Relational Database Management Systems in Business and Organization Strategies

Article *in* Global Disclosure of Economics and Business · July 2020

DOI: 10.18034/gdeb.v9i2.700

1 author:

Harshith Desamsetti
Chater Communications Inc
18 PUBLICATIONS **195** CITATIONS

Relational Database Management Systems in Business and Organization Strategies

Harshith Desamsetti

Software Engineer, MedAllies, Inc., Fishkill, NY 12524, **USA**

*Corresponding Contact:
Email: harshithdesamsetti9@gmail.com

Manuscript Received: 14 Oct 2020 - Revised: 14 Dec 2020 - Accepted: 22 Dec 2020

ABSTRACT

A collection of interconnected, raw data is a database management system. This collection also includes programs or instructions that may be used to access the data. The enormous collection of data is typically referred to as a Database and contains information necessary to access the data. Providing a method to store and retrieve information held in the database is the management system's primary objective. The database makes it possible to store information more conveniently and efficiently. This article discusses the application used in database management systems and how the data in the application are associated by describing the backdrop of DBMS. This paper also discusses the application that is used in database management systems.

Key Words: Variation, Relational Database, Database Management Systems, Organization Strategies

INTRODUCTION

Organizations need accurate, trustworthy data to make decisions. To sustain partnerships, the organization keeps records on numerous aspects. Similar data are termed databases. Database systems are collections of connected files and data interpretation details (Gutlapalli, 2016a). A database system is a computer-based record-keeping system that records and maintains data. A database management system (DBMS) provides access to database data. To define, store, and retrieve database information quickly and effectively, the DBMS is designed. The DBMS communicates with application programs so multiple applications and users can use database data. The DBMS also controls the database, prevents unauthorized people from accessing it, and protects data privacy. Generally, a database organizes linked information. The organized information or database allows data recognition and processing to retrieve or make decisions (Desamsetti, 2016b). People use multiple databases daily. Dictionary, phone directories, library catalogs, etc., are alphabetical or categorized databases (Lal, 2015).

DIFFERENT KINDS OF DATABASES

There are a wide variety of database formats. They can be categorized based on content, including bibliographic, full-text, numerical, and image data. In computers, databases are frequently classified according to organizational strategy (Sathiy, 2017).

The following are examples of some of the most critical organizational databases:

Relational: This strategy utilizes tables to specify the data to be restructured and retrieved in various ways. Tables are the fundamental building blocks of relational databases. In those tables, the data is organized into categories that have been predefined. Each table consists of columns with at least one data category and rows containing a specific data instance corresponding to the types defined in the columns. Rows, columns, and tables are the organizational components a relational database utilizes to structure the data about a particular consumer. These are indexed so that searching with SQL or NoSQL queries is more easily facilitated. The query language known as structured query language (SQL) is used in relational databases' user and application program interfaces. Adding new data categories to relational databases is simple and does not require altering existing applications. A relational database management system, or RDBMS, stores, manages, queries, and retrieves data from a relational database. Users of an RDBMS are typically granted the power to regulate read/write access, specify report generation, and assess utilization of the System (Lal, 2016). Some databases offer compliance with atomicity, consistency, isolation, and durability, or ACID. This helps to ensure that data is accurate and that transactions are carried through in their entirety (Hiroshi et al., 2013).

Distributed: This database keeps its records or files across several sites worldwide. Additionally, data processing is decentralized and replicated across the various nodes that make up the network (Gutlapalli, 2016b). Distributed databases can be homogenous, meaning that all physical locations use the same hardware in the background and operate the same database applications and operating systems. They are also capable of having a diverse makeup. Each location's hardware, System, and database applications may differ in such situations.

Cloud: To create a virtualized environment, these databases are constructed within a public, private, or hybrid cloud. Users are assessed additional costs proportional to the data storage and bandwidth consumed (Lal & Ballamudi, 2017). In addition, they benefit from high availability and scalability on demand. These databases are compatible with applications that run using the software as a service model.

NoSQL: NoSQL databases are an intelligent option when working with significant volumes of distributed data. They can better address the performance challenges that arise with large amounts of data than relational databases. In addition, they are excellent at evaluating massive unstructured data sets and data stored on virtual servers in the cloud. One alternative name for these kinds of databases is "non-relational databases."

Object Oriented: Object-oriented programming languages were used to generate the data included in these databases. They emphasize organizing data and data points instead of actions and rationale. For illustration's sake, an image data record would be considered a data object, not an alphabetic or numeric value.

Graph: These databases fall within the category of the NoSQL database type. They use graph theory techniques in storing, mapping, and querying relationship data. Nodes and edges are the building blocks of graph databases. Entities that connect other nodes are

known as nodes. Interconnections are a common topic of investigation when using these databases. Graph databases are frequently utilized in analyzing data concerning clients as they engage with a company via its web pages and various social media platforms. Graph databases use SPARQL, a declarative programming language and protocol for analytics purposes (Gutlapalli, 2017c). In addition to performing all of the analyses that SQL is capable of, SPARQL may also be used for semantic analysis, which investigates relationships between entities. Because of this, it is excellent for conducting studies on data sets that contain structured and unstructured data. Users of SPARQL can do analytics on data stored in a relational database. These analytics can also consider friend-of-friend relationships, PageRank, and the shortest path.

Hierarchical: The information is stored in a tree-like format inside databases with hierarchical organizational structures. This model depicts connections in the real world, such as instructions for preparing meals, sitemaps for websites, and other such things (Liu et al., 2017). The following are some of the characteristics of a hierarchical model:

- **One-to-many relationship:** The data are organized into a structure that looks like a tree, reflecting the one-to-many relationship between the datatypes.
- **Parent-child relationship:** Every child node has a parent node, even though a parent node can have more than one child node under its care at a time.
- **Deletion problem:** When a parent node is removed, it causes all of its child nodes to be removed.
- **Pointer:** Pointers are used to traverse between the data that has been saved and to connect the parent node to its children.

Network Model: The hierarchical model is expanded upon in the network model by allowing many-to-many links to exist between linked records (Prabhjot & Neha, 2017). This, in turn, implies the existence of numerous parent records. The model is constructed using sets of connected records as building blocks based on mathematical set theory. The following are requirements that a network model must meet. When there are more relationships in this model, it indicates that the data is more interconnected (Lal et al., 2018). Because there are now more connections between records, multiple ways exist to get the same information. This makes gaining access to the data both rapid and straightforward. The network model is worked on by the circular linked list, which executes operations (Desamsetti, 2016a).

Float Data: The float data model has a two-dimensional data element array. This simplistic architecture displays the database as a table with rows and columns (Deming et al., 2018). To access any data, the computer must first read the entire table. The modes could be more robust and efficient as a result. This method is only helpful for working with small data sets since the computer must read the entire flat file into memory to access or modify the data. It is impossible to store a significant quantity of data using this model and to retrieve any data, one must search through the entire table. Because of this, the model is inefficient.

Entity-Relationship: The object of interest is classified as an entity in this database model, and the entity's properties are classified as attributes. These relationships are then utilized to connect the various entities in the model. ER models are created to illustrate relationships in a manner that is accessible to a wide variety of stakeholders. This architecture helps establish databases, which can later be converted into relational model tables (Gutlapalli, 2017b). To put it another way, an ER diagram is used to

display the logical structure of a database straightforwardly. Because the ER model generates a conceptual knowledge of the data, it can be used in the future as a guide to construct the database by following its instructions.

Semi-Structured Data: Because the semi-structured data model is a generalized form of the relational model that enables flexible data representation, it is impossible to differentiate between data and schema in this model because some entities may be missing one or more characteristics. This is because the semi-structured data model offers flexible data representation (Gutlapalli et al., 2019). On the other hand, other entities could have extra properties that make it easier to modify the database schema. Certain entities may be missing particular traits in this paradigm, while others may have an additional property. This strategy makes it possible to store data flexibly. In addition to this, it imparts the characteristics of freedom. It can be utilized to describe interactions between databases that adhere to various schemas (Snehal & Prajakta, 2015).

TYPES OF RELATIONAL DATA

Data Definition Language (DDL): DDL stands for "data storage and definition language," a specialized language that employs particular kinds of declarative statements. The data type of an attribute is specified with the help of the DDL, which is then used to establish the database name and tables (Mandapuram et al., 2018). When using DDL, we use constraints of an attribute by specifying the primary and foreign keys. This generates a data set with a one-of-a-kind identity and ensures that it does not include any null values. The primary focus of the database management system is on the integrity constraints associated with an attribute. These include domain constraints, authorization, referential integrity, and assertions.

Data Manipulation Language (DML): The process of carrying out a task in a database of entities is called data manipulation language (DML). Insertion, deletion, selection, and updating are the tasks that come into play with DML. The user can interact with a substantial amount of data while using DML. Two distinct varieties of programming can be done in this language: procedural programming and declarative programming (Gutlapalli, 2017a). The idea of a procedural call serves as the foundation for the procedural programming language, and the word "procedure" refers to the various functions used in the language. A declarative language is a programming model that specifies the process of expressing a logical calculation without the control flow. This process can be done without the need for an interpreter.

Data Control Language (DCL): Controlling a user's access to authorization information is done with the help of DCL. It comes with the GRANT command as well as the REVOKE command. The admin user grants the child user permission to access their data by executing the grant command on their behalf (Chen et al., 2019). This idea involves an admin user and a kid user. The same administrator uses the revoke command to strip the child user of the previously granted privileges.

PROPERTIES OF ACID IN DBMS SYSTEMS

If a database undergoes adjustments, the data management should continue to be integrated. If the data's integrity is breached, the entire data set will be messed up and ruined (Kashyap et al., 2016). Within a single logical unit of work known as a transaction, it is possible to access and modify a database's contents (Thodupunori & Gutlapalli, 2018). Read and write

operations are used in these transactions to access the data. To guarantee that database transactions are consistent, it is necessary to remember certain qualities. These characteristics are called the ACID properties: atomicity, consistency, isolation, and durability.

- **Atomicity:** "Atomicity" refers to a circumstance in which the entire transaction occurs or does not occur. As a result, no point constitutes a compromise. There is no such thing as a partially completed transaction. Every transaction can be thought of as its independent unit, which either ends in its whole or does not. These two processes are going on right now in this location (Dekkati & Thaduri, 2017). If a transaction is committed, we can see the modifications made. Therefore, the "All or nothing rule" is another word for atomicity. If a transaction fails, we won't be able to view any changes made to the database.
- **Consistency:** Before and after a transaction, we must uphold the integrity criteria for each database to remain consistent. It is a term that refers to the precision of a database (Thaduri & Lal, 2020). Any changes made to the database must always be saved to ensure its integrity is preserved while using a database management system (DBMS). Transactions depend on maintaining data integrity since doing so guarantees that the database will be consistent both before and after the transaction (Desamsetti & Mandapuram, 2017)). The information must be correct at all times.
- **Isolation:** Multiple transactions can occur concurrently with the isolation mechanism, eliminating the possibility of inconsistent database states. A transaction that is independent of any other parties takes place. Any modifications that are made within that transaction are never going to be accessible to any of the other trades unless the exact change that was completed within the transaction has been either committed or stored in the memory. The benefit of isolation ensures that when we run the transactions concurrently, the resultant state will be the same as the state formed when the transactions were done in a particular order and a sequential fashion (Thaduri et al., 2016). Transactions are carried out in a manner that is not dependent on anyone else and are not interrupted. The intermediate state of a transaction is hidden from view from any other transactions.
- **Durability:** When discussing database management systems (DBMS), "durability" refers to the assurance that data will remain in the database after an action is finished. The data should be reliable enough to continue functioning normally if the System becomes corrupted or malfunctions (Daniel, 2016). On the other hand, if it were to vanish, the recovery manager would be responsible for ensuring the database is resilient in the future. Every time we modify the settings, we must execute the COMMIT command to commit those changes. Once the execution of a transaction is finished, the durability feature causes any updates or modifications to the database to be written to and saved on the disk (Dekkati et al., 2019).

FUNCTIONS OF DBMS

According to Codd, a complete database management system (DBMS) possesses eight primary functions.

Data storage, retrieval, and update: Since numerous users may share the same database, the database management system (DBMS) must provide multiple user views and allow users to store, retrieve, and efficiently change data (Pokorny, 2013).

Data dictionary and directory: It is required of the DBMS to keep a data dictionary that is user-accessible.

Transaction integrity: A transaction culminates in a series of stages of delineated commercial activity. For the database management system (DBMS) to preserve transaction integrity, it must give the user or application software the ability to establish transaction boundaries, also known as the logical beginning and end of transactions. The database management system (DBMS) should then commit the modifications for successful transactions and reject the changes for failed operations.

Recovery Services: If there is a problem with the System, the database management system (DBMS) needs to be able to restore the database. Operator errors, disk head accidents, and programming faults are all potential causes of a malfunctioning computer system.

Concurrency Control: Two or more users can attempt to access the same data simultaneously when using a database that numerous users share. Erroneous outcomes are a distinct possibility if two users make a contemporary effort to modify the same data record. The safeguards need to be included within the DBMS to prevent interference or counteract its effects.

Security mechanisms: The data needs to be safeguarded from being misused intentionally or accidentally, as well as from being distracted. The database management system offers means for managing access to data and specifying the types of actions each user can perform.

Data Communication Interface: Users will frequently access a database by utilizing remote terminals of a telecommunications network. A telecommunication monitor processes the flow of transactions going to and coming from the distant terminates. The database management system (DBMS) must have an interface with one or more telecommunication monitors to ensure that all necessary operations are carried out and that the System will assist the end user rather than a burden.

Integrity services: The database management system (DBMS) must give its users the necessary features to help them maintain the data's integrity. It is possible to program various edit checks and integrity limitations into the database management system (DBMS) and its software interfaces. The data dictionary is the typical method by which these tests are carried out.

APPLICATION OF DATABASE MANAGEMENT SYSTEMS

Database management system (DBMS) is an application software that enables users to interface with the System and other applications to record and analyze data. Databases allow online transactions on websites between customers and providers, and they are also used to process internal data within businesses. The management of libraries, aircraft reservation systems, banking, inventory management, and other applications are all examples of possible uses. These applications are briefly described in order, one after the other (Vasin & Yasakov, 2016).

Banking system: The database management system is utilized for customer information maintenance, generating statements, and monitoring day-to-day credit and debit transaction particulars.

Education System: Keeping records of interconnections helps ensure that information can be successfully stored, maintained, and accessed.

The interrelated records can be successfully accessed with relative ease. It maintains data associated with students, tests, scores, attendance, fees, courses, personnel, and payroll, among other things.

Industry: Several locations within an enterprise, such as distribution centers, warehouses, and production units, all use database management systems.

Information regarding the products that are supplied and delivered is kept up to date at the distribution center.

Military: A significant part of the function of the DBMs is to ensure the safety and security of the records of millions of files and soldiers. It offers a high level of protection to ensure the safety of military personnel.

Online Shopping: Across the globe, people are increasingly considering the convenience of online purchasing. When consumers do their shopping online, the information regarding their purchases, as well as their payment data, are saved using DBMS. Every nuance is recorded in the database so that information can be retrieved from it expedient and straightforwardly in the future.

Airline Reservation System: DBMS is where the information on an air flight's departure, arrival, and any delays that may have occurred is stored. The travel experience for the traveler is improved as a result of this.

Railway Reservation System: The DBMS is an essential component of the railway reservation system. Every last aspect of the train's design is described here. Stored using a database management system. The specifics of the train, including the passenger's name, the train's PNR status, and the train's arrival and departure, can be stored in the database and retrieved from it quickly.

Credit card transactions: The transactions made with credit cards throughout the acquisition of the goods and transactions completed online with a credit card are maintained by the database management system (DBMS). The database has been updated with the new credit card information on every occasion after use.

Social Media: These days, a growing number of people are using various forms of social media.

All of the user's information, as well as their opinions, may be found here. Stored within the database management system. Keeping this information is beneficial to contacting a social network operator to verify the information proportional to the number of people using the social networking platform daily (Bodepudi et al., 2019). The whereabouts of users and detailed information regarding users can be obtained and examined without effort by using this data management system for bases.

DBMS PROVIDERS

We can use any number of database management systems, also known as DBMS software (Chudinov et al., 2017), to store information, organize data, and perform data analysis. The following are some of the best choices:

Microsoft Access

Access, Microsoft's database management system, combines a graphical user interface with software development tools and the relational Microsoft Jet Database Engine. Access is also

known as Microsoft Access. It is straightforward because of the intuitive graphical user interface (Mandapuram, 2016). It is included in the Microsoft Office suite of programs in the software's professional and higher editions.MS Access' benefits include the speed and simplicity with which it can create a fully functional relational database management system, the clarity with which data can be imported from a variety of sources, and the simplicity with which it can be customized to meet the needs of both individuals and businesses (Mandapuram, 2017a). One of Microsoft Access's significant limitations is that it is most useful for small and medium-sized enterprises.

MySQL

This relational database management system (RDBMS) is open source and has a client-server architecture. Open source software is program code designed to be modified and used without cost. Let's talk about client-server relationships for a little while. Clients are the devices that use RDBMS software and install it on their computers. Whenever they need access to the data, they create a connection to the RDBMS server.

Simply put, the "client-server" component consists of those two things. MySQL has a simple syntax and a modest level of sophistication in its features. Even developers consider MySQL to be a database because of its natural-sounding language. Scalability should have been felt throughout MySQL's development, meaning it needs to lend itself better to large databases.

Oracle Database

This is the fourth attempt that the company has made to construct a relational database management system. Oracle databases can hold more significant volumes of data, which is very beneficial for large businesses. In addition, it is flexible, which makes it useful for shared structured query language (SQL) and locking. Users have instant access to data items thanks to the relational database foundation that the System utilizes. It constructs a robust engine for synchronous data processing, simultaneously processing data stored in memory. Using Oracle databases straight away is not a recommended course of action. Having Oracle DB engineers who are certified to run it is much recommended. If we intend to use Oracle DB, consider employing individuals trained in the System.

MongoDB

MongoDB is a data storage that can be used for various purposes. This provides multiple features that are not available with other types of databases. The data in MongoDB is mapped to a schema that can be configured. If the requirements of our application shift in any way, we can make speedy adjustments to how our data is stored. This enables us to improve new ideas in a significantly more practical manner. Because MongoDB also provides schema validation, we can decide how much control us want over our database's schema and lock it down accordingly (Mandapuram, 2017b). This suggests that it can manage any data structuring needs we may have in the future. Users of MongoDB can combine documents through the use of references and procedures that are tailored to various requirements.

IBM Db2 DBMS

IBM Db2 was developed by some of the most knowledgeable database specialists in the world. It provides developers, database administrators, and enterprise architects the tools they require for real-time analytics and low-latency transactions, even for the most taxing workloads. Db2 is the tried-and-true hybrid database that provides extreme availability, advanced integrated security, seamless scalability, and intelligent automation for systems that

run the worldwide, from microservices to AI workloads. Db2 also enables seamless scalability, allowing infinitely more data to be stored in the same space. Because the bulk of the Db2 family is now offered on the IBM Cloud Pak for Data platform, either as an add-on or as an included data source service, virtually all of our data can directly be accessed across hybrid cloud or multi-cloud settings to power our AI applications, this is possible because to the fact that AI can now power our AI applications.

Amazon RDS

Amazon Relational Database Service is a managed SQL database service supplied by Amazon Web Services (AWS). RDS is an abbreviation for Amazon Relational Database Service. Amazon RDS supports various database engines to store and organize data. In addition to this, it is capable of supporting actions that are associated with the maintenance of relational databases. These activities include data migration, backup, recovery, and patching. A relational database may be easily set up, managed, and scaled in the cloud with the assistance of Amazon RDS. In addition to automating time-consuming administrative tasks like the provisioning of hardware, the setting up of databases, and the application of patches and backups, it provides affordable and expandable capacity. It frees up more time to focus on our applications, enabling we to give them the necessary rapid response, high availability, security, and compatibility.

PostgreSQL

Free and open-source database administration software PostgreSQL was developed with commercial use in mind. Both relational and non-relational queries, formulated either in SQL or JSON, are supported (Mandapuram & Hosen, 2018). This facilitates flexibility while maintaining SQL compliance. PostgreSQL is a robust object-relational database that is available as open-source software. It has been under continuous development for over 15 years, and thanks to its time-tested architecture, it has earned a strong reputation for dependability, data integrity, and correctness among users. PostgreSQL is supported by all of today's most common operating systems, including Windows, Linux, and UNIX. Oracle and SQL Server are examples of expensive commercial databases that are the only ones that offer advanced data types and performance-enhancing tools. PostgreSQL, on the other hand, comes with these features already built in. Additionally, it is known by the name Postgres.

Apache Cassandra

Apache Cassandra is a distributed database that is extremely scalable and highly functional. It does not have a single point of failure and can handle vast volumes of data across several commodity computers. The NoSQL database family is where it can be found. Apache Cassandra's strength is its ability to manage data in various formats, including organized, semi-structured, and unstructured. Facebook first developed Cassandra; however, in 2008, it was made accessible to the public, and in 2010, it was accepted as one of the highest-level Apache projects. The ability of large organizations to process vast volumes is particularly beneficial to those corporations. As a result, many essential firms, such as Apple, Facebook, and Instagram, are already utilizing it.

CONCLUSION

Despite significant advancements in database management systems, this paper offers an overview of their concepts, applications, and queries for processing. The database-building process involves obtaining, computing, and searching information. The database represents

an idea that has evolved. Implementing database management system concepts enhances hardware and software technology. The advancement of information technology has made labor easier and faster for humans. Rapid growth and development of information technology are expected in the future years worldwide.

REFERENCES

Bodepudi, A., Reddy, M., Gutlapalli, S. S., & Mandapuram, M. (2019). Voice Recognition Systems in the Cloud Networks: Has It Reached Its Full Potential?. *Asian Journal of Applied Science and Engineering*, *8*(1), 51–60. https://doi.org/10.18034/ajase.v8i1.12

Chen, S., Thaduri, U. R., & Ballamudi, V. K. R. (2019). Front-End Development in React: An Overview. *Engineering International*, *7*(2), 117–126. https://doi.org/10.18034/ei.v7i2.662

Chudinov, I. L., Osipova, V. V., Bobrova, Y. V. (2017). The methodology of database design in organization management systems. *Journal of Physics: Conference Series, 803*(1). https://doi.org/10.1088/1742-6596/803/1/012030

DANIEL, O. (2016). THE INTERNET OF THINGS**.** *JOURNAL OF DEMOCRACY, 27*(3), 176-178. HTTPS://DOI.ORG/10.1353/JOD.2016.0042

Dekkati, S., & Thaduri, U. R. (2017). Innovative Method for the Prediction of Software Defects Based on Class Imbalance Datasets. *Technology & Management Review*, *2*, 1–5. https://upright.pub/index.php/tmr/article/view/78

Dekkati, S., Lal, K., & Desamsetti, H. (2019). React Native for Android: Cross-Platform Mobile Application Development. *Global Disclosure of Economics and Business*, *8*(2), 153-164. https://doi.org/10.18034/gdeb.v8i2.696

Deming, C., Dekkati, S., & Desamsetti, H. (2018). Exploratory Data Analysis and Visualization for Business Analytics. *Asian Journal of Applied Science and Engineering*, *7*(1), 93–100. https://doi.org/10.18034/ajase.v7i1.53

Desamsetti, H. (2016a). A Fused Homomorphic Encryption Technique to Increase Secure Data Storage in Cloud Based Systems. *The International Journal of Science & Technoledge*, *4*(10), 151-155.

Desamsetti, H. (2016b). Issues with the Cloud Computing Technology. *International Research Journal of Engineering and Technology (IRJET), 3*(5), 321-323.

Desamsetti, H., & Mandapuram, M. (2017). A Review of Meta-Model Designed for the Model-Based Testing Technique. *Engineering International*, *5*(2), 107–110. https://doi.org/10.18034/ei.v5i2.661

Gutlapalli, S. S. (2016a). An Examination of Nanotechnology's Role as an Integral Part of Electronics. *ABC Research Alert*, *4*(3), 21–27. https://doi.org/10.18034/ra.v4i3.651

Gutlapalli, S. S. (2016b). Commercial Applications of Blockchain and Distributed Ledger Technology. *Engineering International*, *4*(2), 89–94. https://doi.org/10.18034/ei.v4i2.653

Gutlapalli, S. S. (2017a). Analysis of Multimodal Data Using Deep Learning and Machine Learning. *Asian Journal of Humanity, Art and Literature*, *4*(2), 171–176. https://doi.org/10.18034/ajhal.v4i2.658

Gutlapalli, S. S. (2017b). The Role of Deep Learning in the Fourth Industrial Revolution: A Digital Transformation Approach. *Asian Accounting and Auditing Advancement*, *8*(1), 52–56. Retrieved from https://4ajournal.com/article/view/77

Gutlapalli, S. S. (2017c). An Early Cautionary Scan of the Security Risks of the Internet of Things. *Asian Journal of Applied Science and Engineering*, *6*, 163–168. Retrieved from https://ajase.net/article/view/14

Gutlapalli, S. S., Mandapuram, M., Reddy, M., & Bodepudi, A. (2019). Evaluation of Hospital Information Systems (HIS) in terms of their Suitability for Tasks. *Malaysian Journal of Medical and Biological Research*, *6*(2), 143–150. https://mjmbr.my/index.php/mjmbr/article/view/661

Hiroshi, Y., Takeshi, Y., Hiroto, N. (2013). Developing network configuration management database system and its application—data federation for network management. *Telecommunication Systems. New York*, *52*(2), 993-1000. https://doi.org/10.1007/s11235-011-9607-0

Jolana, G., Petra, M. (2018). User-Defined Financial Functions for MS SQL Server. *International Journal of Advanced Computer Science and Applications.* https://doi.org/10.14569/IJACSA.2018.090903

Kashyap, N. K., Pandey, B. K., Mandoria, H. L., Kumar, A. (2016). A Review of Leading Databases: Relational & Non-Relational Database. *i-Manager's Journal on Information Technology*, *5*(2), 34-41.

Kaur, A., Dhindsa, K. S**.** (2016). Performance Evaluation for Crud Operations in NoSQL Databases. *i-manager's Journal on Cloud Computing*, *3*(2), 1-9.

Lal, K. (2015). How Does Cloud Infrastructure Work?. *Asia Pacific Journal of Energy and Environment*, *2*(2), 61-64. https://doi.org/10.18034/apjee.v2i2.697

Lal, K. (2016). Impact of Multi-Cloud Infrastructure on Business Organizations to Use Cloud Platforms to Fulfill Their Cloud Needs. *American Journal of Trade and Policy*, *3*(3), 121–126. https://doi.org/10.18034/ajtp.v3i3.663

Lal, K., & Ballamudi, V. K. R. (2017). Unlock Data's Full Potential with Segment: A Cloud Data Integration Approach. *Technology &Amp; Management Review*, *2*, 6–12. https://upright.pub/index.php/tmr/article/view/80

Lal, K., Ballamudi, V. K. R., & Thaduri, U. R. (2018). Exploiting the Potential of Artificial Intelligence in Decision Support Systems. *ABC Journal of Advanced Research*, *7*(2), 131-138. https://doi.org/10.18034/abcjar.v7i2.695

Liu, X., Zhao, M., Li, S., Zhang, F., Wade, T. (2017). A Security Framework for the Internet of Things in the Future Internet Architecture. *Future Internet*, *9*(3), 27. https://doi.org/10.3390/fi9030027

Mandapuram, M. (2016). Applications of Blockchain and Distributed Ledger Technology (DLT) in Commercial Settings. *Asian Accounting and Auditing Advancement*, *7*(1), 50–57. Retrieved from https://4ajournal.com/article/view/76

Mandapuram, M. (2017a). Application of Artificial Intelligence in Contemporary Business: An Analysis for Content Management System Optimization. *Asian Business Review*, *7*(3), 117–122. https://doi.org/10.18034/abr.v7i3.650

Mandapuram, M. (2017b). Security Risk Analysis of the Internet of Things: An Early Cautionary Scan. *ABC Research Alert*, *5*(3), 49–55. https://doi.org/10.18034/ra.v5i3.650

Mandapuram, M., & Hosen, M. F. (2018). The Object-Oriented Database Management System versus the Relational Database Management System: A Comparison. *Global Disclosure of Economics and Business*, *7*(2), 89–96. https://doi.org/10.18034/gdeb.v7i2.657

Mandapuram, M., Gutlapalli, S. S., Bodepudi, A., & Reddy, M. (2018). Investigating the Prospects of Generative Artificial Intelligence. *Asian Journal of Humanity, Art and Literature*, *5*(2), 167–174. https://doi.org/10.18034/ajhal.v5i2.659

Pokorny, J. (2013). NoSQL databases: a step to database scalability in web environment. *International Journal of Web Information Systems,* 9(1), 69-82. https://doi.org/10.1108/17440081311316398

Prabhjot, P., Neha, S. (2017). Overview of the Database Management System. *International Journal of Advanced Research in Computer Science*, *8*(4).

Sathiy, D.V. V., Shanthini, S., Vinora, A., Mohana, G. P. (2017). An Overview of Database Management Systems and their Applications, along with the queries for processing the System. *SSRG International Journal of Computer Science and Engineering*, 4(3), 1-4. https://doi.org/10.14445/23488387/IJCSE-V4I3P101

Snehal B. S., Prajakta P.C. (2015). Cloud Database Management System (CDBMS). *Compusoft. 4*(1), 1462-1466.

Thaduri, U. R., & Lal, K. (2020). Making a Dynamic Website: A Simple JavaScript Guide. *Technology & Management Review*, *5*, 15–27. https://upright.pub/index.php/tmr/article/view/81

Thaduri, U. R., Ballamudi, V. K. R., Dekkati, S., & Mandapuram, M. (2016). Making the Cloud Adoption Decisions: Gaining Advantages from Taking an Integrated Approach. *International Journal of Reciprocal Symmetry and Theoretical Physics*, *3*, 11–16. https://upright.pub/index.php/ijrstp/article/view/77

Thodupunori, S. R., & Gutlapalli, S. S. (2018). Overview of LeOra Software: A Statistical Tool for Decision Makers. *Technology & Management Review, 3*(1), 7–11.

Vasin, Y. G., Yasakov, Y. V. (2016). Object-oriented topological management system of spatially-distributed databases. *Pattern Recognition and Image Analysis,* 26(4), 734-741. https://doi.org/10.1134/S1054661816040180

--0--

Online Archive:

https://i-proclaim.my/journals/index.php/gdeb/issue/archive