PROJECT 2 – Final Report

Pankaj Walke – 1001781008
Shivam Kumar Sareen – 1001751987
SUMMER 2020 **CSE5330**

Note: ***Entire logic implementation in code is done by both the team members. Both members worked on it together.***

This Code will be implementing ***rigorous 2 phase locked wound-wait concurrency control protocol***. In which, ***wound (Abort)*** or ***wait (Block)*** will be decided based on timestamp of transactionIDs reaching the deadlock.

Programming Language: PYTHON 3
Plugins: mysql-connector,python, pymongo, xml

Database Used: MySQL & MongoDB

Execution Steps:

1. Launch **PyCharm** or any Python IDE
2. Import ***Project2_MongoDB.zip*** file from import options under File tab.
3. Build and Run ***MongoDB-Project2.py/ MongoDB-Project2-Extra Credit 1.py/ MongoDB-Project2-Extra Credit 2.py***
4. Output –
    a. ***(For main Project and Extra Credit 2)*** Tables will be generated in MySQL and Collections will be created in MongoDB
    b. ***(For Extra Credit 1)*** XML file will be created in project directory

Data Structure:
We will JSON format to insert documents in MongoDB. JSON is a Key: Value document

JSON object Construction for **PROJECTS COLLECTION:**

```
//PROJECT COLLECTION
{[
//INDIVIDUAL DOCUMENT
{
PNAME: //Project Name
PNO: //Project Number
DName: //Department name of Project
Employees: // List of Employees working on Project
[
{//Individual Employee Data
ELNAME: //Employee Last Name
EFNAME: //Employee First Name
HOURS: //No. of Hours employee worked
}
{
//employee 2
}
{//employee 3
}]}]}
```

PROJECT 2 – Final Report

Pankaj Walke – 1001781008
Shivam Kumar Sareen – 1001751987
SUMMER 2020 **CSE5330**

JSON object Construction for **EMPLOYEES COLLECTION:**

```
//EMPLOYEES COLLECTION
{[
//INDIVIDUAL DOCUMENT
{
ELNAME: //Employee Last Name
EFNAME: //Employee First Name
DName: //Department name of Employee
PROJECTS: // List of Projects employee is working on
[
{//Individual Project Data
PNAME: //Project Name
PNO:  //Project Number
HOURS: //No. of Hours employee worked
}
{
//project 2
}
{//project 3
}]}]}
```

JSON object Construction for **DEPARTMENT COLLECTION:**

```
//DEPARTMENTS COLLECTION
{[
//INDIVIDUAL DOCUMENT
{
DNAME: //Department Name
DNo: //Department Number
ManageSSN: //SSN of Manager
ManagerStartDate: //Start Date of Manager
Employees: // List of Employees under this Department
[
{//Individual Employee Data
ELNAME: //Employee Last Name
EFNAME: //Employee First Name
HOURS: //No. of Hours employee worked
}
{
//employee 2
}
{//employee 3
}]}]}
```

Code implementation:

STEP 1: Read Data from given **.csv** files and insert it into MySQL database

STEP 2: Query MySQL database created in STEP 1 and fetch resulting tuples in a variable result

```
projectsQuery = "Select PName, PNo, DName from works_on, company.project,
department where project.PDeptNo = department.DNum Group By PName"
mysqlCursor.execute(projectsQuery)

result = mysqlCursor.fetchall()
```

STEP 3: Iterate result for each element and process data and insert into Projects Collection as follows:

```
for x in result:

        //Execute another query to fetch employees working on this Project

        employeeQuery = "Select ELname, EFname, Hours FROM employee, works_on
         Where employee.ESSN = works_on.EmpSSN AND works_on.PNum = " + x[1])
        mysqlCursor.execute(employeeQuery)

        result1 = mysqlCursor.fetchall()
        for y in result 1:

         //Create the dictionary object to insert in Employees List

        employee = { ELname: y[0],EFname: y[1], Hours: y[2]
        empList.append(employee)

//Insert data into projects collection as per structure shown at start

mongodb.insert_many([{
         "PName": x[0],
         "PNo": x[1],
         "DName": x[2],
         "Employees" : emplist
         }]
```

STEP 4: Repeat STEP 3 to create EMPLOYEES and DEPARTMENTS collections
STEP 5: Create XML for PROJECTS and EMPLOYEES using same logic as written in STEP 3 but instead of creating **JSON** object, Create **XML** object and write it to **.xml** file

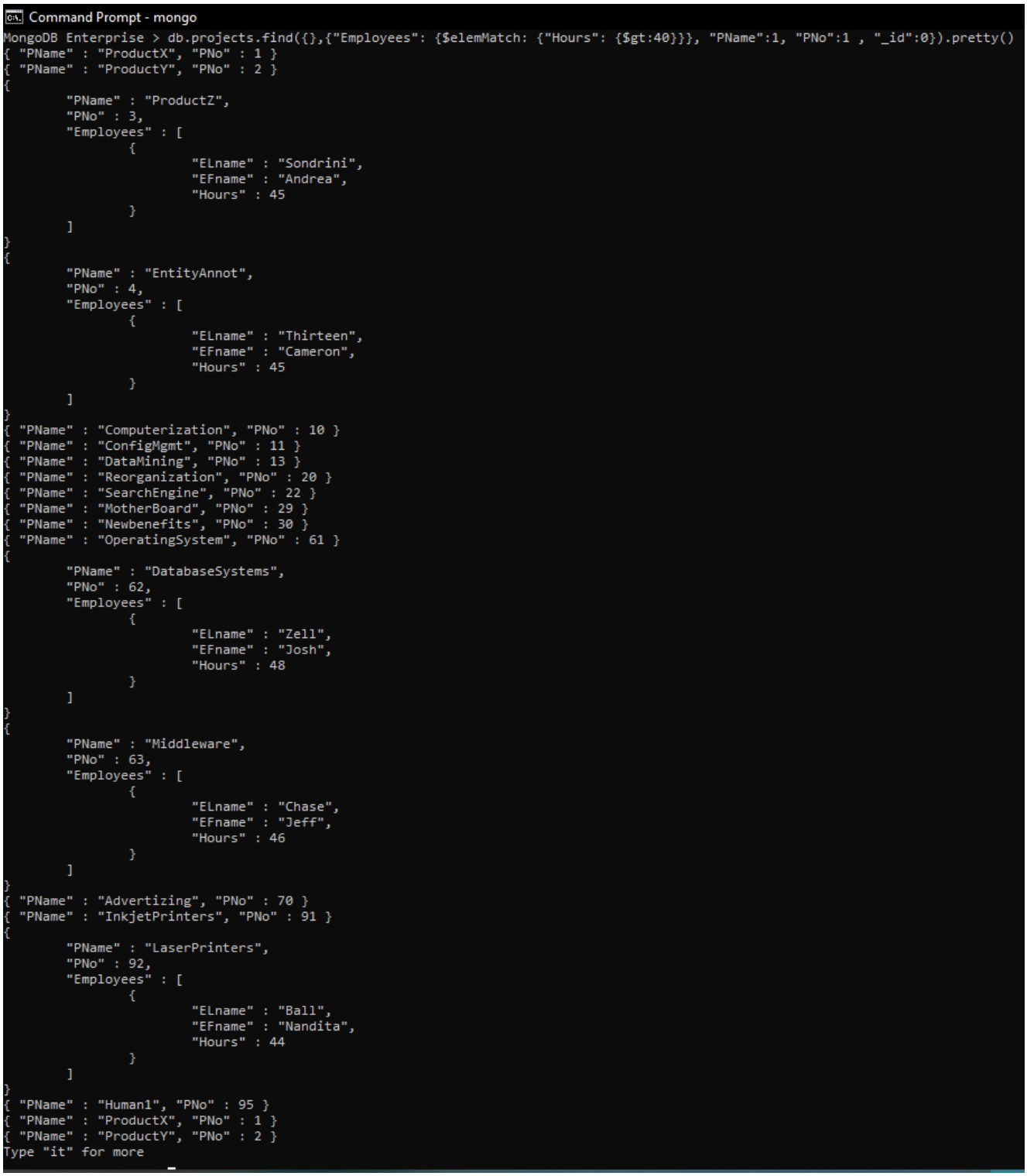
<u>MongoDB Queries:</u>
Following queries can be executed in MongoShell after running the Project Code.

1. **To Find all the Projects where at least one employee has worked for more than 40 hours**

   Query : db.projects.find({},{"Employees": {$elemMatch: {"Hours": {$gt:40}}}, "PName":1, "PNo":1 , "_id":0}).pretty()

   Output:

Pankaj Walke – 1001781008
Shivam Kumar Sareen – 1001751987
SUMMER 2020 **CSE5330**



2. Find details of All Employees working in Research Department

Query : db.employees.find({"DName":"Research"},{"_id": 0,"ELname":1, "EFname":1}).pretty()

Output:

```
Command Prompt - mongo
MongoDB Enterprise > db.employees.find({"DName":"Research"},{"_id": 0,"ELname":1, "EFname":1}).pretty()
{ "ELname" : "Smith", "EFname" : "John" }
{ "ELname" : "Wong", "EFname" : "Franklin" }
{ "ELname" : "English", "EFname" : "Joyce" }
{ "ELname" : "Sondrini", "EFname" : "Andrea" }
{ "ELname" : "Morgan", "EFname" : "Michael" }
{ "ELname" : "Narayan", "EFname" : "Ramesh" }
{ "ELname" : "Miller", "EFname" : "James" }
MongoDB Enterprise >
```

3. Print No. of Employees Working on Each Project

Query : db.projects.aggregate({ $project:{"_id": 0,"PName":1, "PNo":1, workerscount:{$size:"$Employees" }}}).pretty()

Output:

```
Command Prompt - mongo
MongoDB Enterprise > db.projects.aggregate({ $project:{"_id": 0,"PName":1, "PNo":1, workerscount:{$size:"$Employees" }}}).pretty()
{ "PName" : "ProductX", "PNo" : 1, "workerscount" : 4 }
{ "PName" : "ProductY", "PNo" : 2, "workerscount" : 4 }
{ "PName" : "ProductZ", "PNo" : 3, "workerscount" : 5 }
{ "PName" : "EntityAnnot", "PNo" : 4, "workerscount" : 3 }
{ "PName" : "Computerization", "PNo" : 10, "workerscount" : 6 }
{ "PName" : "ConfigMgmt", "PNo" : 11, "workerscount" : 3 }
{ "PName" : "DataMining", "PNo" : 13, "workerscount" : 1 }
{ "PName" : "Reorganization", "PNo" : 20, "workerscount" : 5 }
{ "PName" : "SearchEngine", "PNo" : 22, "workerscount" : 2 }
{ "PName" : "MotherBoard", "PNo" : 29, "workerscount" : 3 }
{ "PName" : "Newbenefits", "PNo" : 30, "workerscount" : 5 }
{ "PName" : "OperatingSystem", "PNo" : 61, "workerscount" : 11 }
{ "PName" : "DatabaseSystems", "PNo" : 62, "workerscount" : 11 }
{ "PName" : "Middleware", "PNo" : 63, "workerscount" : 9 }
{ "PName" : "Advertizing", "PNo" : 70, "workerscount" : 4 }
{ "PName" : "InkjetPrinters", "PNo" : 91, "workerscount" : 10 }
{ "PName" : "LaserPrinters", "PNo" : 92, "workerscount" : 5 }
{ "PName" : "Human1", "PNo" : 95, "workerscount" : 4 }
MongoDB Enterprise >
```

4. Find Employees working on at least 4 projects, and Print Project names

Query : db.employees.find({$expr:{$gte:[{$size:"$Projects"},4]}},{"_id":0, "ELname":1, "EFname":1,"DName":1,"Projects.PName":1}).pretty()

Output:

PROJECT 2 – Final Report

Pankaj Walke – 1001781008
Shivam Kumar Sareen – 1001751987
SUMMER 2020 **CSE5330**