

Enhancing Customer Service Excellence: A Comprehensive Incident Management Solution

Milestone: NoSQL Implementation

Group 2

Aniruthan SA Aravind Anbazhagan

617-238-8857 (Tel of Aniruthan) 629-899-0339 (Tel of Aravind)

[swaminathanarulmur.a@northeastern.edu](mailto:swaminathanarulmur.a@northeastern.edu) [anbazhagan.a@northeastern.edu](mailto:anbazhagan.a@northeastern.edu)

Percentage oS Effort Contributed by Studentl: 50%

# Percentage of Effort Contributed by Student2: 50%

Signature of Student 1: Aniruthan SA

# Signature of Student 2: Aravind Anbazhagan

Submission Date: 12/03/2023

**SQL database tables to CSV files:**

The “dma\_project” database tables were converted into csv files using the below python script.

import pymysql  
import pandas as pd  
  
# Database connection details  
host = 'localhost' # e.g., 'localhost'  
user = 'root'  
password = 'admin123'  
database = 'dma\_project'  
  
# Establishing a database connection  
connection = pymysql.connect(host=host, user=user, password=password, database=database)  
  
try:  
 # Fetching the list of tables  
 with connection.cursor() as cursor:  
 cursor.execute("SHOW TABLES")  
 tables = cursor.fetchall()  
  
 # Export each table to a CSV file  
 for table\_name in tables:  
 query = f"SELECT \* FROM {table\_name[0]}"  
 df = pd.read\_sql(query, connection)  
 df.to\_csv(f"{table\_name[0]}.csv", index=False)  
  
finally:  
 # Closing the database connection  
 connection.close()

**NoSQL implementation:**

* Downloaded and Installed MongoDB compass
* Created a new database dma\_project with three collections namely Resolution, Team\_PDL and Tickets.

A screenshot of a computer

Description automatically generated

* Imported data for the three collections using the csv files generated from SQL database tables using the python script.
* After importing the data, three queries were implemented using aggregation.

When working with the aggregation framework in MongoDB Compass, you can build your pipeline using the graphical user interface. Below are examples of what each of these queries would look like as stages in an aggregation pipeline.

* A Simple Query (as an Aggregation)

To find all documents where `case\_status` is "new":

\*\*Stage 1: `$match`\*\*

```json

{ "case\_status": "new" }

```

* A More Complex Query (as an Aggregation)

To find all documents where `case\_status` is "new" and either `issue\_type` is "wired" or `resolution\_code` is "reboot":

\*\*Stage 1: `$match`\*\*

```json

{

"$and": [

{ "case\_status": "new" },

{

"$and": [

{ "issue\_type": "wired" },

{ "resolution\_code": "reboot" }

]

}

]

}

```

* An Aggregate Query

To group documents by `issue\_type` and count them, only including documents where `case\_status` is "new":

\*\*Stage 1: `$match`\*\*

```json

{ "case\_status": "new" }

```

\*\*Stage 2: `$group`\*\*

```json

{

"\_id": "$issue\_type",

"count": { "$sum": 1 }

}

```

\*\*Stage 3: `$sort`\*\*

```json

{ "count": -1

```

* Saved the three pipelines as shown in the below image.

A screenshot of a computer

Description automatically generated

* Outputs of the queries:

1. **Aggregation:**

A screenshot of a computer

Description automatically generated

1. **Simple:**

{

"\_id": {

"$oid": "656d306774f6a1b94b693723"

},

"ticket\_number": "T000000001",

"case\_status": "new",

"short\_desc": "Short Desc 1",

"case\_desc": "Case Desc 1",

"issue\_type": "wired",

"assigned\_to": "Assignee1",

"resolution\_code": "reboot",

"resolution\_notes": "Note for reboot"

}

{

"\_id": {

"$oid": "656d306774f6a1b94b693724"

},

"ticket\_number": "T000000002",

"case\_status": "new",

"short\_desc": "Short Desc 2",

"case\_desc": "Case Desc 2",

"issue\_type": "wired",

"assigned\_to": "Assignee2",

"resolution\_code": "replacement",

"resolution\_notes": "Note for replacement"

}

Etc…

1. **Complex:**

{

"\_id": {

"$oid": "656d306774f6a1b94b693723"

},

"ticket\_number": "T000000001",

"case\_status": "new",

"short\_desc": "Short Desc 1",

"case\_desc": "Case Desc 1",

"issue\_type": "wired",

"assigned\_to": "Assignee1",

"resolution\_code": "reboot",

"resolution\_notes": "Note for reboot"

}

{

"\_id": {

"$oid": "656d306774f6a1b94b693729"

},

"ticket\_number": "T000000007",

"case\_status": "new",

"short\_desc": "Short Desc 7",

"case\_desc": "Case Desc 7",

"issue\_type": "wired",

"assigned\_to": "Assignee7",

"resolution\_code": "reboot",

"resolution\_notes": "Note for reboot"

}

Etc…