DB2 5331

Intermediate Design Report for Project2

**Team Members**

Neena Gupta - 1001892452

Begum Fatima Zohra -1001880881

**Participation of team Members**

We collaborated on the project. Discussing the best way to implement the problem statement.

**About The Project:**

It is a Menu-driven program to extract data from relational database and load three different documents into the MongoDB

**Data Structures used**

1. List

2. HashMap

**Relational Database**

MySQL

**GUI**

MySQL Workbench for MYSQL

MongoDB Compass for MongoDB

**Programming language**

Java

**File structure:**

1.For creating PROJECT object (document) that will include each employee

Project.java

EmployeeList.java

2.For creating Employee object (document) that will include project each one worked on

Employee.java

ProjectList.java

3.For creating Department object (document) that will include collection of the employees who work for that department

Department.java

DepartmentList.java

**Libraries used for:**

MySQL connector

MongoJavaDriver

GSON

JSON

**Procedure used: Option 1:**Loaded the data in a relational DBMS such as MySQL. Then write three different queries to extract the data needed for each document collection in relational format, and then convert the query result from the relational format into JSON for loading on MongoDB.Created the nested structure using GSON.

**Instructions To Run:**

1. Open a command prompt window and go to the directory where you saved the java program (ProjectMain.java)
2. Type javac ProjectMain.java.
3. Now, type ' java ProjectMain.java' to run your program.
4. You will see the result printed on the window.

**pseudo code**:

1.connect to MySQL and get the data needed using statement.

2. Use a switch case to give options of creating one from project, Employees and department document

switch(choice){

case 1: System.out.println("I am in case 1");

//method call for case 1

createProject(statement);

break;

case 2: System.out.println("I am in case 2");

//method call for case 2

createEmployees(statement);

break;

case 3: System.out.println("I am in case 3");

//method call for case 3

createDepartment(statement);

break;

case 4: System.out.println("Exiting the application");

System.exit(0);

default: System.out.println("Incorrect input!!! Please re-enter choice from the menu");

}

3. When particular switch case is called say createProject(statement)

Result set of SQL query is stored in Hash Maps using while looping

HashMap<String, Project> makeJson

= new HashMap<String, Project>();

while(resultSet.next()) {

String pname = resultSet.getString("PNAME");

if (makeJson.containsKey(pname)){

Project projectName = makeJson.get(resultSet.getString("PNAME"));

EmployeeList employeeList2 = new EmployeeList(resultSet.getString("EMP\_FNAME"), resultSet.getString("EMP\_LNAME"), resultSet.getString("HOURS"));

projectName.getEmployee().add(employeeList2);

}

else {

Project project = new Project();

List<EmployeeList> employeeList = new ArrayList<>();

EmployeeList employeeList1 = new EmployeeList(resultSet.getString("EMP\_FNAME"), resultSet.getString("EMP\_LNAME"), resultSet.getString("HOURS"));

project.setPname(resultSet.getString("PNAME"));

project.setPnumber(resultSet.getString("PNUMBER"));

project.setDname(resultSet.getString("DNAME"));

employeeList.add(employeeList1);

project.setEmployee(employeeList);

makeJson.put(pname,project);

}

}

4. Convert HashMap into JSON using GSON

List<String> jsonList = new ArrayList<String>();

for (Map.Entry entry : makeJson.entrySet()) {

GsonBuilder gsonBuilder = new GsonBuilder();

Gson gson = gsonBuilder.create();

String JSONObject = gson.toJson(entry.getValue());

jsonList.add(JSONObject);

log(JSONObject);

}

System.out.println(jsonList);

5.Connect to MongoDB. Create a projects document and insert projects document into MongoDB as collections.

MongoCollection<Document> collection = database.getCollection("PROJECTS");

List<Document> jsonDocument = new ArrayList<Document>();

for (String object : jsonList) {

Document jsnObject = Document.parse(object);

jsonDocument.add(jsnObject);

}

collection.insertMany(jsonDocument);

6. Close the MongoDB and MYSQL connection.

7. Same way createEmployees(statement) and createDepartment(statement) is created.