INTRODUCTION_DBMS

CONCEPTS & APPLICATION

Introduction

DBMS refers to the technology of storing and retrieving users data with utmost efficiency along with appropriate security measures.

Database

A database is a collection of related data and data is a collection of facts and figures that can be processed to produce information. It includes transient data likes marks obtained by student where we can conclude about toppers and average marks. It is also defined as collection of logically interrelated data and description of this data, designed to meet the information needs for organization.

A database is simply an organized collection of related data, typically stored on disk and accessible by possibly many concurrent users. Databases are generally separated into application areas. Example: One database may contain HR(employee and payroll) data; another may contain sales data; another may contain accounting data; and soon. Databases are managed by a DBMS.

Database Management System

DBMS is a set of program/procedure that manages the database and provide an access to database in a form required by any application program.

It is a collection of programs that allows users to specify the structure of a database, to create , query and modify the data in the database and to control access to it. (Example: limit access to the database so that only relevant staff can access details of enrolling students.)

DBMS software are MySQL, Oracle, MongoDB, Microsoft Access etc.

Database System

Both database and database management system in combined form is called database system. It shows collection of related files with details about their definition, interpretation, manipulation and maintenance.

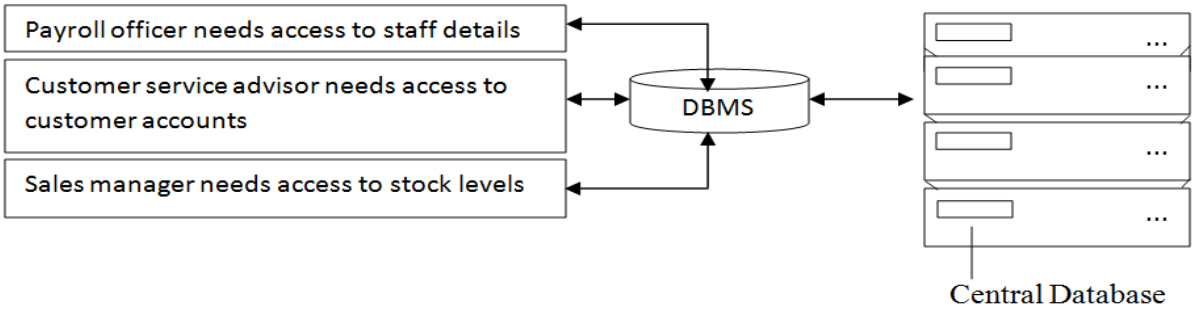


Figure : Database System

Applications of DBMS

Databases are widely used. Some of representative applications are:

1. **Banking :** For customer information, accounts and loans and banking transaction.
2. **Universities :** For student registrations and grades.
3. **Online shopping :** Everyone wants to shop from home,. Everyday new products are added and sold only with the help of DBMS. Purchase information, invoice bills and payment, all of these are done with the help of DBMS.
4. **Airlines :** For reservation and schedule information.
5. **Credit & transaction :** For purchase on credit cards and generation of monthly statements.
6. **Library Management System :** Maintain all the information related to book issue dates, name of the book, author and availability of the book.
7. **Telecommunication :** For keeping records of call made, generating monthly bills, maintaining balances on prepaid calling cards.
8. **Sales:** For customer, product and purchase information.
9. **Finance :** For storing information about purchase information and purchase of financial instruments such as stocks and bonds.
10. **Manufacturing :** For management of supply chain and for tracking production of items in factories, inventories of items and orders for items.

OBJECTIVES & EVOLUTION

Database management system (DBMS) is a software that defines a database, stores data supports a query language, produces reports and creates data language, produces reports and creates system are designed to manage large bodies of information.

Objectives :

DBMS applications must be capable of solving challenging problems of different an organizations.

List of objectives are -

- Provide for mass storage of relevant data
- Making easy access to data for the authorized user
- Providing prompt response to users request for data
- Eliminate redundantly (duplicate) data
- Allow multiple users to database system
- Provide data integrity
- Protect the data from physical harm and unauthorized access
- Serving different types of users
- Provide security with user access privilege
- Combining interrelated data to generate report
- Provide multiple views for some data

Mass storage

DBMS can store a lot of data in it for all the big firms. DBMS is really ideal technology to use. It can store thousands of records in it & one can fetch all that data whenever it is needed.

Remove Duplicate

If you have lots of data then data duplicity will occur for sure at any instance. DBMS guarantees it that there will be no data duplicity among all the records. While storing new records, DBMS makes sure that same data was not inserted before.

Data Protection

Information such as bank details, employees salary details and sale purchase details should always be kept secured. Also all the companies need their data secured from unauthorized use. DBMS gives a master level security to their data. No one can alter or modify the information without the privileges of using that data.

Data backup & Recovery

Sometimes database failure occurs so there is no option like one can say that all the data has been lost. There should be a backup of database so that on database failure it can be recovered. DBMS has the ability to back up and recover all the data in database.

Everyone can work on database

There is no need to be a master of programming language if you want to work on DBMS. Any accountant who is having less technical knowledge can work on DBMS. All the definition and descriptions are given in it so that even a non-technical background person can work on it.

Integrity

Integrity means your data is authentic and consistent. DBMS has various validity checks that make your data completely accurate and consistent.

Platform Independent

One can run DBMS at any platform. No particular platform is required to work on database management system.

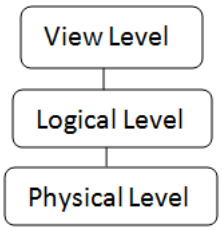
EVOLUTION

Assignment One

DATA ABSTRACTION & DATA INDEPENDENCY

Data Abstraction

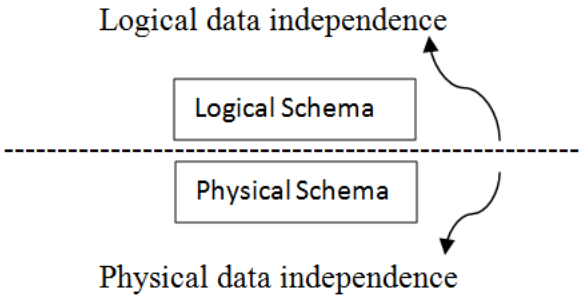
For the system to be usable, it must retrieve data efficiently. The need for efficiency has led designer to use complex data structures to represent data in the database. Developers hide complexity from users through several levels of abstraction to simplicity user interactions with the system.



Physical Level : The lowest level of abstraction describes how the data are actually stored.
Logical Level : The next-higher level of abstraction describes what data are stored in the database and what relationship exist among those data.
View Level : The highest level of abstraction describes only part of the entire database. The view level of abstraction exists to simplicity their interaction with the system. The system may provide many view for the database.

Data Independence

A database system normally contains a lot of data in addition to users data. Example : it stores data about data, known as metadata, to locate and retrieve data easily. It is rather difficult to modify or update a set of metadata once it is stored in the database. But as a DBMS expands, it needs to change over time to satisfy the requirements of the users. if the entire data is dependent, it would become a tedious and highly complex job.



Metadata itself follows a layered architecture, so that when we change data at one layer, it does not affect the data at another level. This data is independent but mapped to each other.
Data independence can be defined as the capacity to change the schema at one level of a database system without having to change the schema at the next higher level.

Data independence occurs because when the schema is changed at some level, the schema at the next higher level remains unchanged; only the mapping between the two levels is changed.
There are two types of data independence

1. **Logical data independence**
2. **Physical data independence**

Logical data independence

Logical data is data about database, that is , it stores information about how data is managed inside. For example, a table (relation) stored in the database and all its constraints, applied on that relation.
Logical data independence is a kind of mechanism which liberalize itself from actual data stored on the disk. If we do some changes on table format, it should not change data residing on the disk.
(It is the capacity to change the conceptual schema without having to change external schemas or application program.)
We may change the conceptual schema to expand the database to change constraints, or to reduce the database.
Example : The addition or removal of new entities, attributes or relationships to the conceptual schema or having to rewrite existing application programs.

Physical data independence

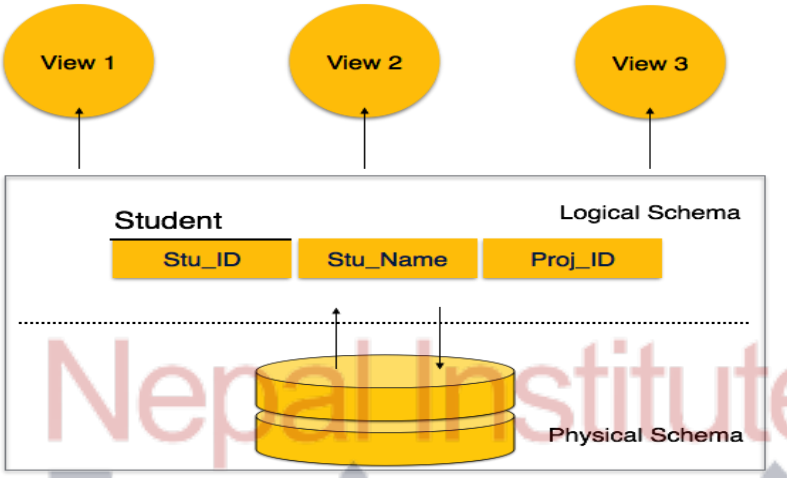
All the schemas are logical and the actual data is stored in bit format on the disk. Physical data independence is the power to change the physical data without impacting the schema or logical data.
Example : In case we want to change or upgrade the storage system itself. Suppose we want to replace hard-disk with SSD, it should not have any impact on the logical data or schemas.
(It is the capacity to change the internal schema without having to change the conceptual schema. Hence the external schema need not be changed as well as changes to the internal schema may be needed because some physical files had to be reorganized.)
Example : A change to the internal schema such as using different file organization, or storage structures, storage devices or indexing strategy, should be possible without having to change the conceptual or external schemas.

SCHEMAS & INSTANCES

DB Schemas

It is the skeleton that represents the logical view of the entire database. It defines how the data is organized and how the relations among them are associated. Its formulates all the constraints that are to be applied on the data.

A database schema defines its entities and the relationship among them. It contains a descriptive detail of the database, which can be depicted by means of schema diagrams. It is the database designers who design the schema to help programmers understand the database and make it useful.



A database schema can be divided broadly into two categories −

- **Physical Database Schema** − This schema pertains to the actual storage of data and its form of storage like files, indices, etc. It defines how the data will be stored in a secondary storage.
- **Logical Database Schema** − This schema defines all the logical constraints that need to be applied on the data stored. It defines tables, views, and integrity constraints.

Database Instance

The data in the database at a particular moment in time is called a database state or snapshot. It is also called the current set of occurrences or instances in the database. Database instances tend to change with time. A DBMS ensures that its every instance (state) is in a valid state by diligently following all the validations, constraints and conditions that the database designers have imposed.

CONCEPT OF DDL, DML, DCL & TCL

A database system provides a data definition language to specify the database schema and a data manipulation language to express database queries and updates.

Data definition language

Deals with database schema and description of how the data should reside in the database.
example:
CREATE - to create database
ALTER - alters the structure of the existing database
DROP - delete objects from the database

TRUNCATE - remove all records from a table, including all spaces allocated for the records are removed.
RENAME - to rename database instances

Data manipulation language

DML is a language that enables users to access or manipulate data as organized by the appropriate data models.
DML commands are not auto-committed. It means changes are not permanent to database they can be rolled back.
Data manipulation is retrieval of information insertion of new information, deletion of information or modification of information stored in the database.
example :
SELECT - retrieve data from the database
INSERT - insert data into a table
UPDATE - updates existing data in table
DELETE - delete all records from a database table

There are two types of **DML**

1. **Procedural DML :** Require a user to specify what data are needed and how to get those data
2. **Non procedural DML :** Require a user to specify what data are needed without specifying how to get those data.

Data control language

It includes commands such as GRANT and REVOKE mostly concerned with rights, permissions and other controls of the database system.
GRANT - allows users access privileges to database
REVOKE - withdraw users access privileges given using the GRANT command.

Transaction control language

- Deals with a transaction within a database
- TCL commands are to keep a check on other commands and their affect on the database
- These commands can annul changes made by other. It can also make changes permanent
  example : COMMIT - to permanently save
  ROLLBACK - to undo changes
  SAVE POINT - to save temporarily

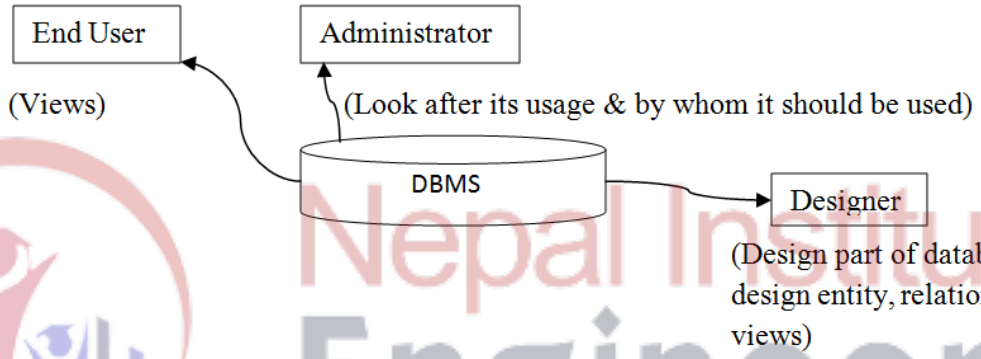
DBMS Characteristics

Traditionally, data was organized in file formats. DBMS was a new concept then, and all the research was done to traditional style of data management.
Characteristics are :
Real world entity

- More realistic and users real world entities to design its architecture
- uses behavior and attributes too.
  example : school may use student as entity and their age as attributes

Relation based tables : allows entities and relations among them to form tables.
Isolation of data and application
Less redundancy
Consistency
Query language
ACID properties
Multi user and concurrent users
Multiple views
Security

Users
Different users with different right and permission for different purposes
Some retrieve data and some back it up



Advantages of database

- Controls redundancy
- Improved data sharing
- Data integrity
- Security (more secure by defining authorization for different users whenever sensitive data access is attempted)
- Data consistency
  - By eliminating data redundancy, inconsistency problem is also reduced/removed
- Efficient data access
  - All access of data through DBMS providing key for effective data processing
- Data independence
  - DBMS provides interface between the application program and the data. When changes are made to data representation, the meta data obtained by DBMS is changed but DBMS is continuous to provide the data to application program in the previously used way. DBMS handles the task of transformation of data whenever necessary. Reduced application development maintenance time.

Disadvantages

- Increased complexity
- Requirement of new & specialized manpower
- Large size of DBMS

File Management System

An abstraction to store, retrieve, manage and update a set of files.. It keeps track on the file and also manage them.

Difference between DBMS & FPS

DBMS	FPS
Minimal data redundancy	Data redundancy exists
Data inconsistency does not exist	Inconsistency exists here
Database access is easier	Database access is comparatively different
Transaction like insert, delete, view, updating etc are possible in database	Transaction are not possible
Concurrent access and recovery techniques are possible	Concurrent access and recovery techniques are not possible
Data remains secure	Data are not completely secure
Administrators store the relationship in the form of structural tables	File manager store all relationship in directories i.e. file system

DBMS : Data redundancy
Data inconsistency
Difficulty in accessing data
Data isolation

FPS : Security problem
Atomicity problem
Concurrent-access anomalies
Integrity problems

DATA : Raw & isolated facts about an entity (recorded)
example: text, audio, video, image, map etc..

INFORMATION : Processed, meaningful, usable data

DBMS Architecture

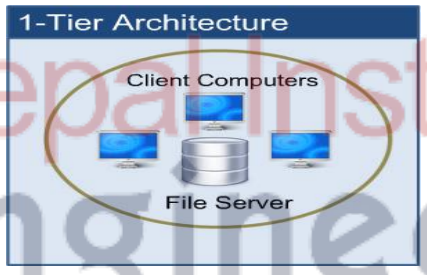
Database architecture uses programming languages to design a particular type of software for businesses or organizations. Database architecture focuses on the design, development, implementation and maintenance of computer programs that store and organize information for businesses, agencies and institutions. A database architect develops and implements software to meet the needs of users.

The design of a DBMS depends on its architecture. It can be centralized or decentralized or hierarchical. The architecture of a DBMS can be seen as either single tier or multi-tier. The tiers are classified as follows :

1. **1-tier architecture**
2. **2-tier architecture**
3. **3-tier architecture**
4. **n-tier architecture**

1-tier architecture:

One-tier architecture involves putting all of the required components for a software application or technology on a single server or platform.

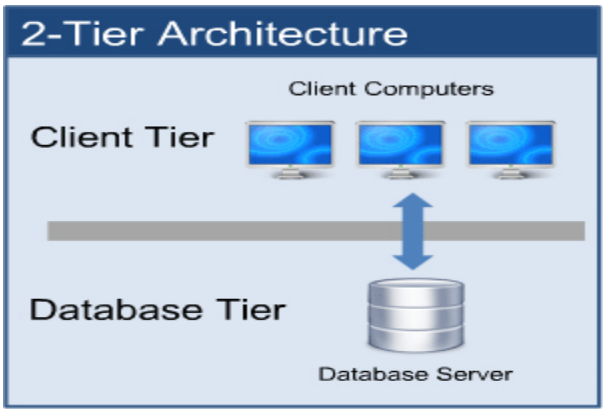


1-tier architecture

Basically, a one-tier architecture keeps all of the elements of an application, including the interface, Middleware and back-end data, in one place. Developers see these types of systems as the simplest and most direct way.

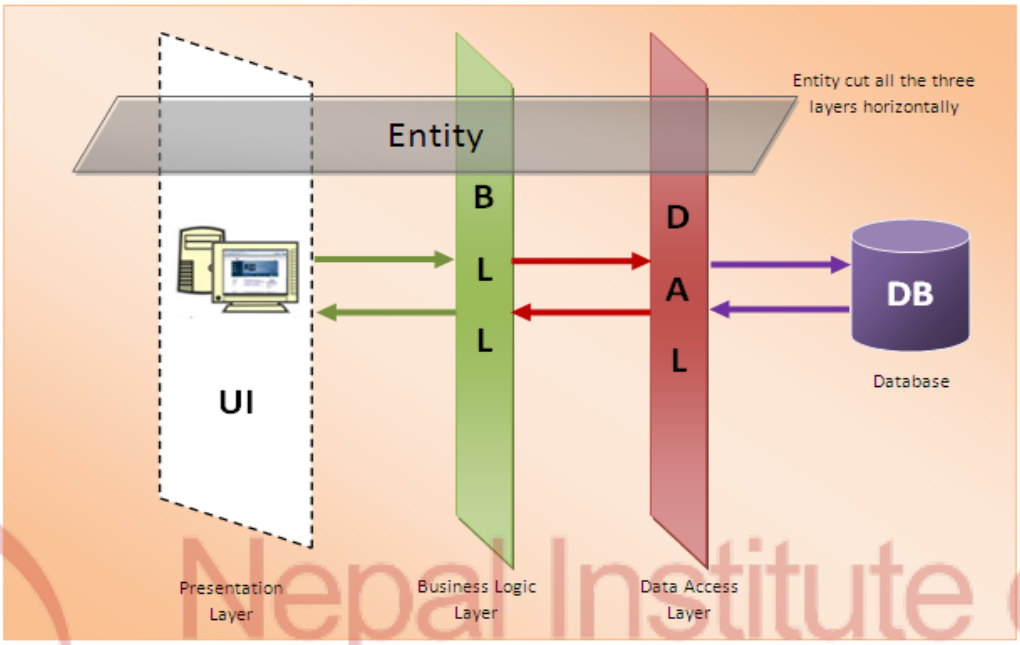
2-tier architecture:

The two-tier is based on Client Server architecture. The two-tier architecture is like client server application. The direct communication takes place between client and server. There is no intermediate between client and server.



3-tier architecture:

A 3-tier architecture separates its tiers from each other based on the complexity of the users and how they use the data present in the database. It is the most widely used architecture to design a DBMS.



[Basic 3-Tire architecture]

3-tier architecture

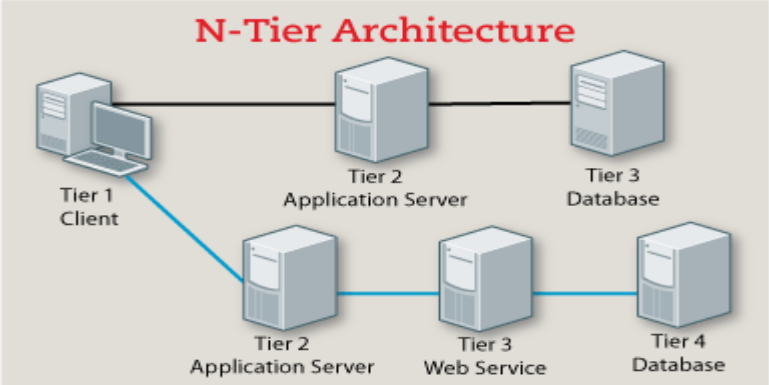
This architecture has different usages with different applications. It can be used in web applications and distributed applications. The strength in particular is when using this architecture over distributed systems.

- **Database (Data) Tier** − At this tier, the database resides along with its query processing languages. We also have the relations that define the data and their constraints at this level.
- **Application (Middle) Tier** − At this tier reside the application server and the programs that access the database. For a user, this application tier presents an abstracted view of the database. End-users are unaware of any existence of the database beyond the application. At the other end, the database tier is not aware of any other user beyond the application tier. Hence, the application layer sits in the middle and acts as a mediator between the end-user and the database.
- **User (Presentation) Tier** − End-users operate on this tier and they know nothing about any existence of the database beyond this layer. At this layer, multiple views of the database can be provided by the application. All views are generated by applications that reside in the application tier.

n-tier architecture:

N-tier architecture would involve dividing an application into three different tiers. These would be the

1. logic tier,
2. the presentation tier, and
3. the data tier.



It is the physical separation of the different parts of the application as opposed to the usually conceptual or logical separation of the elements in the model-view-controller (MVC) framework. Another difference from the MVC framework is that n-tier layers are connected linearly, meaning all communication must go through the middle layer, which is the logic tier. In MVC, there is no actual middle layer because the interaction is triangular; the control layer has access to both the view and model layers and the model also accesses the view; the controller also creates a model based on the requirements and pushes this to the view. However, they are not mutually exclusive, as the MVC framework can be used in conjunction with the n-tier architecture, with the n-tier being the overall architecture used and MVC used as the framework for the presentation tier.

OLD QUESTIONS ASKED PREVIOUSLY(Marks/Year)

1. What do you mean by scheme and instances? Mention the different levels of data abstraction and explain.(4/2075)

2. Why is data independence importance in data modeling? Differentiate between schema and instances. (4/2073 Bhadra)

Advantages of Data independence

The advantages of data independency in DBMS are as follows:

- *Ability of improving performance*
- *Alterations in data structure does not requires alterations in application programs*
- *Implementation details can be hidden from the users*
- *Reduction of incongruity*
- *Tractability in improvement system*
- *Affordable prices of maintaining system*

- *Providing the best services to the users*
- *Permit users to focus on general structures*
- *Enforcement of standards*
- *Improvement of security*
- *The state of being undamaged or undivided can be improved*

Differences Between Schema and Instance

- *A schema is the design representation of a database whereas instance is the snapshot of a database at a particular moment.*
- *Instance changes very frequently, whenever data is removed or added in the database. As against, the changes in schema occurs rarely.*
- *For example, schema and instance can be easily perceived by analogy to a program. At the time of writing a program in a programming language, the variables of that program is declared at first, this is analogous to the schema definition. Additionally, each variable in a program must have some values associated at a particular time; this is similar to an instance.*

3. What do you mean by data abstraction? List the various levels of data abstraction and briefly explain? (1+3/2073 Magh)

4. Why data independence is important in data modeling? Differentiate between physical and logical data independence.(4/2072 Ashwin)

Logical Data Independence	***Physical Data Independence***
Logical Data Independence is mainly concerned with the structure or changing the data definition.	*Mainly concerned with the storage of the data.*
It is difficult as the retrieving of data is mainly dependent on the logical structure of data.	*It is easy to retrieve.*
Compared to Logic Physical independence it is difficult to achieve logical data independence.	*Compared to Logical Independence it is easy to achieve physical data independence.*
You need to make changes in the Application program if new fields are added or deleted from the database.	*A change in the physical level usually does not need change at the Application program level.*
Modification at the logical levels is significant whenever the logical structures of the database are changed.	*Modifications made at the internal levels may or may not be needed to improve the performance of the structure.*
Concerned with conceptual schema	*Concerned with internal schema*
Example: Add/Modify/Delete a new attribute	*Example: change in compression techniques, hashing algorithms, storage devices, etc*

5. What are the drawbacks of file system to store data?(4/2072 Magh)

Drawbacks of File system

- ***Data redundancy:*** *Data redundancy refers to the duplication of data, lets say we are managing the data of a college where a student is enrolled for two courses, the same student details in such case will be stored twice, which will take more storage than needed. Data redundancy often leads to higher storage costs and poor access time.*
- ***Data inconsistency:*** *Data redundancy leads to data inconsistency, lets take the same example that we have taken above, a student is enrolled for two courses and we have student address stored twice, now lets say student requests to change his address, if the address is changed at one place and not on all the records then this can lead to data inconsistency.*
- ***Data Isolation:*** *Because data are scattered in various files, and files may be in different formats, writing new application programs to retrieve the appropriate data is difficult.*
- ***Dependency on application programs:*** *Changing files would lead to change in application programs.*
- ***Atomicity issues:*** *Atomicity of a transaction refers to "All or nothing", which means either all the operations in a transaction executes or none.*

  For example: Let's say Steve transfers 100$ to Negan's account. This transaction consists multiple operations such as debit 100$ from Steve's account, credit 100$ to Negan's account. Like any other device, a computer system can fail let's say it fails after first operation then in that case Steve's account would have been debited by 100$ but the amount was not credited to Negan's account, in such case the rollback of operation should occur to maintain the atomicity of transaction. It is difficult to achieve atomicity in file processing systems.
- ***Data Security:*** *Data should be secured from unauthorized access, for example a student in a college should not be able to see the payroll details of the teachers, such kind of security constraints are difficult to apply in file processing systems.*

6. What difficulties would you face if you used file system directly to implement a database application? What is physical data independence? (3+1/2071 Bhadra)

7. Distinguish between a database and a DBMS. What is the advantages of separating the logical level and physical level in database design?(2+2/2071 Magh)

Difference between DBMS and Database

A database is a collection of organized data whereas the system that manages a collection of databases is called a Database Management System.

The database holds the records, fields and cells of data whereas the DBMS is the tool used to manipulate the data inside the database.

However, the term database is increasingly used as shorthand for Database Management System. To make the distinction simple, consider and operating system and the individual files stored in the system. Just like you need an operating system to access and modify files in the system, you need a DBMS to manipulate databases stored in the database system.

Next

The diagrammatic representation of the logical and physical data independence is as a shown below:

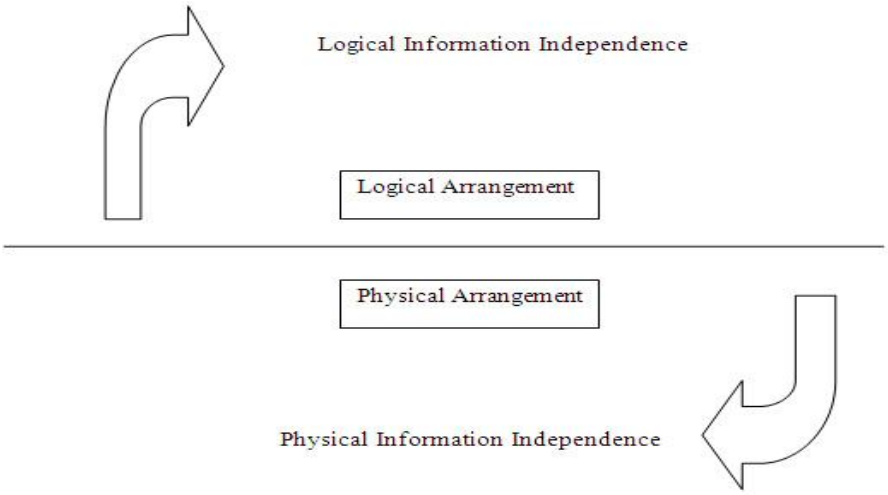


Fig 3: Types of data independence

Logical Data Independence

Logical data independence points out that the conceptual pattern can be altered by undamaging the current external patterns or schemas. The external level and conceptual level has mapping in between them and it takes all the made alterations. It also protects and isolates application programs from actions like combination of dual records into a single record or separating a single record into two or more records.

Logical data independence is a complex parameter to succeed when compared to the physical data independence because it needs pliancy in the scheme of database. For example, two users A and B select the same fields 'student name' and 'student roll number' then user A adds a new column to the field i.e. 'student age' then this does not affect the external view of user B but the internal patterns of both the users will be changed.

Physical Data Independence

Physical data independence points out the physical storing patterns changes by undamaging conceptual structures or arrangements. The presence of internal level in the architecture of database and the operation of changes from the conceptual level to internal level achieves the physical data independence.

Mapping between conceptual level and internal level provides a way to propagate from conceptual records to physical or stored records. If sophistication is made in the physical devices then likewise changes should be made in mapping of conceptual level and internal level which maintains conceptual level unchanged. To make conceptual schema as physically independent of data then external patterns defined on conceptual schema should be physical data independent

9. Briefly explain different levels of data abstraction in a database system.(4/2070 Bhadra)
10. Explain difference between DDL, DML and DCL along with examples (4/2070 Magh).
11. Briefly highlight your significant differences between a file-processing system and a DBMS.(4/2069 Bhadra)
12. Describe the levels of abstraction of database. What do you understand by physical data independence? (4+2/2068 Magh)
13. What are the drawbacks of using file systems to store data? Explain. (6/2068 Bhadra)