Database Management Systems—An Efficient, Effective, and Augmented Approach for Organizations

Anushka Sharma, Aman Karamchandani, Devam Dave, Arush Patel, and Nishant Doshi

Abstract Big and small firms, organizations, hospitals, schools, and other commercial offices are generating moderate to huge amounts of data regularly and need to constantly update and manage these data. These data are not only used at that instance, but generally, the retrospective analysis of data helps tremendously to improve the business strategies and the marketing trends. With time, these data may grow and become unmanageable if handled conventionally, like the file system. These factors resulted in the introduction of the terms database and database management system. Hierarchical, network, relational, and object-oriented approaches of DBMS are discussed in this paper. A highlight of the new-generation database approach called NoSQL is also included in this paper along with an insight into augmented data management. A model based on the database design for the Study in India Program is discussed. It is followed by a graphical user interface developed in Java for the same which ensures the ease of access to the database.

1 Introduction

A database is a collection set of interrelated data that can be inserted, retrieved, and deleted from it. A database management system (DBMS) is an application that stores and manages the data. Some of the popular relational DBMS like MySQL, Oracle, PostgreSQL, and SQLite have been discussed in the paper. A DBMS can be used for personal as well as commercial use. For example, a university can have its own IT admin application, while the traffic control system, supermarkets, or telecommunication sector have applications that are used at a huge level. In the following paper, we have talked about the variety of ways in which a DBMS can be used in a corporate setting, as well as the plethora of significant advantages they provide over the traditional file system. The aim of this paper is to enlighten the readers on why DBMS is the way going forward, as well as giving a brief

A. Sharma · A. Karamchandani · D. Dave · A. Patel · N. Doshi (✉)
Department of Computer Science and Engineering, School of Technology, Pandit Deendayal Energy University, Gandhinagar, India
e-mail: Nishant.Doshi@sot.pdpu.ac.in

T. Senjyu et al. (eds.), *ICT with Intelligent Applications*, Smart Innovation, Systems and Technologies 248, https://doi.org/10.1007/978-981-16-4177-0_47

explanation to the future trends in addition to the future technologies related to database management systems.

Outline: The following is the structure of the paper. Section 2 gives us the literature survey, which tells us about the history of DBMS briefly. Section 3 showcases the various advantages database management systems have to offer over a typical file system. Various types of databases are highlighted in Sect. 4. Section 5 has highlighted some of the most used database software available in the market at present. Section 6 has provided some useful insights regarding the future trends of DBMS, and the various software are mentioned in Sect. 5. Section 7 gives a brief explanation of augmented data management, an up-and-coming technology closely ingrained to DBMS. In Sect. 8, a database that we designed for the Study in India Program is presented; with Sect. 9 showcasing the graphical user interface (GUI), we designed to go along with it. Finally, Sect. 10 discusses the entire paper and concludes the work.

2 Literature Survey

Before the evolution of the database, every information is used to be recorded on papers. The evolution of DBMS started from somewhere in the mid of the twentieth century by Charles Bachman who designed the first DBMS system. Further, in 1970, Edgar Codd, who was working with IBM's information management system (IMS), wanted to get a better search engine for his research. This resulted in the introduction of relational databases, popularly based on 12 Codd's rules on his name. In 1976, Peter Chen defined the entity-relationship model also known as the ER model very much useful to define relationships among entities. From 1980 onward, the relational model became a widely accepted database component. Object-oriented DBMS was developed in 1985, and gradually, Microsoft MS access and Internet database applications were introduced with rising demand to manage data. To give support to standard input data formats, in 1997, XML was integrated into DBMS products by many vendors. Relational databases are the best solutions to store structured data, but in this era, in response to the major challenges faced by the internet data and the need for faster search engines for unstructured data, NoSQL proved to be useful.

3 Advantages of Database

The file system manages data using files in the disk. In this, data are entered by the user in Excel sheets, Word documents, or PowerPoint presentations with specific names under certain files and folders. Hence, database proves itself to be better due to the following cases:

- Data redundancy: It refers to the duplication of the same data at different places. For example, if an employee's department gets changed then it has to get updated in all the places where this information is stored. This also shows that the same data are present in multiple locations or files. Data redundancy can increase the size of the database, thus making the search or access process time slower.
- Data inconsistency: In the example stated above, if an employee has been a part of two different projects then his details have been stored twice. Now, if he asks to change his email address, it is possible that out of the two, one instance of his email address gets changed and the other would stay intact. This would lead to data inconsistency due to the presence of different email addresses of the same person in the office.
- Data searching: In order to search for an employee, project or any specific attribute, it is much easier to carry it out in DBMS. On the other hand, in file system, entire search operations need to be made to carry out the same.
- Data isolation: In the file system, data will be scattered among multiple files of different formats such as Excel, PPT, and Word. Here, it would be difficult to create a program to carry out an operation. This is majorly faced by some offices where they do not use a database and are dependent on only the documents.
- Data integrity: In DBMS, the integrity of the data is maintained due to the integrity constraints. Data integrity makes sure that processes such as insertion, deletion, and updating of records are successful. For example, the phone number of any employee cannot exceed ten digits in India.
- Data security: The date of the employees and other confidential information remain more secure in the DBMS as compared to the file system. The security of data is of the utmost importance to an office.
- Atomicity of updates: If a system failure occurs while a process or update is being carried out in DBMS, this would lead to the database being in an inconsistent state. For example, if the increment or any other important reformation of some information of an employee is being carried out then due to some system failure or electricity cut-off, the system shuts down. No problem would occur in the database upon reopening.
- Concurrent access by multiple users: If a database does not support concurrent access then it would be impossible for the database to be in control. For example, the admin office of a company is handled by different users. So, if there is a concurrent by the users, the information would be updated simultaneously without any problem.

4 Types of Database

A database comprises a set of interrelated data. The linking of data is explained through different types of databases. These databases are discussed below:

4.1 Hierarchical Database

In the 1960s, IBM developed an hierarchical data model, the data model in which the data are stored in a tree structure called the hierarchical data model. Data are stored as records that are connected to other records through links. A record is a collection of fields that can store one value each by defining the type of record, and the type of the field can be defined. In the hierarchical data model, it is required by rule that each parent can have only one child whereas each parent record can have more than one child records. To get data from such a data structure, the whole tree needs to be traversed starting from the root. Shared-memory, shared-disk, and shared-nothing architectures are different type of architectures that are combined by the hierarchical database architecture. An interconnected network connects the nodes present in the system at the highest level, not sharing the memory with one another. Therefore, a shared-nothing architecture is present at the highest level, whereas each node of the system comes under the shared-memory architecture, including some processors [1].

4.2 Network Database

The network database system was the first system to use multi-parent structure. It was developed to replace the traditional-used hierarchical database model which in contrast had a single parent system. This structure, when visualized, forms a web-like network consisting of multiple parent–child relations. This allows the network database to have well-modeled relationships between its entities. The lack of flexibility given by the hierarchical database model was overcome; hence, in the early 1970s, it was highly implemented and used. It provided more flexibility in terms of parent–child relationships, as it offered multiple parents to a single record (entity). This made accessing the data fast and fairly simple. This was the first model to introduce many-to-many relationship between tables. This kind of many-to-many relation (having multiple links) applied in two ways: the schema and the database, which itself is a generalized graph representation of the records connected by relationships.

This model failed in the long run due to the introduction of relational database model. It was difficult to maintain and was not as flexible as the relational model. Also, it was only accessible by well-qualified engineers and programmers. Still, it needed the programmer to understand the data structure well in order to make it more efficient.

4.3 Relational Database

The most widely used and the most popular database is the relational database. In this kind of database, the data are stored in the form of rows and columns in a table with a

specified table name. As the name suggests, the data in the tables are related to each other; hence, these tables are called relations. The rows and columns are the tuples and the attributes of the table, respectively. The information stored in the relations is called a record. Every table has an attribute that uniquely defines the records under it. This field or attribute is called the primary key. Since the data are organized in the form of rows and columns, the information can be retrieved easily since the data are stored in a comparatively systematic manner. Relational database management system has proved to be the most easily designed database. The open-source software such as MySQL, PostgreSQL, SQLite, and many more has made the database storage and management process easier, especially for start-ups and for small companies who are opting for RDBMS. The data in the database are accessible through the queries entered by the user which is obtained by the means of relational algebra and relational calculus. An advantage of RDBMS is that a user can view the data in whichever way or order he wants just by writing the query as per his requirements. All the advantages of DBMS are present in RDBMS making it ideal for office use. DBMS, as discussed above, includes hierarchical, network, and object-oriented. But, none of these DBMS stores data in a tabular manner. In addition to this, these systems do not apply any security when it comes to data manipulation. RDBMS follows ACID properties, i.e., atomicity, consistency, isolation, and durability. Also, RDBMS can store and handle a comparatively large amount of data and can support access to multiple users.

4.4 Object-Oriented Database

In an object-oriented database, information and data are represented by objects, which are instances of classes. An object-oriented database can be called a database having the capabilities of a relational database along with the capabilities of object-oriented programming. The object-oriented database management system has three main parts to it which are object structure, object classes, and object identity. The term object-oriented database management system (OODBMS) first came into play circa 1985. Several research projects have been done on the subject, with the most notable one being ORION [2].

The main advantage that object-oriented databases have over other databases is that for one, they handle complex data in a better and more effective manner as compared to other databases. As a result of object-oriented programming languages such as C++ and Java being used in these databases, less coding is required, and as a result, the coding used is more efficient. One other advantage is that data can be accessed quickly and the response time is comparatively less due to the use of pointers while searching for something. In the memory sector, OODBMS does wonderfully well, taking into account the use of single-level memory and automatic-method storage.

While there are several advantages of the object-oriented database, there are a handful of disadvantages as well. Perhaps, the most major drawback of using object-oriented databases is that they do provide adequate tools for querying and reporting

the database. A feature of the object-oriented is like double-edged sword-object-oriented programming. On one hand, there are advantages to using the object-oriented programming languages, on the other hand, object-oriented databases are limited to these specific languages and other languages cannot be implemented. The major application of object-oriented databases is done in scientific fields as these fields require complex data to be handled in the most efficient as well as the most effective way possible such as engineering, high energy physics, and molecular biology. Examples of object-oriented databases are Smalltalk, Perst, etc. [3].

5 Database Software

Database software helps small and big firms and companies to manage the data efficiently. It provides a front end, simple interface through which, even people with a non-technical background can carry out the tasks. Database software can perform various tasks such as storing, managing, changing, searching, and extracting the data within a database. A database system as such, also secures the information in the database which is crucial for all companies. It also increases the overall accuracy as well as efficiency of using databases. Some of the most popular database software presently are Oracle, MySQL, PostgreSQL, MongoDB, etc. We have shown some statistics and qualities of these software (Fig. 1 and Table 1).

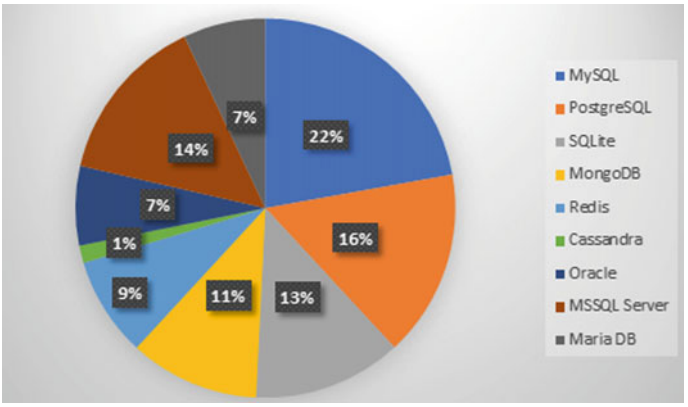

Fig. 1 Database comparison in terms of usage [6]

Table 1 Software comparison chart [13]

Software	Data structure	Scaling capacity	Usage ease
MySQL	SQL	Vertical, complex	Easy
MariaDB	SQL	Vertical	Easy
Oracle	SQL, multi-model	Vertical	Hard
PostgreSQL	SQL, object-oriented	Vertical	Hard
MSSQL server	T-SQL	Vertical, complex	Hard
MongoDB	NoSQL, document-oriented	Horizontal	Easy
Redis	NoSQL, key-value	Horizontal	Easy
Cassandra	NoSQL, column-oriented	Horizontal	Hard

6 Future Trends

The database software market continues to expand with the constant generation of data. With the emergence of big data, the amount of semi-structured and unstructured data has critically escalated. "IDC Technologies has predicted that in 2025, the world will create and replicate 163ZB of data, representing a tenfold increase from the amount of data created in 2016." [4] The relational database model (or RDBMS) is capable of handling only the structured type of data. NoSQL, known as Not Only SQL, is the solution for handling unstructured or semi-structured data and provides a mechanism of faster storage and retrieval of data. It is considered to be the next-generation database. Some NoSQL databases provide suggestively higher data throughput than traditional RDBMSs. MongoDB, Neo4j, and Amazon DynamoDB are some of the examples of NoSQL databases [5].

However, the latest trends include adopting a hybrid database structure instead of a single database structure. In this way, the users can utilize the capabilities of both, SQL and NoSQL. This bridges the relational database with the non-relational database (Table 2).

Table 2 Features comparison [6]

Feature	NoSQL	Relational database
Schema	Dynamic	Pre-defined
Properties	CAP (consistency, availability, and partition tolerance)	ACID
Scalability	Horizontal	Vertical
Consistency and reliability	Poor	Good
Availability	Good	Good
Data	High amount of unstructured data	Low to medium amount of structured data
Example	Cassandra and MongoDB	PostgreSQL and Oracle

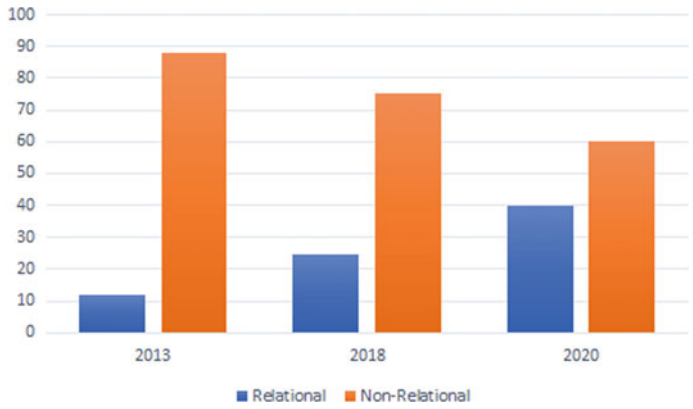

Fig. 2 Relational versus non-relational database [6]

Security is going to be one of the most prominent complications in the near future. IDC states the data to be managed has reached 40ZB in 2020, and only half of that is protected. For protection, data logging is necessary. Since the data are getting generated from many different places such as web servers, file servers, and credit card payment gateways, it is not structured in the same way for each of them, making it very difficult to design a relational database for the same [7]. Therefore, a hybrid model can be optimized here.

Management of database is a tedious and time-consuming task for the most administrators. Hence, automating database systems is an emerging trend. AI is used to simplify maintenance which includes handling, updating, and upgrading. This is known as augmented data management. Also, with cloud service providers (CSPs) offering more options, organizations (and DBAs) are embracing their current infrastructure with the cloud services. Some of the databases, also working on cloud platforms, are Microsoft Azure, Amazon Web. Thus, for small firms and organizations, the relational database has always been useful, but as and when the number of data increases, there is a shift from SQL to NoSQL [8, 9] (Fig. 2).

7 Augmented Data Management

Augmented data management is a newly emerging technology in the vast expanse of data management and data analytics. This process stems from the capabilities of two of the most popular and robust technologies that the world has to offer at this moment in time, artificial intelligence and machine learning. The main purpose of augmented data management is to increase the number of automated tasks, and thus, reducing manual tasks such as handling and configuring as well as tampering of huge amounts of data. This includes various processes which include copying, synchronizing, restoring, migrating, archiving, and backing up of data, and the most

tedious ones being cleaning and preserving the quality of data. Gartner, a global research and advisory firm, has predicted that by 2022, 45% of the manual tasks of database management would be reduced due to the introduction of machine learning and automated service-level management and that owing to the introduction of this AI-enabled automation, and 20% of the specialized data management professionals will no longer be in need. According to the firm, the technology is expected to influence effectively every area of data management, including database management systems data integration, metadata management, and various others [10].

An immense drawback that is faced by data scientists and data analysts is the tremendous amount of time that is spent on unnecessary manual tasks such as the ones mentioned above. According to an analysis carried out by AIM, they spend 80% of their time in the maintenance and organization of data. The implementation of augmented data management hopes to reduce this time significantly and thus increasing the efficiency of not only the data of an organization or company, but also the time and productivity of its employees as well as the financial costs of operations by decreasing the data complexities. Another facet in the kaleidoscope of advantages that augmented data management provides to a company is ensuring authentic data as human errors can lead to distorted data which can, in some cases, lead to major consequences.

Augmented data management can be used in almost every sector where database management systems are in place. For example, if a pharmaceutical company or bank wants to take decisions based on machine learning, the necessity to make sure that the data are of the highest quality is of the utmost importance. In a financial firm, any specific trading portfolio is judged on metadata. Augmented data management assists in the management of such metadata. An industry that is going to benefit greatly from this technology is the media and entertainment industry. Machine learning and AI are the roots of new-age digital advertising, which is projected which is estimated to increase by 23% by 2023, according to Statista. It can also be used to monitor and control the security and privacy of online users who decide to upload photos and/or videos online. With the substantial increase in the number of media being uploaded to the net, augmented data management can also help with the relevant data storage in specific segments. Thus, augmented data management is one of the most exciting upcoming prospects to keep a keen eye on in the next couple of years [11].

8 Office Management Using Relational Database Approach for Study in India Program (SIP)

This database is designed for the Study in India Program. This database helps the current way of storing data to change into a smart and simpler way of storing data. This project emphasizes more on smart and efficient work rather than the tedious data entries and repeated documentation of every new Study in India Program. Since,

the amount of data generated is not in TB or ZB, it is more feasible to develop a relational database for SIP.

8.1 About SIP

The Study in India Program is an integral part university which is known for establishing connection with foreign universities. Study in India Program gives the exposure of the Indian culture, business, psychology, and tradition to the international students from various countries [12].

8.2 SIP Requirements

SIP generates a lot of data related to the students, faculties, and university which is further used to prepare reports and post-analysis. To ease this activity, we have established an interface to generate an integrated database management system. This database will help SIP to ease the office activities documentation, reporting activities and to perform the retrospective analysis from past datasets.

From multiple discussions with SIP team members and our own experience and knowledge regarding SIP, following major requirements at the highest level of abstraction are identified:

- To store the itinerary/time table of SIP in database to keep the record for the future reference.
- To store student, faculty, and university details.
- To store the feedback by the students.
- To keep record of student and volunteer attendance.
- To store pictures or their links.
- To store and manage other useful information like documents/PPTs/lectures.
- To keep record and the information of the managers that the future programs might need.
- Stay details for SIP.

8.3 SIP Database Design

The entity–relationship diagram (ER diagram) is a model that shows the relationships between the tables. It is the initial step while designing a database. Here, there are four types of relationships, one to one, one to many, and many to one. We have not used many to many, as it was not required. The tables have primary keys and some have composite keys (their combination becomes the primary key for the table). Also, there are foreign keys, which are the primary keys of some other table and may or

may not be the primary key of the table it is related to. The dotted lines represent non-identifying relationship, in which the primary key of the parent table is not the primary key of the child table. The solid lines represent identifying relationship, in which the parent table and the child table have the same primary key. (refer pg. 7 for ER diagram).

9 Graphical User Interface

Graphical user interface forms an interface between the user and the devices by using icons and other graphical notations. After the designing part, i.e., the creation of the database management system, the project moves ahead. The focus is now on making an application that is user-friendly and can be easily accessed by an office or company. Without a GUI, access to a database is quite cumbersome. A person needs to know the SQL statements to insert, delete, update, or search in the database. GUI makes this process easy, as the database gets linked to Java (GUI) and develops an application that is secure and can be modified as and when required. SWING graphical user interface is a part of the Java framework and has been used to develop a desktop application that can be used in any system and any platform.

Firstly, a code is written to establish a connection with the required database. Here, Java Eclipse IDE (integrated development environment) is used with SQLite (database). The first page of the application is the login page which makes sure that only the registered people or the people who have the correct username and password are the only ones who are allowed to access the menu page (Fig. 3).

The menu page is the main content page which provides a handle to work upon five main options given on the menu bar (see Fig. 4). International students, volunteers, and patrons tabs give four options—insert, delete, update, and search student. The insertion and deletion of data are done using PreparedStatement, in which the general SQL statement is passed. For the search and update operations, ResultSet along with PreparedStatement is used along with SQL update query.

Contact MenuItem is for the purpose of getting contact information about the main admin.

The SIPs MenuItem contains the information related to a specific SIP. As the name suggests, the Change Password button gives the user an option to change the password. The password can be changed only when a person enters the right username and old password. Register New gives an option to the admin or the other users to register for new users.

In conclusion, the application can perform all the SQL queries along with easy access to the database. This application also ensures the security of the database along with the atomicity of updates.

The most important factor is reusability. Just by changing the names of the label and buttons, and establishing a new database connection, other offices or companies can also use the same GUI. Other advantages such as data redundancy and searching are also there which make DBMS GUI more reliable than the trivial file system

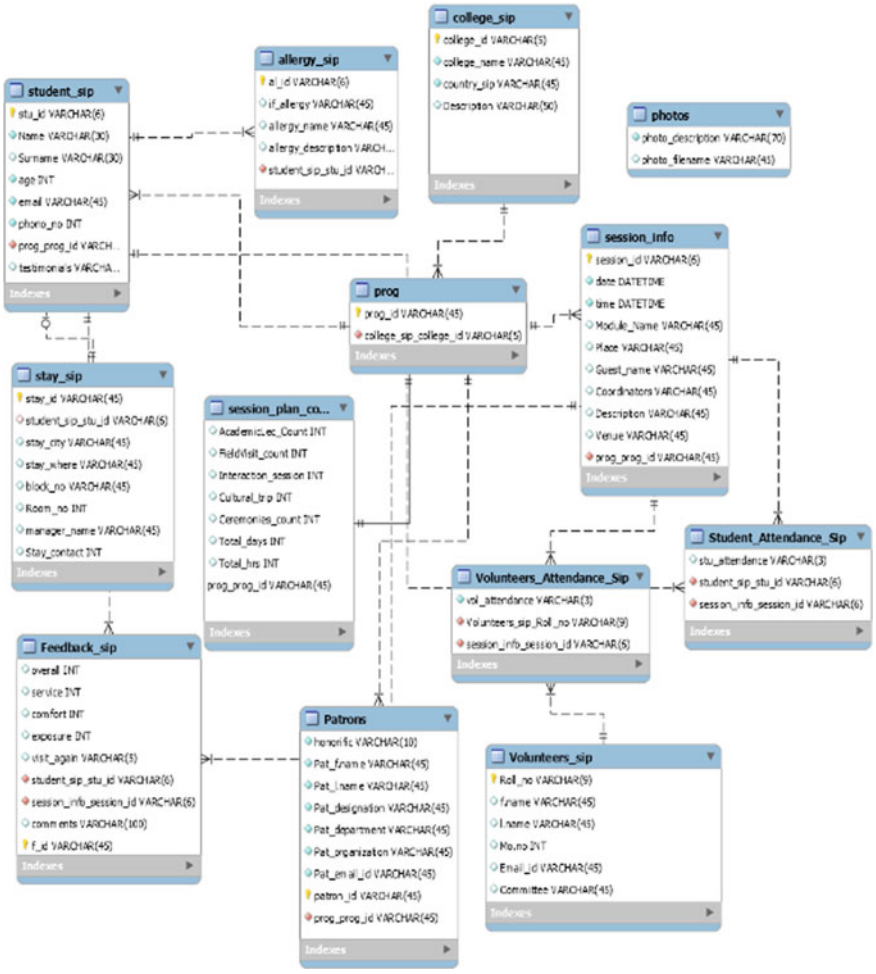

Fig. 3 ER diagram

method. This was just an example of a small office. The same GUI can be used for huge companies and organizations to store thousands of records. But, this also has disadvantages when it comes to a large level; i.e., companies like Google use NoSQL, because of the huge amount of data as the data are increasing at a fast rate and thus SQL might become a less reliable source.

Fig. 4 Menu page

10 Conclusion

The database provides the best solution for the data generated from big and small firms, businesses, schools, colleges, and other commercial offices. Post-analysis of data is required to improve the business strategies, future planning, and to know about the customer's liking and marketing trends. The database is proven to be more advantageous than old file-based systems. Network, hierarchical, relational, and object-oriented-based databases are the popular approaches to store and manage the database. In the era of big data where a huge amount of structured, semi-structured, and unstructured data needs to be managed, the NoSQL approach has emerged and has become popular day-by-day. A small database application for office use is discussed in this paper which is based on the relational database approach. Implementation to store and retrieve the actual data into the database has been done using Java—JDBC.

Acknowledgements We would like to extend our gratitude to Prof. Nigam Dave, Head of Office of International Relations, PDPU, and Dr. Ritu Sharma, Associate Professor, PDPU, for providing insight into SIP requirements. We are immensely grateful to them for guiding us through our project and providing us with information as and when required.

References

1. Silberschatz, S., Korth, H.F., Sudarshan, S.: Database system concepts
2. Object database page. https://en.wikipedia.org/wiki/Object_database

3. Ketabchi, M.A., Mathur, S., Risch, T., Chen, J.: Comparative analysis of RDBMS and OODBMS—a case study
4. Database Trends and Application. http://www.dbta.com/Editorial/News-Flashes/Key-Database-Trends-Now-and-for-the-Future-131888.aspx, last accessed 2019/5/21
5. ChristofStrauch, Prof. Walter Kriah: NoSQL Database
6. Dbzone webpage. https://www.dbzone.org (for figure 1,2)
7. NoSQL and hybrid databases. https://www.stratoscale.com/blog/dbaas/hybrid-databases-combining-relational-nosql/
8. Sethi, B., Mishra, S., Patnaik, P.K.: A study of NoSQL database. Int. J. Eng. Res. Technol. (IJERT) (2014)
9. Padhy, R.P., Patra, M.R., Satapathy, S.C.: RDBMS to NoSQL: reviewing some next-generation non-relational database's. (IJAEST) Int. J. Adv. Eng. Sci. Technol. (2011)
10. https://www.gartner.com/en/conferences/apac/data-analytics-india/gartner-insights/rn-top-10-data-analytics-trends/augmented-data-management
11. AnalyticsIndiaMag webpage. https://analyticsindiamag.com/how-startups-can-leverage-augmented-data-management-to-drive-business/,last accessed 2019/10/14
12. PDPU official site. https://www.pdpu.ac.in/exposure-program.html
13. Comparing Database Management Systems. https://www.altexsoft.com/blog/business/comparing-database-management-systems-mysql-postgresql-mssql-server-mongodb-elasticsearch-and-others. Last Accessed 20 June 2019