What is Data?

The facts that can be recorded and which has implicit meaning known as 'Data'
Example: Student Entity.
1. Student Name
2. Student Roll Number
3. Student Address

Database

- It's a collection of interrelated data.
- These can be stored in the form of tables.
- Example: Student database consist the field as StudentName, StudentRollNumber and StudentAddress

File System

- File system basically a way of arranging the files in a storage medium like hard disk
- File system consists of different files which are grouped in to directories.
- The directories further contain other folders and files

Database Management System

- Database management system is a software which is used to manage the database. For example: MySQL, Oracle etc.
- DBMS provides interfaces to perform various operations like database creation, storing data in it, updating data, creating a table in the database and lot more.
- It provides protection and security to the database

Advantages of DBMS

- Controls Database redundancy
- Data sharing
- Easily maintenance
- Reduce time.
- Back Up
- Multiple user interfaces

Disadvantages of DBMS

- Cost of hardware and software
- Complexity.
- Dependence on technology.
- Performance issues.
- Data integration.
- Security.
- Data loss/corruption.

Different types of Database users

- Database administrators: is a person or a team who manages the database. They are able to create new id and password for the user need access to the database. One who provides security to the database
- Naïve / Parametric end users: The users doesn't have DBMS knowledge but they frequently use database applications in their daily life to get the desired results. Eg: Railway ticket booking user or Clerk in a bank.
- System analyst: User analyze the requirement of end users. They check all the requirements are satisfied or not.
- Sophisticated users can be the engineers, scientists, business analysts who are familiar with the database. They can develop their own database applications according to the requirement
- Database designers: they are the users who design the structure of the database which includes tables, indexes, views, constraints, triggers, stored procedures.
- Application programmers: back end programmers who write code for the application programs. They are computer professionals.

Instances and schema

- Instances: Instances are the collection of information stored in a particular moment.
- Schema: Schema is the overall description of the database. The basic structure of data will be stored in the database is called schema

Database Objects in DBMS

A database object is any defined object in a database that is used to store or reference data. Anything which we make from create command is known as Database Object. It can be used to hold and manipulate the data.

Some of the examples of database objects are:

- Table Basic unit of storage; composed rows and columns
- View Logically represents subsets of data from one or more tables
- Sequence - Generates primary key values
- Index - Improves the performance of some queries
- Synonym Alternative name for an object

Syntax for creating a table:

CREATE TABLE [schema.] table (column datatype [DEFAULT expr][, ...]);

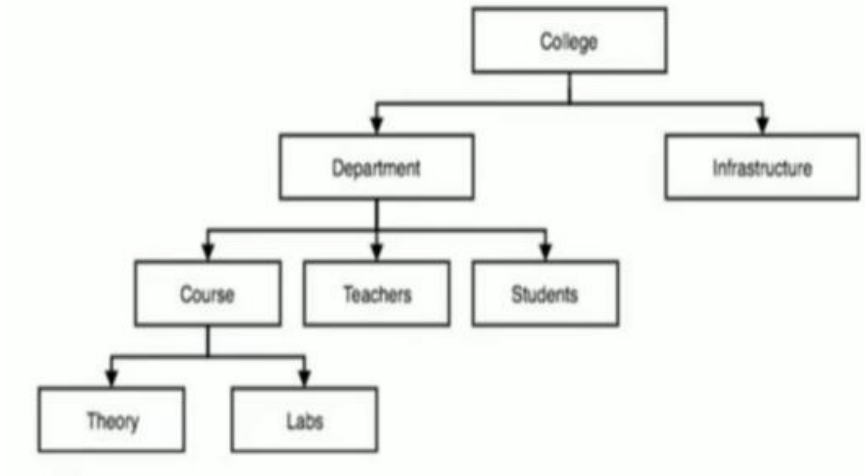
DBMS Database models

A database model defines the logical design and structure of a database and it defines how the data will be stored, accessed, and updated in a database management system.

- Hierarchical Model
- Network Model
- Entity- Relationship Model
- Relational model

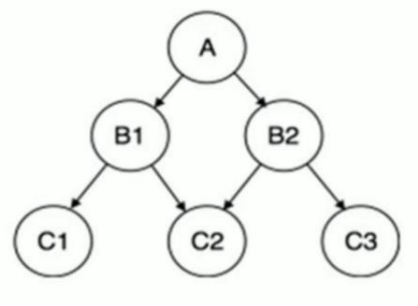
Hierarchical Model

- This database model organizes data in to a tree like structure, with a single root, to
- which all the other data is linked.
- The hierarchy starts from the Root data and expands like a tree, adding child nodes to the parent nodes.
- In this model, a child node will only have a single parent node.
- This model efficiently describes many real world relationships like index of a book recipes etc.



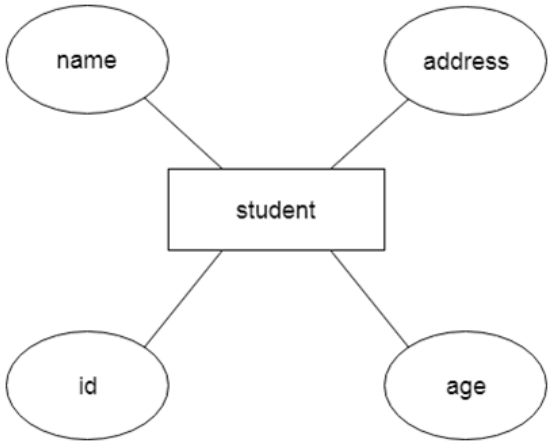
Network Model

- This is an extension of Hierarchical model.
- In this model data is organized more like a graph and are allowed to have more than one parent node.
- This database model was used to map many-to-many data relationships.
- This was the most widely used database mode, before relational Model was introduced.



ER Model

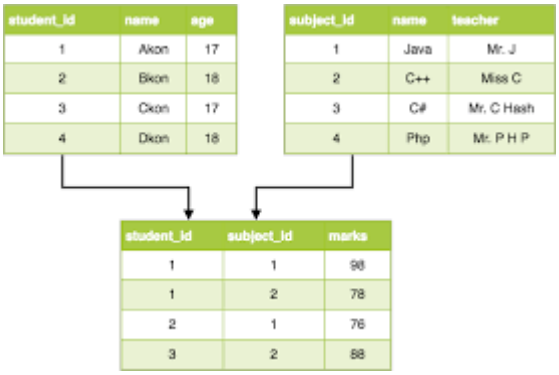
- In this database model, relationships are created by dividing the object of interest in to entity and its characteristics into attributes.
- Different entities are related using relationships.
- Entity- An entity in an ER model is a real world entity having the properties called attributes.
- Relationship: the logical association among entities is called relationship.



Relational Model

- In this model, data is organized in two-dimentional tables and the relationship is maintained by storing a common field.
- The basic structure of data in the relational model in tables.

- Tables are also known as relations in relational model and rows are called tuples.



student_id	name	age
1	Akon	17
2	Bkon	18
3	Ckon	17
4	Dkon	18

subject_id	name	teacher
1	Java	Mr. J
2	C++	Miss C
3	C#	Mr. C Hash
4	Php	Mr. P H P

student_id	subject_id	marks
1	1	98
1	2	78
2	1	76
3	2	88

Keys in DBMS

- KEYS are attributes or sets of attributes that enable the identification of a row or tuple within a relation or table.
- They establish connections between different tables and assist in uniquely identifying a row by
- utilizing one or more columns in the table.
- Keys play a vital role in locating distinct records or rows within a table, and they serve as a means to find a unique record or row within the database.

Employee

ID	Name	SSN	Salary	Phone	Email
101	John	AA	50000	12	j@sw
102	Robin	BB	60000	13	r@yh
103	Alya	CC	35000	14	a@hm
104	Yusuf	DD	68000	15	y@ch
105	John	EE	62000	89	j@in
106	Raj	FF	45000	87	r@au
107	Jayant	GG	25000	45	j@us
108	John	HH	35000	15	j@de
109	Neil	II	25000	12	n@uk

Types of Keys in DBMS

1. Primary Key: This type of keys in DBMS refers to a column that uniquely identifies all the records within that table. A table has one primary key only, and this key must not contain repeated or duplicated values across its rows. Each value within the primary key must be unique, with no repetitions allowed.

Properties of a Primary Key:

- The Primary Key field shouldn't be left NULL; the Primary Key column must contain a value.
- In that column, no two rows in the table may contain identical values.

- If a foreign key in a DBMS refers to the primary Key, no value may be altered or modified in this primary key column.

2. Candidate keys: play a vital role in upholding the integrity and consistency of a database. The purpose of this key in DBMS is to guarantee each row's uniqueness and independent identification within a table. Additionally, candidate keys enforce relationships between tables, ensuring data integrity and maintaining overall database consistency.

Properties of a candidate key

- It must have distinct values.
- A candidate key in SQL can have a variety of qualities, but it must not include null values and must have at least the bare minimum of fields to guarantee uniqueness. Create a unique identifier for each table record.

3. Super Key: The collection of all keys enabling us to recognize every row in the table is a super key. This type of key in DBMS specifies that all the table columns that may identify the columns uniquely function as the super keys.

Properties of a super key

- A super key must ensure that each record in a table is unique.
- Nevertheless, the minimal collection of characteristics that can guarantee uniqueness should be a super key.
- Multiple subsets that are likewise regarded as super keys can exist for a super key.

4. Foreign Key: To create connections between two accessible tables, we employ a foreign key. Every value in a column or collection must match the primary key in the referential table for the foreign key to function. We can preserve referential integrity and data integrity with the aid of a foreign key.

Properties of a foreign key

- It is a key that serves as both a secondary key and a primary key in two different tables.
- At any given time, it combines two or more relations.
- They serve as cross-references for the tables.

5. Alternate Key: A key in DBMS might be selected as the main Key in a table in multiple ways. Any key that has the potential to replace the primary key but is not yet the primary key is considered an alternate key. It's a prospective main key that hasn't been selected yet.

Properties of an alternate key

- Alternate keys refer to all keys that are not main keys.
- It's a backup key.
- It has two or more fields that allow it to recognize two or more records. These criteria are reiterated.

Data Normalization

Data normalization is a fundamental concept in database management systems (DBMS) that aims to reduce data redundancy and improve data integrity by organizing data into well-structured tables. The normalization process involves breaking down large tables into smaller, more manageable ones and establishing relationships between them.

Here's an overview of the normalization process typically divided into different normal forms:

First Normal Form (1NF):

Ensures that each column in a table contains atomic (indivisible) values and there are no repeating groups of columns.

Example: Breaking down a table containing information about students and their courses into two separate tables, one for students and one for courses, with a relationship established between them.

Second Normal Form (2NF):

Requires that a table is in 1NF and all non-key attributes are fully functional dependent on the primary key.In simpler terms, each non-key attribute must depend on the entire primary key, not just part of it.

Example: If a table has a composite primary key (made up of multiple columns), ensure that each non-key attribute depends on the entire composite key, not just one part of it.

Third Normal Form (3NF):

Builds on 2NF by ensuring that no transitive dependencies exist in the table. A transitive dependency occurs when a non-key attribute depends on another non-key attribute rather than the primary key.

Example: If a table contains attributes like student ID, course ID, and instructor name, ensure that instructor name depends only on the course ID and not on the student ID.

Fourth Normal Form (4NF) and Beyond:

These forms address more complex dependencies such as multi-valued dependencies and join dependencies. They are less commonly encountered in typical database design scenarios but may be relevant for specialized cases.

The normalization process aims to eliminate anomalies such as insertion, deletion, and update anomalies by ensuring that data is organized efficiently and redundancies are minimized. It helps maintain data integrity and facilitates easier maintenance and querying of the database.

Data Redundancy

Data redundancy in a database management system (DBMS) refers to the duplication of data within the database. It occurs when the same piece of data is stored in multiple places unnecessarily. While some redundancy might be unavoidable or even intentional for performance or other reasons, excessive redundancy can lead to several problems:

Wasted Storage Space: Redundant data occupies extra storage space within the database. This not only increases the storage requirements but also incurs additional costs, especially in scenarios where databases are stored on expensive storage systems.

Data Inconsistency: When the same data is stored in multiple places, there is a risk of inconsistencies arising if the data is updated in one place but not in others. Inconsistencies can lead to incorrect query results and undermine the integrity of the database.

Update Anomalies: Redundant data can result in update anomalies, where updating a piece of data in one place requires updates in multiple places. Failure to update all instances of redundant data can lead to inconsistencies and errors.

Dividing SQL into Sublanguages

1. DDL - Data Definition Language
2. DQL Data Query Language
3. DML - Data Manipulation Language
4. DCL Data Control Language
5. TCL Transaction Control Language

Dividing SQL into Sublanguages

Data Query Language (DQL): The Data Query Language is the sublanguage responsible for reading, or querying, data from a database. In SQL, this corresponds to the SELECT

- SELECT Name FROM Accounts;

Data Manipulation Language (DML): The Data Manipulation Language is the sublanguage responsible for adding, editing or deleting data from a database. In SQL, this corresponds to the INSERT, UPDATE, and DELETE

- INSERT INTO Accounts (Name, Balance) VALUES ('Evan Johnson', 3200);
- UPDATE Accounts SET Balance = 3000 WHERE Name = 'Evan Johnson';
- DELETE FROM Accounts WHERE Name = 'Evan Johnson';

Data Definition Language (DDL): The Data Definition Language is the sublanguage responsible for defining the way data is structured in a database. In SQL, this corresponds to manipulating tables through the

- CREATE TABLE, ALTER TABLE, and DROP TABLE

Data Control Language (DCL): The Data Control Language is the sublanguage responsible for the administrative tasks of controlling the database itself, most notably granting and revoking database permissions for users. In SQL, this corresponds to the GRANT, REVOKE, and DENY commands, among others.

- GRANT SELECT TO 'mark'@'localhost'
- DENY SELECT FROM 'mark'@'localhost'
- REVOKE SELECT FROM 'mark'@'localhost'