What is a Database?

A database is a shared collection of logically related data designed to meet the information needs of an organization. Consider a franchise company, for example. The franchise stores various pieces of information such as customer details, inventory data, employee records, and sales figures. This data is logically related and stored in a database to facilitate easy access, management, and analysis.

Uses of Databases

Data Storage: Databases are used to store vast amounts of data in an organized manner, ensuring data can be retrieved efficiently.

Data Analysis: Businesses use databases to analyze trends and patterns in their data, aiding in decision-making processes.

Keeping Records: Databases are essential for maintaining records of transactions, inventory, employee details, and other critical information.

Web Apps: Modern web applications rely heavily on databases to store user information, content, and other data needed to operate effectively.

CRUD Operations in Databases

CRUD stands for Create, Read, Update, and Delete. These operations are fundamental to interacting with a database.

- **Create:** Adding new records to the database.
- **Read:** Retrieving data from the database.
- **Update:** Modifying existing records in the database.
- **Delete:** Removing records from the database.

What If All the Data Stored in the Servers Is Lost?

The loss of all data stored in servers can be catastrophic, leading to operational disruptions, financial losses, and reputational damage. For instance, in 2017, Equifax suffered a data breach that exposed sensitive information of over 140 million customers, leading to severe consequences.

Properties of a Database

Integrity: Ensures accuracy and consistency of data.

Security: Protects data from unauthorized access and breaches.

Availability: Ensures data is accessible when needed.

Concurrency: Manages simultaneous data access to ensure data integrity.

Independence of Apps: Data is independent of the applications that use it, allowing for flexibility and scalability.

Difference Between SQL and NoSQL

SQL Databases: Relational databases that use structured query language (SQL) for defining and manipulating data. They are table-based and suitable for complex queries.

NoSQL Databases: Non-relational databases that store data in various formats like documents, key-value pairs, graphs, or wide-column stores. They are designed for large-scale data storage and real-time web applications.

What is a DBMS?

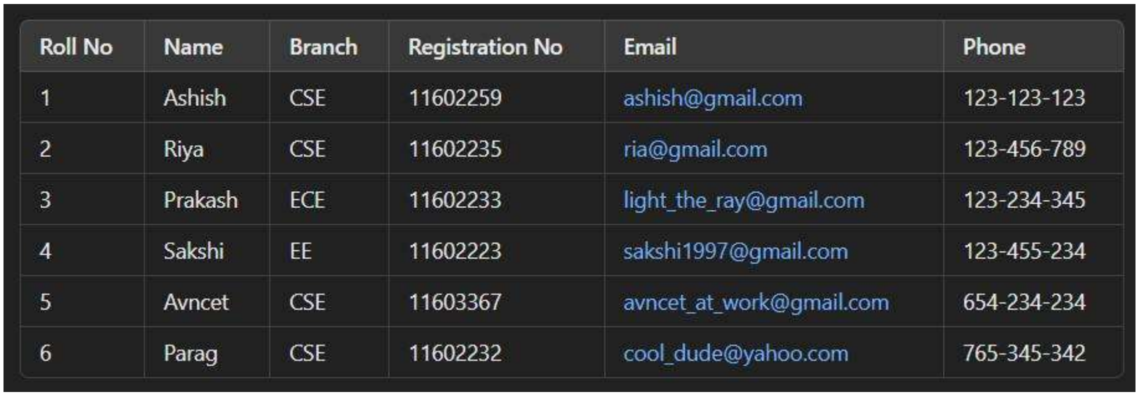
A Database Management System (DBMS) is software that interacts with users, applications, and the database itself to capture and analyze data. It provides tools to define, manipulate, retrieve, and manage data in a database.

Schema

A schema is the logical structure of a database, defining how data is organized and how the relations among them are associated. It includes definitions of tables, fields, relationships, views, indexes, and other elements.

Keys in Databases

Keys are fundamental to database design as they ensure that each record within a table can be uniquely identified. They enforce integrity and establish relationships between tables. Let's explore different types of keys using the following data table:



Roll No	Name	Branch	Registration No	Email	Phone
1	Ashish	CSE	11602259	ashish@gmail.com	123-123-123
2	Riya	CSE	11602235	ria@gmail.com	123-456-789
3	Prakash	ECE	11602233	light_the_ray@gmail.com	123-234-345
4	Sakshi	EE	11602223	sakshi1997@gmail.com	123-455-234
5	Avncet	CSE	11603367	avncet_at_work@gmail.com	654-234-234
6	Parag	CSE	11602232	cool_dude@yahoo.com	765-345-342

Types of Keys

Super Key:

A set of one or more columns that can uniquely identify a record in a table.

Example: Roll No, Registration No, Email, Phone, Roll No + Name, Roll No + Branch.

Candidate Key:

A minimal super key; i.e., a super key with no redundant attributes.

Example: Roll No, Registration No.

Primary Key:

A candidate key chosen to uniquely identify records in a table. It cannot contain NULL values.

Example: Registration No.

Alternate Key:

Candidate keys that are not chosen as the primary key.

Example: Roll No., Branch, Email, Phone, Name etc.

Composite Key:

A key that consists of two or more attributes that uniquely identify a record.

Example: Roll No + Email, Roll No + Name + Branch + Registration No + Phone.

Foreign Key:

A key used to link two tables. It is a field (or collection of fields) in one table that refers to the primary key in another table.

Example: If there were another table for departments, Branch in the Employees table could be a foreign key linking to the Departments table.

Cardinality of Relations

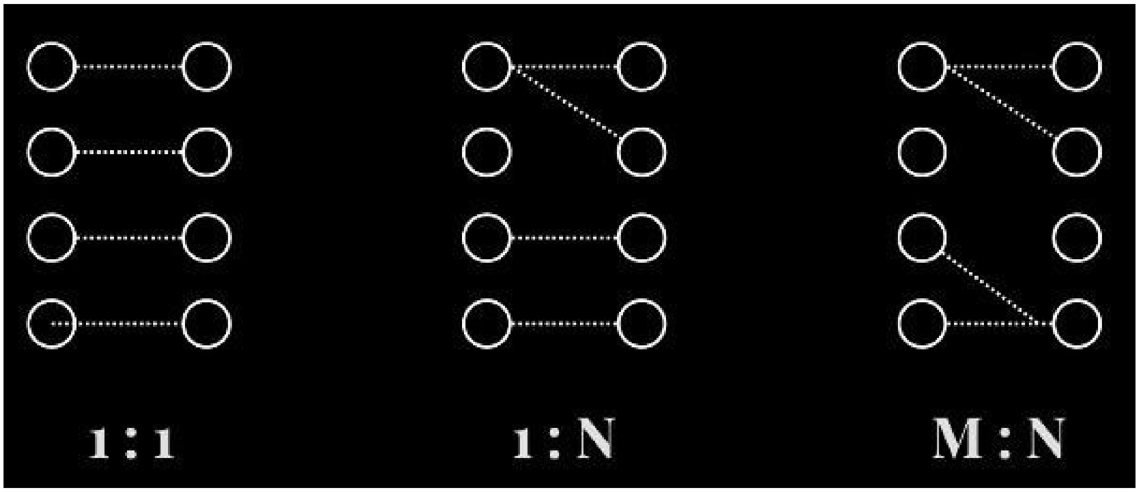
Cardinality refers to the uniqueness of data values contained in a particular column (attribute) of a database table. It can be:

One-to-One: Each record in Table A is linked to a single record in Table B.

One-to-Many: A single record in Table A is linked to multiple records in Table B.

Many-to-One: Multiple records in Table A are linked to a single record in Table B.

Many-to-Many: Multiple records in Table A are linked to multiple records in Table B.



Conclusion

Understanding keys is crucial for database design and management. They ensure data integrity, establish relationships between tables, and facilitate efficient data retrieval and manipulation. By using keys effectively, you can design robust databases that support the complex information needs of modern applications and businesses.