id: ObjectId("6113304b69f35f25f0d12820")
DNAME: "'Research'"
MGR_START_DATE: " '22-MAY-1978'"
MANAGER_LNAME: Array
EMP_DETAILS: Array
_id: ObjectId("6113304b69f35f25f0d12821")
DNAME: "Administration'"
MGR_START_DATE: " '01-JAN-1985'"
MANAGER_LNAME: Array
EMP_DETAILS: Array
_id: ObjectId("6113304b69f35f25f0d12822")
DNAME: "'Headquarters'"
MGR_START_DATE: " '19-JUN-1971'"
MANAGER_LNAME: Array
EMP_DETAILS: Array
_id: ObjectId("6113304b69f35f25f0d12823")
DNAME: "'Software'"
MGR_START_DATE: " '15-MAY-1999'"
MANAGER_LNAME: Array
EMP_DETAILS: Array
94°F
ENG
19:26
11-08-2021


File Edit Selection View Go Run Terminal Help
RESULT_DEPARTMENT.xml - DB2 - Visual Studio Code
EXPLORER
DB2
__pycache__
.idea
JSON_FILES
myenv
XML_FILES
RESULT_DEPARTMENT.xml
RESULT_EMPLOYEE.xml
RESULT_PROJECT.xml
~$OJECT_FINAL_REPORT.docx
DEPARTMENT.csv
EMPLOYEE.csv
export_data.py
extracredit.py
main.py
PROJECT_FINAL_REPORT.pdf
PROJECT.csv
querydocument.txt
README.txt
requirements.txt
sample.xml
WORKS_ON.csv
main.py
RESULT_DEPARTMENT.xml
README.txt
extracredit.py
export_data.py
XML_FILES > RESULT_DEPARTMENT.xml
<?xml version="1.0" ?>
<all>
<item>
<_id>
<key name="$oid">6113304b69f35f25f0d12820</key>
</_id>
<DNAME>'Research'</DNAME>
<MGR_START_DATE> '22-MAY-1978'</MGR_START_DATE>
<MANAGER_LNAME>
<item>
<E_LNAME> 'Wong'</E_LNAME>
</item>
</MANAGER_LNAME>
<EMP_DETAILS>
<item>
<E_FNAME>'Franklin'</E_FNAME>
<E_LNAME> 'Wong'</E_LNAME>
<SALARY>40000</SALARY>
</item>
<item>
<E_FNAME>'John'</E_FNAME>
<E_LNAME> 'Smith'</E_LNAME>
<SALARY>30000</SALARY>
</item>
<item>
<E_FNAME>'Ramesh'</E_FNAME>
<E_LNAME> 'Narayan'</E_LNAME>
<SALARY>38000</SALARY>
</item>
<item>
<E_FNAME>'Joyce'</E_FNAME>
<E_LNAME> 'English'</E_LNAME>
<SALARY>25000</SALARY>
</item>
OUTLINE
86°F
ENG
23:08
11-08-2021