**CSE 5331 – project 2 report**

**Team Members:**

Saahil Pralhad Anande(1001855308)

Unnathi Reddy Nalla (1001828087)

**Programing language:** Python

**Database:** Mysql, mongodb

**Libraries**: json, mysql.connector, pymongo, dicttoxml, ast

**Code structure: Option 1:** Load the data in a relational DBMS such as MySQL

1. To implement this project we first create 4 table in MYSQL. Employee, department, project, workson.
2. After creating the table run the python file name data\_to\_mysql which would read different text file and upload its data to mysql
3. When the data is uploaded to mysql run the python file convert\_to\_json\_and\_mongo\_import which would extract the data from mysql database and perform some query to upload the nested document to json and mongodb.
4. Connect to mongodb in locahost and perform queries

**Loaded data in Mysql:**

We have created tables for each given input text files and loaded data into respective tables in Mysql.

* **Employee Table:**

Create Table using the following query-

CREATE TABLE employee (

emp\_first varchar(255),

emp\_last varchar(255),

emp\_ssn int,

emp\_birth varchar(255),

emp\_address varchar(255),

emp\_gender varchar(50),

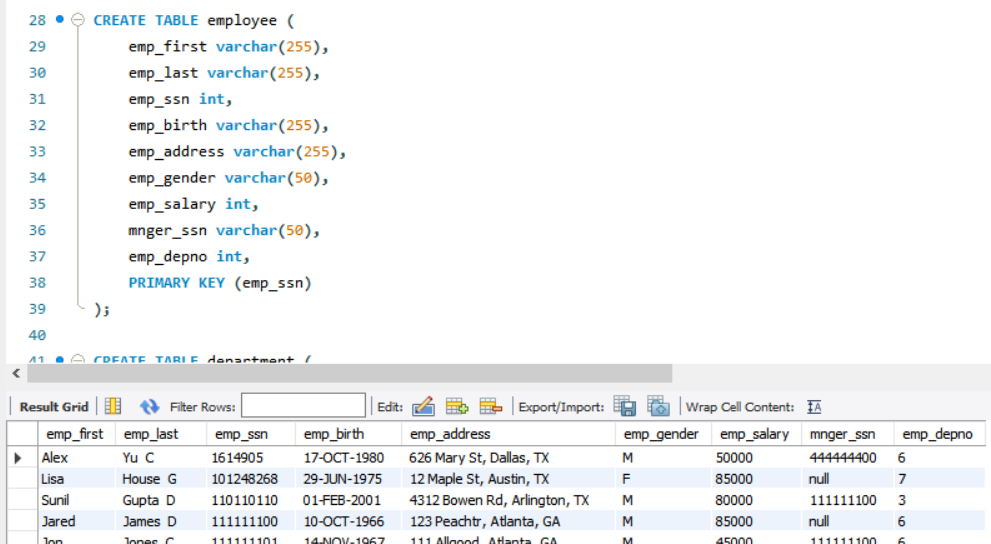
emp\_salary int,

mnger\_ssn varchar(50),

emp\_depno int,

PRIMARY KEY (emp\_ssn)

);



The table is late loaded using data\_to\_mysql python program.

* **Department table:**

Created table department in Mysql with the following query:

CREATE TABLE department (

d\_name varchar(255),

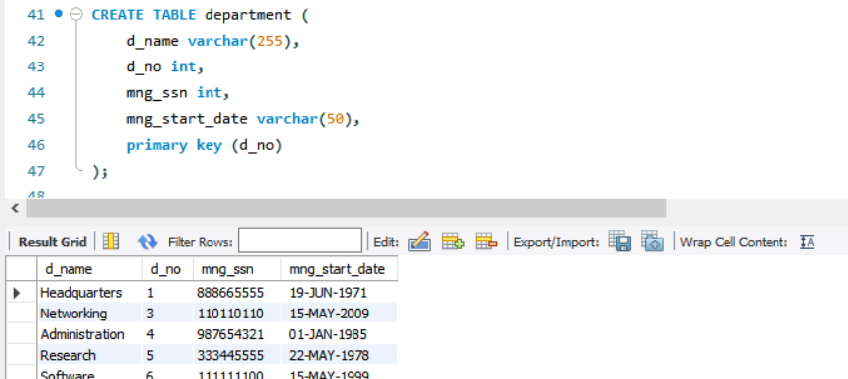
d\_no int,

mng\_ssn int,

mng\_start\_date varchar(50),

primary key (d\_no)

);



The table is late loaded using data\_to\_mysql python program.

* **Project table:**

Created table project in Mysql with the following query:

CREATE TABLE project (

p\_name varchar(255),

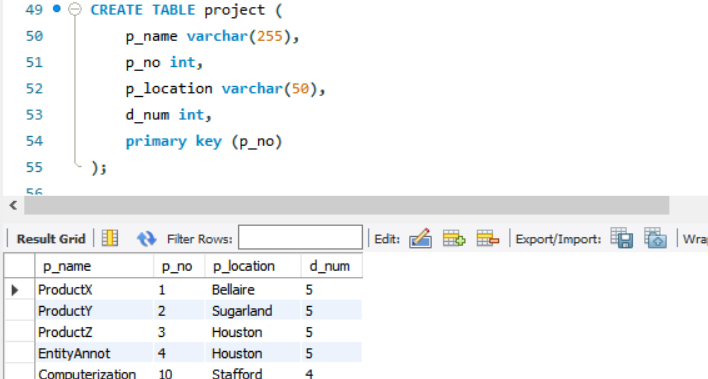
p\_no int,

p\_location varchar(50),

d\_num int,

primary key (p\_no)

);



* **Works on table:**

Created table workson in Mysql with the following query:

CREATE TABLE workson

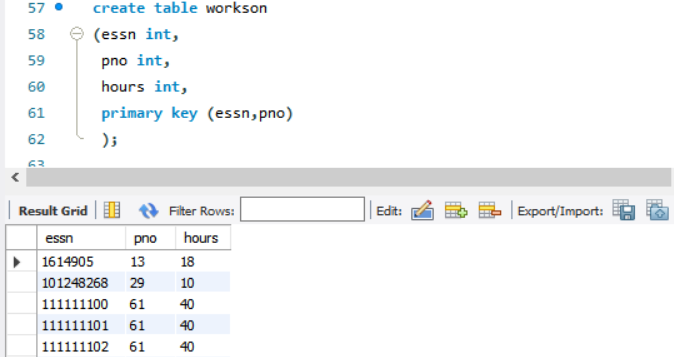
(essn int,

pno int,

hours int,

primary key (essn,pno)

);



The table is late loaded using data\_to\_mysql python program.

**Pseudo code:**

1. **Create tables employee, department, project, workson**.

**#extract and load the mysql table**

emplist = []

with open(FILENAME.txt', 'r') as file: # reading the input file  
 for line in file.readlines():

emplist = line.split(",")

#get row values for each table

**#Connect to mysql**

myConn = pymysql.connect(host=hostname, user=username, passwd=password, db=database )

#write and execute query to insert data in tables.

cursor = newconnection.cursor()  
query = "INSERT INTO employee (emp\_first,emp\_last , emp\_ssn, emp\_birth, emp\_address,emp\_gender,emp\_salary,mnger\_ssn,emp\_depno) VALUES ('"+firstname+"','"+lastname+" "+middlename+"','"+emp\_ssn+"','"+dateofbirth+"','"+address+", "+city+", "+state+"','"+gender+"','"+salary+"','"+mng\_ssn+"','"+depno+"'); "  
cursor.execute(query)  
newconnection.commit()

**#use alter table and set foreign keys**

alter table department add foreign key (mng\_ssn) references employee(emp\_ssn);

alter table project add foreign key (d\_num) references department(d\_no);

alter table workson add foreign key (essn) references employee(emp\_ssn), add foreign key (pno) references project(p\_no);

1. **Once the table has been loaded write queries to join multiple tables.**

mycursor = newconnection.cursor()

query1 = "select p\_no,p\_name,d\_num,emp\_first,emp\_last,hours from project p join department d on p.d\_num = d.d\_no join workson w on w.pno=p.p\_no join employee e on e.emp\_ssn = w.essn order by p.p\_no"

mycursor.execute(query1)

myresult = mycursor.fetchall()

query2 = "select e.emp\_ssn,e.emp\_first,e.emp\_last,d.d\_name,p.p\_name,p.p\_no,w.hours from employee e join workson w on e.emp\_ssn=w.essn join project p on p.p\_no = w.pno join department d on d.d\_no = e.emp\_depno order by e.emp\_ssn"

mycursor.execute(query)

employee\_col\_result = mycursor.fetchall()

query3 = "select d.d\_no, d.d\_name, e.emp\_last, d.mng\_start\_date,e.emp\_last, e.emp\_first, e.emp\_salary from department d join employee e on d.d\_no =e.emp\_depno order by d.d\_no;"

mycursor.execute(query)

department\_col\_result = mycursor.fetchall()

1. Create a dictionaries for project document ‘project\_document\_collection’ for P\_no, P\_name, D\_no and collection of employees emp\_first, emp\_last and hours
2. Create a dictionaries for employee document ‘employee\_document\_collection’ for emp\_first, emp\_last, D\_no and collection of projects P\_no, P\_name and Hours.
3. Create a dictionaries for department document ‘department\_document\_collection’ for d\_name, manger\_last, manger\_start and collection for employees emp\_first, emp\_last, salary.
4. Iterate the fetched data and append the result into the respective dictionaries. For the project document, if the p\_no already exists in the dictionary then we only append the employee collection details else we will append all the project details as well as the employee collection details. For the employee document, if the emp\_ssn already exists in the dictionary then we only append the project collection details else we will append all the employee details as well as the project collection details.

**#for creating project document collection**

jsonresult = {}  
for p\_no, p\_name,d\_num,emp\_first, emp\_last, hours in myresult:  
 if p\_no in jsonresult:  
 emp = {  
 "Employee\_first\_name" : emp\_first,  
 "Employee\_last\_name" : emp\_last,  
 "hours" : hours  
 }  
 jsonresult[p\_no]["employee"].append(emp)  
 else:  
 jsonresult[p\_no] = {  
 "Project\_name" : p\_name,  
 "project\_number" : p\_no,  
 "department\_number" : d\_num,  
 "employee" : [{  
 "Employee\_first\_name" : emp\_first,  
 "Employee\_last\_name" : emp\_last,  
 "hours" : hours  
 }]  
 }

**#for creating employee document collection**

jsonresult = {}  
for emp\_ssn, emp\_first, emp\_last,d\_name,p\_name, p\_no, hours in employee\_col\_result:  
 if emp\_ssn in jsonresult:  
 pro = {  
 "project\_name" : p\_name,  
 "project\_number" : p\_no,  
 "Hours" : hours  
 }  
 jsonresult[emp\_ssn]["project"].append(pro)  
 else:  
 jsonresult[emp\_ssn] = {  
 "Employee\_first\_name" : emp\_first,  
 "Employee\_last\_name" : emp\_last,  
 "department\_name" : d\_name,  
 "project" : [{  
 "project\_name" : p\_name,  
 "project\_number" : p\_no,  
 "Hours" : hours  
 }]  
 }

**#for creating department document collection**

jsonresult = {}

for d\_no, d\_name, emp\_last,mng\_start\_date,emp\_last, emp\_first, emp\_salary in employee\_col\_result:

if d\_no in jsonresult:

empy = {

"Employee\_first\_name" : emp\_first,

"Employee\_last\_name" : emp\_last,

"Salary" : hours

}

jsonresult[d\_no]["employee"].append(pro)

else:

jsonresult[d\_no] = {

"Department\_name" : d\_name,

"Manager\_lastname" : emp\_last,

"Manager\_start\_date" : mng\_start\_date,

"employee" : [{

"Employee\_first\_name" : emp\_first,

"Employee\_last\_name" : emp\_last,

"Salary" : hours

}]

}

**#** **Convert the project\_document\_collection, employee\_document\_collection and department\_document\_collection to json using json\_dumps. Open a file and write the generated json string into department.json, project.jason, employee.jason file**

jsonconv = json.dumps(jsonresult)

**#write to file project\_document.jason**

file="project\_document.json"

with open(file, 'w') as out:

out.write(jsonconv + '\n')

**#write to file employee\_document.jason**  
file="employee\_document.json"  
with open(file, 'w') as out:  
 out.write(jsonconv + '\n')

**#write to file department\_document.jason**  
file="department\_document.json"  
with open(file, 'w') as out:  
 out.write(jsonconv + '\n')

# **Convert the project\_document\_collection, employee\_document\_collection and department\_document\_collection to XML using DICT TO XML. Open a file and write the generated json string into department\_xml.xml, project\_XML.xml, employee\_XML.xml file**

**#xml file converstion**  
convo = literal\_eval(jsonfile)  
xmlfile = dicttoxml(convo)  
  
**#write to file employee\_document.xml**  
with open("EMPLOYEE\_XML.xml", 'w') as out:  
 out.write(str(xmlfile, 'utf-8'))

**#insert the converted Jason to mongodb**

for x in jsonresult:  
 myclient = pymongo.MongoClient(mongodb\_host)  
 mydb = myclient[mongodb\_dbname]  
 mycol = mydb["project\_document"]  
 mycol.insert\_one(jsonresult[x])

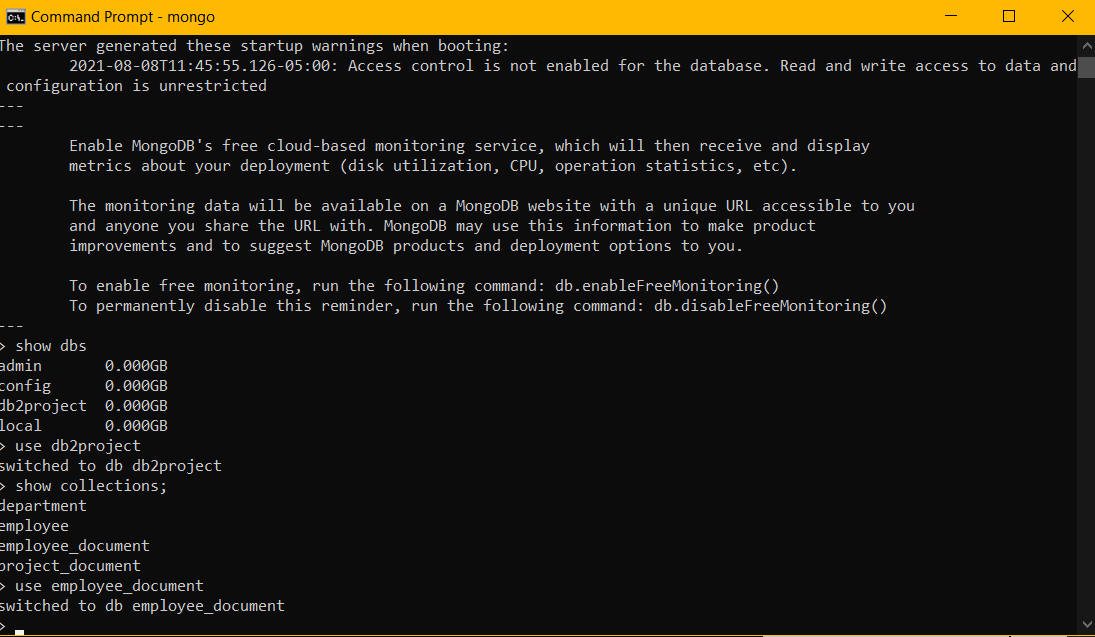
for x in jsonresult:  
 myclient = pymongo.MongoClient(mongodb\_host)  
 mydb = myclient[mongodb\_dbname]  
 mycol = mydb["employee\_document"]  
 mycol.insert\_one(jsonresult[x])

for x in jsonresult:  
 myclient = pymongo.MongoClient(mongodb\_host)  
 mydb = myclient[mongodb\_dbname]  
 mycol = mydb["department\_document"]  
 mycol.insert\_one(jsonresult[x])

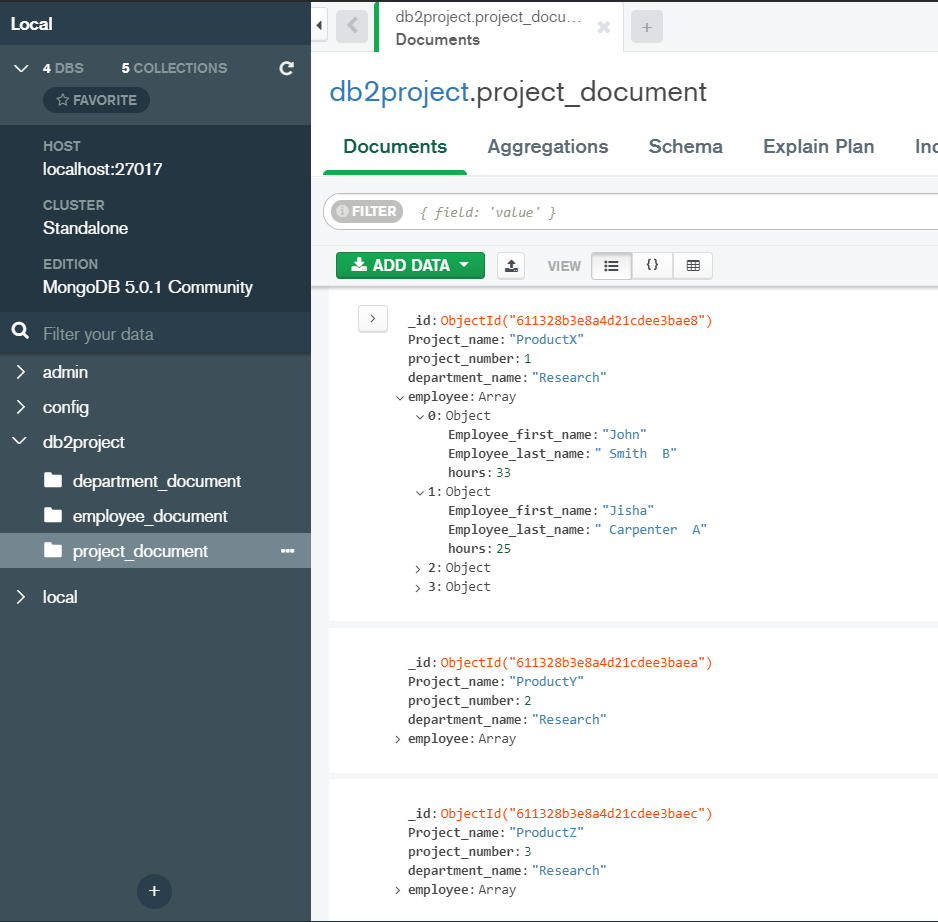
**#now we get mongo collection uber db2databse named project\_document, employee\_document , department\_document**

**#we perform queries on these collections**

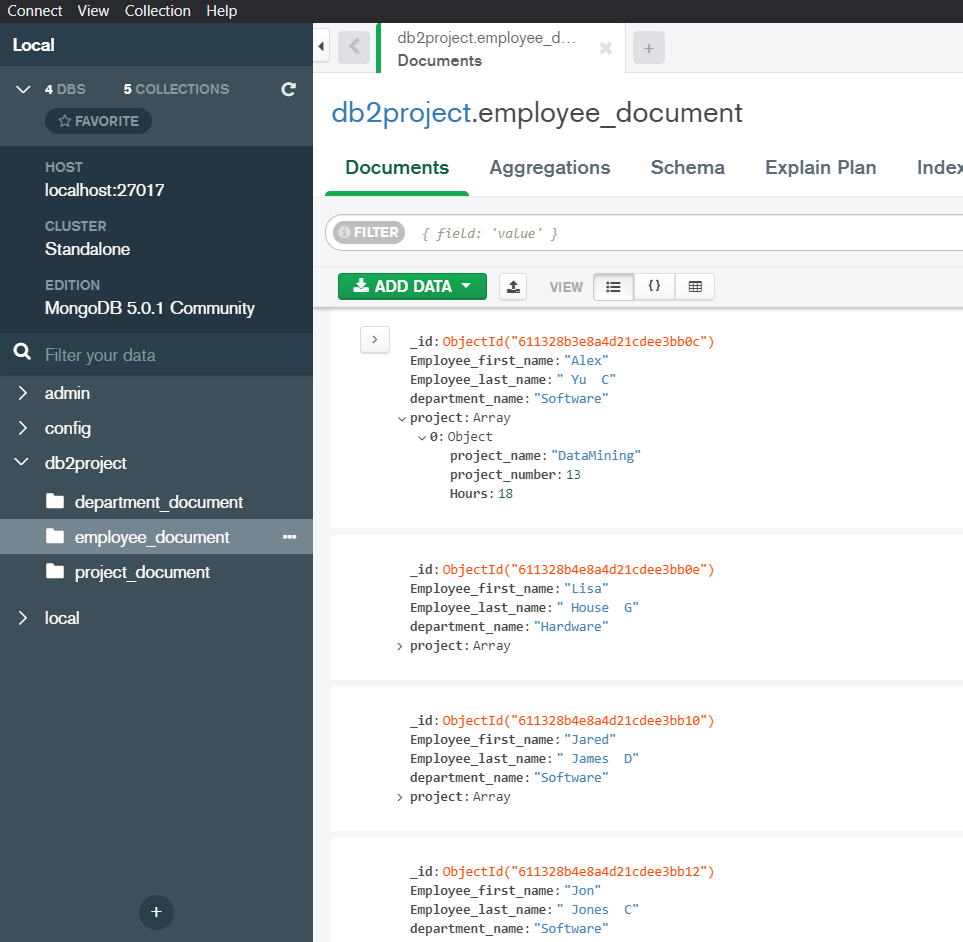
**Mongodb collections:**

****

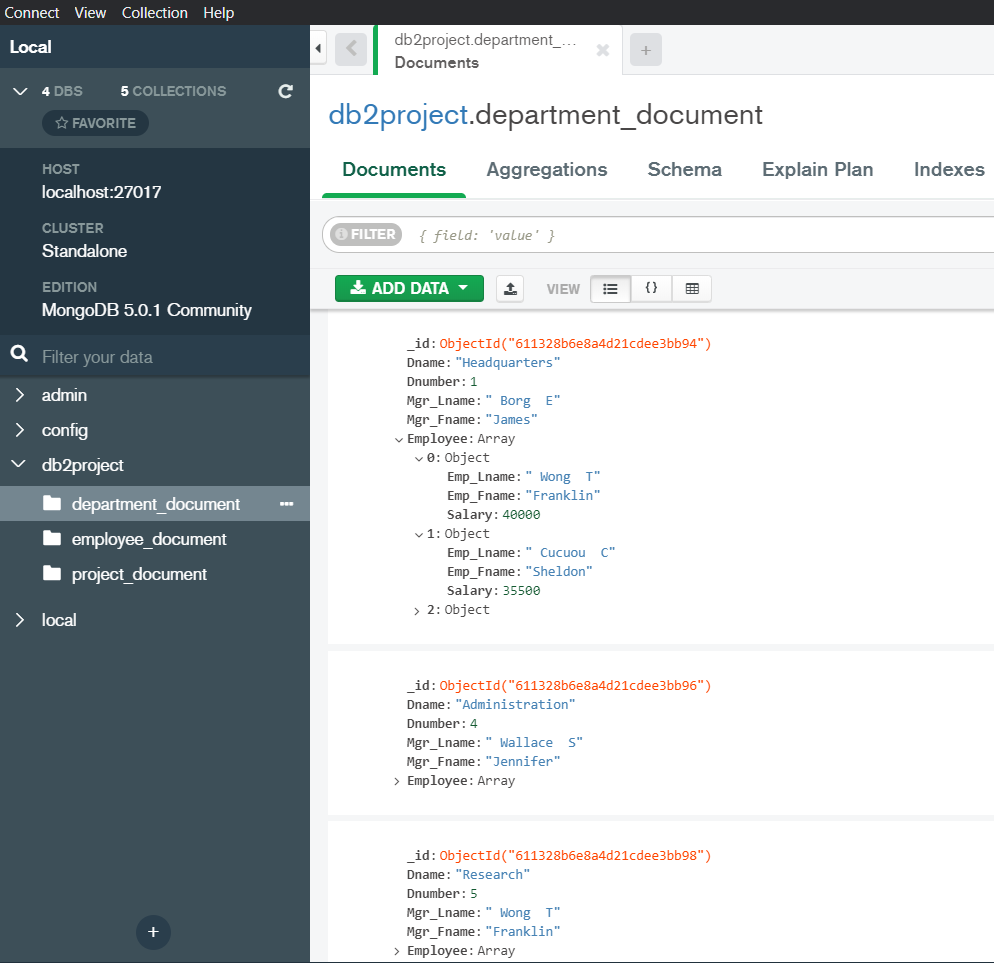
* Created project collection named ‘project\_document’



* Created employee collection named ‘employee\_document’



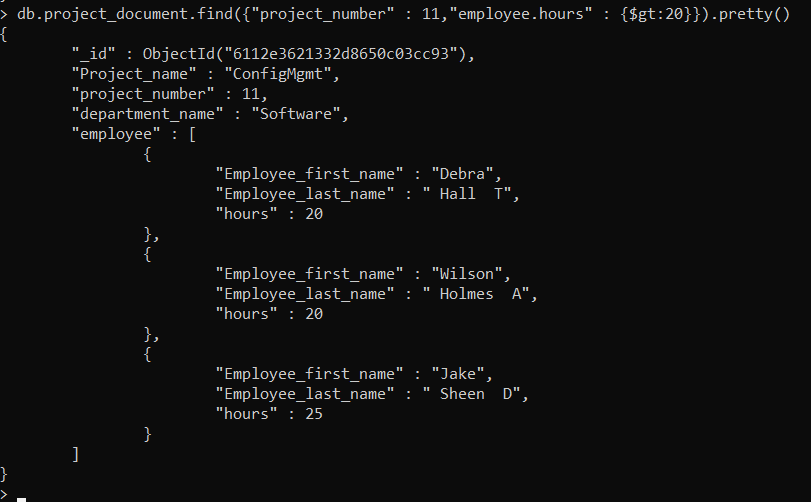
* Created collection named ‘department\_document’

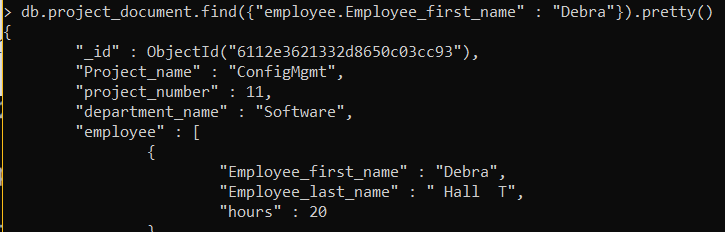


**Queries on project collection:**

1. Query whose project number is 11

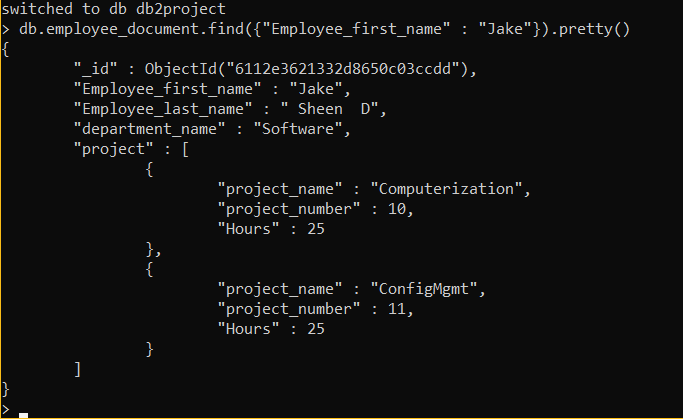


1. Query whose project number is 11 and employees in that department work greater than 20 hours.
2. Query whose Employee name in the project collection is Debra.



**Queries on Employee collection:**

1. Query whose First name is “Jake”.



1. Query whose department name is “Software” and project number in that department is less than 12

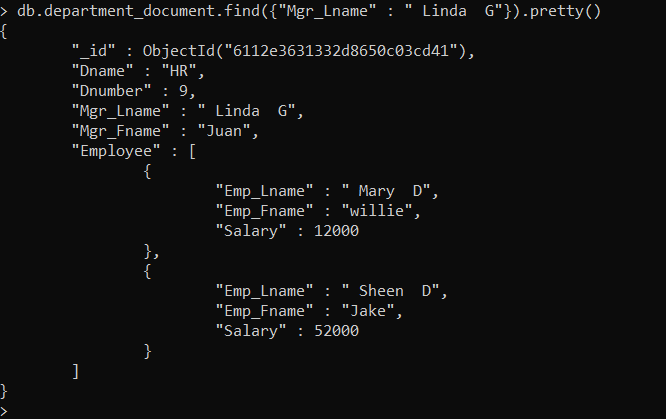


1. Query whose department name is software and employees in that department work greater than 40 hours.

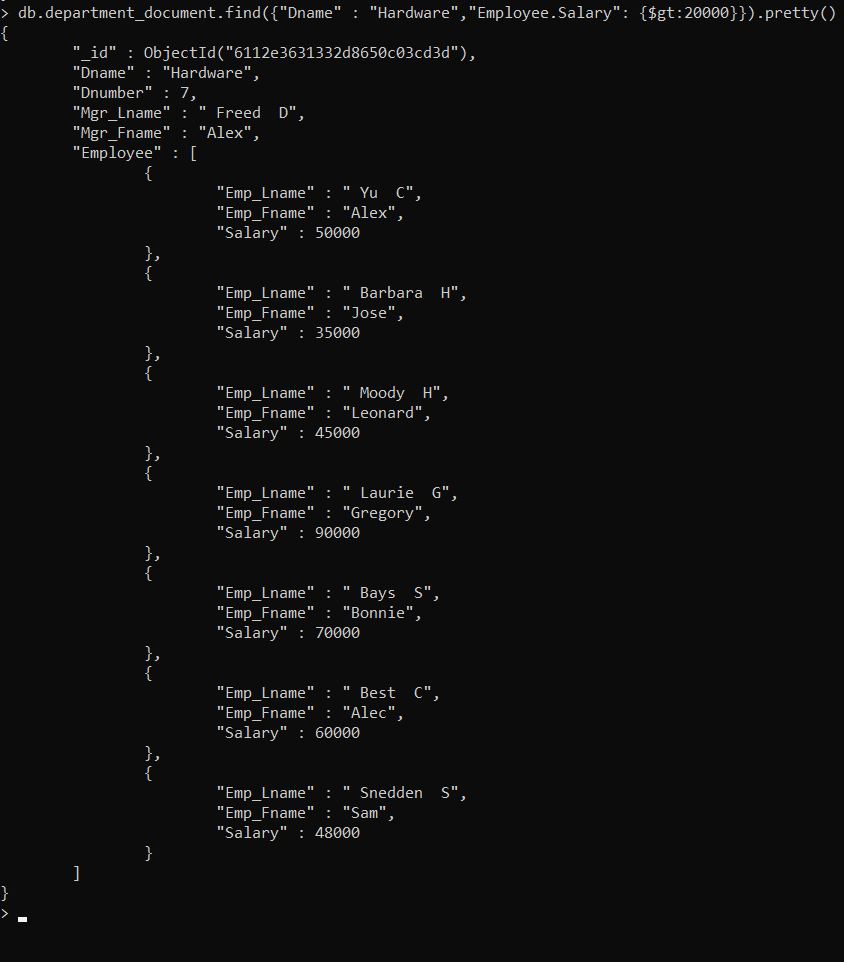


**Queries on Department collection:**

1. Query whose Manager last name is “Linda G”



2. Query whose department name is “hardware” and employee salary is greater that 200000



**Contribution by Each Team Member:**

Saahil anande worked on sql joins and mongo import

Unnathi reddy worked on nested Json and xml file structure and sql import.