Module No.1

Introduction to Database Systems

Introduction, Characteristics of databases, File system v/s Database system, Data abstraction and data Independence, DBMS system architecture, Database Administrator

Introduction

A database is a composed gathering of information, generally stored and accessed electronically from a computer system. A database management system (DBMS) is framework programming for making and overseeing databases. The DBMS gives clients and engineers an exact strategy to make, recoup, update and regulate data. A Data Management System makes it makes it attainable for end clients to make, read, update and delete data in a database. The DBMS essentially fills in as an interface between the database and end customers or application programs, ensuring that data is dependably dealt with and remains viably accessible. The reason behind the system is to store and change data into information to help decide.

The DBMS oversees significant things: the data, the database engine that empowers data to be accessed, bolted and altered; and the database outline, which describes the database's rational structure. These three essential parts helps give synchronization, security, data trustworthiness and uniform association methodologies. Numerous database management systems are moreover responsible for robotized rollbacks, restarts and recovery similarly as the logging and analyzing of development. The most helpful thing about the structure is that it gives a brought unified viewpoint on data that can be accessed by various customers, from different regions, in a measured way.

Characteristics of Database Management System are:

1. Provides security and expels excess
2. Self-portraying nature of a database framework
3. Insulation among projects and information deliberation
4. Support of various perspectives on the information
5. Sharing of information and multiuser exchange handling
6. DBMS enables substances and relations among them to frame tables
7. It follows the ACID concept (Atomicity, Consistency, Isolation, and Durability).
8. DBMS supports multi-client condition that enables clients to access and control information in parallel

Application of DBMS

There are different fields where a database management system is utilized. Following are a few applications which utilize the information base administration framework –

1. **Railway Reservation System –**
   In the rail route reservation framework, the information base is needed to store the record or information of ticket appointments, status about train's appearance, and flight. Additionally, if trains get late, individuals become acquainted with it through the information base update.

2. **Library Management System –**
   There are lots of books in the library so; it is difficult to store the record of the relative multitude of books in a register or duplicate. Along these lines, the data set administration framework (DBMS) is utilized to keep up all the data identified with the name of the book, issue date, accessibility of the book, and its writer.

3. **Banking –**
   Database the executive's framework is utilized to store the exchange data of the client in the information base.

4. **Education Sector –**
   Presently, assessments are led online by numerous schools and colleges. They deal with all assessment information through the data set administration framework (DBMS). In spite of that understudy's enlistments subtleties, grades, courses, expense, participation, results, and so forth all the data is put away in the information base.

5. **Credit card exchanges –**
   The database Management framework is utilized for buying on charge cards and age of month to month proclamations.

6. **Social Media Sites –**
   We all utilization of online media sites to associate with companions and to impart our perspectives to the world. Every day, many people group pursue these online media accounts like Pinterest, Facebook, Twitter, and Google in addition to. By the utilization of the data set administration framework, all the data of clients are put away in the information base and, we become ready to interface with others.

7. **Broadcast communications –**
   Without DBMS any media transmission organization can't think. The Database the executive's framework is fundamental for these organizations to store the

call subtleties and month to month postpaid bills in the information base.

8. **Account –**
   The information base administration framework is utilized for putting away data about deals, holding and acquisition of monetary instruments, for example, stocks and bonds in a data set.

9. **Online Shopping –**
   These days, web-based shopping has become a major pattern. Nobody needs to visit the shop and burn through their time. Everybody needs to shop through web based shopping sites, (for example, Amazon, Flipkart, Snapdeal) from home. So all the items are sold and added uniquely with the assistance of the information base administration framework (DBMS). Receipt charges, installments, buy data these are finished with the assistance of DBMS.

10. **Human Resource Management –**
    Big firms or organizations have numerous specialists or representatives working under them. They store data about worker's compensation, assessment, and work with the assistance of an information base administration framework (DBMS).

11. **Manufacturing –**
    Manufacturing organizations make various kinds of items and deal them consistently. To keep the data about their items like bills, acquisition of the item, amount, inventory network the executives, information base administration framework (DBMS) is utilized.

12. **Airline Reservation System –**
    This framework is equivalent to the railroad reservation framework. This framework additionally utilizes an information base administration framework to store the records of flight takeoff, appearance, and defer status.

Disadvantages of File processing system:

- **Slow access time –**
  Direct access of files is very difficult and one needs to know the entire hierarchy of folders to get to a specific file. This involves a lot of time.
  Data is isolated in File Processing System and data is stored in different files. These files can be in different formats. If you want to extract data from two file then you are required to which part of the file is needed and how they are related to each other.

But still in spite of so many disadvantages, File Processing System is still good for small organizations because it does not require costly softwares and programmers to handle it.

- **Presence of redundant data –**
  The same data can be present in two or more files which takes up more disc space.
  For Example: A student is having record in college library and in Examination department. Then his name, roll number, fathers name and class will be same in both the departments. Also these departments are not dependent on each other. So it create lots of duplicates value about that student and when he needs any change for his name or class then he has to go to both the departments to make these changes happen otherwise it will create problem for him.

- **Inconsistent Data –**
  Due to data redundancy, same data stored at different places might not match to each other.
  For Example: If you change student name in library then his name should be changed in all the departments related to the student.

- **Data Integrity Problems –**
  The data present in the database should be consistent and correct. To achieve this, the data should must satisfy certain constraints.
  For Example**:** The maximum marks of the student can never be more than 100.

- **Difficulty in recovery of corrupt data –**
  Recovery or backup of lost and corrupt data is nearly impossible in case of File Processing System.

- **Lack of Atomicity –**
  Operations performed in the database must be atomic i.e. either the operation takes place as a whole or does not take place at all.
  For Example: If you are buying a ticket from railway and you are in the process of money transaction. Suddenly, your internet got disconnected then you may or may not have paid for the ticket. If you have paid then your ticket will be booked and if not then you will not be charged anything. That is called consistent state, means you have paid or not.

- **Problem in Concurrent Access –**
  When a number of users operates on a common data in database at the same time then anomalies arise, due to lack of concurrency control.

- **Unauthorized Access and lack of security–**
  Anyone who gets access to the file can read or modify the data.

For Example: If a student can access his data in the college library then he can easily change books issued date. Also he can change his fine detains to zero.

There are the following differences between DBMS and File systems:

Basis	DBMS Approach	File System Approach
Meaning	DBMS is a collection of data. In DBMS, the user is not required to write the procedures.	The file system is a collection of data. In this system, the user has to write the procedures for managing the database.
Sharing of data	Due to the centralized approach, data sharing is easy.	Data is distributed in many files, and it may be of different formats, so it isn't easy to share data.
Data Abstraction	DBMS gives an abstract view of data that hides the details.	The file system provides the detail of the data representation and storage of data.
Security and Protection	DBMS provides a good protection mechanism.	It isn't easy to protect a file under the file system.
Recovery Mechanism	DBMS provides a crash recovery mechanism, i.e., DBMS protects the user from system failure.	The file system doesn't have a crash mechanism, i.e., if the system crashes while entering some data, then the content of the file will be lost.
Manipulation Techniques	DBMS contains a wide variety of sophisticated techniques to store and retrieve the data.	The file system can't efficiently store and retrieve the data.
Concurrency Problems	DBMS takes care of Concurrent access of data using some form of locking.	In the File system, concurrent access has many problems like redirecting the file while deleting some information or updating some information.
Where to use	Database approach used in large systems which interrelate many files.	File system approach used in large systems which interrelate many files.
Cost	The database system is expensive to design.	The file system approach is cheaper to design.

Data Redundancy and Inconsistency	Due to the centralization of the database, the problems of data redundancy and inconsistency are controlled.	In this, the files and application programs are created by different programmers so that there exists a lot of duplication of data which may lead to inconsistency.
Structure	The database structure is complex to design.	The file system approach has a simple structure.
Data Independence	In this system, Data Independence exists, and it can be of two types. o Logical Data Independence o Physical Data Independence	In the File system approach, there exists no Data Independence.
Integrity Constraints	Integrity Constraints are easy to apply.	Integrity Constraints are difficult to implement in file system.
Data Models	In the database approach, 3 types of data models exist: o Hierarchal data models o Network data models o Relational data models	In the file system approach, there is no concept of data models exists.
Flexibility	Changes are often a necessity to the content of the data stored in any system, and these changes are more easily with a database approach.	The flexibility of the system is less as compared to the DBMS approach.
Examples	Oracle, SQL Server, Sybase etc.	Cobol, C++ etc.

Data Abstraction and data independence:

Database systems comprise complex data structures. In order to make the system efficient in terms of retrieval of data, and reduce complexity in terms of usability of users, developers use abstraction i.e. hide irrelevant details from the users. This approach simplifies database design.

There are mainly 3 levels of data abstraction:

1. **Physical:** This is the lowest level of data abstraction. It tells us how the data is actually stored in memory. The access methods like sequential or random access and file organization methods like B+ trees and hashing are used for the same. Usability, size of memory, and the number of times the records are factors that we need to know while designing the database.

Suppose we need to store the details of an employee. Blocks of storage and the amount of memory used for these purposes are kept hidden from the user.

2. **Logical:** This level comprises the information that is actually stored in the database in the form of tables. It also stores the relationship among the data entities in relatively simple structures. At this level, the information available to the user at the view level is unknown.

We can store the various attributes of an employee and relationships, e.g. with the manager can also be stored.

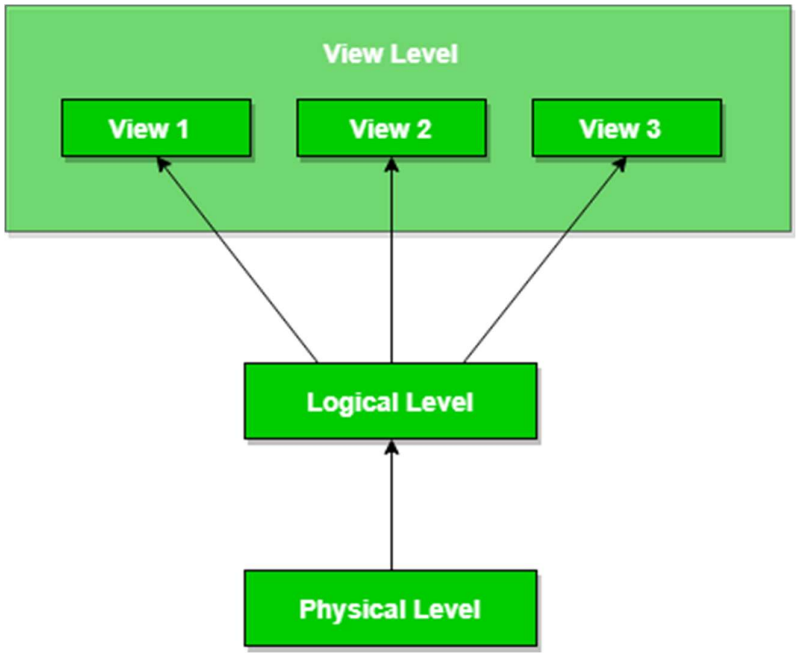
3. **View:** This is the highest level of abstraction. Only a part of the actual database is viewed by the users. This level exists to ease the accessibility of the database by an individual user. Users view data in the form of rows and columns. Tables and relations are used to store data. Multiple views of the same database may exist. Users can just view the data and interact with the database, storage and implementation details are hidden from them.

Example: In case of storing customer data,

Physical level – it will contains block of storages (bytes,GB,TB,etc)

Logical level – it will contain the fields and the attributes of data.

View level – it works with CLI or GUI access of database



The main purpose of data abstraction is to achieve data independence in order to save the time and cost required when the database is modified or altered.

Data Independence

Data Independence is mainly defined as a property of DBMS that helps you to change the database schema at one level of a system without requiring to change the schema at the next level. it helps to keep the data separated from all program that makes use of it.

We have namely two levels of data independence arising from these levels of abstraction :

Physical level data independence: It refers to the characteristic of being able to modify the physical schema without any alterations to the conceptual or logical schema, done for optimization purposes, e.g., the Conceptual structure of the database would not be affected by any change in storage size of the database system server. Changing from sequential to random access files is one such example. These alterations or modifications to the physical structure may include:

- Utilizing new storage devices.
- Modifying data structures used for storage.
- Altering indexes or using alternative file organization techniques etc.

Logical level data independence: It refers characteristic of being able to modify the logical schema without affecting the external schema or application program. The user view of the data would not be affected by any changes to the conceptual view of the data. These changes may include insertion or deletion of attributes, altering table structures entities or relationships to the logical schema, etc.

Database Architecture

A Database Architecture is a representation of DBMS design. It helps to design, develop, implement, and maintain the database management system. A DBMS architecture allows dividing the database system into individual components that can be independently modified, changed, replaced, and altered. It also helps to understand the components of a database.

A Database stores critical information and helps access data quickly and securely. Therefore, selecting the correct Architecture of DBMS helps in easy and efficient data management.

Types of DBMS Architecture

1-Tier Architecture

2-Tier Architecture

3-Tier Architecture

There are mainly three types of DBMS architecture:

1-Tier Architecture

1 Tier Architecture in DBMS is the simplest architecture of Database in which the client, server, and Database all reside on the same machine. A simple one tier architecture example would be anytime you install a Database in your system and access it to practice SQL queries. But such architecture is rarely used in production.

Fig. 1-Tier Architecture

2-Tier Architecture

A 2 Tier Architecture in DBMS is a Database architecture where the presentation layer runs on a client (PC, Mobile, Tablet, etc.), and data is stored on a server called the second tier. Two tier architecture provides added security to the DBMS as it is not exposed to the end-user directly. It also provides direct and faster communication.

In the above 2 Tier client-server architecture of database management system, we can see that one server is connected with clients 1, 2, and 3.

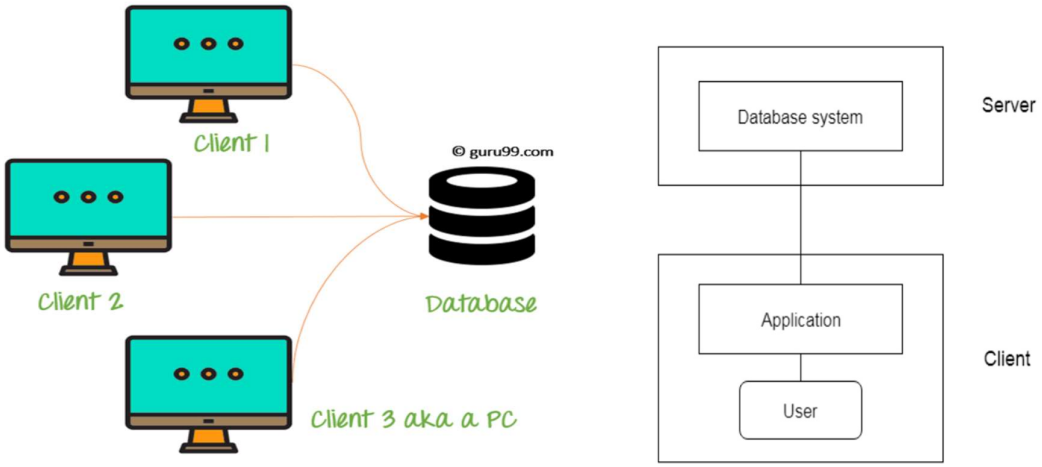


Fig. 2 Tier Architecture

A Contact Management System created using MS- Access.

3-Tier Architecture

A 3 Tier Architecture in DBMS is the most popular client server architecture in DBMS in which the development and maintenance of functional processes, logic, data access, data storage, and user interface is done independently as separate modules. Three Tier architecture contains a presentation layer, an application layer, and a database server.

3-Tier database Architecture design is an extension of the 2-tier client-server architecture. A 3-tier architecture has the following layers:

- Presentation layer (your PC, Tablet, Mobile, etc.)
- Application layer (server)
- Database Server

The Application layer resides between the user and the DBMS, which is responsible for communicating the user's request to the DBMS system and send the response from the DBMS to the user. The application layer(business logic layer) also processes functional logic, constraint, and rules before passing data to the user or down to the DBMS.

The goal of Three Tier client-server architecture is:

- To separate the user applications and physical database
- To support DBMS characteristics
- Program-data independence
- Supporting multiple views of the data

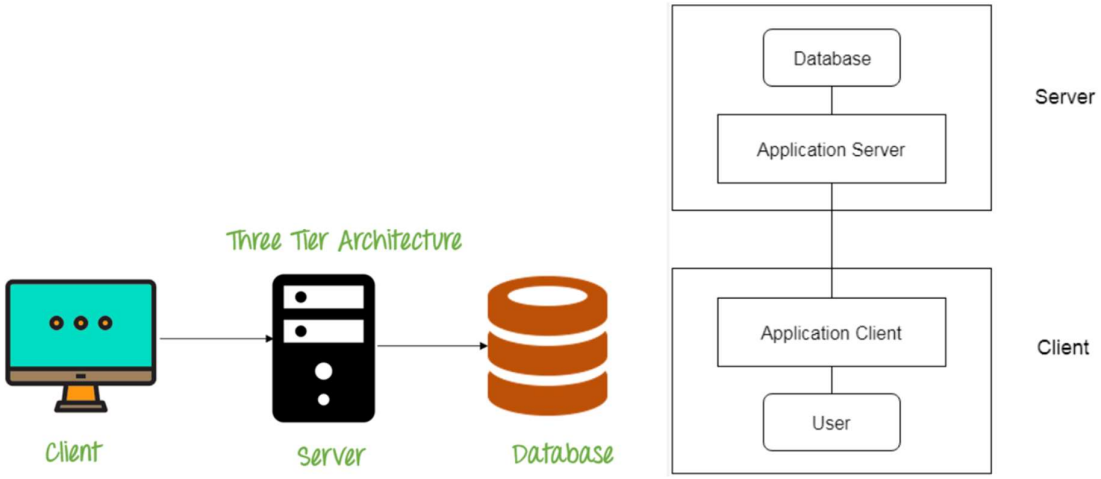


Fig. 3 Tier Architecture

Any large website on the internet, including google.com

DBMS System Structure

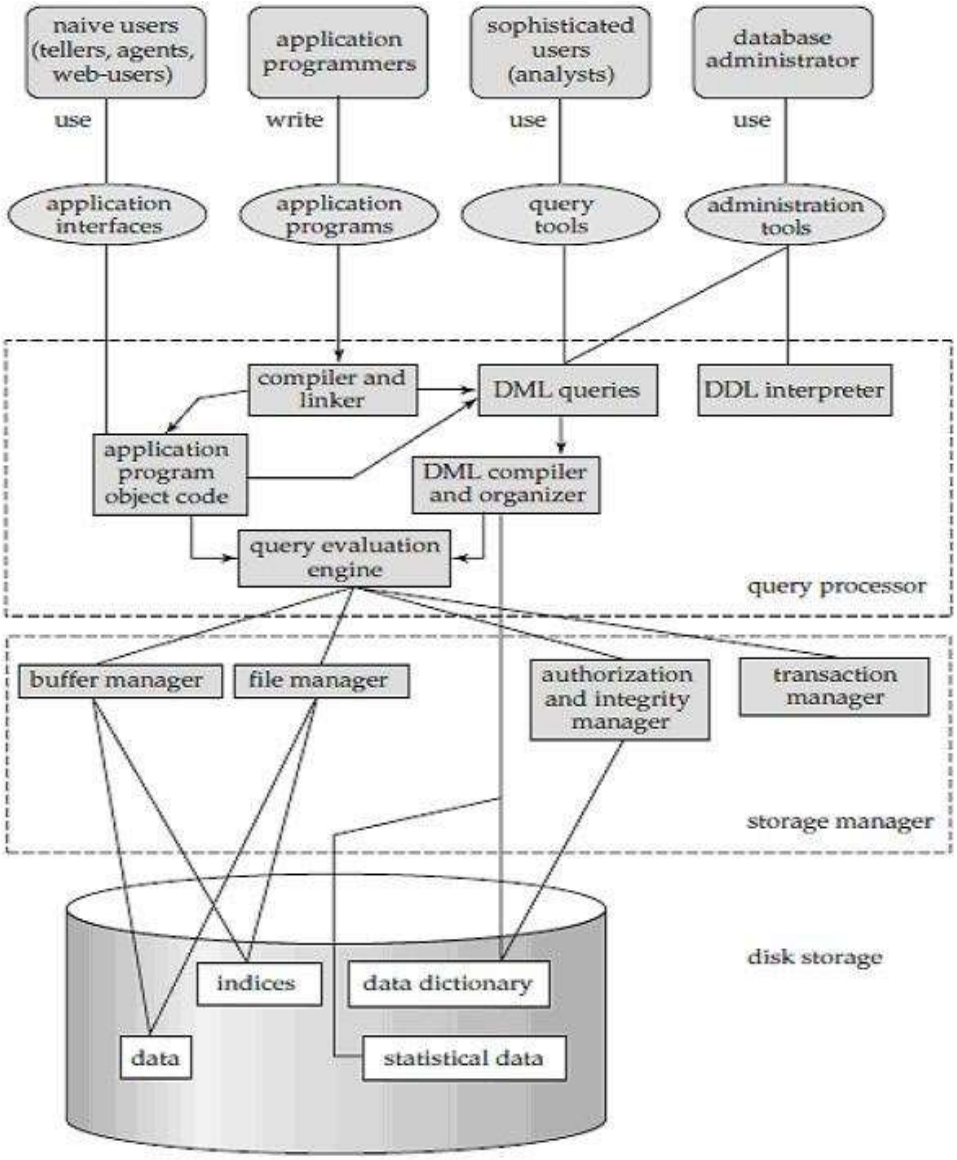


Fig. DBMS System Structure

The database system is divided into three components: Query Processor, Storage Manager, and Disk Storage.

1. Query Processor: It interprets the requests (queries) received from end user via an application program into instructions. It also executes the user request which is received from the DML compiler.

Query Processor contains the following components –

DML Compiler: It processes the DML statements into low level instruction (machine language), so that they can be executed.

DDL Interpreter: It processes the DDL statements into a set of table containing meta data (data about data).

Embedded DML Pre-compiler: It processes DML statements embedded in an application program into procedural calls.

Query Optimizer: It executes the instruction generated by DML Compiler.

2. Storage Manager: Storage Manager is a program that provides an interface between the data stored in the database and the queries received. It is also known as Database Control System. It maintains the consistency and integrity of the database by applying the constraints and executing the DCL statements. It is responsible for updating, storing, deleting, and retrieving data in the database. It contains the following components –

Authorization Manager: It ensures role-based access control, i.e,. checks whether the particular person is privileged to perform the requested operation or not.

Integrity Manager: It checks the integrity constraints when the database is modified.

Transaction Manager: It controls concurrent access by performing the operations in a scheduled way that it receives the transaction. Thus, it ensures that the database remains in the consistent state before and after the execution of a transaction.

File Manager: It manages the file space and the data structure used to represent information in the database.

Buffer Manager: It is responsible for cache memory and the transfer of data between the secondary storage and main memory.

3. Disk Storage: It contains the following components –

Data Files: It stores the data.

Data Dictionary: It contains the information about the structure of any database object. It is the repository of information that governs the metadata.

Indices: It provides faster retrieval of data item.

Users of DBMS:

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

Database Administrator (DBA) : Database Administrator (DBA) is a person/team who defines the schema and also controls the 3 levels of database. The DBA will then create a new account id and password for the user if he/she need 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 data base.
- 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.

Naive 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.

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.

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.

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.

Responsibilities of DBA:

- Creating and maintaining database standards and policies

- Supporting database design, creation, and testing activities
- Managing the database availability and performance, including incident and problem management
- Administering database objects to achieve optimum utilization
- Defining and implementing event triggers that will alert on potential database performance or integrity issues
- Performing database housekeeping, such as tuning, indexing, etc.
- Monitoring usage, transaction volumes, response times, concurrency levels, etc.
- Identifying reporting, and managing database security issues, audit trails, and forensics
- Designing database backup, archiving, and storage strategy