Unit-4
Introduction to Database management system

4.1 Database: In the digital age, data plays a crucial role in every aspect of our lives. From managing business information to tracking personal activities, the need for efficient and organized data storage is more significant than ever. This is where databases come into play. In this chapter, we'll delve into the fundamentals of databases, exploring their definition, types, and the key components that make them an indispensable part of modern computing.

4.1.1 What is a Database?

Data is a collection of a distinct small unit of information. It can be used in a variety of forms like text, numbers, media, bytes, etc. it can be stored in pieces of paper or electronic memory, etc. Data is information that can be translated into a form for efficient movement and processing. Data is interchangeable.

A database is a structured collection of data that is organized and stored in a way that allows easy access, retrieval, and management. Unlike traditional methods of data storage, such as spreadsheets or text files, databases provide a systematic and efficient way to handle large volumes of data. They are designed to support data integrity, security, and scalability.

4.1.2 Types of Databases

There are various types of databases used for storing different varieties of data:

1) Centralized Database

It is the type of database that stores data at a centralized database system. It comforts the users to access the stored data from different locations through several applications. These applications contain the authentication process to let users access data securely. An example of a Centralized database can be Central Library that carries a central database of each library in a college/university.

Advantages of Centralized Database

- It has decreased the risk of data management, i.e., manipulation of data will not affect the core data.
- Data consistency is maintained as it manages data in a central repository.
- It provides better data quality, which enables organizations to establish data standards.

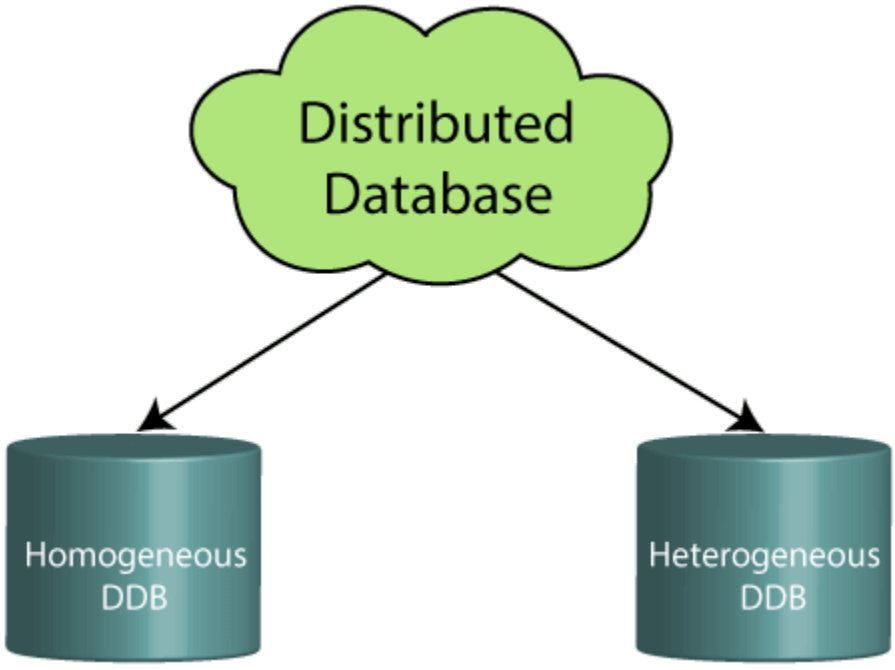
- It is less costly because fewer vendors are required to handle the data sets.

Disadvantages of Centralized Database

- The size of the centralized database is large, which increases the response time for fetching the data.
- It is not easy to update such an extensive database system.
- If any server failure occurs, entire data will be lost, which could be a huge loss.

2) Distributed Database

Unlike a centralized database system, in distributed systems, data is distributed among different database systems of an organization. These database systems are connected via communication links. Such links help the end-users to access the data easily. **Examples** of the Distributed database are Apache Cassandra, HBase, Ignite, etc.



- **Homogeneous DDB:** Those database systems which execute on the same operating system and use the same application process and carry the same hardware devices.
- **Heterogeneous DDB:** Those database systems which execute on different operating systems under different application procedures, and carries different hardware devices.

Advantages of Distributed Database

- Modular development is possible in a distributed database, i.e., the system can be expanded by including new computers and connecting them to the distributed system.
- One server failure will not affect the entire data set.

3) Relational Database

This database is based on the relational data model, which stores data in the form of rows(tuple) and columns(attributes), and together forms a table(relation). A relational database uses SQL for storing, manipulating, as well as maintaining the data. E.F. Codd invented the database in 1970. Each table in the database carries a key that makes the data unique from others. **Examples** of Relational databases are MySQL, Microsoft SQL Server, Oracle, etc.

Properties of Relational Database

There are following four commonly known properties of a relational model known as ACID properties, where:

A means Atomicity: This ensures the data operation will complete either with success or with failure. It follows the 'all or nothing' strategy. For example, a transaction will either be committed or will abort.

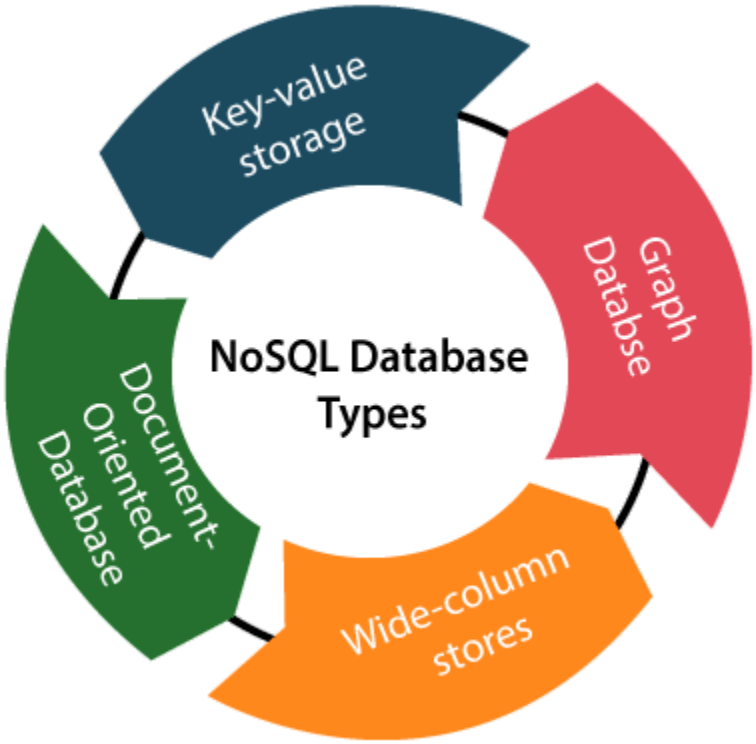
C means Consistency: If we perform any operation over the data, its value before and after the operation should be preserved. For example, the account balance before and after the transaction should be correct, i.e., it should remain conserved.

I means Isolation: There can be concurrent users for accessing data at the same time from the database. Thus, isolation between the data should remain isolated. For example, when multiple transactions occur at the same time, one transaction effects should not be visible to the other transactions in the database.

D means Durability: It ensures that once it completes the operation and commits the data, data changes should remain permanent.

4) NoSQL Database

Non-SQL/Not Only SQL is a type of database that is used for storing a wide range of data sets. It is not a relational database as it stores data not only in tabular form but in several different ways. It came into existence when the demand for building modern applications increased. Thus, NoSQL presented a wide variety of database technologies in response to the demands. We can further divide a NoSQL database into the following four types:



a. **Key-value storage:** It is the simplest type of database storage where it stores every single item as a key (or attribute name) holding its value, together.
b. **Document-oriented Database:** A type of database used to store data as JSON-like document. It helps developers in storing data by using the same document-model format as used in the application code.
c. **Graph Databases:** It is used for storing vast amounts of data in a graph-like structure. Most commonly, social networking websites use the graph database.
d. **Wide-column stores:** It is similar to the data represented in relational databases. Here, data is stored in large columns together, instead of storing in rows.

Advantages of NoSQL Database

- It enables good productivity in the application development as it is not required to store data in a structured format.
- It is a better option for managing and handling large data sets.
- It provides high scalability.
- Users can quickly access data from the database through key-value.

5) Cloud Database

A type of database where data is stored in a virtual environment and executes over the cloud computing platform. It provides users with various cloud computing services (SaaS, PaaS, IaaS, etc.) for accessing the database. There are numerous cloud platforms, but the best options are:

- Amazon Web Services(AWS)
- Microsoft Azure
- Kamatera
- PhonixNAP

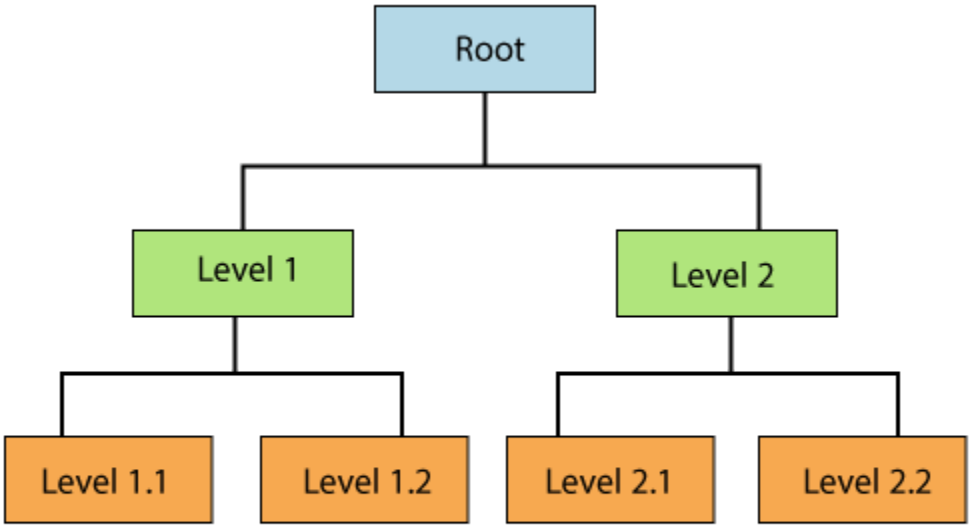
- ScienceSoft
- Google Cloud SQL, etc.

6) Object-oriented Databases

The type of database that uses 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.

7) Hierarchical Databases

It is the type of database that stores data in the form of parent-children relationship nodes. Here, it organizes data in a tree-like structure.



Hierarchical Database

Data get stored in the form of records that are connected via links. Each child record in the tree will contain only one parent. On the other hand, each parent record can have multiple child records.

8) Network Databases

It is the database that typically follows the network data model. Here, the representation of data is in the form of nodes connected via links between them. Unlike the hierarchical database, it allows each record to have multiple children and parent nodes to form a generalized graph structure.

9) Personal Database

Collecting and storing data on the user's system defines a Personal Database. This database is basically designed for a single user.

Advantage of Personal Database

- It is simple and easy to handle.
- It occupies less storage space as it is small in size.

10) Operational Database

The type of database which creates and updates the database in real-time. It is basically designed for executing and handling the daily data operations in several businesses. For example, An organization uses operational databases for managing per day transactions.

11) Enterprise Database

Large organizations or enterprises use this database for managing a massive amount of data. It helps organizations to increase and improve their efficiency. Such a database allows simultaneous access to users.

Advantages of Enterprise Database:

- Multi processes are supportable over the Enterprise database.
- It allows executing parallel queries on the system.

4.1.3 Key Components of a Database

A database is a structured collection of data that is organized and managed in a way that facilitates efficient retrieval, storage, and manipulation. The key components of a database system include:

1. Data:
    a. Raw Facts: The actual information that is stored in the database. It can include text, numbers, dates, images, and more.
    b. Entities: These are the objects or things that the database is designed to track. For example, in a database for a library system, entities could include books, borrowers, and transactions.
2. Database Management System (DBMS):

    Software: The software that provides an interface for interacting with the database. It manages tasks such as data storage, retrieval, and security. Examples include MySQL, Oracle, Microsoft SQL Server, and PostgreSQL.

3. Tables:

    Structure: Data is organized into tables, which represent entities. Each table consists of rows and columns. Rows represent individual records, and columns represent attributes or fields.

4. Fields (Attributes):

    Columns: These are individual data elements within a table that represent the attributes of the entity. For example, in a "Customers" table, columns could include "CustomerID," "FirstName," "LastName," etc.

5. Records (Rows):

   Tuples: Each row in a table represents a record or a tuple. It contains data for each attribute defined by the columns. In a "Customers" table, each row would represent information about a specific customer.

6. Keys:
   a. Primary Key: A unique identifier for each record in a table. It ensures that each record can be uniquely identified and helps in establishing relationships between tables.
   b. Foreign Key: A field in one table that refers to the primary key in another table. It establishes a link between the two tables.
7. Relationships:

   Connections: Relationships define how tables are related to each other. Common types include one-to-one, one-to-many, and many-to-many relationships.

8. Indexes:

   Optimization: Indexes improve the speed of data retrieval operations on a database. They are created on one or more columns of a table, allowing the database management system to locate and retrieve data more quickly.

9. Queries:

   SQL: Structured Query Language is used to interact with and retrieve data from the database. Users and applications can use queries to perform various operations like selecting, updating, inserting, or deleting data.

10. Integrity Constraints:

    Rules: These are rules that ensure the consistency and accuracy of data in the database. Examples include unique constraints, check constraints, and referential integrity.

11. Transactions:

    Atomicity, Consistency, Isolation, Durability (ACID): These are properties that ensure the reliability of database transactions. Transactions are sequences of one or more database operations that are executed as a single unit.

12. Views:

Virtual Tables: Views are virtual tables based on the result of a SELECT query. They allow users to access and manipulate data without directly modifying the underlying tables.

These components work together to provide a structured and efficient way to store, retrieve, and manage data in a database system

4.2 DBMS: A Database Management System is software that facilitates the creation, maintenance, and use of databases. DBMS stands for Database Management System. It is software that provides an interface for interacting with databases and is designed to manage, organize, and retrieve data stored in a database. The primary functions of a DBMS include:

1. **Data Definition:** This involves defining the structure of the data to be stored in the database. It includes creating tables, specifying data types, and defining relationships between tables.
2. **Data Manipulation**: DBMS allows users to insert, update, delete, and retrieve data from the database. This is typically done using SQL (Structured Query Language) commands.
3. **Data Retrieval:** Users can query the database to retrieve specific information based on certain criteria. SQL is also commonly used for this purpose.
4. **Data Security**: DBMS provides mechanisms to control access to the database, ensuring that only authorized users can view or modify certain data.
5. **Data Integrity:** DBMS enforces integrity constraints to ensure the accuracy and consistency of data. This includes constraints such as unique keys, primary keys, foreign keys, etc.
6. **Concurrency Control:** DBMS manages access to the database by multiple users simultaneously, ensuring that transactions are executed in a way that maintains data consistency.
7. **Backup and Recovery:** DBMS includes features to backup the database periodically and recover data in case of system failures or errors.
8. **Data Independence**: DBMS provides a level of abstraction between the physical storage of data and the way users perceive the data. This allows changes to the database structure without affecting the applications that use the data.

There are various types of DBMS, including relational database management systems (RDBMS), object-oriented database management systems (OODBMS), and NoSQL databases, each designed to handle different types of data and applications. Some popular DBMSs include MySQL, PostgreSQL, Oracle Database, Microsoft SQL Server, and MongoDB.

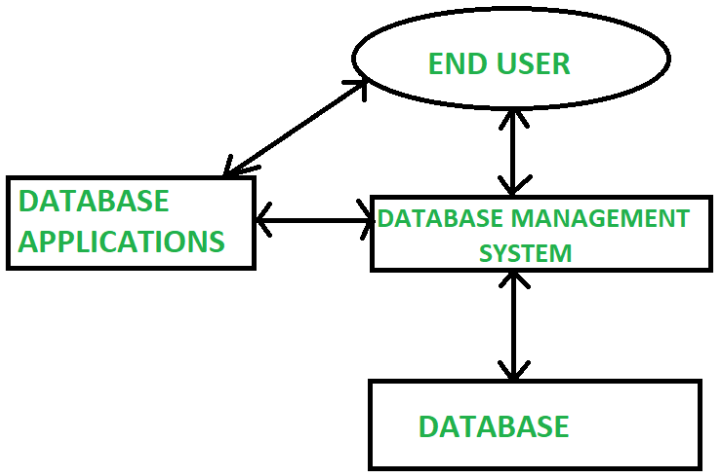


Figure: Block diagram of DBMS

4.3 Why Database-File System Vs DBMS: Databases are crucial for several reasons:

1. Data Organization: Databases provide a structured way to organize vast amounts of information. They arrange data into tables, relationships, and schemas, allowing for efficient storage and retrieval.
2. Data Integrity and Consistency: Databases ensure data accuracy, consistency, and integrity through mechanisms like constraints, validations, and normalization. This maintains reliable and trustworthy information.
3. Data Accessibility: Databases facilitate easy and quick access to information. With optimized query processing, users can retrieve specific data without sifting through volumes of unorganized data.
4. Concurrency Control: They manage simultaneous access to data by multiple users, ensuring that transactions occur reliably without interfering with each other.
5. Security: Databases offer various security features, including user authentication, access control, encryption, and audit trails, safeguarding sensitive information from unauthorized access or tampering.
6. Scalability: As data grows, databases can scale to accommodate larger volumes without compromising performance, ensuring that systems can handle increasing amounts of information.
7. Data Recovery and Backup: They provide mechanisms for data backup and recovery, safeguarding against data loss due to system failures, human errors, or disasters.
8. Decision Making: Databases enable data-driven decision-making processes by providing relevant, accurate, and timely information for analysis, reporting, and forecasting.
9. Support for Applications: Many applications rely on databases to store and retrieve information. Whether it's a banking system, an e-commerce platform, or a healthcare application, databases serve as the backbone for these systems.

10. Regulatory Compliance: Databases help organizations comply with various regulations regarding data storage, privacy, and security by implementing measures to adhere to legal requirements.

Difference between File System and DBMS

Basics	File System	DBMS
Structure	The file system is a way of arranging the files in a storage medium within a computer.	DBMS is software for managing the database.
Data Redundancy	Redundant data can be present in a file system.	In DBMS there is no redundant data.
Backup and Recovery	It doesn't provide Inbuilt mechanism for backup and recovery of data if it is lost.	It provides in house tools for backup and recovery of data even if it is lost.
Query processing	There is no efficient query processing in the file system.	Efficient query processing is there in DBMS.

Consistency	There is less data consistency in the file system.	There is more data consistency because of the process of normalization.
Complexity	It is less complex as compared to DBMS.	It has more complexity in handling as compared to the file system.
Security Constraints	File systems provide less security in comparison to DBMS.	DBMS has more security mechanisms as compared to file systems.
Cost	It is less expensive than DBMS.	It has a comparatively higher cost than a file system.

Data Independence	There is no data independence.	In DBMS data independence exists, mainly of two types: 1) Logical Data Independence. 2)Physical Data Independence.
User Access	Only one user can access data at a time.	Multiple users can access data at a time.
Meaning	The users are not required to write procedures.	The user has to write procedures for managing databases
Sharing	Data is distributed in many files. So, it is not easy to share data.	Due to centralized nature data sharing is easy
Data Abstraction	It give details of storage and representation of data	It hides the internal details of Database

Integrity Constraints	Integrity Constraints are difficult to implement	Integrity constraints are easy to implement
Attributes	To access data in a file , user requires attributes such as file name, file location.	No such attributes are required.
Example	Cobol, C++	Oracle, SQL Server

The main difference between a file system and a DBMS (Database Management System) is the way they organize and manage data:

1. File systems are used to manage files and directories, and provide basic operations for creating, deleting, renaming, and accessing files. They typically store data in a hierarchical structure, where files are organized in directories and subdirectories. File systems are simple and efficient, but they lack the ability to manage complex data relationships and ensure data consistency.
2. On the other hand, DBMS is a software system designed to manage large amounts of structured data, and provide advanced operations for storing, retrieving, and manipulating data. DBMS provides a centralized and organized way of storing data, which can be accessed and modified by multiple users or applications. DBMS offers advanced features like data validation, indexing, transactions, concurrency control, and backup and recovery mechanisms. DBMS ensures data consistency, accuracy, and integrity by enforcing data constraints, such as primary keys, foreign keys, and data types.

In summary, file systems are suitable for managing small amounts of unstructured data, while DBMS is designed for managing large amounts of structured data, and offers more advanced features for ensuring data integrity, security, and performance.

4.4 Database Applications:

- **Telecom:** There is a database to keeps track of the information regarding calls made, network usage, customer details etc. Without the database systems it is hard to maintain that huge amount of data that keeps updating every millisecond.
- **Industry**: Where it is a manufacturing unit, warehouse or distribution centre, each one needs a database to keep the records of ins and outs. For example distribution centre should keep a track of the product units that supplied into the centre as well as the products that got delivered out from the distribution centre on each day; this is where DBMS comes into picture.
- **Banking System**: For storing customer info, tracking day to day credit and debit transactions, generating bank statements etc. All this work has been done with the help of Database management systems. Also, banking system needs security of data as the data is sensitive, this is efficiently taken care by the DBMS systems.
- **Sales:** To store customer information, production information and invoice details. Using DBMS, you can track, manage and generate historical data to analyse the sales data.
- **Airlines**: To travel though airlines, we make early reservations, this reservation information along with flight schedule is stored in database. This is where the real-time update of data is necessary as a flight seat reserved for one passenger should not be allocated to another passenger, this is easily handled by the DBMS systems as the data updates are in real time and fast.
- **Education sector:** Database systems are frequently used in schools and colleges to store and retrieve the data regarding student details, staff details, course details, exam details, payroll data, attendance details, fees details etc. There is a large amount of inter-related data that needs to be stored and retrieved in an efficient manner.
- **Online shopping:** You must be aware of the online shopping websites such as Amazon, Flipkart etc. These sites store the product information, your addresses and preferences, credit details and provide you the relevant list of products based on your query. All this involves a Database management

system. Along with managing the vast catalogue of items, there is a need to secure the user private information such as bank & card details. All this is taken care of by database management systems.

4.5 Database Users: Database users are categorized based on their interaction with the database. These are seven types of database users in DBMS.

1. **Database Administrator (DBA) :** Database Administrator (DBA) is a person/team who defines the schema and also controls the 3 levels of the database. The DBA will then create a new account id and password for the user if he/she needs to access the database. DBA is also responsible for providing security to the database and he allows only the authorized users to access/modify the database. DBA is responsible for the problems such as security breaches and poor system response time.
    - DBA also monitors the recovery and backup and provide technical support.
    - The DBA has a DBA account in the DBMS which called a system or superuser account.
    - DBA repairs damage caused due to hardware and/or software failures.
    - DBA is the one having privileges to perform DCL (Data Control Language) operations such as GRANT and REVOKE, to allow/restrict a particular user from accessing the database.
2. **Naive / Parametric End Users :** Parametric End Users are the unsophisticated who don't have any DBMS knowledge but they frequently use the database applications in their daily life to get the desired results. For examples, Railway's ticket booking users are naive users. Clerks in any bank is a naive user because they don't have any DBMS knowledge but they still use the database and perform their given task.
3. **System Analyst :**

   System Analyst is a user who analyzes the requirements of parametric end users. They check whether all the requirements of end users are satisfied.
4. **Sophisticated Users :** Sophisticated users can be engineers, scientists, business analyst, who are familiar with the database. They can develop their own database

applications according to their requirement. They don't write the program code but they interact the database by writing SQL queries directly through the query processor.

5. **Database Designers :** Data Base Designers are the users who design the structure of database which includes tables, indexes, views, triggers, stored procedures and constraints which are usually enforced before the database is created or populated with data. He/she controls what data must be stored and how the data items to be related. It is responsibility of Database Designers to understand the requirements of different user groups and then create a design which satisfies the need of all the user groups.
6. **Application Programmers :** Application Programmers also referred as System Analysts or simply Software Engineers, are the back-end programmers who writes the code for the application programs. They are the computer professionals. These programs could be written in Programming languages such as Visual Basic, Developer, C, FORTRAN, COBOL etc. Application programmers design, debug, test, and maintain set of programs called "canned transactions" for the Naive (parametric) users in order to interact with database.
7. **Casual Users / Temporary Users :** Casual Users are the users who occasionally use/access the database but each time when they access the database they require the new information, for example, Middle or higher level manager.
8. **Specialized users :** Specialized users are sophisticated users who write specialized database application that does not fit into the traditional data-processing framework. Among these applications are computer aided-design systems, knowledge-base and expert systems etc.

4.6 Introduction to SQL:

SQL, or Structured Query Language, is a programming language designed for managing and manipulating relational databases. It serves as a standard interface for interacting with relational

database management systems (RDBMS) and is widely used for tasks such as querying data, updating records, inserting new data, and managing database structures. SQL provides a set of powerful and standardized commands that allow users to communicate with databases and perform various operations.

Here's a brief introduction to some fundamental aspects of SQL:

Purpose of SQL:

- SQL is primarily used to interact with relational databases, which are organized collections of data stored in tables with predefined structures.
- It facilitates the management and retrieval of data by providing a standardized way to define, query, and manipulate relational databases.

Basic SQL Commands:

- SELECT: Retrieves data from one or more tables.

```
SELECT column1, column2 FROM table_name WHERE condition;
```

- INSERT: Adds new records to a table.

```
INSERT INTO table_name (column1, column2) VALUES (value1, value2);
```

- UPDATE: Modifies existing records in a table.

```
UPDATE table_name SET column1 = value1 WHERE condition;
```

- DELETE: Removes records from a table.

```
DELETE FROM table_name WHERE condition;
```

Data Definition Language (DDL):

- SQL includes DDL commands for defining and managing the structure of a database.
- Examples include CREATE TABLE for creating tables, ALTER TABLE for modifying table structures, and DROP TABLE for deleting tables.

Data Manipulation Language (DML):

- DML commands are used to manipulate data stored in the database.
- Examples include SELECT for querying data, INSERT for adding new records, UPDATE for modifying existing records, and DELETE for removing records.

Data Control Language (DCL):

- DCL commands manage access to the database.
- Examples include GRANT to provide specific privileges to users and REVOKE to remove privileges.

Data Query Language (DQL):

- DQL is used to retrieve information from the database.
- The primary DQL command is SELECT, which allows users to specify the columns they want to retrieve and apply various conditions.

Clauses and Conditions:

- SQL queries often involve the use of clauses such as WHERE for filtering data, ORDER BY for sorting results, GROUP BY for grouping data, and HAVING for filtering grouped data.

Joins:

- SQL supports different types of joins (e.g., INNER JOIN, LEFT JOIN, RIGHT JOIN) to combine data from multiple tables based on specified relationships.

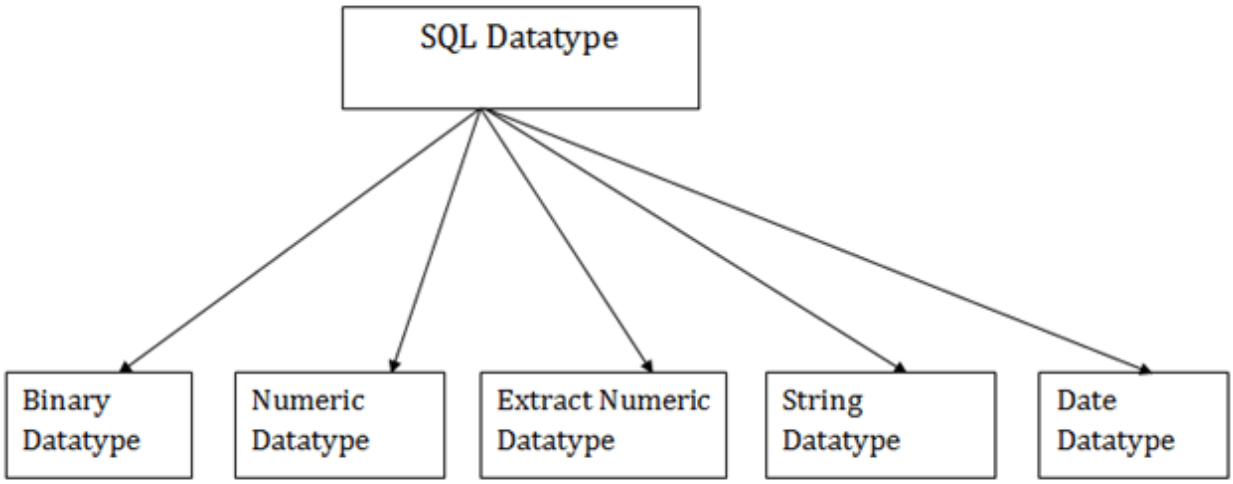
Indexes:

- Indexes are used to optimize data retrieval by providing a quick lookup mechanism.
- They can be created on one or more columns of a table.

SQL is a declarative language, meaning that users specify the desired outcome, and the database management system determines the most efficient way to execute the query. It is a powerful tool for managing and querying relational databases, making it a fundamental skill for database administrators, developers, and analysts.

4.7 Data Types: SQL Datatype is used to define the values that a column can contain.

- Every column is required to have a name and data type in the database table.



1. Binary Datatypes: There are Three types of binary Datatypes which are given below:

Data Type	Description
binary	It has a maximum length of 8000 bytes. It contains fixed-length binary data.
varbinary	It has a maximum length of 8000 bytes. It contains variable-length binary data.
image	It has a maximum length of 2,147,483,647 bytes. It contains variable-length binary data.

2. Approximate Numeric Datatype : The subtypes are given below:

Data type	From	To	Description
float	-1.79E + 308	1.79E + 308	It is used to specify a floating-point value e.g. 6.2, 2.9 etc.

real	-3.40e + 38	3.40E + 38	It specifies a single precision floating point number

3. Exact Numeric Datatype

The subtypes are given below:

Data type	Description
int	It is used to specify an integer value.
smallint	It is used to specify small integer value.
bit	It has the number of bits to store.
decimal	It specifies a numeric value that can have a decimal number.
numeric	It is used to specify a numeric value.

4. Character String Datatype

The subtypes are given below:

Data type	Description
char	It has a maximum length of 8000 characters. It contains Fixed-length non-unicode characters.
varchar	It has a maximum length of 8000 characters. It contains variable-length non-unicode characters.
text	It has a maximum length of 2,147,483,647 characters. It contains variable-length non-unicode characters.

5. Date and time Datatypes

The subtypes are given below:

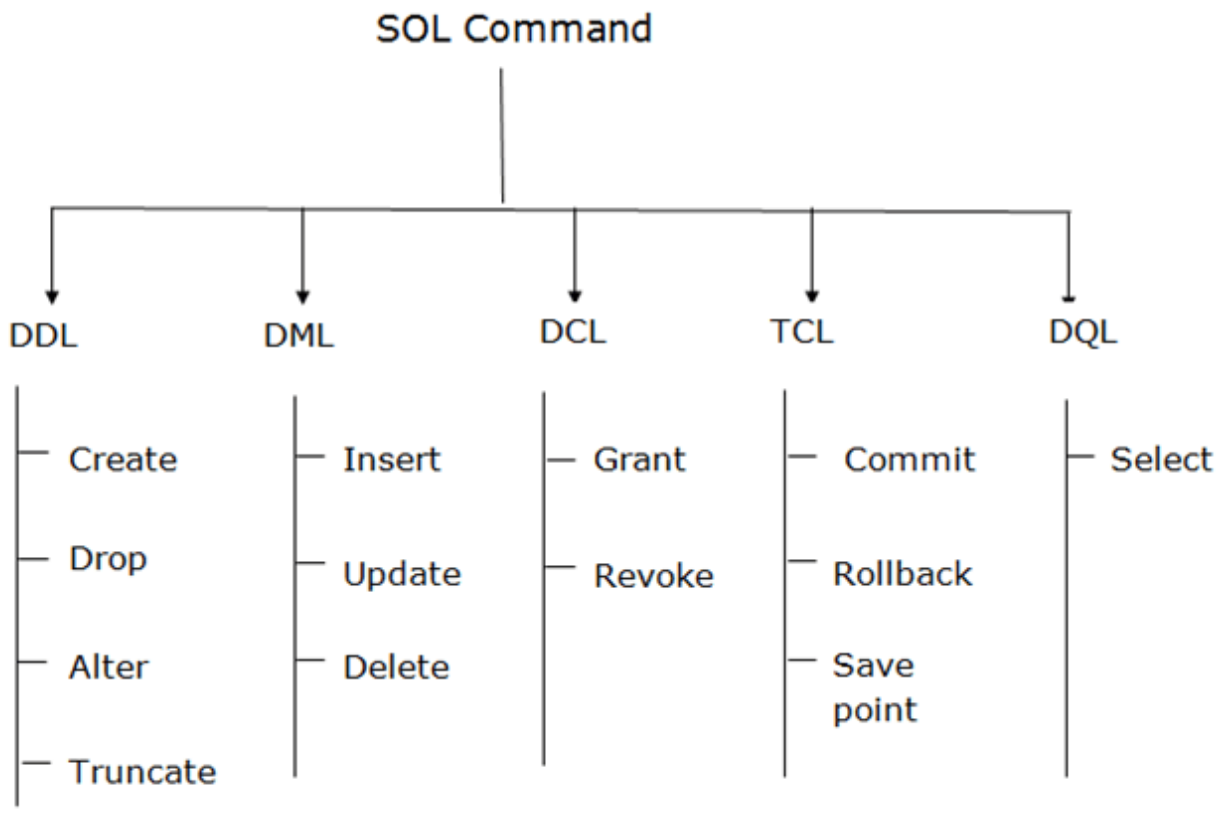
Datatype	Description

date	It is used to store the year, month, and days value.
time	It is used to store the hour, minute, and second values.
timestamp	It stores the year, month, day, hour, minute, and the second value.

4.8 Classification of SQL:

SQL commands are instructions. It is used to communicate with the database. It is also used to perform specific tasks, functions, and queries of data. SQL can perform various tasks like create a table, add data to tables, drop the table, modify the table, set permission for users.

SQL (Structured Query Language) can be broadly classified into several categories based on the types of operations or tasks it performs. Here are the main classifications of SQL:



1. Data Definition Language (DDL)

- DDL changes the structure of the table like creating a table, deleting a table, altering a table, etc.
- All the command of DDL are auto-committed that means it permanently save all the changes in the database.

Here are some commands that come under DDL:

- CREATE
- ALTER
- DROP
- TRUNCATE

a. CREATE It is used to create a new table in the database.

Syntax:

CREATE TABLE TABLE_NAME (COLUMN_NAME DATATYPES[,....]);

Example:

CREATE TABLE EMPLOYEE(Name VARCHAR2(20), Email VARCHAR2(100), DOB DATE);

b. DROP: It is used to delete both the structure and record stored in the table.

Syntax

DROP TABLE table_name;

Example

DROP TABLE EMPLOYEE;

c. ALTER: It is used to alter the structure of the database. This change could be either to modify the characteristics of an existing attribute or probably to add a new attribute.

Syntax:

To add a new column in the table

ALTER TABLE table_name ADD column_name COLUMN-definition;

To modify existing column in the table:

ALTER TABLE table_name MODIFY(column_definitions....);

EXAMPLE

1. ALTER TABLE STU_DETAILS ADD(ADDRESS VARCHAR2(20));
2. ALTER TABLE STU_DETAILS MODIFY (NAME VARCHAR2(20));

d. TRUNCATE: It is used to delete all the rows from the table and free the space containing the table.

Syntax:

TRUNCATE TABLE table_name;

Example:

TRUNCATE TABLE EMPLOYEE;

2. Data Manipulation Language

- DML commands are used to modify the database. It is responsible for all form of changes in the database.
- The command of DML is not auto-committed that means it can't permanently save all the changes in the database. They can be rollback.

Here are some commands that come under DML:

- INSERT
- UPDATE
- DELETE

a. INSERT: The INSERT statement is a SQL query. It is used to insert data into the row of a table.

Syntax:

```
INSERT INTO TABLE_NAME (col1, col2, col3,.... col N) VALUES (value1, value2, value3, .... valueN);
```

Or

```
INSERT INTO TABLE_NAME  VALUES (value1, value2, value3, .... valueN);
```

For example:

```
INSERT INTO javatpoint (Author, Subject) VALUES ("Sonoo", "DBMS");
```

b. UPDATE: This command is used to update or modify the value of a column in the table.

Syntax:

```
UPDATE table_name SET [column_name1= value1,...column_nameN = valueN] [WHERE CONDITION]
```

For example:

```
UPDATE students  SET User_Name = 'Sonoo'  WHERE Student_Id = '3'
```

c. DELETE: It is used to remove one or more row from a table.

Syntax:

```
DELETE FROM table_name [WHERE condition];
```

For example:

```
DELETE FROM javatpoint WHERE Author="Sonoo";
```

3. Data Control Language

DCL commands are used to grant and take back authority from any database user.

Here are some commands that come under DCL:

- Grant
- Revoke

a. Grant: It is used to give user access privileges to a database.
Example

```
GRANT SELECT, UPDATE ON MY_TABLE TO SOME_USER, ANOTHER_USER;
```

b. Revoke: It is used to take back permissions from the user.
Example

```
REVOKE SELECT, UPDATE ON MY_TABLE FROM USER1, USER2;
```

4. Transaction Control Language

TCL commands can only use with DML commands like INSERT, DELETE and UPDATE only. These operations are automatically committed in the database that's why they cannot be used while creating tables or dropping them.
Here are some commands that come under TCL:

- COMMIT
- ROLLBACK
- SAVEPOINT

a. Commit: Commit command is used to save all the transactions to the database.
Syntax:

```
COMMIT;
```

Example:

```
DELETE FROM CUSTOMERS WHERE AGE = 25;
COMMIT;
```

b. Rollback: Rollback command is used to undo transactions that have not already been saved to the database.
Syntax:

```
ROLLBACK;
```

Example:

```
DELETE FROM CUSTOMERS WHERE AGE = 25;
ROLLBACK;
```

c. SAVEPOINT: It is used to roll the transaction back to a certain point without rolling back the entire transaction.
Syntax:

```
SAVEPOINT SAVEPOINT_NAME;
```

5. Data Query Language

DQL is used to fetch the data from the database.
It uses only one command:

- SELECT

SELECT: This is the same as the projection operation of relational algebra. It is used to select the attribute based on the condition described by WHERE clause.
Syntax:

SELECT expressions FROM TABLES WHERE conditions;

For example:

SELECT emp_name FROM employee WHERE age > 20;