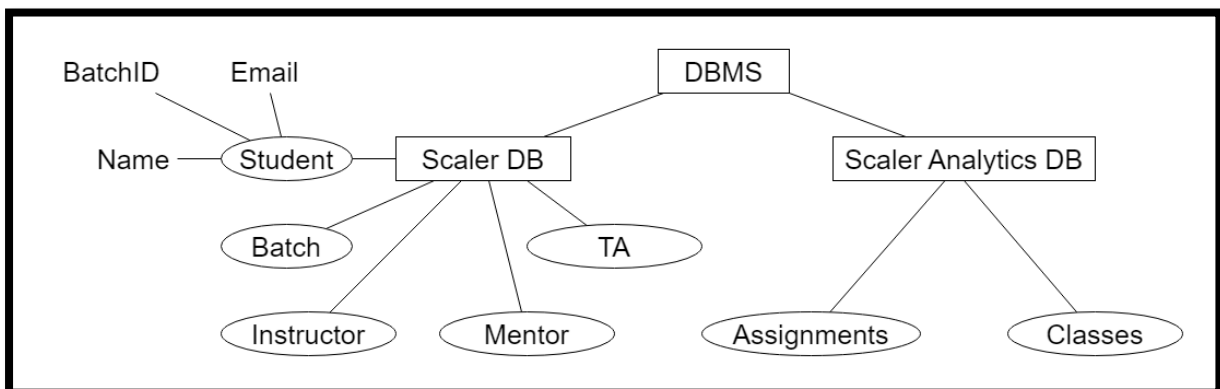


Airbase: A place to store aircrafts.

Database: A place to related data.



Entity: An entity is a real-world object or concept that has a distinct identity and can be uniquely identified.

In a database, an entity is often represented as a table.

Each row in the table represents an instance of the entity, and each column represents an attribute or property of the entity.

Attributes: Attributes are characteristics or properties of an entity.

For example, in a database for a "Person" entity, attributes could include "Name," "Age," "Address," etc.

Attributes are used to describe the details of each entity instance.

Q1: Is file system a DBMS?

Answer: File system is considered as a primitive database but it has lots of cons.

Pros

Available out of box.

Not complex.

Cons

Querying is difficult.

Security is a concern. Anyone can open the file system and read the confidential information.

Joining is not possible.

Redundant data. If an instructor leaves the scaler, we delete his information from the instructor table but it persists in the batch table.

No structuring for data.

Concurrency not possible.

DBMS

DBMS stands for "Database Management System." It is a software application or system that allows users to interact with a database. DBMS serves as an intermediary between the database and the users or applications, providing an organized and efficient way to store, retrieve, manage, and manipulate data.

Here are some key aspects of DBMS:

Data Storage: DBMS stores data in a structured manner, often using tables or other data structures. This structured storage allows for easy retrieval and organization of data.

Data Retrieval: Users can query the database to retrieve specific data using query languages like SQL (Structured Query Language). DBMS handles the translation of user queries into actions on the database.

Data Manipulation: DBMS enables users to add, update, and delete data in the database while maintaining data integrity and consistency.

Data Security: DBMS provides access control and security features to protect the data from unauthorized access and ensure data privacy.

Data Integrity: It enforces data integrity constraints to maintain the accuracy and consistency of data in the database.

Concurrency Control: DBMS manages simultaneous access to the database by multiple users or applications, ensuring that transactions do not interfere with each other.

Backup and Recovery: DBMS includes backup and recovery mechanisms to safeguard data against loss or corruption.

Scalability: DBMS systems can scale to handle large volumes of data and users. They are used in various environments, from small businesses to large enterprises.

Common examples of DBMS systems include:

Relational Database Management Systems (RDBMS): These are based on the relational data model and use tables to store data. Examples include MySQL, PostgreSQL, Oracle Database, and Microsoft SQL Server.

NoSQL Database Management Systems: These are designed for handling unstructured or semi-structured data. Examples include MongoDB, Cassandra, and Redis.

NewSQL Database Management Systems: These combine the benefits of traditional RDBMS with the scalability and performance of NoSQL databases. Examples include Google Spanner and CockroachDB.

In-Memory Database Management Systems: These store data in the computer's main memory (RAM) for faster data access. Examples include Redis and Memcached.

Important terms related with RDBMS

Relation: Table

Attribute: Column

Tuple: Row

Relationships: Relationships define how entities are related to each other within the database. For example, in a university database, the "Student" entity might have a relationship with the "Course" entity through the "Enrollment" relationship, indicating that students are enrolled in courses. Relationships help establish the connections and associations between entities.

Cardinality: Cardinality specifies the number of instances of one entity that can be related to the number of instances of another entity through a relationship. Common cardinality types include "one-to-one," "one-to-many," and "many-to-many."

Attributes vs. Relationships: Attributes describe the properties of an entity instance, while relationships describe how entities are connected or associated.

Degree: Number of columns.

Cardinality: Number of rows.

SQL: Structured Query Language

It is used to query RDBMS. It allows us to do crud operations.

1. Each row must be unique. At least one value should be different.

Student

Name	Marks
Ankit Kumar	90
Ankit Kumar	90

Student

ID	Name	Marks
1	Ankit Kumar	90
2	Ankit Kumar	90

2. Values in the cell should be atomic i.e. single entity.

Student

ID	Name	Marks	Phone
1	Ankit Kumar	90	x, y, z
2	Ankit Kumar	90	a

Student-Phone

SID	Phone
1	X
1	y
1	z
2	a

3. Order of columns does not matter.
4. Order of rows does not matter.

Properties of RDBMS:

Primary Key: Each table in an RDBMS typically has one or more columns designated as the primary key. The primary key ensures that each row in the table is uniquely identified. No two rows can have the same primary key value.

Uniqueness: The values in the primary key column(s) must be unique across all rows in the table. This means that no duplicate values are allowed in the primary key column(s).

All Rows Should Be Unique: In an RDBMS, each row within a table must have a unique identifier. This uniqueness is typically enforced through the use of a primary key, which ensures that no two rows in the table have the same key value. This uniqueness is a crucial property for data integrity and for establishing relationships between tables.

A Cell (Attribute) Cannot Be Multivalued: In the relational model, each attribute (column) in a table should contain only a single value for each

row. This property is known as atomicity, and it ensures that data is well-structured and organized. Each cell in a table holds a single, indivisible piece of data. For example, if you have a "Birthdate" column, each cell in that column should contain only one date value.

Data Integrity: The enforcement of primary key constraints ensures data integrity within the database. It prevents the insertion of duplicate data into the table.

Indexing: RDBMS systems often create an index on the primary key column(s) to facilitate faster data retrieval. This index speeds up queries that involve searching for specific rows by their primary key values.

Foreign Keys: Foreign keys are used to establish relationships between tables in a relational database. A foreign key in one table typically references the primary key of another table. This relationship ensures data consistency and integrity.

Join Operations: The uniqueness of primary keys simplifies join operations. Joining tables based on their primary key and foreign key relationships is a common practice in querying relational databases.

Referential Integrity: RDBMS systems enforce referential integrity, which means that foreign key values must correspond to valid primary key values in the referenced table. This prevents orphaned or inconsistent data.

Cascading Actions: RDBMS systems allow you to define cascading actions for foreign keys. For example, you can specify that when a referenced row is deleted, all related rows in other tables should also be deleted (CASCADE), or you can set actions like SET NULL or SET DEFAULT.

Normalization: Normalization is a process in RDBMS design that aims to minimize data redundancy and dependency. Primary keys play a crucial role in normalizing the database schema.

The concept of RDBMS is inspired from

Mathematical Foundations: Dr. Codd's work was heavily influenced by mathematical set theory and logic. He formulated a set of mathematical rules and principles for managing data in a way that eliminated data redundancy and ensured data integrity.

Data Structures and Algorithms: HashSet<Tuple>

Keys

"keys" refer to attributes or sets of attributes that are used to uniquely identify and organize data within a database. Keys play a vital role in ensuring data integrity, enabling efficient data retrieval, and establishing relationships between tables.

Types of keys

Attributes	Keys	Super Keys	Candidate Key	Primary Key
Fname	X	X	X	X
Fname, Lname	X	X	X	X
Fname, Email	✓	✓	X	X
Fname, Lname, Email	✓	✓	X	X
Email	✓	✓	✓	X
Email, Phone	✓	✓	X	X
Fname, Lname, Email, Phone	✓	✓	X	X
Phone	✓	✓	✓	X
ID	✓	✓	✓	✓

Super key: Super key is just pet name of Keys.

Candidate Key: Minimal Super key is a candidate key.

Primary Key: Any key which is chosen by the DB Architect is called primary key.

Note:

1. We should not pick attribute of an entity as primary key because phone and email can be change by the user.
2. Strings occupy more memory.
3. Strings matching is heavy operation.

Foreign Key: A foreign key is a column or set of columns in one table that refers to the primary key of another table. It establishes relationships between tables, defining how data in one table relates to data in another. Foreign keys enforce referential integrity, ensuring that data remains consistent across related tables.

Students

ID	Name	Phone
1	Sumit	123, 456
2	Rahul	789

Students

ID	Name
1	Sumit
2	Rahul

Student_Phone

SID	Phone
1	123
1	456
2	789

Composite key:

A composite key is a key that consists of two or more columns combined to uniquely identify rows. While individual columns may not be unique on their own, their combination in the composite key is unique.

Mathematically, we can say that

Primary key Size > 1