What is Data?

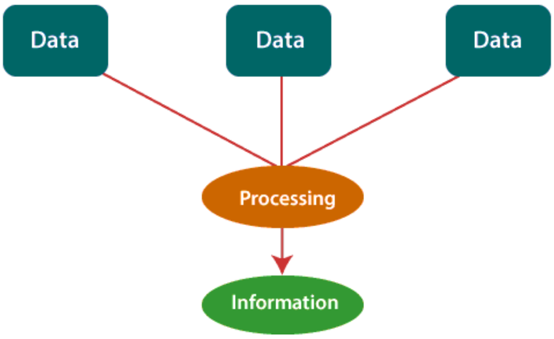
Data is a real-world entity or an object. Data is a distinct piece of information or facts that need to be processed. It can be in any form like text, number, picture, measurements, and bytes.

Example: Ankit, Delhi, 12, 80.

What is Information?

When data are processed, organized, structured, and interpreted in a given context, so as to make them useful and meaningful, they are called information.

Example: Name - Ankit, **City** - Delhi, **Class** – 12, **Marks** – 80.



What is Database?

A database is an organized collection of inter-related data, which helps in *insertion, deletion, update and retrieval* of data efficiently. The database is also used to organize the data or information in the form of tables, views, schemas, reports, etc.

Note: Using the database, you can easily access, update, and delete any information.

Database systems are basically developed for large amount of data. When dealing with huge amount of data, there are two things that require optimization: **Storage of data** and **retrieval of data**.

Storage: According to the principles of database systems, the data is stored in such a way that it acquires lot less space as the redundant data (duplicate data) has been removed before storage.

Fast Retrieval of data: Along with storing the data in an optimized and systematic manner, it is also important that we retrieve the data quickly when needed. Database systems ensure that the data is retrieved as quickly as possible.

Database Management System (DBMS)

DBMS is a software that manages the data for **efficient storage and fast retrievals**. MySQL, IBM Db2, Oracle, PostgreSQL etc. are all DBMS software that manages the data.

DBMS is used in various **applications such as telecom, banking, sales, airlines, education, online shopping** etc.

DBMS also secures the data from unauthorized access as well as corrupt data insertions. It allows multiple users to access data simultaneously while maintaining the **data consistency and data integrity**.

DBMS allows following operations to the authorized users of the database:

Data Definition: Creation of table, table schema creation, removal of table definition etc. comes under data definition. It is basically a layout of the table and their relation with the other tables in the database.

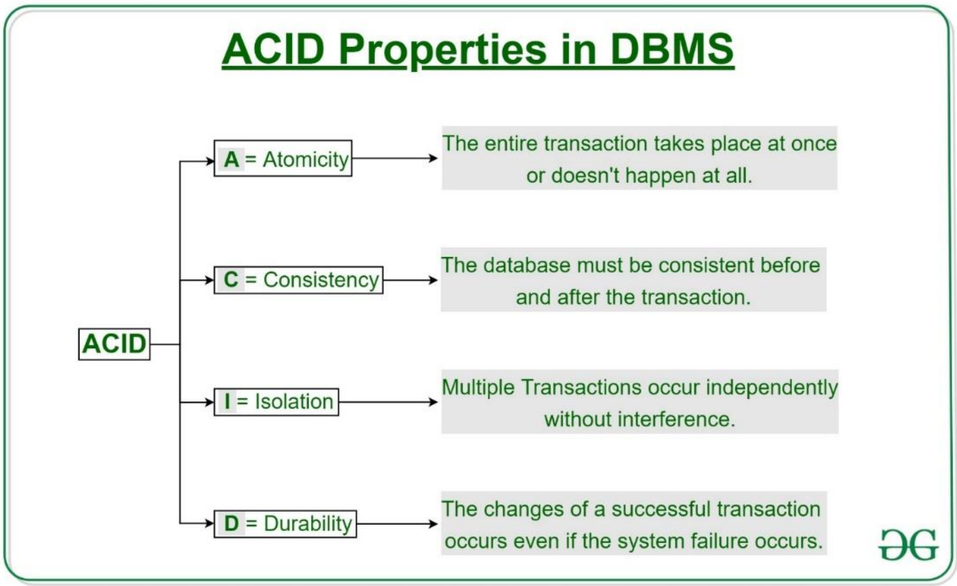
Data Modification: DBMS allows users to insert, update and delete the data from the tables. These tables contains rows and columns, where row represents a record of data while column represents attributes of the records.

Data Retrieval: DBMS allows users to fetch data from the database.

User administration: DBMS also allows user management such as organizing users in different groups with different access levels. Granting users access to certain tables in database, revoking access from certain users etc.

Characteristics of DBMS

- Stores the data in such a way so that the relation between data is still maintained in the database.
- Allows fast retrieval.
- It can handle multiple accessing the database at the same time.
- It maintains data integrity by following ACID properties of the database.



ACID Properties in DBMS

ACID
A = Atomicity
The entire transaction takes place at once or doesn't happen at all.
C = Consistency
The database must be consistent before and after the transaction.
I = Isolation
Multiple Transactions occur independently without interference.
D = Durability
The changes of a successful transaction occurs even if the system failure occurs.

- It provides data security by managing user access.
- DBMS allows automatic backup of database to handle accidental corruption or deletion of data.
- It allows scaling of database as per the need.
- It allows data rollback and redone in case of a data operation failure.

Advantages of DBMS

- **Handles Database redundancy:** The major disadvantage of file based system of storing the data is data redundancy, same data is stored in multiple files. DBMS handles data redundancy to manage the storage space efficiently.
- **Data sharing:** DBMS allows data sharing so that data can be shared between multiple users of the same organization efficiently.
- **Data Maintenance:** DBMS performs regular data checks and automatic backup.
- **Performance:** Provides better performance for operations such as read, insert, update and deletion of data.
- **Backup:** It maintains backup of the database so that in case of a failure, database can be recovered to the previous state using the backup.
- **Multiple users:** It allows multiple users to access the data at the same time.

Disadvantages of DBMS

- **Hardware and Software Cost:** Although DBMS has several advantages over file system of data management, however all this comes with a cost. DBMS needs a dedicated hardware and software system to manage the database.

- **Need large Storage:** DBMS is usually used in the large organizations that require large amount of data stored in the devices.
- **Complexity:** Database management system is complex and not easy to implement.
- **Requires learning:** In order to manage database, user require learning the concepts of DBMS which require additional time and resources that an organization has to bear.

DBMS applications

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 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 center, each one needs a database to keep the records of ins and outs. For example distribution center should keep a track of the product units that supplied into the center as well as the products that got delivered out from the distribution center 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 analyze 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.

Drawbacks of File system

- **Data redundancy:** Data redundancy refers to the duplication of data, lets say we are managing the data of a college where a student is enrolled for two courses, the same student details in such case will be stored twice, which will take more storage than needed. Data redundancy often leads to higher storage costs and poor access time.
- **Data inconsistency:** Data redundancy leads to data inconsistency, lets take the same example that we have taken above, a student is enrolled for two courses and we have student address stored twice, now lets say student requests to change his address, if the address is changed at one place and not on all the records then this can lead to data inconsistency.
- **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.
- **Dependency on application programs:** Changing files would lead to change in application programs.
- **Atomicity issues:** Atomicity of a transaction refers to “All or nothing”, which means either all the operations in a transaction executes or none.

For example: Let's say Steve transfers 100$ to Negan's account. This transaction consists multiple operations such as debit 100$ from Steve's account, credit 100$ to Negan's account. Like any other device, a computer system can fail let's say it fails after first operation then in that case Steve's account would have been debited by 100$ but the amount was not credited to Negan's account, in such case the rollback of operation should occur to maintain the atomicity of transaction. It is **difficult to achieve atomicity in file processing systems**.

- **Data Security:** Data should be secured from unauthorized access, for example a student in a college should not be able to see the payroll details of the teachers, such kind of security constraints are difficult to apply in file processing systems.

Advantage of DBMS over file system

There are several advantages of Database management system over file system. Few of them are as follows:

- **No redundant data**: Redundancy removed by data normalization. No data duplication saves storage and improves access time.
- **Data Consistency and Integrity**: the root cause of data inconsistency is data redundancy, since data normalization takes care of the data redundancy, data inconsistency also been taken care of as part of it
- **Data Security**: It is easier to apply access constraints in database systems so that only authorized user is able to access the data. Each user has a different set of access thus data is secured from the issues such as identity theft, data leaks and misuse of data.
- **Privacy**: Limited access means privacy of data. DBMS can grant and revoke access to the database on user level that ensures who is accessing which data. It also helps user to manage the constraints on database, this ensures which type of data can be entered into the table.
- **Easy access to data** – Database systems manages data in such a way so that the data is easily accessible with fast response times. Even if the database size is huge, the DBMS can still provide faster access and updation of data.
- **Easy recovery**: Since database systems keeps the backup of data, it is easier to do a full recovery of data in case of a failure. This is very useful especially for almost all the organizations, as the data maintained over time should not be lost during a system crash or failure.
- **Flexible**: Database systems are more flexible than file processing systems. DBMS systems are scalable, the database size can be increased and decreased based on the amount of storage required. It also allows addition of additional tables as well as removal of existing tables without disturbing the consistency of data.

Disadvantages of DBMS

- DBMS implementation cost is high compared to the file system
- Complexity: Database systems are complex to understand
- Performance: Database systems are generic, making them suitable for various applications. However this feature affect their performance for some applications

What is a DBMS (Database Management System)?

Database management system is a software that maintains the data on a system. It allows the user to perform various operations on the data such as read, write, update etc. DBMS typically maintains the data on the system in a form of file.

What is a RDBMS (Relational Database Management System)?

RDBMS stores the data in form of tables, these tables are interconnected to each other which helps in identifying the relation between the data stored in different tables. It stores the data efficiently and the operations on the data stored in RDBMS are faster compared to the traditional file based data management system.

Difference between DBMS vs RDBMS

DBMS	**RDBMS**
Data is stored in a **files**.	Data is stored in a **tables**.
DBMS **doesn’t support Normalization**.	RDBMS **supports normalization** of tables, which reduces t data redundancy and avoid the database from multiple anomalies.
DBMS **doesn’t have a proper security** of the database.	RDBMS allows to set permissions on tables, which **prevents unauthorized access**. It also allows constraints to be set which make sure which data can be entered into the table.
In DBMS, data is stored in files so the data stored in different file is isolated and there is **no relation** between the data stored in different files.	In RDBMS, data is stored in tables and tables can have a relationship with other tables. This helps in identifying the **relationship between data stored** in different tables.
DBMS doesn’t support distributed database.	RDBMS supports distributed database.
Data redundancy is an issue in DBMS.	RDBMS removes data redundancy using normalization.
DBMS is suitable for small organization where data size is small and there is no need to scale the data in future.	RDBMS is suitable for large organizations where the size of the data is huge.
It support single user.	It supports multiple users.
Software and hardware requirements are low.	Software and hardware requirements are high since the size o the data is big.

DBMS	RDBMS
DBMS examples are: XML, MS Access etc.	RDBMS examples are: IBM Db2, Oracle, MySQL etc.

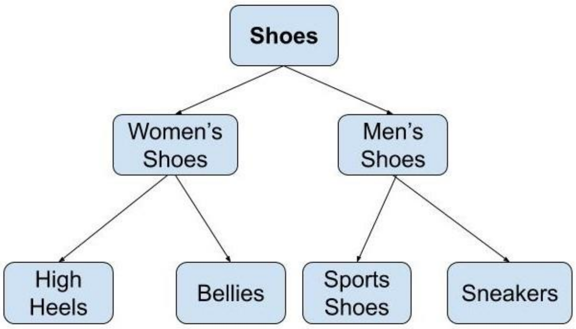
Data Model

Data Model gives us an idea that how the final system will look like after its complete implementation. It defines the data elements and the relationships between the data elements. Data Models are used to show how data is stored, connected, accessed and updated in the database management system.

Types of Data Model

Hierarchical Model

Hierarchical Model was the first DBMS model. This model organizes the data in the hierarchical tree structure. The hierarchy starts from the root which has root data and then it expands in the form of a tree adding child node to the parent node. This model easily represents some of the real-world relationships like food recipes, sitemap of a website etc. ***Example:*** We can represent the relationship between the shoes present on a shopping website in the following way:



Hierarchical Model

Features of a Hierarchical Model

1. ***One-to-many relationship:*** The data here is organized in a tree-like structure where the one-to-many relationship is between the datatypes. Also, there can be only one path from parent to any node. ***Example:*** In the above example, if we want to go to the node *sneakers* we only have one path to reach there i.e through men's shoes node.
2. ***Parent-Child Relationship:*** Each child node has a parent node but a parent node can have more than one child node. Multiple parents are not allowed.
3. ***Deletion Problem:*** If a parent node is deleted then the child node is automatically deleted.

Advantages of Hierarchical Model

- It is very simple and fast to traverse through a tree-like structure.

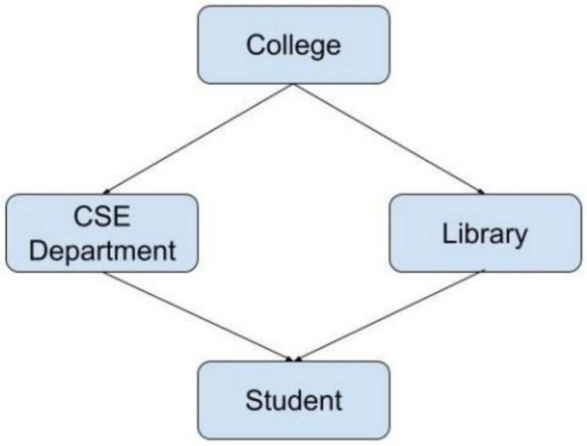
- Any change in the parent node is automatically reflected in the child node so, the integrity of data is maintained.

Disadvantages of Hierarchical Model

- Complex relationships are not supported.
- As it does not support more than one parent of the child node so if we have some complex relationship where a child node needs to have two parent node then that can't be represented using this model.
- If a parent node is deleted then the child node is automatically deleted.

Network Model

This model is an extension of the hierarchical model. It was the most popular model before the relational model. This model is the same as the hierarchical model, the only difference is that a record can have more than one parent. It replaces the hierarchical tree with a graph. ***Example:*** In the example below we can see that node student has two parents i.e. CSE Department and Library. This was earlier not possible in the hierarchical model.



Network Model

Features of a Network Model

1. ***Ability to merge more Relationships:*** In this model, as there are more relationships so data is more related. This model has the ability to manage one-to-one relationships as well as many-to-many relationships.
2. ***Many paths:*** As there are more relationships so there can be more than one path to the same record. This makes data access fast and simple.

Advantages of Network Model

- The data can be accessed faster as compared to the hierarchical model. This is because the data is more related in the network model and there can be more than one path to reach a particular node. So the data can be accessed in many ways.
- As there is a parent-child relationship so data integrity is present. Any change in parent record is reflected in the child record.

Disadvantages of Network Model

- As more and more relationships need to be handled the system might get complex. So, a user must be having detailed knowledge of the model to work with the model.

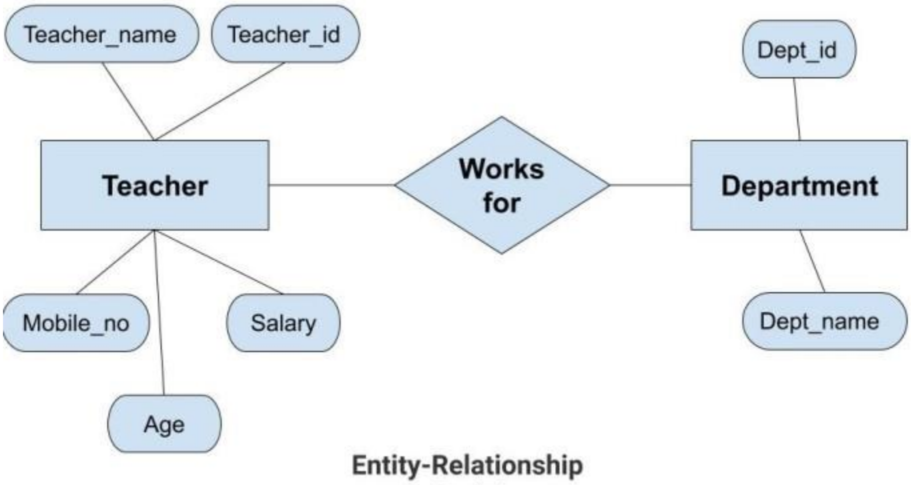
- Any change like updation, deletion, insertion is very complex.

Entity-Relationship Model

Entity-Relationship Model or simply ER Model is a high-level data model diagram. In this model, we represent the real-world problem in the pictorial form to make it easy for the stakeholders to understand. It is also very easy for the developers to understand the system by just looking at the ER diagram. We use the ER diagram as a visual tool to represent an ER Model. ER diagram has the following three components:

- ***Entities:*** Entity is a real-world thing. It can be a person, place, or even a concept. *Example:* Teachers, Students, Course, Building, Department, etc are some of the entities of a School Management System.
- ***Attributes:*** An entity contains a real-world property called attribute. This is the characteristics of that attribute. *Example:* The entity teacher has the property like teacher id, salary, age, etc.
- ***Relationship:*** Relationship tells how two attributes are related. *Example:* Teacher works for a department.

Example:



Entity-Relationship Model

In the above diagram, the entities are Teacher and Department. The attributes of ***Teacher*** entity are Teacher_Name, Teacher_id, Age, Salary, Mobile_Number. The attributes of entity ***Department*** entity are Dept_id, Dept_name. The two entities are connected using the relationship. Here, each teacher works for a department.

Features of ER Model

- ***Graphical Representation for Better Understanding:*** It is very easy and simple to understand so it can be used by the developers to communicate with the stakeholders.
- ***ER Diagram:*** ER diagram is used as a visual tool for representing the model.
- ***Database Design:*** This model helps the database designers to build the database and is widely used in database design.

Advantages of ER Model

- ***Simple:*** Conceptually ER Model is very easy to build. If we know the relationship between the attributes and the entities we can easily build the ER Diagram for the model.

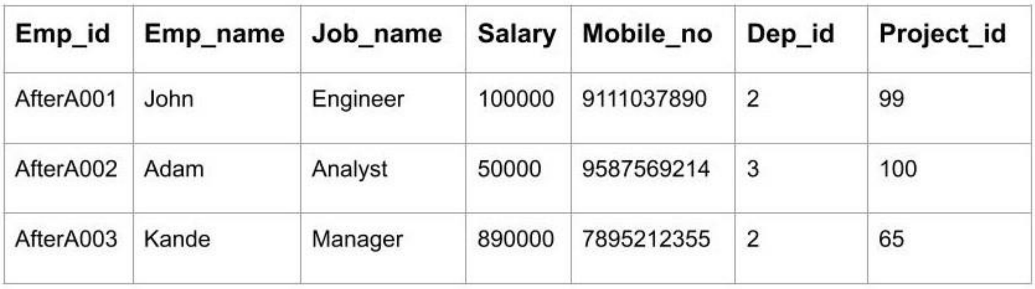
- ***Effective Communication Tool:*** This model is used widely by the database designers for communicating their ideas.
- ***Easy Conversion to any Model:*** This model maps well to the relational model and can be easily converted relational model by converting the ER model to the table. This model can also be converted to any other model like network model, hierarchical model etc.

Disadvantages of ER Model

- ***No industry standard for notation:*** There is no industry standard for developing an ER model. So one developer might use notations which are not understood by other developers.
- ***Hidden information:*** Some information might be lost or hidden in the ER model. As it is a high-level view so there are chances that some details of information might be hidden.

Relational Model

Relational Model is the most widely used model. In this model, the data is maintained in the form of a two-dimensional table. All the information is stored in the form of row and columns. The basic structure of a relational model is tables. So, the tables are also called *relations* in the relational model. ***Example:*** In this example, we have an Employee table.



Emp_id	Emp_name	Job_name	Salary	Mobile_no	Dep_id	Project_id
AfterA001	John	Engineer	100000	9111037890	2	99
AfterA002	Adam	Analyst	50000	9587569214	3	100
AfterA003	Kande	Manager	890000	7895212355	2	65

EMPLOYEE TABLE

Features of Relational Model

- ***Tuples:*** Each row in the table is called tuple. A row contains all the information about any instance of the object. In the above example, each row has all the information about any specific individual like the first row has information about John.
- ***Attribute or field:*** Attributes are the property which defines the table or relation. The values of the attribute should be from the same domain. In the above example, we have different attributes of the *employee* like Salary, Mobile no, etc.

Advantages of Relational Model

- ***Simple:*** This model is simpler as compared to the network and hierarchical model.
- ***Scalable:*** This model can be easily scaled as we can add as many rows and columns we want.
- ***Structural Independence:*** We can make changes in database structure without changing the way to access the data. When we can make changes to the database structure without affecting the capability to DBMS to access the data we can say that structural independence has been achieved.

Disadvantages of Relational Model

- ***Hardware Overheads:*** For hiding the complexities and making things easier for the user this model requires more powerful hardware computers and data storage devices.

- ***Bad Design:*** As the relational model is very easy to design and use. So the users don't need to know how the data is stored in order to access it. This ease of design can lead to the development of a poor database which would slow down if the database grows.

KEYS IN DBMS

Keys are one of the basic requirements of a relational database model. It is widely used to identify the tuples (rows) uniquely in the table. We also use keys to set up relations amongst various columns and tables of a relational database.
A key in DBMS is an attribute or a set of attributes that help to uniquely identify a tuple (or row) in a relation (or table). Keys are also used to establish relationships between the different tables and columns of a relational database. Individual values in a key are called key values.

TYPES OF KEYS IN DBMS

What is a Primary Key?

PRIMARY KEY in DBMS is a column in a table that uniquely identify every row in that table. The Primary Key can't be a duplicate meaning the same value can't appear more than once in the table. A table cannot have more than one primary key.
Rules for defining Primary key:

- Two rows can't have the same primary key value
- It is must for every row to have a primary key value.
- The primary key field cannot be null.
- The value in a primary key column can never be modified or updated if any foreign key refers to that primary key.

Example:

In the following example, StudID is a Primary Key.

StudID	Roll No	First Name	LastName	Email
1	11	Tom	Price	abc@gmail.com
2	12	Nick	Wright	xyz@gmail.com
3	13	Dana	Natan	mno@yahoo.com

What is the Alternate key?

ALTERNATE KEYS is a column or group of columns in a table that uniquely identify every row in that table. A table can have multiple choices for a primary key but only one can be set as the primary key. All the keys which are not primary key are called an Alternate Key.
Example:

In this table, StudID, Roll No, Email are qualified to become a primary key. But since StudID is the primary key, Roll No, Email becomes the alternative key.

StudID	Roll No	First Name	LastName	Email
1	11	Tom	Price	abc@gmail.com
2	12	Nick	Wright	xyz@gmail.com
3	13	Dana	Natan	mno@yahoo.com

What is a Candidate Key?

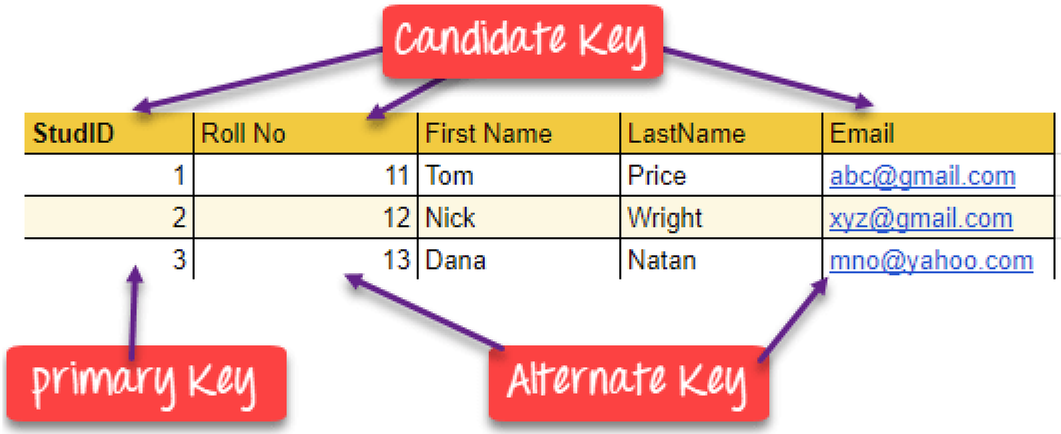
CANDIDATE KEY in SQL is a set of attributes that uniquely identify tuples in a table. Candidate Key is a super key with no repeated attributes. The Primary key should be selected from the candidate keys. Every table must have at least a single candidate key. A table can have multiple candidate keys but only a single primary key.

Properties of Candidate key:

- It must contain unique values
- Candidate key in SQL may have multiple attributes
- Must not contain null values
- It should contain minimum fields to ensure uniqueness
- Uniquely identify each record in a table

Candidate key Example: In the given table Stud ID, Roll No, and email are candidate keys which help us to uniquely identify the student record in the table.

StudID	Roll No	First Name	LastName	Email
1	11	Tom	Price	abc@gmail.com
2	12	Nick	Wright	xyz@gmail.com
3	13	Dana	Natan	mno@yahoo.com



StudID	Roll No	First Name	LastName	Email
1	11	Tom	Price	abc@gmail.com
2	12	Nick	Wright	xyz@gmail.com
3	13	Dana	Natan	mno@yahoo.com

Candidate Key in DBMS

What is the Foreign key?

FOREIGN KEY is a column that creates a relationship between two tables. The purpose of Foreign keys is to maintain data integrity and allow navigation between two different instances of an entity. It acts as a cross-reference between two tables as it references the primary key of another table.

Example:

DeptCode	DeptName
001	Science
002	English
005	Computer

Teacher ID	Fname	Lname
B002	David	Warner
B017	Sara	Joseph
B009	Mike	Brunton

In this key in dbms example, we have two table, teacher and department in a school. However, there is no way to see which teacher work in which department.

In this table, adding the foreign key in Deptcode to the Teacher name, we can create a relationship between the two tables.

Teacher ID	DeptCode	Fname	Lname
B002	002	David	Warner
B017	002	Sara	Joseph
B009	001	Mike	Brunton

This concept is also known as Referential Integrity.

Note: Candidate keys are chosen from super keys and one of these candidate keys will further become *Primary Key*. ***The primary key selection is done by the Data Base Administrator according to the frequency of queries.***

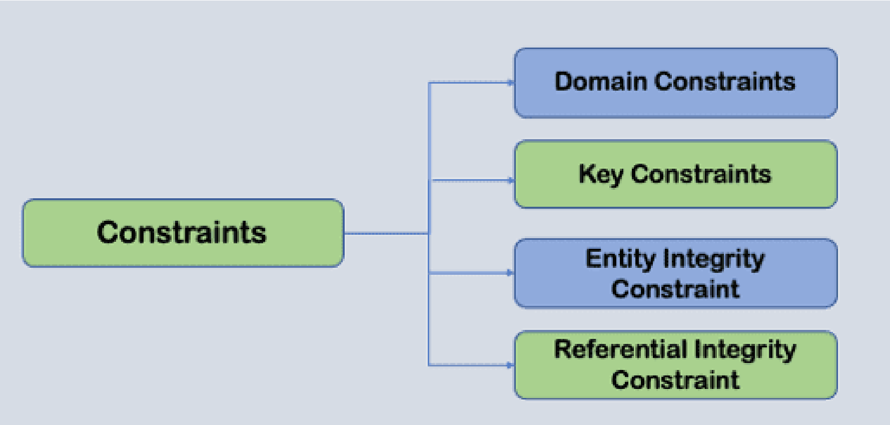
The candidate key also helps in determining prime and non-prime attributes. The columns present in a candidate key are known as prime attributes in DBMS, and the columns that are not present in any candidate key are called non-prime attributes.

Constraints:

Constraints are the rules that we can apply on the type of data in a table. Constraints allow us to limit the data that can be stored within a column of a table. You can use constraints to limit the data that is stored in a particular column, a whole table.

- Constraints ensure that data entered by the user into columns must be within the criteria specified by the condition
- For example, if you want to maintain only unique IDs in the employee table or if you want to enter only age under 18 in the student table etc

Types of Constraints In DBMS



1. Domain Constraints in DBMS

Domain constraint puts constraints on domain or set values for an attribute. It states that the attribute value must be the atomic value of its domain.

ID (*Primary Key)	Department ID	Marks
1	10	70
2	11	80
3	12	90

4	10	**Unknown**

In this Student Table '***Unknown*** 'value violates domain constraints as in marks attribute only numbers are allowed.

2. Key Constraints in DBMS

There are a number of key constraints in dbms that ensure that an entity or record is uniquely or differently identified in the database.

Types of key constraints in DBMS

- NOT NULL: ensures that the specified ***column doesn't contain a NULL value.***
- UNIQUE: ***provides a unique/distinct values*** to specified columns.
- DEFAULT: ***provides a default value to a column*** if none is specified.
- CHECK: checks ***for the predefined conditions before inserting*** the data inside the table.
- PRIMARY KEY: it ***uniquely identifies a row*** in a table.
- FOREIGN KEY: ensures ***referential integrity*** of the relationship

3. Entity Integrity Constraint in DBMS

The entity integrity constraint states that any attribute of primary keys must not contain a null value in any relation and should be unique. This is due to the fact that it is the key that is employed to determine specific rows within a relation. And if the primary key is an empty value, it is impossible to recognize those rows.

ID (*Primary Key)	Department ID	Marks
1	10	70
2	11	80
3	12	90
	10	100

Id is Primary key thus 4th row violates Entity Integrity Constraint as id is null for this record.

4. Referential Integrity Constraint in DBMS

This restriction is enforced when a foreign reference refers to the primary key of another relation. It stipulates that all values taken by the foreign keys must be either available in another relation to the primary key.

Student Table

ID (*Primary Key)	Department ID (*Foreign Key)	Marks
1	10	70
2	11	80

3	**14**	90
4	10	100

Department Table

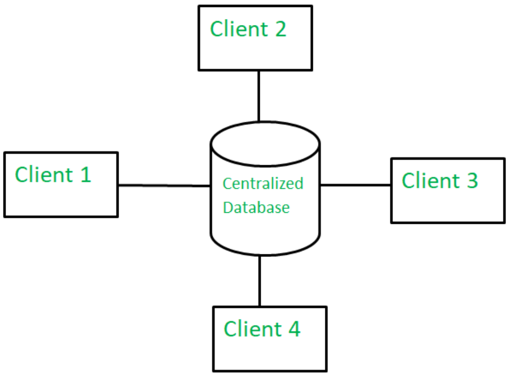
Department ID (*Primary Key)	Name
10	CSE
11	ME
12	ECE
13	Civil

In this above example Student table is linked to Department table via Department id but department id which is foreign key doesn't have 14 in department table thus violating Referential Integrity Constraint.

TYPES OF DATABASE

1. Centralized Database:

A centralized database is basically a type of database that is stored, located as well as maintained at a single location only. This type of database is modified and managed from that location itself. This location is thus mainly any database system or a centralized computer system. The centralized location is accessed via an internet connection (LAN, WAN, etc). This centralized database is mainly used by institutions or organizations.



Advantages:

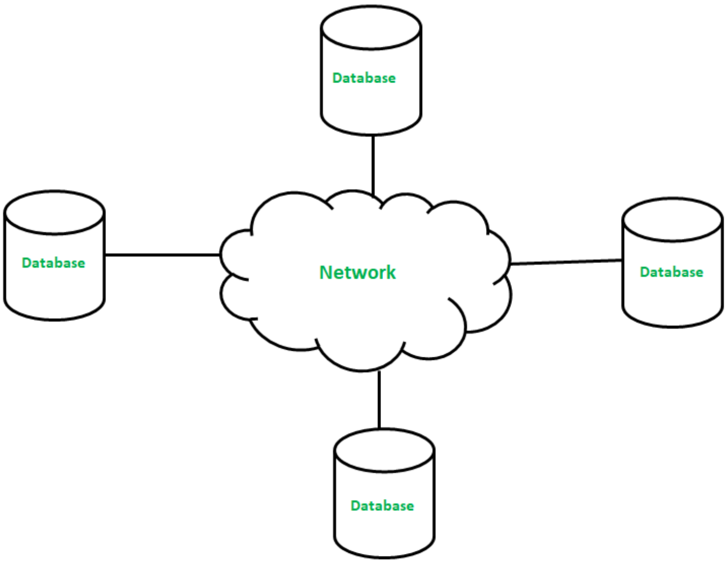
- Since all data is stored at a single location only thus it is easier to access and coordinate data.
- The centralized database has very minimal data redundancy since all data is stored in a single place.
- It is cheaper in comparison to all other databases available.

Disadvantages:

- The data traffic in the case of a centralized database is more.
- If any kind of system failure occurs in the centralized system then the entire data will be destroyed.

2. Distributed Database:

A distributed database is basically a type of database which consists of multiple databases that are connected with each other and are spread across different physical locations. The data that is stored in various physical locations can thus be managed independently of other physical locations. The communication between databases at different physical locations is thus done by a computer network.



Advantages:

- This database can be easily expanded as data is already spread across different physical locations.
- The distributed database can easily be accessed from different networks.
- This database is more secure in comparison to a centralized database.

Disadvantages:

- This database is very costly and is difficult to maintain because of its complexity.
- In this database, it is difficult to provide a uniform view to users since it is spread across different physical locations.

Examples of centralized databases

Examples of a centralized database are a desktop or server CPU or mainframe computer that users access through a computer network such as a LAN or WAN.

Examples of distributed databases

Some common examples of distributed databases include:

- Apache Ignite
- Apache Cassandra
- Apache HBase
- Couchbase Server
- Amazon SimpleDB
- Clusterpoint
- FoundationDB

Who is a Database Administrator?

A database administrator, or DBA, is someone who is in charge of making sure a database runs smoothly. As a challenging role that requires focus, logic, and an enthusiastic personality that can cope under pressure, the job necessitates a variety of skills. DBAs must work within an organization to monitor, repair, and develop databases.

This job necessitates a high level of expertise from a single person or group of people. Most database administrators are trained to diagnose the system-wide database and repair any issues that arise to ensure that the data remains consistent and well-defined

Typical duties include:

- Managing database access and permissions.
- Ensuring that databases meet users' requirements.
- Coordinating with programmers, applications/operational staff, IT project managers and other technical staff.
- Reviewing and managing database security, integrity and backup procedures.
- Writing operating manuals and providing training and support for database use.
- Implementing processes for resolving faults and data loss.
- Writing disaster recovery plans.
- Archiving data.

SKILLS

A DBA should have the following abilities:

- Exceptional problem-solving and analytical abilities
- Good communication.
- Teamwork, and negotiation skills
- Good organizational skills
- Understanding of the major data manipulation languages as well as database design principles.
- The ability to work under pressure and to meet tight deadlines.
- Adaptability and flexibility
- A dedication to ongoing professional development
- The ability to establish and maintain positive working relationships with coworkers and customers
- Business awareness and comprehension of IT business requirements
- Ability to keep up with new technology developments
- Working knowledge of information legislation, such as the Data Protection Act