Shavon Parker
10/8/2021
A590
Jie Wang

Writing Assignment 2

Compare the database types in terms of popularity, performance, speed and data latency, security, system compatibility, scalability, and capability of handling big data.

There are around 11 different types of databases that exist today. They are centralized, distributed, NoSQL, relational, cloud, network, object-oriented, and hierarchical.

A centralized database is exactly like it sounds; it stores data in a centralized way. This means for example a central library that carries a central database of each library in a college [1]. The advantages of this type of database are that it has a decreased risk of data management [1]. By using this type of database provides better data quality, this enables organizations to establish data standards [1]. In relation to speed and data latency this database isn't so ideal because of the size of the centralized database is large, this increases the response time for fetching the data [1]. As far as security goes, the centralized database applications contain the authentication process to let users access data securely [1]. Another challenge this type of database has that it is not very easy to update a database of the extensive size as the centralized database [1].

The distributed database is different from the centralized database because data is distributed among different database systems of an organization [1]. The distributed database can be divided into two subsections: homogeneous and heterogeneous [1]. Homogeneous are those database systems which execute on the same operating system and use the same application process and carry the same hardware devices [1]. Heterogeneous are those database systems which execute on different operating systems under different application procedures and carries different hardware devices [1]. In relation to speed this database is amazing because it can be broken down where if one server was to fail it would not affect the entire data set [1]. Another benefit is that this database can be easily updated and expanded because of the system being distributed system, it can run independent to one another [1].

The relational database is based on the relational data model, which stores data in the form of rows also know as tuple, and columns known as attributes [1]. Relational databases use SQL for storing, manipulating, as well as maintaining the data, each table in the database carries a key that makes the data unique from others [1]. The relational database holds to four commonly known properties known as ACID which stands for Atomicity, Consistency, Isolation, and Durability [1].

The NoSQL database is the type of database that is used for storing a wide range of data sets [1]. A NoSQL database should not be mistaken for a relational database simply because it stores data not only in tabular form but in several different ways [1]. The NoSQL database can be divided into four subsection types: key-value, graph, document-oriented, and wide-column. Key-value storage is the simplest type of database storage, it stores every single item as a key holding its value, together [1]. Graph storage is used for storing vast amounts of data in a graph-like structure, this is mostly used with social networking websites [1]. Next there is document-oriented is used to store data as JSON-like document, it also helps developers in storing data by using the same document-model format as used in the application code [1]. Finally wide-column storage is similar to the data represented in relational databases because data is stored in large columns together, instead of storing in rows [1]. This type of database is good for productivity in the application development because is not required to store data in a structured format [1]. This type of database is the better option for managing and handling large data sets [1].

The cloud database is where data is stored in a virtual environment and executes over the cloud computing platform [1]. There are numerous cloud platforms and some of the best- and well-known options are: Amazon Web Services, Microsoft Azure, Kamatera, PhonixNAP, ScienceSoft, and Google Cloud SQL [1]. Since this database uses various cloud computing services it is the optimal choice if you have an excessive number of users trying to access the database at any given time.

Object-oriented databases use the object-based data model approach for storing data in the database system, the data is represented and stored as objects which are similar to the objects used in the object-oriented programming language [1]. The data is represented and stored in the form of objects [2]. These objects have members such as fields, properties, and methods, objects also have a life cycle that includes the creation of an object, use of an object, and deletion of an object [2].

Object based databases bring permanent persistence to objects, and objects can be store in this way forever [2]. Since object-oriented databases uses a layer of object-relational mapping that maps database schemas with object in code, reading and mapping an object database data to the objects is direct without any API or OR tool, which equates to faster data access and better performance [2]. Some common and popular object databases are Cache, ConceptBase, Db4o, ObjectDB Object Database, ObjectDatabase++, Objectivity/DB, ObjectStore, Versant Object Database, WakandaDB, and Object-relational databases [2].

Next is the hierarchical database which stores data in the form of parent-children relationship nodes, it organizes data in a tree-like structure [1]. The tree-like structure is composed in a way where there is root data and then at the next level the second level and the third layer are subsections of the previous branching off of it [1]. Each child of the tree can only have one parent; however, each parent record can have multiple children [1].

Network databases typically follows the network data model, the representation of data is in the form of nodes connected via links between them [1]. It is different from hierarchical database because it allows each record to have multiple children and parent nodes (What Are Object-Oriented Databases And Their Advantages, 2019) to form a generalized graph structure [1].

Personal databases collect and store data on the user’s system defines a personal database, it is basically designed for a single user [1]. This database is ideal because it is simple and easy to handle, as well as it occupies less storage space because of how small it is [1]. This database as you can imagine isn’t designed to accommodate multiple users on a database [1].

Personal databases work in a way where they support one application, they have fewer tables, involve only one computer, and usually have a simple design [3]. Some advantages include fast processing because it is based on a local computer the data can be processed faster and more reliable in terms of handling [3]. Another advantage is higher security this is possible because the data is stored on a personal computer and the data does not need any special security arrangement for authorization of data [3].

Operational database creates and updates the database in real-time, basically designed for executing and handling the daily data operations in several businesses [1]. Most organizations use operational databases for managing day per day transactions [1]. Elements in a operational database can be added and removed on the fly, these databases can be either SQL or NoSQL-based, where the latter is geared toward real time operations [4]. Key characteristics of operational databases is their orientation toward real-time operations, compared with conventional databases that rely on batch processing [4].

Enterprise databases are used by large organizations or enterprises for managing a massive amount of data [1]. It helps those organizations to increase and improve their efficiency, letting multiple users access the database simultaneous [1]. Benefits of this type of database includes multi processes are being supported over the enterprise database and it allows executing parallel queries on the system [1].

Compare column-based and row-based database.

There are two ways to organize relational databases by row oriented or column oriented. The first style I will discuss is row-oriented databases. Row oriented databases are databases that organize data by record, keeping all of the data associated with a record next to each other in memory [5]. This type is the traditional way of organizing data and still provide some key benefits for storing data quickly, optimized for reading and writing rows efficiently [5]. The most common databases that use row orientation is Postgres and MySQL [5].

In a row-oriented database, the data is stored row by row, such that the first column of a row will be next to the last column of the previous row [5]. Data written in this way allows the database to write a row quickly because all that needs to be done to write to it is to tack on another row to the end of the data [5].

Row oriented databases are still commonly used for online transactional processing style applications since they can manage writes to the database as well [5]. For the online analytical processing use cases need a database that can support ad hoc querying of the data [5]. Row oriented databases are fast at retrieving a row or a set of rows but when performing an aggregation, it brings extra data into memory which is slower than only selecting the columns that you are performing the aggregation on, the number of disks the row-oriented database might need to access is usually larger [5]. Adding data to a row-oriented database is quick and easy, getting data out of it can require extra memory to be used and multiple disks to be accessed [5].

In row-oriented databases, indexes can be created but data is rarely stored in multiple sort orders.

The next style of databases are column-oriented databases. Column oriented databases are databases that organize data by field, keeping all of the data associated with a field next to each other in memory [5]. Columnar databases have grown in popularity and provide performance advantages to querying data, optimized for reading and computing on columns efficiently [5]. Some common column-oriented databases are Redshift, BigQuery, and Snowflake [5].

Since data warehouses were created to support the act of analyzing data, those types of databases are read optimized [5]. In a C-Store, columnar, or column-oriented database, the data is stored such that each row of a column will be next to other rows from that same column [5]. If data is stored on a single disk, then the column oriented database will have the same problem as the row oriented database, column oriented databases will have significant benefits when stored on separate disks [5]. The writable store has the data sorted in the order it was added, to make adding data into it easier. We can easily append the relevant fields to our database [5].

When reading for a column stored database to get the sum of the ages the computer only needs to go to one disk and sum all the values inside of it, no extra memory needs to be pulled in and it accesses a minimal number of disks [5].

When encoding with run-length encoding, it allows you to replace any sequence of the same value with a count and value indicator [5]. This type of encoding becomes even more powerful when you create projections with columns that are sorted since all values that are the same are next to each other [5].

In column-oriented databases you can have the data stored in an arbitrary number of ways, there are benefits beyond query performance [5]. These differences sort ordered columns are referred to as projections and they allow the system to be more fault tolerant, since the data is stored multiple times [5]. Read optimized store can also have multiple projections, it also has tuple mover which manages the relevant updates because of this it has to navigate the multiple projections and insert the data in the proper places [5]. This architecture means that while the data is being updated and partially added data must be ignored by queries until the update is complete [5].

References

[1] Types of Databases. (n.d.). Retrieved from javaTpoint: https://www.javatpoint.com/types-of-databases

[2] What Are Object-Oriented Databases And Their Advantages. (2019, September 6). Retrieved from C# Corner: https://www.c-sharpcorner.com/article/what-are-object-oriented-databases-and-their-advantages2/

[3] Akhtar, Z. (2021, June 21). *Personal Database Functions, Advantages & Disadvantages*. Retrieved from DatabaseTown: https://databasetown.com/personal-database-functions-advantages/

[4] Operational Database (ODB). (n.d.). Retrieved from techopedia: https://www.techopedia.com/definition/5711/operational-database-odb

[5] Blake Barnhill, M. D. (2021, August 9). *Row vs Column Oriented Databases*. Retrieved from Data School: https://dataschool.com/data-modeling-101/row-vs-column-oriented-databases/