ADBMS

UNIT - I :

Introduction : Definition Of Data, Database And DBMS – Need For Database – Advantages Of Relational Database Systems - ACID Properties - Database Related Terms : Concurrency, Consistency, Data Availability, Reliability, Transparency, Data Integrity And Data Security

Definition Of Data, Database And DBMS

DATA :

- Data are **characteristics or information, usually numerical**, that are collected through observation.
- Data are a **set of values of qualitative or quantitative variables about one or more persons or objects**, while a datum is a single value of a single variable.

DATABASE:

- A *database* is a **structured collection of records or data or information that is organized so that it can be rapidly and easily accessed, managed and updated**. .

DBMS:

- Database Management System (DBMS) is a **system software for creating and managing databases.**
- DBMS **provides users and programmers with a systematic way to create, retrieve, update and manage data.**
- DBMS **serves as an interface between the database and end users or application programs, ensuring that data is consistently organized and remains easily accessible.**

Need For Database

- We need databases because **they can store organized data in a systematic manner which allows us to query, sort and manipulate data in various ways and aid decision making.**

Need For Database Management Systems

- DBMS **provides functionalities that eliminates disadvantages of File-based database system**.
- Database Management Systems are needed for:
  a) **Creation of a database.**
  b) **Retrieval of information from the database.**
  c) **Updating the database.**
  d) **Managing a database.**
  e) **Overcome the disadvantages of File-based database system.**

ADVANTAGES OF DATABASE SYSTEMS

1. An organized and comprehensiveness of recording the result of the firm's activities.
2. A receiver of data will be able to meet the information requirement of the MIS users.
3. Reduced data redundancy.
4. Reduced updation errors and increased consistency (reliability).
5. Greater data integrity (truthfullness) and independence from applications programs.
6. Improved data access to users through use of host and query languages.
7. Improved data security, data availability and recoverability
8. Reduced data entry, storage, and retrieval costs.
9. Standard can be enforced – Data stored in a Standard format are desirable during interchange or migration between systems.
10. Concurrent access and conflicting requirements by multiple users handled efficiently.

Advantages Of Relational Database Systems

Advantages :

1. **Structural Independence: - Any changes to the database structure, does not affect the way we are accessing the data.**

   If there is no structural independence, then modifying or altering the file like adding a field or deleting a field would result in several applications become unworkable.

2. **Simplicity: - This model is designed based on the logical data. It does not consider how data are stored physically in the memory.**

3. **Because of Simplicity and Data Independence, Relational Data Model is easy to maintain and access.**

   Data independence is the ability to modify the scheme without affecting the programs and the application to be rewritten. Data is separated from the programs, so that the changes made to the data will not affect the program execution and the application.

Advantages Of Relational Database Systems

4. **Supports Structured Query Language – SQL. Hence it helps the user to retrieve and modify the data in the database. By the use of SQL, user can get any specific information from the database.**
5. **The relational model makes database management systems more independent of any particular application.**

Disadvantages:

1. ***High hardware cost*: - In order to separate the physical data information from the logical data, more powerful system hardware – memory is required. This makes the cost of database high.**
2. **Designing till the infinitesimal (minute) level, will lead to complexity in the database.**

Characteristics of Relational Database Management System

1. Stores Any Kind of Data and data about the Data (Meta data)
2. Supports ACID Properties - Atomicity, Consistency, Isolation, and Durability.
3. Supports Concurrent Use of Database
4. *Provides insulation between Data and Application*
5. Provides efficient Backup and Recovery facilities.
6. Represents Complex Relationship Between Data
7. *Provides Security*
8. *Supports reduction of Data redundancy.*
9. *Supports strong query language.*

ACID PROPERTIES - ATOMICITY

Atomicity guarantees that when a transaction takes place either all actions of a transaction be executed completely, or, in case of some failure, partial effects of each incomplete transaction be undone.

- ***Transaction-Management Component*** of the database system **ensures Atomicity**
- ***The database system keeps track (on disk) of the old values of any data on which a transaction performs a write, and, if the transaction does not complete its execution, the database system restores the old values to make it appear as though the transaction never executed.***
- ***Features of Atomicity:***
  - ***A transaction is a unit of operation - either all the transaction's actions are completed or none***
  - ***Atomicity is maintained in the presence of deadlocks, CPU or Disk Failures, Database Software Failures or application software failures.***
  - ***Atomicity can be turned off at the system or session level***

ACID PROPERTIES - CONSISTENCY

Consistency in database systems ensures **that any given database transaction must change data only in allowed ways.**

- **Data cannot be written that would violate the database's own rules for valid data.**
- **If a certain transaction occurs that attempts to introduce inconsistent data, the entire transaction is rolled back and an error returned to the user.**

Consistency guarantees *that* the database is consistent before and after execution of a transaction.

- ***Ensuring consistency for an individual transaction is the responsibility of the application programmer who codes the transaction.*** **This task may be facilitated by automatic testing of integrity constraints.**
- **Execution of a transaction in isolation *(that is, with no other transaction executing concurrently)* preserves the consistency of the database.**
- **Consistency ensures that any changes to values in an instance are consistent with changes to other values in the same instance.**

ACID PROPERTIES - ISOLATION

***Isolation guarantees* insulation of transactions from the effects of other concurrently executing transactions.**

- **Concurrency-Control Component** of the database system **ensures isolation.**

- **If several transactions are executed concurrently, their operations may interleave in some undesirable way, resulting in an inconsistent state.**
  - i.e., **if updates to shared data are not controlled there is possibility for transactions to see inconsistent intermediate states created by updates of other transactions. Such a situation can result in erroneous updates to data stored in the database.**

- **The isolation property of a transaction ensures that the concurrent execution of transactions results in a system state that is equivalent to a state that could have been obtained had these transactions executed one at a time in some order.**

ACID PROPERTIES - DURABILITY

Durability guarantees that, once a transaction completes successfully, all the updates that it carried out on the database persist, even if there is a system failure after the transaction completes.

- ***Recovery-Management Component*** of the database system **ensures Durability.**
- ***We can guarantee durability by ensuring that either:***
  1. **The updates carried out by the transaction have been written to disk before the transaction completes.**
  2. **Information about the updates carried out by the transaction are written to disk is sufficient to enable the DBMS to reconstruct the updates when the database system is restarted after the failure.**

- **Durability refers to the ability of the system to recover committed transaction updates if either the system or the storage media fails.**

ACID PROPERTIES - DURABILITY

Features to be considered for Durability:

Recovery to the most recent successful *commit after*

- ***a database software failure***
- ***an application software failure***
- ***a CPU failure***
- ***a disk failure***
- ***a data disk failure***

Database Related Terms : Concurrency

- ***Definition***. CONCURRENCY IS **THE ABILITY OF A DATABASE TO ALLOW MULTIPLE USERS TO ACCESS OR CHANGE SHARED DATA AT THE SAME TIME**.
- The **greater the number of concurrent user processes that can execute without blocking each other, the greater the concurrency of the database system.**
- The **ability to offer concurrency is unique to databases**.
  - Spreadsheets cannot offer several users the ability to view and work on the different data in the same file, because once the first user opens the file it is locked to other users. Other users can read the file, but may not edit data.

Database Related Terms : Concurrency

General Principle : (*Almost all databases deal with concurrency the same way.*)

1. **Changed but unsaved data is held in some sort of temporary log or file.**
2. **Once it is saved, it is then written to the database's physical storage in place of the original data.**
3. **As long as the user performing the change, has not saved the data, only he should be able to view the data he is changing.**
4. **All other users querying for the same data should view the data that existed prior to the change.**
5. **Once the user saves the data, new queries should reveal the new value of the data**

Database Related Terms : Consistency

- Consistency in database systems **refers to the requirement that any given database transaction must change data only in allowed ways.**
- **Any data written to the database must be valid according to all defined rules, including constraints, cascades, triggers, and any combination thereof.**
- *Consistency, in the context of databases, states that :*
  - **Data cannot be written that would violate the database's own rules for valid data.**
  - **If a certain transaction occurs that attempts to introduce inconsistent data, the entire transaction is rolled back and an error returned to the user.**
- **Consistency rules** are vitally important while creating databases,
  - They are the **representation of the business rules for which the database is being created**.
  - They **make the application developers' work easier-** ***It is usually much easier to define consistency rules at the database level rather than defining them in the application that connects to the database.***

  Example : a bank account number must follow a specific pattern- it must begin with a 'C' for checking account or 'S' for savings account, then followed by 14 digits that are picked from the date and time, in the format YYYYMMDDHHMISS.

Database Related Terms : Data Availability

- Data Availability **describe or implement products and services that ensure that data continues to be available at a required level of performance in any situation.**
- Data Availability is the **process of ensuring that data is available to end users and applications, when and where they need it.**
- Data Availability **defines the extent to which data is readily usable with IT and management procedures, tools and technologies required to enable and manage**
- Data Availability is primarily **used to create service level agreements (SLA) , which define and guarantee the service provided .**
- Data Availability is **done by implementing *Data / Storage Redundancy, Data Security, Network Optimization, Data Security* .**
  - ***Storage area networks (SAN), network attached storage and RAID-based storage systems are popular storage management technologies for ensuring data availability.***

Note : RAID is a technology that is used to increase the performance and/or reliability of **data storage**. The abbreviation stands for either Redundant Array of Independent Drives or Redundant Array of Inexpensive Disks, which is older and less used. A **RAID system** consists of two or more drives working in parallel.

Database Related Terms : Reliability

- A DATABASE IS **MORE RELIABLE IF IT IS FAULT TOLERANT.**
  1. ***Fault tolerance*** *is the* ***ability of a Database to continue functioning when part of it fails.***
  2. **Fault tolerance is achieved by Designing the database with A high degree of redundancy.** ***If any single database fails, the redundant database takes its place with no appreciable downtime.***
- Reliability **represents the probability of Database to perform their required functions for a desired period of time without failure in specified environments with a desired confidence.**
- **Reliability explains the time that it will take the component to fail while it is operating.**
- Reliability **does not explains any repair actions that may take place.**
- Reliability **does not reflect how long it will take to get the unit under repair back into working condition.**

Database Related Terms : Reliability

- Reliability measures are **generally used to calculate the probability of failure of a database.**
- *The **reliability can affect Database's availability***

- **Mean Time Between Failures (*MTBF*).**

  Measure used to define a Database's reliability is MTBF

  MTBF is the **average time interval, usually expressed in thousands or tens of thousands of hours, that elapses before a component fails and requires service.**

  MTBF = **(TOTAL ELAPSED TIME - SUM OF DOWNTIME)/NUMBER OF FAILURES**

- **Mean Time To Repair (*MTTR*).**

  MTTR is **the average time interval (expressed in hours) that it takes to repair a failed component.**

Database Related Terms : Data Security

- **Data Security refers to protective digital privacy measures that are applied to prevent unauthorized access to computers, databases and websites.**
- **Data Security features protects databases from data corruption.**
  - **Examples : backups, data masking and data erasure**

    (Data deletion leaves data recoverable, while data erasure is permanent)

    (Data masking is the process of hiding original data with modified content).
- **A key Data Security technology measure is encryption, *where digital data, software/hardware, and hard drives are encrypted and therefore rendered unreadable to unauthorized users and hackers.***
- **Authentication is one of the common method of practising Data Security.**
  - **With authentication, users must provide a password, code, biometric data, or some other form of data to verify identity before access to a system or data is granted.**

Database Related Terms : Data Integrity

- DATA INTEGRITY **is the overall completeness, accuracy and consistency of data stored in a database.**
- Data Integrity is usually **imposed during the database design phase through the use of standard procedures and rules.**
- Data Integrity can be **maintained through the use of various error-checking methods and validation procedures.**
- Data integrity is **imposed within a database when it is designed and is authenticated through the ongoing use of error checking and validation routines**.
- **The following three integrity constraints are used in a relational database structure to achieve data integrity**:

1. **ENTITY INTEGRITY**: Every table must have its own primary key and that must be unique and not null.
2. **REFERENTIAL INTEGRITY**: **The foreign key value would refer to a primary key value of another table.**
3. **DOMAIN INTEGRITY**: All columns in a relational database are in a defined domain.

Database Related Terms : Data Integrity

- Data Integrity **ensures that all data in a database can be traced and connected to other data.**
  1. This ensures that everything is recoverable and searchable.
  2. Having a well-defined and well-controlled data integrity, system increases stability, performance, reusability and maintainability.
- **Software developers can define integrity constraints to enforce business rules on data when entered into an application.**
  - Business rules specify conditions and relationships that must always be true, or must always be false.
- **For Data Integrity**
  1. **Data values are standardized according to a data model and/or data type.**
  2. **All characteristics of the data must be correct – including business rules, relations, dates, definitions and lineage.**

Database Related Terms : Transparency

- **A transparency is an aspect of the distributed system, that is hidden from the user (programmer, system developer, user or application program).**
- **A transparency is provided by including some set of mechanisms in the distributed system at a layer below the interface where the transparency is required.**
- **A number of basic transparencies have been defined for a distributed system. It is important to realize that not all of these are appropriate for every system, or are available at the same level of interface.**
- ***Some important kinds of Transparencies.***

1. ***Access Transparency –***
   - **There should be no apparent difference between local and remote access methods. The explicit communication may be hidden.**

     Example :

     - **From a user's point of view, access to a remote service such as a printer should be identical with access to a local printer.**
     - **From a programmers point of view, the access method to a remote object may be identical to access a local object of the same class.**

Database Related Terms : Transparency

2. *Location Transparency* –
   - The details of the topology of the system should be of no concern to the user.
     - The location of an object in the system may not be visible to the user or programmer.
     - Names may give no hint as to location.
3. *Performance Transparency* –
   - The configuration of the system should not be apparent to the user in terms of performance.
     - This may require complex resource management mechanisms.
     - It may not be possible at all in cases where resources are only accessible via low performance networks..
4. *Scaling Transparency* –
   - A system should be able to grow without affecting application algorithms.
     - A system should also be capable of scaling down to small environments where required, and be space and/or time efficient as required.

Database Related Terms : Transparency

5. *Migration Transparency* –

   If objects (processes or data) migrate to provide better performance, or reliability, or to hide differences between hosts, then Migration should be hidden from the user.

6. *Replication Transparency* –

   If the system provides replication (for availability or performance reasons) it should not concern the user.

7. *Concurrency Transparency* –

   Users and Applications should be able to access shared data or objects without interference between each other.

   - This requires very complex mechanisms in a distributed system.
   - Example: a distributed printing service must provide the same atomic access per file as a central system so that printout is not randomly interleaved.