Avi Jain, et. al. International Journal of Engineering Research and Applications
www.ijera.com
ISSN: 2248-9622, Vol. 10, Issue 8, (Series-V) August 2020, pp. 46-53

RESEARCH ARTICLE OPEN ACCESS

Benchmarking of Hybrid Model for Food Delivery Web Application Development

Avi Jain[1], Smit Shah[2], Prayag Patel[3], Akshat Shah[4], Sneha Rohra[5]
[1,2] *(Department of Computer Science, K. J. Somaiya College of Engineering, Mumbai-400047*
[3] *(Department of Electronics and Telecommunication, Thadomal Shahani Engineering College, Mumbai-400050)*
[4,5] *(Department of Information Technology, Thadomal Shahani Engineering College, Mumbai-400050)*

ABSTRACT
Database system comprises a collection of programs running on computer and thus helping the user to store, manage and protect information. Relational database is widely used in the industry for more than 30 years. It is common for a company to use the relational database for the database in their system. The commonly used Relational Database Management Systems are MySQL, Oracle, Microsoft SQL server and PostgreSQL. Each RDBMS is well known as it's in practice and each has their own superiority. Experts in particular database systems are also available to work on.MySQL is known for its faster query execution than the other relational databases. Even though PostgreSQL have a relatively slower performance than MySQL, PostgreSQL has more features that can help users in managing data with ease.With large amounts of increasing datasets it becomes absolutely necessary to manage and process all the data with the help of these database systems. Therefore we propose a hybrid database system of MySQL, PostgreSQL and MongoDB to improve the performance of a web applicationfor 'Food Delivery System'. The ability of the relational databasesin storing sensitive data complements the ability of NoSQL in processing high amounts of data. The proposed hybrid database system is implemented using MySQL and PostgreSQLfor relational databases and MongoDB for NoSQL databases

Keyword – Benchmarking, Hybrid Database, Hybrid Food Delivery, MongoDB, SQL

Date of Submission: 15-08-2020 Date of Acceptance: 01-09-2020

I. INTRODUCTION

Food Management System is an application which will help restaurants to optimize and control over their restaurants. For the management point of view, the manager will be able to control the restaurant by having all the reports to hand and able to see the records of each employee and orders. Our experimental setup helps to optimize the results by separating different fields in the Food Management System into either NoSQL database system or SQL database system.

Database system comprises a collection of programs running on computer and thus helping the user to store, manage and protect information. Relational database is widely used in the industry for more than 30 years. It is common for a company to use the relational database for the database in their system. The commonly used Relational Database Management Systems are MySQL, Oracle, Microsoft SQL server and PostgreSQL. Each RDBMS is well known as it's in practice and each has their own superiority. Experts in particular database systems are also available to work on.

MySQL is known for its faster query execution than the other relational databases. Even though PostgreSQL have a relatively slower performance than MySQL, PostgreSQL has more features that can help users in managing data with ease.

With large amounts of increasingdatasets, it becomes absolutely necessary to manage and process all the data with the help of these database systems. Therefore, we propose a hybrid database system of MySQL, PostgreSQL and MongoDB to improve the performance of a web applicationfor 'Food Delivery System'. The ability of the relational databasesin storing sensitive data complements the ability of NoSQL in processing high amounts of data. The proposed hybrid database system is implemented using MySQL and PostgreSQLfor relational database and MongoDB for NoSQL database.

II. LITERATURE REVIEW

Hybrid database model is a database system that uses combination of two or more differentdatabase models in a system. This model is preferred because at times the data stored is not

alwayssuitable to be processed using single database model.Our research started by referring various papers about similar topics and briefing papers closelyrelated to study of SQL and NoSQL databases and how MongoDB has an upper hand in efficientlyhandling increasing dataloads[2].Traditionally, applications have relied on relational databases suchas MySQL, Oracle, or Microsoft SQL Server, which offer improved storage efficiency and dataretrieval speeds compared to flat files [3], [4], [5]. However, as the rate of data generation andacquisition has increased, novel approaches for data storage have emerged [6], [7]. In particular,document-oriented and other NoSQL databases have rapidly gained adoption in the commercialsector [7].Lokesh Kumar et al[8] performed analysis and comparison of MongoDB with MySQLdatabase by mainly focusing on efficient handling of increasing unstructured data and replacementof traditional MySQL databases with MongoDB to compare the two databases in order to justify thepertinence of MongoDB over MySQL. Cook[9] tried to combine two databases that have differentmodels: object oriented database and relational database. From the experiments, Cook founds somebenefits in using multiple database models in a system in terms of Flexibility, Increased performance,Logical distribution which inturn enhance the experience on web usage by the user.Hybrid database is not a solution for every problem, but it is an option in making a database that canstore normalized data and can run aggregated query with good performance [10]. Most of theEnterprise Resource Planning solutions have a stronghold for relational databases, but unfortunatelythey are lacking the flexibility where users can customize entry forms, without performing updationon the database schema. By introducing a NoSQL document database into the system, users cancreate as well as edit forms quickly, as required. The data in this case will be stored as documents.Some relational database vendors like Microsoft SQL Server 2016 have recognized the need for sucha blended solution, and implemented something similar to a document database inside of theirrelational database for storing JSON documents inside cells, which eases up workflow. [11]

III. METHODOLOGY

The study of the different databases is done and different datasets of varying amounts of data are inserted into all the tables of all the Database Systems (MongoDB, PostgreSQL, MySQL).Then the data is extracted using the CodeIgnitor framework in Windows 10 Operating system. This simple web page showcases the performance of all three database systems.18 tables are created in all the three database systems and 5 different datasetscontaining tuple values are inserted and the results relating to time and memory consumed are studied for all the three databases. All of the dataset is created using a Python Script and is inserted into the database using the 'Psycopg2' library.CSV files can't be directly inserted into all the databases and henceforth the needto write code was needed.Codeigniter is set up using PHP and Xampp.

1. Start
2. Connect to database
3. Open a new csv in write mode
4. For range n:
    a. Insert random number into database
    b. Insert random names
    c. Insert random dates
5. Check csv
6. Commit to database
7. Repeat these steps for different databases
8. Stop

(Where n= 1k, 5k, 10k, 25k, 50k tuples)

(Random quantities are inserted in each table according to attributes)

Fig 1. Algorithm

IV. TABLES AND FIGURES

Read and Write Performance:

TABLE COUNT - 18
First Dataset 1,000 data tuples
Second Dataset 5,000 data tuples
Third Dataset 10,000 data tuples
Fourth Dataset 25,000 data tuples
Fifth Dataset 50,000 data tuples

The list of the tables and their structures are described below

Table 1 Users

Field Name	Data Type	Size
Name	Varchar	20
Password	Varchar	32
idEmployee	Int	
idQuestion	Int	

Answer	Varchar	100

Table 2 Cancellation

Field Name	Data Type
idCancellation	INT
idOrder	INT

Table 3 Category

Field Name	Data Type	Size
idCategory	INT	
Name	Varchar	32
idParent	INT	

Table 4 Customer

Field Name	Data Type	Size
idCustomer	INT	
Name	Varchar	150
Bdate	Date	
Mobile_no	Double	
Email_id	Varchar	45

Table 5 Designation

Field Name	Data Type	Size
idDesignation	INT	
Name	Varchar	45
Salary	INT	

Table 6 Employee

Field Name	Data Type	Size
idEmployee	Varchar	150
Name	Varchar	400
Address	Varchar	30
City	Varchar	30
State	Varchar	
Mobile_no	Double	
idDesignation	INT	

Table 7 Feedback

Field Name	Data Type	Size
idFeed_back	INT	
Suggestion	Varchar	200
idEmployee	INT	
idCompany	INT	

Table 8 Food Items

Field Name	Data Type	Size
idFood_item	INT	
Name	Varchar	45
Price	INT	
idCategory	INT	

Table 9 Ingredients

Field Name	Data Type
idFood_Item	INT
idVegetable	INT
Quantity	INT

Table 10 Order Type

Field Name	Data Type	Size
idOrder_type	INT	
Name	Varchar	45

Table 11 Order

Field Name	Data Type
idOrder	INT
idTable	INT
idCustomer	INT
idEmployee	INT
idStatus	INT
idOrder_type	INT
Date	Date

Table 12 Order Details

Field Name	Data Type	Size
idSub_Order	INT	
idOrder	INT	
idFood_item	INT	

Special	Varchar	45

Table 13 Shift Timing

Field Name	Data Type
idShift_time	INT
idEmployee	INT
Start_Time	Datetime
End_Time	Datetime

Table 14 Tables

Field Name	Data Type	Size
idTable	INT	
idEmployee	Varchar	45

Table 15 Table booking

Field Name	Data Type
idTable_booking	INT
idTable	INT
idCustomer	INT
idStatus	INT
Date	Date

Table 16 Questions

Field Name	Data Type	Size
idQuestion	INT	
Questions	Varchar	100

All of these tables were stored in all the three databases and for all the 5 datasets(1.000 - 50,000) results were recorded.

Food_Items and Order-Type were kept constant throughout all the datasets since it isn't likely to increase after a particular limit. The limit was set to 500 and 15 respectively.

V. RESULT AND DISCUSSION

Here are the graphs of some important tables and their results and efficiency analysis.The X-axis symbolizes the datasets, namely 1,000 to 50,000 respectively.The Y-axis symbolizes the time taken to retrieve them in seconds.

Users table comes in the SQL category but since it has so many tuples a larger retrieval is done betterby MongoDB but MySQL also proves to be efficient herewhich can beobserved in Fig 2.

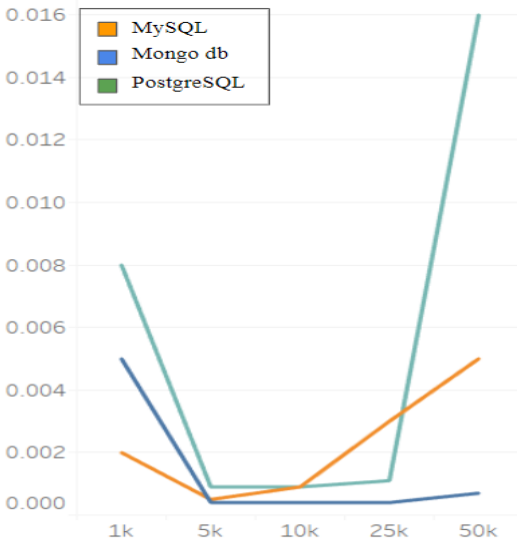

Fig 2. Users

Cancellationtable is required for any cancellation of the orders and in Fig.3 we see a spike in the beginning for MongoDB but in later stages it proves to be better thanboth PostgreSQL and MySQL as both of them show gradual increase.

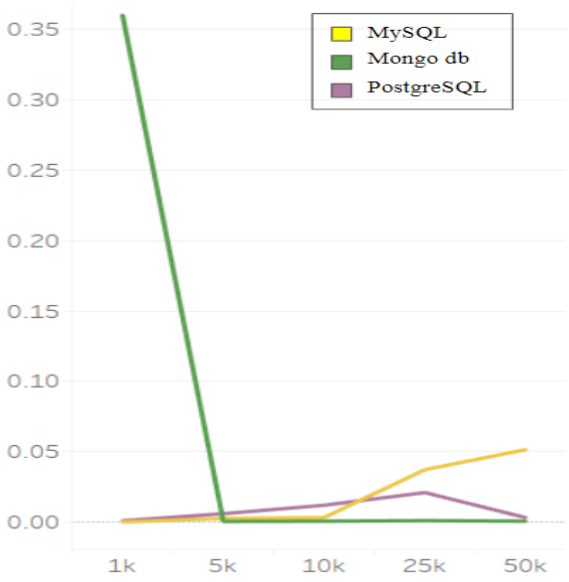

Fig 3. Cancellation

Fig. 4 represents the graph of category table and the best results can be seen for PostgreSQL.MySQL takes a big spike whereas MongoDB doesn't prove to be that efficient.

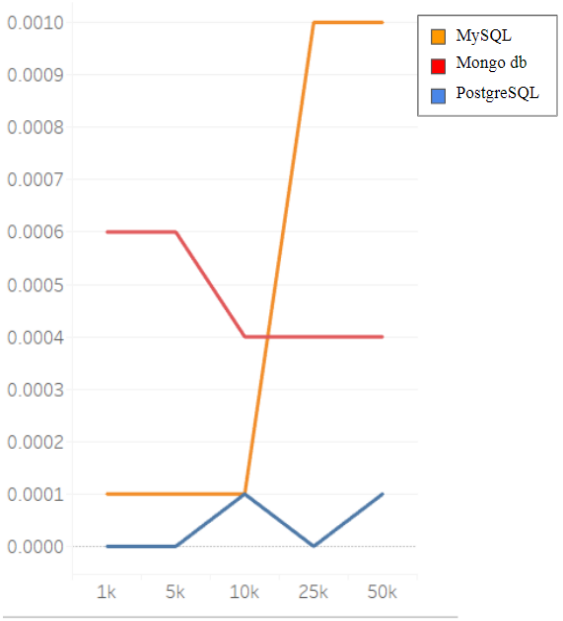

Fig 4. Category

PostgreSQL increases exponentially in the customers table but we see that MongoDB is the best suitable for this table be it any dataset.MySQL gives irregular results here.

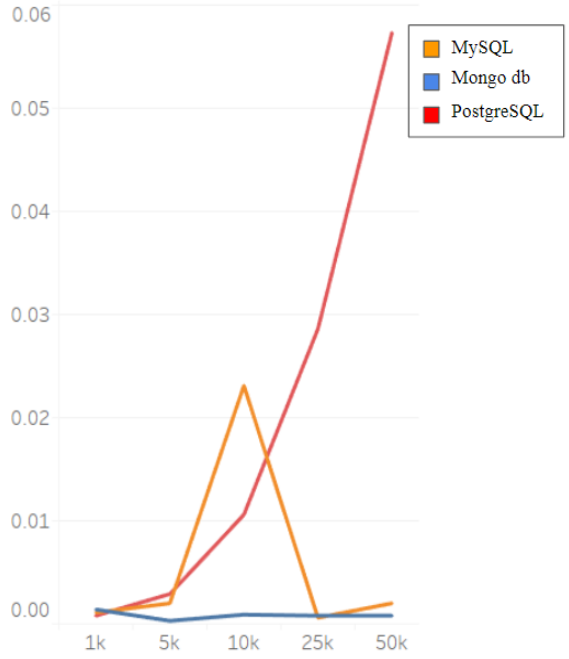

Fig 5. Customers

Employees as shown in fig.6 is SQL data and is best retrieved in MySQL.

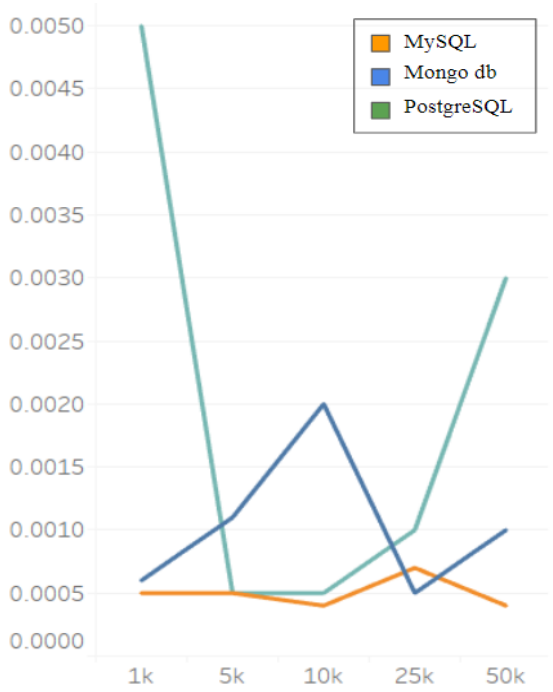

Fig 6. Employee

Feedback, Orders, Order_Details and Questions are all non-structural that we classify as NoSQL data and hence in all the graphs MongoDB proves to be the most efficient.

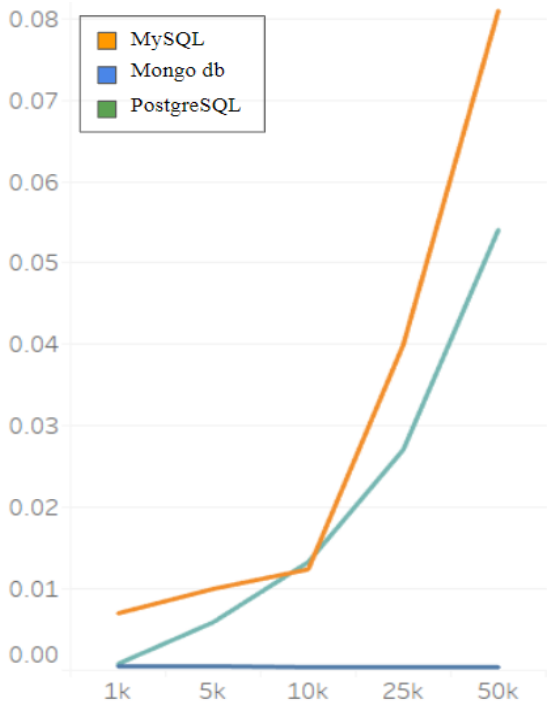

Fig 7. Feedback

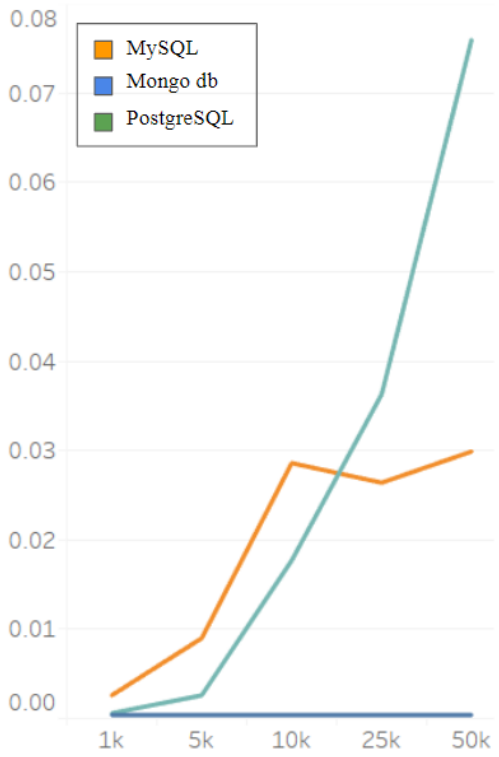

Fig 8. Orders

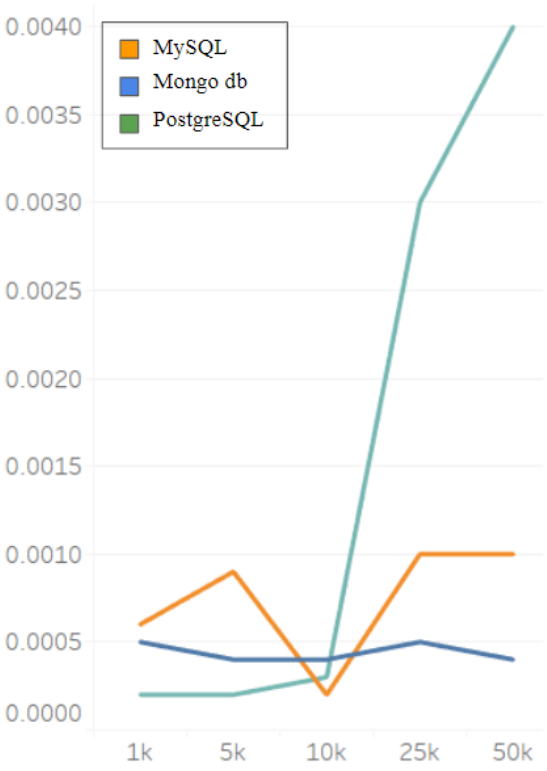

Fig 10. Tables

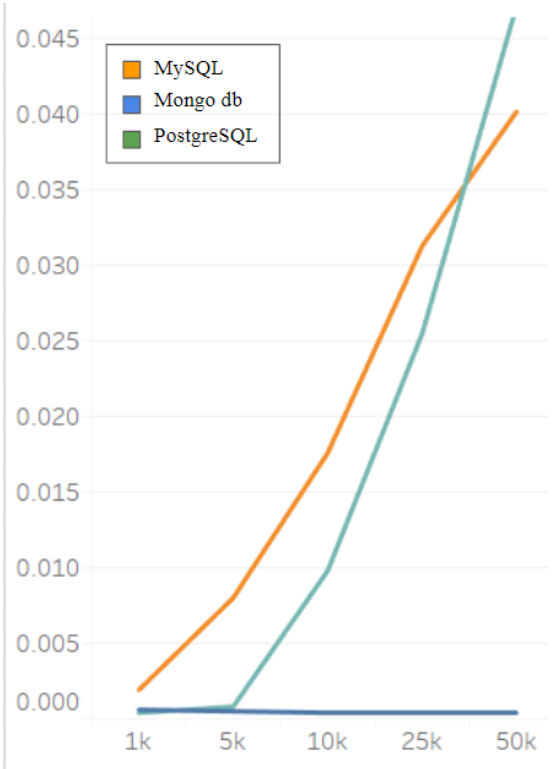

Fig 9. Order Details

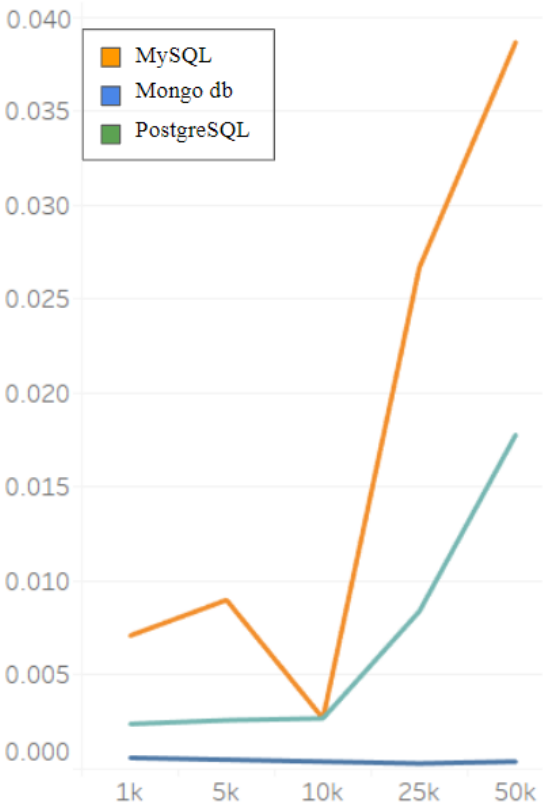

Fig 11. Questions

A hybrid database becomes important because nosql data retrieval is faster in MongoDB and large data retrieval of SQL can be done in MySQL. Tables like Feedback, Orders and Questions contain non-structural data which is fastest retrieved in MongoDB where as tables like Customers, Users, Employees are structural and can be retrieved in in MySQL or PostgreSQL. Many applications nowadays are using Hybrid Database System because it uses two or more Database Models in one system. This is preferred because not all the stored data is suitable for just one type of a Database model. Therefore we have used an Object Oriented Database, Relational Database and one NoSQL model. We discovered multiple benefits by using a hybrid system:

• **Flexibility.** In complex data types, the data may need to be forced to fit the storage model that is being used in the system. Hybrid database models can increase the flexibility of the system to store different types of data.

• **Increased performance.** Leveraging the strengths of multiple storage models improves the system performance.

• **Logical distribution.** Data Distribution is better and the number of forced data is reduced due to the flexibility of hybrid database models.

• **Built for the web.** The increased performance and the flexibility on storing the data can enhance the experience of web usage by the user.

A Hybrid Database model will not be suitable for every problem but a system where we need to store different types of normalized data and run aggregate queries with commendable performance, that's where our system shines the best.

In our application we see that for different datasets of users, PostGRES works best whereas for tables like no - fixed scheme MongoDB gives us the best results.

VI. CONCLUSION AND FUTURE WORK

The main aim of this paper is to give an overview of Hybrid database systems, about how it has declined the dominance of using just one type of a database system for an application, with its background and characteristics.

From the experimental results and analysis, it can be concluded that the hybrid database model of MySQL, PostgreSQL and MongoDB improves the web application performance on large database sizes. MongoDB has a relatively better write time than MySQL or PostgreSQL; but PostgreSQL write procedure is more consistent. In terms of disk space, CPU and RAM usage, the hybrid model database uses less disk space than MySQL or PostgreSQL. But it comes with a price of higher RAM requirements.

The current evaluations are performed on a single system; the repetition of this evaluation through experiments is suggested as on another operating system and different web frameworks the performance will definitely vary.

ACKNOWLEDGEMENTS

This research would not be possible without the help of Prof. Vaibhav Vasani, Prof. PradnyaBhangale ,SanketSahasrabudhe, Krunal Dabhi, Parth Shah and Virag Jain for helping with setup and guiding with research materials.

REFERENCES

Proceeding Papers and Journal Papers:

[1]. Gregorius Ongo, Gede Putra Kusuma,Hybrid Database System of MySQL and MongoDB in Web Application Development, 2018 International Conference on Information Management and Technology (ICIMTech)

[2]. Mayur M Patil , AkkamahadeviHanni,CHTejeshwar, Priyadarshini Patil , "A qualitative analysis of the performance of MongoDB vs MySQL Database based on insertion and retrieval operations using a web/android application to explore Load Balancing – Sharding in MongoDB and its advantages", International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC 2017)

[3]. E.F. Codd, A relational model of data for large shared data banks Commun. ACM, 13 (1970), pp. 377-387

[4]. M.M. Astrahan, M.W. Blasgen, D.D. Chamberlin, K.P. Eswaran, J.N. Gray, P.P. Griffiths, et al. System R: relational approach to database management, ACM Trans. Database Syst., 1 (1976), pp. 97-137

[5]. W.J. Kent, C.W. Sugnet, T.S. Furey, K.M.

[6]. Roskin, T.H. Pringle, A.M. Zahler, et al. The human genome browser at UCSC Genome Res., 12 (2002), pp. 996-1006

[7]. R. Cattell, Scalable SQL and NoSQL data stores, SIGMOD Rec., 39 (2011), pp. 12-27

[8]. N. Leavitt, Will NoSQL databases live up to their promise? Computer, 43 (2010), pp. 12-14

[9]. Lokesh Kumar, Computer Science & Engineering, Vivekananda Global University Jaipur, Rajasthan, India; Dr.ShaliniRajawat, Computer Science & Engineering, Vivekananda Global University Jaipur, Rajasthan, India; Krati Joshi, Computer Science & Engineering, Vivekananda Global

University Jaipur, Rajasthan, India – "Comparative analysis of NoSQL(MongoDB) with MySQL database"- IJMTER, ISSN-2393-8161.

[10]. R. Cook, "Is a hybrid database in your future?," SunWorld, February 1997. [Online]. Available:02-1997/swol-02-objects.html

[11]. [J. Madison, "Building a Hybrid Data Warehouse Model," Oracle, April 2007. [Online].

Avi Jain, et. al. "Benchmarking of Hybrid Model for Food Delivery Web Application Development." *International Journal of Engineering Research and Applications (IJERA)*, vol.10 (08), 2020, pp 46-53.