Databases

Concepts

Nexart store there daily generated data in a a database before processing it in a database. Organized data that have been stored as soft copies in a computer is defined as a database. In order to manage a database, a Database Management System is required.

They are categorized into two categories and they are Relational, Non-Relational, NewSQL, InMemory, Time-Series and Object oriented Databases. NewSQL database is considered as an enhanced version of a relational database. Scalability acts as the icing for these databases. Google Spanner and CockraochDB are examples for NewSQL databases. Main memory is used by Memory databases to store data. Perfect instances for NewSQL databases are Redis and Memcached. Data that is generated by IoT devices and financial transactional data are stored in Time-Series databases and the data are time stamped. InfluxDB and TimeScaleDB are examples for TimeSeries databases. Data are stored as objects in Object-Oriented databases.db40 and ObjectDB are instances for Object-Oriented Databases.

Single-Tier, Two-Tier, Three-Tier and N-Tier are database Architectures. In a Single-Teir architecture the database and application are hosted in the same machine and these architectures are deployed in development/testing environments. A direct communication between the client and the server is maintained in a Two-Tier architecture and it’s applications are applications which falls in the range of small to medium. An intermediary layer is added to enhance the capabilities of a Three-Teir architecture and they are used in web applications, enterprise applications etc. As an enhancement Three-Tier designs add an intermediary layer. N-Tier architecture is used to address complex commercial issues that are related to databases. This architecture surpasses the the third architecture by adding more layers. Mostly conglomerate applications use N-Tier architecture for their systems.

The components of the database are Database Engine, Database Schema, Query Processor, Storage Manager, Transaction Manager, Concurrency Control Manager and Recovery Manager. The Database Engine is responsible for accessing and managing the database and this service is considered as a core service. The Logical structure of a database is defined as the Logical Schema of a database and it includes tables, views etc. The mechanism which is responsible for interpreting and executing queries is the Query Selector. The storage manager plays the role of managing the disk space. The transaction manager make sure that all the transactions happen according to the properties of ACID. The consistency of simultaneous data access are managed by the concurrency control manager. When a database failure occurs, the recover manager restores the database back to its normal condition.

According to the design aspect of databases, they can be divided into three categories and they are Conceptual Design, Logical Design and Physical Design. A perfect example for a conceptual design would be entity relationship diagrams. Theses diagrams illustrates the data model. The Logical Design demonstrates the database in a form of tables,views etc.

In order avoid redundancies and enhance data integrity normalization is carried out. 1st, 2nd, 3rd and 4th normalization are examples for normalization types.

The data are efficiently retrieved by using the database indexing mechanism. Primary, secondary, clustered and non-clustered indexing are instances for database indexes. Authentication, Authorization, Encryption and Backup and Recovery mechanisms take care of the security just in case of a security breach. Authentication is responsible for verifying the identity of the user while the actions that the verified user can perform is granted by authorization. Encryption mechanism encrypt the data using hashing algorithms before storing or transporting them in the database. Backup and recovery mechanisms make sure that the data can restored back to its normal condition in case of data loss or corruption. Full backup, incremental backup and differential backup are examples for database backup methods. The entire database is copied to a storage device in full backup method. Incremental backups copies the recently altered data to a storage device. The data changed since the last full backup is copied to a storage device in differential backup method. Restoring data to a certain point in time is defined as point-in-time recovery. Recovering recently committed data by examining the transaction logs are part of the log based recovery method. Cloud, Distributed and Graph databases are considered as modern trends of databases.

The accurate processing of database transactions is assured by the properties of ACID and they atomcity, consistency, isolation and durability. A transaction is treated as a one unit in Atomocity while all the standards are maintained under consistency. Isolation mechanisum compartmentalize each transaction until they terminated. Fault tolerances in transactions are ensured by the durability property.The next section draws out the advantages and disadvantages of relational and non-relational databases.

Relational and Non-Relational Databases

Structured data are stored in relational databases using columns and rows while semi-structured and unstructured data are stored non-relational databases. Structured Query Language is used by relational databases to perform all the database operations while Not Structured Query Language is used by non-relational databases to perform all the database operations. MySQL, PostgresSQL, Oracle and SQL Server are examples for relational databases. There are four types of non-relational databases and they are Document Stores, Key-Value Stores, Column Stores and Graph Databases. MongoDB and CouchDB are examples for Document Stores while Redis and DynamoDB are instances for Key-Value Stores. Cassandra and HBase are examples for column store nosql databases. Neo4j and ArangoDB are best examples for graph database. The table below higlights main differences between SQL and NOSQL.

|  |  |
| --- | --- |
| SQL | NOSQL |
| Used in relational databases | Used in non-relational databases |
| Has static schema design and structure | Has dynamic schema design and structure |
| Handle complex queries | Handle large volumes of data |
| Equipped with vertical scalability | Equipped with horizontal scalability |
| Follows ACID property | Follows CAP property |

Relational and non-relational databases such as MSSQL and MongoDB have ACID in common. Though a transaction is complete or rolled back, atmocity ensures that the database is maintained in a consistent state in MSSQL. Atomocity is applied in a single document update process in MongoDB. Constraints, triggers and rule mechanisms ensure that consistency is maintained in MSSQL. Consistency is applied for multi-documents in MongoDB and it is offered via flexible schema design which enables read and write operations. Transparency of uncommitted changes to other transactions is maintained via isolation mechanism in MSSQL. Snapshot mechanism in MongoDB ensures isolation in multi-document transactions. Transactions logs and database backups ensure that durability is maintained in MSSQL while journaling and replica sets ensures durability In MongoDB. The benefits and drawbacks of relational and non-relational databases are discussed in the next section.

Strength and Weaknesses of Relational and Non-Relational Databases

The table given below spotlights the pros and cons of relational databases.

|  |  |
| --- | --- |
| Advantages | Disadvantages |
| Provides a structure to manage data | Limited Scalability |
| Compatible with ACID | Performance issues |
| Extensive Query Capabilities | Difficulty in changing the schema |
| Avoiding data redundancy and enhancing data integrity | High Cost |
| Capability of mapping complex relationships between entities | Capability of mapping complex relationships between entities |
| Technical Support | Complexity in applications |
| Security | Unability of handling datasets that scaling. |

The advantages and disadvantages of non-relational databases are emphasized in the table given below.

|  |  |
| --- | --- |
| Advantages | Disadvantages |
| Scalability | Not Consistent |
| Availability | Consistency Issues |
| Higer Performance | Limited Query Capabilities |
| Flexibility | ACID not supported |
| Handling of structured and unstructured data | Limited support |
| Low Expenditure | Complexity in terms of data modeling |

We have already acknowledged the benefits and drawbacks of relational and non-relational database. Next step would be to select the right database type for your application and the next chapter addresses about it.

Right Database Option for an application

Applications use relational or non-relational databases to store data according to the requirements of the project. As we acknowledged before relational and non-relational databases use SQL and NOSQL to perform all database operations. If an application has structured data with a static schema and requires multirow transactions, it is advisable to deploy a relational database such as MSSQL. The developers still can go with non-relational database but it can limit the functionality of the application sometimes. It may take a toll on the project budget sometimes. If a solution has a dynamic schema and holds semi-structured and unstructured data, it would be an ideal to use a non-relational database such as MongoDB. The characteristics of a non-relational database supports an enterprise application which is scaling in a rapid rate. A relational database can be used for this application but it will downgrade the capabilities of the application even if the option is less costly. NOSQL databases are ideal for solutions which require a hierarchical storage structures, scalable features and functions and where relationships between entities are not important. Maintenance of a database is important for an application to run smoothly without causing any issues. Operations such importing and exporting data occur in the management phase. Hence these tasks are repetitive, an automation is required to boost up the productivity of the system. The next chapter suggests an enhancement to improve efficiency of the Nexart system.