**DATA:**

Any raw information is known as data.

**INFORMATION:**

Processed and meaningful data that provides knowledge. (Which removes the meaninglessness of data)

**META DATA:**

Descriptive information about other data, such as its structure, origin, or format. (Data of data)

**DATABASE:**

A structured collection of data that is organized for efficient retrieval and management. (Where we store our data)

**DBMS:**

Software that manages and facilitates access to databases.

**DDL:**

Set of commands for defining and managing database structure. (Creating tables and setting up their properties)

**DML:**

Set of commands for manipulating and retrieving data within a database. (Such as insert update delete)

**DATA BASE INSTANCE:**

A snapshot of a database at a particular moment, including its data and schema.

**DBA:**

A professional responsible for managing and maintaining a database system.

**ANALYZING TOOLS:**

Software or applications used to examine and interpret data for insights.

**DATA CENTER:**

A data center is a facility that houses the computers and other equipment that store and process data.

**F5 LOAD BALANCER:**

A device that distributes network traffic across multiple servers to ensure efficient use and prevent overload. (Device use to balance load on all servers)

**DLP (DATA LEAK PROTECTION):**

Measures and policies to prevent unauthorized access or sharing of sensitive data.

**TUPPLE:**

A set of values representing a single record or data point in a database table. (Single record is known as tupple.)

**ENTITY:**

A distinct and identifiable object, concept, or thing about which data is stored. (Physical or nonphysical)

**RECORD:**

A complete set of fields or attributes representing a single instance of an entity in a database.

**ATTRIBUTE:**

A characteristic or property that describes an entity, contributing to the definition of its data. (Information of entity)

**DATA BASE SCHEMA:**

The overall structure that defines the organization of data in a database, including tables, relationships, and constraints.

**ER DIAGRAM:**

A visual representation of database entities, their attributes, and relationships.

**TYPES OF ATTRIBUTES:**

Descriptive characteristics of entities, including simple, composite, derived, and multivalued attributes.

**PRIMARY KEY:**

A unique identifier for a record in a database table, ensuring data integrity and facilitating retrieval. (Unique attribute and it cannot be null)

**FOREIGN KEY:**

A field in a database table that refers to the primary key in another table, establishing a link between them.

**CANDIDATE KEY:**

A unique key that could be chosen as the primary key for a table.

**SUPER KEY:**

A set of one or more attributes that, taken collectively, can uniquely identify a record in a table. (A key which is eligible for a candidate key is known as super key)

**COMPOSITE KEY:**

A primary key composed of multiple attributes to uniquely identify a record in a table.

**DATABASE MANAGER:**

A database manager is a person who is responsible for developing and maintaining an organizations' database systems that store and organize data for companies.

**DRAW BACKS OF MY SQL:**

Limitations of the MySQL database system, which may include lack of support for certain features, performance issues, or scalability challenges.

**ERP (ENTERPRISE RESOURCE PLANNING):**

Integrated software systems for managing and automating business processes across an organization.

**CRM (CUSTOMER RESPONSE MANAGEMENT):**

Strategies, processes, and technologies to manage and analyze customer interactions throughout the customer lifecycle.

**SQL:**

A domain-specific language for managing and manipulating relational databases.

**SQL+:**

An Oracle-specific extension of SQL that includes additional features and capabilities.

**RDB (RELATIONAL DATA BASE):**

A type of database that organizes data into tables with predefined relationships between them.

**NON RELATIONAL DATA BASE:**

A type of database that does not use the tabular structure of relational databases, often used for handling large amounts of unstructured data.

**RMAN (ORACLE RECOVERY MANAGER):**

Oracle's tool for backup and recovery of databases.

**CLOUD SERVERS:**

Virtual servers hosted in a cloud computing environment, offering scalable and flexible computing resources.

**ALTERNATE KEY:**

A candidate key that is not selected as the primary key for a table.

**LOGICAL DIAGRAM:**

A visual representation of the logical structure and relationships within a system or database. (Rough diagram)

**PHYSICAL LAYER:**

Records saved in db.

**LOGICAL LAYER:**

The conceptual organization and structure of data or systems, independent of the physical implementation.

**APPLICATION LAYER:**

The Application layer resides between the user and the DBMS, which is responsible for communicating the user's request to the DBMS system and send the response from the DBMS to the user.

**DRIVE ATTRIBUTES:**

Attributes which are derived from other attributes for example we can drive country from the address.

**DSL (data structure language):**

Structure of the data base (primary, foreign etc)

**DATABASE ANOMLIES:**

Irregularities or inconsistencies (error) in data that can occur during database operations.

**DATA BASE NORMALIZATION:**

A process in database design to reduce redundancy and dependency by organizing data into tables.

**1NF:**

Ensures that each table cell contains a single, indivisible value.

**2NF:**

Eliminates partial dependencies by ensuring all non-prime attributes are fully functionally dependent on the primary key.

**3NF:**

Removes transitive dependencies by ensuring non-prime attributes are not dependent on other non-prime attributes.

**3.5NF:**

A concept extending beyond 3NF to address certain exceptional cases in normalization.

**4NF:**

Addresses multi-valued dependencies in data.

**5NF:**

Deals with cases where information can be derived from other information in multiple ways.

**COMMIT:**

Making changes permanent in a database. (Permanently delete)

**TRANSITIVITY:**

In database relationships, the property where if A is related to B and B is related to C, then A is related to C.

**DATA CLEANING:**

The process of identifying and correcting errors or inconsistencies in a database. (Check lagana ke koi value null ya 0 tou nhi hai ya kissi number ki jagha koi text tou nhi arraha agr hai tou us data ko side kardo)

**NF DIAGRAM:**

A diagram illustrating the normalization levels (1NF, 2NF, etc.) applied to a database.

**ORACLE CRYSTAL REPORT:**

A reporting tool used with the Oracle database for designing and generating reports. (Use to generate pdf of the page which is showing on the data base)

**LOSSY DECOMPOSITION:**

Breaking down a relation into smaller relations, potentially losing certain dependencies.

**SPORIOUS TUPPLE:**

Unwanted record.

**NOSQL:**

A type of database management system that does not rely on a traditional relational database model. (We can store any kind of data for example YouTube Facebook etc)

**DATABASE CONSTRAINT:**Rules or conditions imposed on data columns to ensure data integrity.

**MULTIPLICITY:**

In database relationships, the count or range of occurrences of one entity related to another.

**STRONG ENTITY TYPE:**

An entity type with a primary key, capable of existing independently.

**WEAK ENTITY TYPE:**

An entity type that relies on another entity for existence and does not have a primary key attribute.

**ENTITY OCCURRENCE:**

An individual instance or occurrence of an entity.

**ENTITY TYPE:**

A category or class of entities sharing common attributes and relationships in a database.

**THE PURPOSE OF NORMALIZATION:**

To organize data in a relational database to reduce redundancy, dependency, and anomalies, ensuring data integrity, efficiency, and ease of maintenance.

**HOW NORMALIZATION CAN BE USED WHEN DESIGNING A RELATIONAL DATABASE:**

Normalization involves decomposing a database into tables, defining relationships, and applying normalization rules (1NF, 2NF, 3NF, etc.) to ensure efficient storage, minimize redundancy, and maintain data consistency.

**THE POTENTIAL PROBLEMS ASSOCIATED WITH REDUNDANT DATA IN BASE RELATIONS:**

Redundant data can lead to update anomalies, where changes must be made in multiple places, insertion anomalies, where incomplete information cannot be added, and deletion anomalies, where removing data unintentionally removes related information.

**THE CONCEPT OF FUNCTIONAL DEPENDENCY, WHICH DESCRIBES THE RELATIONSHIP BETWEEN ATTRIBUTES:**

A functional dependency describes the relationship between attributes in a relation, where the value of one attribute uniquely determines the value of another.

**THE CHARACTERISTICS OF FUNCTIONAL DEPENDENCIES USED IN NORMALIZATION:**

Reflexive: Every attribute is functionally dependent on itself.

Augmentation: If A determines B, then AC determines BC.

Transitive: If A determines B and B determines C, then A determines C.

**HOW TO IDENTIFY FUNCTIONAL DEPENDENCIES FOR A GIVEN RELATION:**

Identify key attributes and candidate keys.

Examine how non-key attributes depend on the key(s) or other non-key attributes. ‘

Define functional dependencies based on observed relationships.

**DATA REDUNDANCY:**   
Data redundancy occurs when the same piece of data is stored in multiple places within a database.

**KEEPING DATA SAFE IN DATABASES/DATABASE SECURITY**

Data is like treasure for companies, so we have to make sure it's protected. Some of the data is really important for the company's plans, so we need to keep it safe and secret.

Database Security is like having strong locks and guards for this treasure. We use special tools and plans to keep the data safe from on-purpose or accidental problems.

And it's not just about the data in the database. If there's a security problem, it might cause issues in other parts of the system, and that could end up affecting the database too. So, it's like keeping a whole fortress safe, not just the treasure inside!

**DATABASE SECURITY**

Involves measures to avoid:

Theft and fraud

Loss of confidentiality (secrecy)

Loss of privacy

Loss of integrity

Loss of availability

**Keeping Databases Safe**

Database security is all about preventing different problems. Here are the main things we want to avoid:

1. **Theft and Fraud:** We don't want anyone stealing or cheating with our data.

2. **Loss of Confidentiality (Secrecy):** We need to make sure that only the right people can see certain information, and others can't.

3. **Loss of Privacy:** Everyone's personal information should be kept private and not shared without permission.

4. **Loss of Integrity:** The data should stay accurate and unchanged. We don't want someone messing with it and giving us the wrong information.

5. **Loss of Availability:** The data should always be available when we need it. We don't want the information to be lost or unreachable.

**THREAT**

A threat is any situation or event that can harm a system and, as a result, cause problems for an organization. These situations or events can be on purpose or by accident, and they have the potential to negatively impact the smooth functioning of a system and the overall well-being of an organization.

**or**

 Any situation or event, whether intentional or unintentional, that will adversely affect a system and consequently an organization.

**PROTECTING COMPUTERS: SAFETY MEASURES**

These are like safety measures for computers. They include:

**Authorization:** Making sure only the right people can use certain things.

**Access controls:** Deciding who can get into different parts of the computer.

**Views:** Setting up what different people can see on the computer.

**Backup and recovery:** Saving copies of important stuff and knowing how to get it back if something goes wrong.

**Integrity:** Keeping the information accurate and not letting anyone mess with it.

**Encryption:** Turning information into secret code so only the right people can understand it.

**RAID technology:** Using special technology to keep data safe and not lost if something goes bad.

**AUTHORIZATION**

Authorization means giving someone the right or permission to access a computer system or its parts. It's like allowing them to do certain things. Before this, there's a check called authentication to make sure the person trying to access is really who they say they are.

**ACCESS CONTROL**

Access control is about deciding who gets permission to do certain things in a computer system. It involves giving and taking away special rights. These rights, called privileges, allow a user to do specific actions like reading, writing, or changing things in a database. People are given these privileges based on what they need to do fortheir jobs.

**DISCRETIONARY ACCESS CONTROL (DAC)**

In computer systems, many use a way of deciding who can do what, and one common method is called Discretionary Access Control (DAC). It's like having control over who can access and do things in a database.

In the language of databases (like SQL), they use commands like GRANT to give special rights to users, allowing them to do specific things. On the flip side, there's the REVOKE command, which takes away these rights if needed. It's a way to manage and control who can do what in a database.

**WEAKNESSES OF DAC AND INTRODUCTION TO MAC IN SIMPLE TERMS**

While Discretionary Access Control (DAC) is useful, it has a weakness: someone who is not allowed can trick someone who is allowed into sharing important information.

To strengthen security, there's another way called Mandatory Access Control (MAC). In MAC, there are strict rules for the entire system that users can't change on their own.

Each piece of data in the database is given a security class, and each user is assigned a clearance level. Rules are then set for who can read or write each piece of data based on these security levels. This ensures that sensitive data can't be shared without the proper clearance.

It's like having strong, unchangeable rules for everyone to make sure important information stays safe. However, the standard language for databases (SQL) doesn't have built-in support for MAC.

MAC was introduced not as a response to DAC failure but as a means to enhance security by addressing specific vulnerabilities and challenges associated with individual user control in certain high-security environments. The choice between DAC and MAC depends on the specific needs and security requirements of an organization or system.

**BELL-LAPADULA MODEL FOR MAC:**

The Bell-LaPadula Model is a popular Mandatory Access Control (MAC) model used for enforcing confidentiality in computer security. It is described in terms of various elements:

**OBJECTS:**

These are things in the system like relations, views, tuples, and attributes – essentially the components of the database.

**SUBJECTS:**

These are entities that can access objects, such as users and programs.

**SECURITY CLASSES AND CLEARANCES:**

The classification of data and the level of clearance a subject has to access certain information.

**CLASSIFICATION LEVELS:**

The model uses four classification values: U (unclassified), C (confidential), S (secret), and TS (top secret).

**ORDERED CLASSIFICATIONS:**

These classifications are ordered, meaning there's a hierarchy: TS > S > C > U. For example, TS (top secret) data has a higher security level than S (secret) data.

**SECURITY RULE:**

The primary rule in the Bell-LaPadula Model is the "no read up, no write down" rule. This means a subject with a certain clearance level cannot read data at a higher security level (up), and it cannot write data to a lower security level (down).

The model is designed to prevent unauthorized access to classified information, ensuring that information flows only in a way that maintains security clearances. It's widely used in government and military environments where strict confidentiality is crucial.

**The Bell-LaPadula** Model is a security model that ensures confidentiality in computer systems. It classifies data into levels (unclassified, confidential, secret, top secret) and establishes a hierarchy. The main rule is "no read up, no write down," meaning a user can't access higher-level data but can access lower-level data. This model is commonly used in high-security environments to prevent unauthorized access to sensitive information.

**View in Databases:**

A view in databases is like a virtual table created by performing specific operations on existing tables. It doesn't physically exist in the database but is generated on demand when a user requests it. Essentially, it's a dynamic result derived from one or more relational operations applied to the base relations (tables). Views allow users to see a customized perspective of the data without altering the actual underlying data in the database.

**BACKUP AND JOURNALING IN DATABASES:**

**BACKUP:**

- Backup is the process of regularly creating a copy of the database along with log files (and sometimes programs) and storing it on offline storage media.

- This copy serves as a safety net in case of data loss, system failure, or other emergencies. It allows for the restoration of the database to a previous state.

**JOURNALING:**

- Journaling is the process of keeping a log file (or journal) that records all changes made to the database.

- This log file maintains a record of modifications, additions, or deletions to the data, providing a chronological history of database transactions.

- Journaling is crucial for effective recovery in the event of a failure, as it allows for replaying the logged transactions to bring the database back to a consistent and correct state.

**DATA TRANSMISSION SECURITY AND ENCRYPTION:**

To securely transmit data over insecure networks, a cryptosystem is used, consisting of:

- **Encryption Key:** A key used to encrypt the original data (plaintext).

- **Encryption Algorithm:** A method, along with the encryption key, to transform plaintext into ciphertext.

**- Decryption Key:** A key used to decrypt the ciphertext back into plaintext.

- **Decryption Algorithm:** A method, along with the decryption key, to transform ciphertext back into plaintext.

**Symmetric Encryption:**

- Symmetric encryption uses the same key for both encryption and decryption.

- It relies on secure communication lines for exchanging the key.

- The challenge is that users often lack access to secure communication lines, and for real security, keys need to be as long as the message.

**Data Encryption Standard (DES)**

- DES is a symmetric encryption scheme, where the same key is used for both encryption and decryption.

- It is a standard encryption algorithm developed by IBM.

- DES has been widely used, but its security strength has been questioned over time, leading to the development of more advanced encryption algorithms.

In summary, for secure data transmission, encryption is essential, and while symmetric encryption like DES has been widely used, evolving technologies have led to the development of more secure encryption methods and key exchange protocols.

**RAID (REDUNDANT ARRAY OF INDEPENDENT DISKS) TECHNOLOGY:**

**Fault-Tolerance:**

- Hardware running the Database Management System (DBMS) should be fault-tolerant, meaning it continues to operate even if one component fails.

- Suggests having redundant components that seamlessly integrate into the system in case of a failure.

**Critical Fault-Tolerant Components:**

- Include disk drives, disk controllers, CPU, power supplies, and cooling fans.

- Disk drives are particularly vulnerable, with shorter times between failures compared to other components.

**Disk Array for Reliability and Performance:**

- Solution involves a large disk array comprising several independent disks.

-**Data Striping:**Data is divided into equal-size partitions (striping units) and distributed across multiple disks, increasing performance.

**Improving Reliability:**

**Redundancy through Parity or Error-Correcting Scheme:** Redundant information is stored across disks using a parity scheme or an error-correcting scheme.

- This enhances reliability by enabling the system to recover from the failure of a single disk without losing data.

In summary, RAID technology enhances fault tolerance in database systems by employing redundant components, utilizing data striping for performance improvement, and implementing redundancy and error-correction mechanisms to enhance reliability, especially in the case of disk failures.

**RAID 0: STRIPING (NO REDUNDANCY):**

**Operation:** Data is divided into blocks and distributed across multiple disks.

**Advantages:** Improved performance as data can be read/written from/to multiple disks simultaneously.

**Drawback:** No redundancy; if one disk fails, data is lost as there's no backup.

**RAID 1: MIRRORING (FULL REDUNDANCY):**

**Operation:** Data is duplicated on two disks (mirrored).

**Advantages:** Full redundancy; if one disk fails, the other has an exact copy.

**Drawback:** Costlier as it requires twice the storage capacity.

In summary, RAID 0 focuses on performance by striping data across disks but lacks redundancy, while RAID 1 prioritizes data redundancy by mirroring data on two disks, ensuring high fault tolerance but at the expense of storage efficiency.

**RAID 5: STRIPING WITH DISTRIBUTED PARITY:**

**Operation:**

- Data is striped across multiple disks.

- Parity information (error-checking data) is distributed across all disks.

-**Advantages:**

- Good balance of performance and fault tolerance.

- Efficient use of disk space (no need for a complete duplicate).

**-Drawback:**

- Slower write performance due to parity calculations.

**RAID 10 (OR RAID 1+0): MIRRORING AND STRIPING:**

**-Operation:**

- Combination of RAID 1 (mirroring) and RAID 0 (striping).

- Data is mirrored (duplicated) and then striped.

**-Advantages:**

- High performance from striping.

- High fault tolerance from mirroring.

**-Drawback:**

- Requires a minimum of four disks, and half of the total capacity is used for mirroring.

In summary, RAID 5 provides a good balance between performance and fault tolerance by striping data and using distributed parity, while RAID 10 combines mirroring and striping for high performance and fault tolerance but requires more disks and storage space.

**DBMSS AND WEB SECURITY:**

Internet communication relies on TCP/IP and HTTP, which were not initially designed with security in mind, allowing anyone monitoring traffic to read it. To ensure secure transmission over the Internet, measures are needed to address privacy, integrity, authenticity, non-fabrication, and non-repudiation:

1. **Privacy:**

- Information should be inaccessible to anyone except the sender and receiver.

2. **Integrity:**

- Data should not be changed during transmission.

3. **Authenticity:**

- The receiver should be sure that the information came from the claimed sender.

4. **Non-Fabrication:**

- Assurance that the received data is genuine and not fabricated.

5. **Non-Repudiation:**

- The sender cannot deny sending the information.

**Security Measures Include:**

1. **Proxy Servers:**

- Act as intermediaries between clients and servers, enhancing security and performance.

2. **Firewalls:**

- Control and monitor incoming and outgoing network traffic based on predetermined security rules.

3. **Message Digest Algorithms and Digital Signatures:**

- Ensure data integrity by generating fixed-size hash values (message digests) and using digital signatures for authentication.

4. **Digital Certificates:**

- Electronic documents that bind the identity of a person or entity to a public key.

5. **Kerberos:**

- Authentication protocol for secure communication over a non-secure network.

6. **Secure Sockets Layer (SSL) and Secure HTTP (SHTTP):**

- Protocols that provide secure communication over a computer network, commonly used for securing online transactions.

These measures collectively enhance the security of DBMSs and web communication, safeguarding information during transmission.

**DBMS BLADE SERVERS:**

Blade servers for databases are like compact, interchangeable computer slices that work together to efficiently manage and store data in a database management system.

**MULTIFACTOR AUTHENTICATION:**

Multifactor authentication means using more than one way to prove you are who you say you are, like combining a password with a fingerprint or a code from your phone.

**FIREWALL:**

A firewall is like a digital barrier that protects your computer or network, deciding what data is allowed in or out and blocking harmful stuff from getting through.

**HOT SWAPPABLE:**

Hot swappable means you can change or replace a component (like a hard drive) in a system without turning it off, keeping things running smoothly without any interruption.