Introduction: Database-System Applications- Purpose of Database Systems - View of Data --Database Languages - **Relational Databases** - Database Design -Object- Based and Semi structured Databases - Data Storage and Querying Transaction Management -Data Mining and Analysis - Database Architecture - Database Users and Administrators - History of Database Systems.

**What is database?**

* A database is an organized collection of structured information, or data, typically stored electronically in a computer system.
* A database is usually controlled by a [database management system (DBMS)](https://www.oracle.com/in/database/what-is-database.html#WhatIsDBMS).
* Databases support storage and manipulation of data.
* Databases make data management easy.
* You can organize data into tables, rows, columns, and index it to make it easier to find relevant information.
* There are many databases available like MySQL, Sybase, Oracle, MongoDB, SQL Server, etc.

**Database System Application**

* Database applications are software programs designed to collect, manage information efficiently.
* **Database Management System (DBMS)** is software for storing and retrieving users' data while considering security measures.
* The DBMS accepts the request for data from an application and instructs the operating system to provide the specific data.
* In large systems, a DBMS helps users and other third-party software to store and retrieve data.
* DBMS allows users to create their own databases as per their requirement.
* It provides an interface between the data and the software application.

Applications where we use Database Management Systems are:

**Telecom**: There is a database to keeps track of the information regarding calls made, network usage, customer details etc. Without t he database systems it is hard to maintain that huge amount of data that keeps updating every millisecond.

**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.

**Airlines**: To travel though airlines, we make early reservations; this reservation information along with flight schedule is stored in database.

**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 hell lot 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, Flip kart 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.

**Purpose of Database**

* Database systems are designed to manage large numbers of information.
* Management of data involves both defining[structures](https://www.toppr.com/guides/business-management-entrepreneurship/organizing/structure-of-organization/) for storage of information and providing mechanisms for the manipulation of information.
* In addition, the database system must ensure the safety of the information stored, despite system crashes or attempts at unauthorized access.
* If data are to be shared among several users, the system must avoid possible anomalous results.

**Data redundancy and inconsistency:**Since different programmers create
the files and application programs over a long period, the various files are
likely to have different structures and the programs may bewritten in several
programming languages. Moreover, the same information may be duplicated
in several places (files). For example, if a student has a double major (say,
music and mathematics) the address and telephone number of that student
may appear in a file that consists of student records of students in the Music
department and in a file that consists of student records of students in the
Mathematics department. This redundancy leads to higher storage and access
cost. In addition, it may lead to**data inconsistency**; that is, the various copies
of the same datamayno longer agree. For example, a changed student address
may be reflected in the Music department records but not elsewhere in the
system.

**Data isolation:** Because data are scattered in various files, and files may
be in different formats, writing new application programs to retrieve the
appropriate data is difficult.

**Security problems:** Not every user of the database system should be able
to access all the data. For example, in a university, payroll personnel need
to see only that part of the database that has financial information. They do
not need access to information about academic records. But, since application
programs are added to the file-processing system in an ad hoc manner,
enforcing such security constraints is difficult.

**View of Data**

* Abstraction is one of the main features of database systems.
* Hiding irrelevant details from user and providing abstract view of data to users, helps in easy and efficient **user-database** interaction.
* There are [three level of DBMS architecture](https://beginnersbook.com/2018/11/dbms-three-level-architecture/), The top level of that architecture is “view level”.
* The view level provides the “**view of data**” to the users and hides the irrelevant details such as data relationship, database schema, [constraints](https://beginnersbook.com/2015/04/constraints-in-dbms/), security etc from the user.
* To fully understand the view of data, you must have a basic knowledge of data abstraction and instance & schema.

**Data Abstraction**

Database systems are made-up of complex data structures. To ease the user interaction with database, the developers hide internal irrelevant details from users. This process of hiding irrelevant details from user is called data abstraction.

We have three levels of abstraction:

1. **Physical level:** This is the lowest level of data abstraction. It describes how data is actually stored in database. You can get the complex data structure details at this level.
2. **Logical level:** This is the middle level of 3-level data abstraction architecture. It describes what data is stored in database.
3. **View level:** Highest level of data abstraction. This level describes the user interaction with database system.

# Instance and schema

# Instance The data stored in database at a particular moment of time is called instance of database. Database schema defines the variable declarations in tables that belong to a particular database; the value of these variables at a moment of time is called the instance of that database.

For example, let’s say we have a single table student in the database, today the table has 100 records, so today the instance of the database has 100 records. Let’s say we are going to add another 100 records in this table by tomorrow so the instance of database tomorrow will have 200 records in table. In short, at a particular moment the data stored in database is called the instance that changes over time when we add or delete data from the database.

## Schema Design of a database is called the schema. Schema is of three types: Physical schema, logical schema and view schema.

For example: In the following diagram, we have a schema that shows the relationship between three tables: Course, Student and Section. The diagram only shows the design of the database, it doesn’t show the data present in those tables. Schema is only a structural view (design) of a database as shown in the diagram below.



1. The design of a database at physical level is called **physical schema**, how the data stored in blocks of storage is described at this level.
2. Design of database at logical level is called **logical schema**, programmers and database administrators work at this level, at this level data can be described as certain types of data records gets stored in data structures, however the internal details such as implementation of data structure is hidden at this level (available at physical level).
3. Design of database at view level is called **view schema**. This generally describes end user interaction with database systems.

# Database Language

* A DBMS has appropriate languages and interfaces to express database queries and updates.
* Database languages can be used to read, store and update the data in the database.

## Types of Database Language



### DATA DEFINITION LANGUAGE

* **DDL** stands for **D**ata **D**efinition **L**anguage. It is used to define database structure or pattern.
* It is used to create schema, tables, indexes, constraints, etc. in the database.
* Using the DDL statements, you can create the skeleton of the database.
* Data definition language is used to store the information of metadata like the number of tables and schemas, their names, indexes, columns in each table, constraints, etc.

### DATA MANIPULATION LANGUAGE

* **DML** stands for **D**ata **M**anipulation **L**anguage. It is used for accessing and manipulating data in a database. It handles user requests.

### DATA CONTROL LANGUAGE

* **DCL** stands for **D**ata **C**ontrol **L**anguage. It is used to retrieve the stored or saved data.
* The DCL execution is transactional. It also has rollback parameters.

### TRANSACTION CONTROL LANGUAGE

* TCL is used to run the changes made by the DML statement. TCL can be grouped into a logical transaction.

**Database Design**

It is a collection of processes that facilitate the designing, development, implementation and maintenance of enterprise data management systems. Properly designed database are easy to maintain, improves data consistency and are cost effective in terms of disk storage space. The database designer decides how the data elements correlate and what data must be stored.

The main objectives of database designing are to produce **logical** and **physical** designs models of the proposed database system.

 **The logical model** concentrates on the data requirements and the data to be stored independent of physical considerations. It does not concern itself with how the data will be stored or where it will be stored physically.

**The physical data** design model involves translating the logical design of the database onto physical media using hardware resources and software systems such as database management systems (DBMS).



## Why Database Design is Important?

It helps produce database systems

1. That meet the requirements of the users
2. Have high performance.

Database designing is crucial to **high performance** database system.

**Object- Based and Semi structured Databases**

**Object- Based** is a database in which the information is represented in form of object as used in object-oriented programming. It is different from rational database. This type of database is used when there is complex data or/and multiple data relationships. It have a many-to-many object relationship. It should not be used when there are few join tables and there are large volume of simple transaction data.

### Features of Object Oriented Database:

* It support transactions.
* It supply querying in bulk data.
* Concurrent Access
* Security

**Semi-Structured Database**

It is the data are in the form of structured data that does not conform with the formal structure of data models associated with rational databases or other form of data. Therefore, it is also known as self-describing structure.

### Types of Semi-Structured Database:

* XML semi-structured database
* JSON (JavaScript Object Notation)semi-structured database

**Advantages of Semi-Structured Database**

* It can show the information of data source that is not constrained by schema.
* It is used to view structured data as semi-structured data.
* The data transfer format may be portable.

**Data Mining and Analysis**

In general terms, **“Mining”** is the process of extraction of some valuable material from the earth e.g. coal mining, diamond mining etc. In the context of computer science, **“Data Mining”** refers to the extraction of useful information from a bulk of data or [data warehouses](https://www.geeksforgeeks.org/data-warehousing/).

Now a days, data mining is used in almost all the places where a large amount of data is stored and processed. For example, banks typically use ‘data mining’ to find out their prospective customers who could be interested in credit cards, personal loans or insurances as well. Since banks have the transaction details and detailed profiles of their customers, they analyze all this data and try to find out patterns which help them predict that certain customers could be interested in personal loans etc.

**Main Purpose of Data Mining**

* Basically, the information gathered from Data Mining helps to predict hidden patterns, future trends and behaviors and allowing businesses to take decisions.
* Technically, data mining is the computational process of analyzing data from different perspective, dimensions, angles and categorizing/summarizing it into meaningful information.
* Data Mining can be applied to any type of data e.g. Data Warehouses, Transactional Databases, Relational Databases, Multimedia Databases, Spatial Databases, Time-series Databases, World Wide Web.

**Data Mining as a whole process**

The whole process of Data Mining comprises of three main phases:

1. Data Pre-processing – Data cleaning, integration, selection and transformation takes place

2. Data Extraction – Occurrence of exact data mining

3. Data Evaluation and Presentation – Analyzing and presenting results.

**Database Architecture**

Database architecture uses programming languages to design a particular type of software for businesses or organizations. Database architecture focuses on the design, development, implementation and maintenance of computer programs that store and organize information for businesses, agencies and institutions. A database architect develops and implements software to meet the needs of users.

The design of a DBMS depends on its architecture. It can be centralized or decentralized or hierarchical. The architecture of a DBMS can be seen as either single tier or multi-tier. The tiers are classified as follows :

1. 1-tier architecture
2. 2-tier architecture
3. 3-tier architecture
4. n-tier architecture

**1-tier architecture:**

One-tier architecture involves putting all of the required components for a software application or technology on a single server or platform.



Basically, a one-tier architecture keeps all of the elements of an application, including the interface, Middleware and back-end data, in one place. Developers see these types of systems as the simplest and most direct way.

**2-tier architecture:**

The two-tier is based on Client Server architecture. The two-tier architecture is like client server application. The direct communication takes place between client and server. There is no intermediate between client and server.



**3-tier architecture:**

A 3-tier architecture separates its tiers from each other based on the complexity of the users and how they use the data present in the database. It is the most widely used architecture to design a DBMS.



**n-tier architecture:**

N-tier architecture would involve dividing an application into three different tiers. These would be the

1. logic tier,
2. the presentation tier, and
3. the data tier.



**DATABASE USERS AND ADMINISTRATORS**

**Database Users**

Database users are the one who really use and take the benefits of database. There will be different types of users depending on their need and way of accessing the database.

**Application Programmers –** They are the developers who interact with the database by means of DML queries. These DML queries are written in the application programs like C, C++, JAVA, Pascal etc. These queries are converted into object code to communicate with the database. For example, writing a C program to generate the report of employees who are working in particular department will involve a query to fetch the data from database. It will include a embedded SQL query in the C Program.

**Sophisticated Users –** They are database developers, who write SQL queries to select/insert/delete/update data. They do not use any application or programs to request the database. They directly interact with the database by means of query language like SQL. These users will be scientists, engineers, analysts who thoroughly study SQL and DBMS to apply the concepts in their requirement. In short, we can say this category includes designers and developers of DBMS and SQL.

**Specialized Users –** These are also sophisticated users, but they write special database application programs. They are the developers who develop the complex programs to the requirement.

**Stand-alone Users –** These users will have stand –alone database for their personal use. These kinds of database will have readymade database packages which will have menus and graphical interfaces.

**Native Users –** these are the users who use the existing application to interact with the database. For example, online library system, ticket booking systems, ATMs etc which has existing application and users use them to interact with the database to fulfill their requests.

**Database Administrators**

The life cycle of database starts from designing, implementing to administration of it. A database for any kind of requirement needs to be designed perfectly so that it should work without any issues. Once all the design is complete, it needs to be installed. Once this step is complete, users start using the database. The database grows as the data grows in the database. When the database becomes huge, its performance comes down. Also accessing the data from the database becomes challenge. There will be unused memory in database, making the memory inevitably huge. These administration and maintenance of database is taken care by database Administrator – DBA.

A DBA has many responsibilities. A good performing database is in the hands of DBA.

**Installing and upgrading the DBMS Servers:** – DBA is responsible for installing a new DBMS server for the new projects. He is also responsible for upgrading these servers as there are new versions comes in the market or requirement. If there is any failure in upgradation of the existing servers, he should be able revert the new changes back to the older version, thus maintaining the DBMS working. He is also responsible for updating the service packs/ hot fixes/ patches to the DBMS servers.

**Design and implementation: –** Designing the database and implementing is also DBA’s responsibility. He should be able to decide proper memory management, file organizations, error handling, log maintenance etc for the database.

**Performance tuning: –** Since database is huge and it will have lots of tables, data, constraints and indices, there will be variations in the performance from time to time. Also, because of some designing issues or data growth, the database will not work as expected. It is responsibility of the DBA to tune the database performance. He is responsible to make sure all the queries and programs works in fraction of seconds.

**Migrate database servers: –** Sometimes, users using oracle would like to shift to SQL server or Netezza. It is the responsibility of DBA to make sure that migration happens without any failure, and there is no data loss.

**Backup and Recovery: –** Proper backup and recovery programs needs to be developed by DBA and has to be maintained him. This is one of the main responsibilities of DBA. Data/objects should be backed up regularly so that if there is any crash, it should be recovered without much effort and data loss.

**Security: –** DBA is responsible for creating various database users and roles, and giving them different levels of access rights.

**Documentation: –** DBA should be properly documenting all his activities so that if he quits or any new DBA comes in, he should be able to understand the database without any effort. He should basically maintain all his installation, backup, recovery, security methods. He should keep various reports about database performance.