Difference between DBMS and File processing system

1)Data Redundancy and Inconsistency
2)Difficulty in accessing the data
3)Data Isolation
4)Integrity Problems
5)Concurrent Access Anomalies
6)Security Problems
7)Atomicity Problem

Introduction to Database Management System

$(A'+B')'$

Bases	DBMS	Flat file system
Definition	DBMS is a collection of interrelated data and software programs to access those data.	Flat file system stores data in a plain text file. Here, the records are specified in a single line.
Data redundancy	There is no problem of data redundancy.	There is main problem of data redundancy.
Cost	DBMS software are very costly and also regular update makes it costly.	Flat file are cost effective.
Use	Mostly, large organizations use DBMS who can afford it and have a large number of client and employees to be managed.	Small organizations use it as it is cost effective and who have to deal with small number of clients and employees.
Views	Views are created and an employees can't see all information available, hence there is security.	Any information can be seen by anyone, hence there is no security.

DBMS VS Flat file system

View level

View 1

View 2

View N

Conceptual level

Logical level

Internal level

Physical level

Stored database

Data abstraction

- It is a process in which developers hide complexities of data management from users through several level of abstraction.
- It defines views; which user can view which part.
- It only shows a part of database that a user needs.

Data abstraction

- It is the lowest level of abstractions which describes how the data are stored in Database.
- DBMS developer is concerned with this level.
- This level emphasis on minimizing the number of disks access so that data can be retrieved very fast.
- It describes data structures and access method in detail.
- Storage allocation e.g. B tree, B^+ tree, hashing, data compression and encryption techniques are describe.

$(A'+B')'$

Physical Level

- This level describes what data are stored in the database and what relationships exist among those data.
- The Logical level of abstraction is used by the database administrators
- Describes the stored data in terms of the data model.

Logical Level

- Every user don't need to access all the information stored in the database.
- This level is basically concerned with dividing the database according to the need of the database users.
- It simplifies the interaction of the users with the system.

$(A'+B')'$

View Level

- Data Definition Language(DDL):
  - Database language that is used to create, drop or modify database schema is called DDL.
  - It is used by Database Administrators(DBA) to specify the conceptual schema.
  - DDL interpreter converts DDL statements into equivalent low level statements understood by the DBMS.
  - Normally create, alter, and drop statements are DDL statements.
  - DDL statements make changes in the schema

Database Languages (DDL, DML,DCL)

Example: For create command

```
create table Student
(
sid char(4),
sname varchar(50),
standard integer
);
```

Example: For drop command

```
drop Student;
```

Example: For alter command

```
alter table Student
ADD COLUMN address varchar(20)
;
```

DDL cont'd

- Database language that enables insert, update, delete, and retrieval of data from the database is called Data Manipulation Language.
- DML complier converts DML statements into equivalent low level statements that the database understands.
- Normally insert, update, delete, select are DML commands.
- DML reflects change in the instance, not the schema.

$(A'+B')'$

DML

Example: For insert

Insert into Student
values("A-101",
"Ramesh", 12);

Example: for update

Update Student
Set class = 11
Where sid = "A-101"
;

Example: for select

Select *
From Student

Example: For delete

Delete from standard
Where sname = "Kriti"

Note: In SQL, cases are insensitive. So, instead of Student one can write StUdEnT as well.

Also, for integer values "12" is incorrect but 12 is correct.

And, for char and varchar "Kriti" is correct and Kriti is incorrect.

DML cont'd

- Data Control Language(DCL):

  Used to control the accessibility of users to the database.

  Example are Grant, Revoke.

  Grant SELECT privilege on a table to a specific user:

  GRANT SELECT ON employees TO Amit;

  Grant INSERT, UPDATE, DELETE privileges on a table to a specific user with the option to grant those privileges to other users :

  GRANT INSERT, UPDATE, DELETE ON products TO manager WITH GRANT OPTION;

(A-1B)

DCL

- Data Control Language(DCL):

Revoke SELECT privilege on a table from a specific user :

REVOKE SELECT ON employees FROM Amit;

Revoke all privileges on a table from a specific user :

REVOKE ALL PRIVILEGES ON products FROM manager;

$(A'+B')'$

DCL

- Users are distinguished by the way they interact with the database system.

1. Naïve users:

They interact with the system by invoking one of the application programs written previously. Naive users are bank teller, receptionist, etc.

2. Application programmers:

They are specializes computer professionals who write the application programs for naïve users .

Database users and administrators

3. Sophisticated end-users:

Interact with the system without writing program. They make request by writing query in a database query language.

4. Specialized end-users:

These users typically have a deeper understanding of the database schema, data structures, and querying languages, allowing them to perform advanced tasks beyond basic data entry or retrieval. They are sophisticated users who write special database applications programs like CAD systems, knowledge based and expert systems, complex data systems(audio/video), etc.

5. System Analysts:

Determine the requirements of end users, especially naïve end-users and develop specifications for canned transaction that meet these requirements.

Database users and administrators cont'd

Database Administrators:

DBA is a person having central control over data and the associated application programs. He is the one who can define schema, install new software, enforcing security to the system, etc.

Major responsibilities of DBA are:

1. Define schema and modify as per needed
2. Install new software for efficient operations
3. Enforcing and monitoring security
4. Analyzing the data stored in the DBMS
5. Keeping the backup of the DBMS periodically
6. Ensuring the state of the hardware and software

Database users and administrators cont'd

$(A'+B')'$

- Collection of operations that form a single logical unit of work are called transaction. A transaction typically consists of one or more database operations, such as inserts, updates, or deletions, that are executed as a single, indivisible unit.
- Transaction management ensures that the database system have ACID properties.

A= Atomicity

C= Consistency

I = Isolation

D = Durability

$(A'+B')'$

Transaction Management

1. Atomicity:

This property ensures that a transaction is treated as a single, indivisible unit of work. Either all the operations within the transaction are completed successfully, or none of them are applied to the database. If any part of the transaction fails (due to an error or any other reason), the entire transaction is rolled back, and the database is restored to its state before the transaction began.

Properties of Transaction Management (ACID property)

2. Consistency:

Consistency ensures that the database remains in a valid state both before and after the execution of a transaction. In other words, the integrity constraints, data integrity rules, and referential integrity constraints defined in the database schema must be preserved during the execution of a transaction. This property ensures that the database does not enter an inconsistent state, even in the presence of failures or concurrent transactions.

Properties of Transaction Management (ACID property)

3. Isolation:

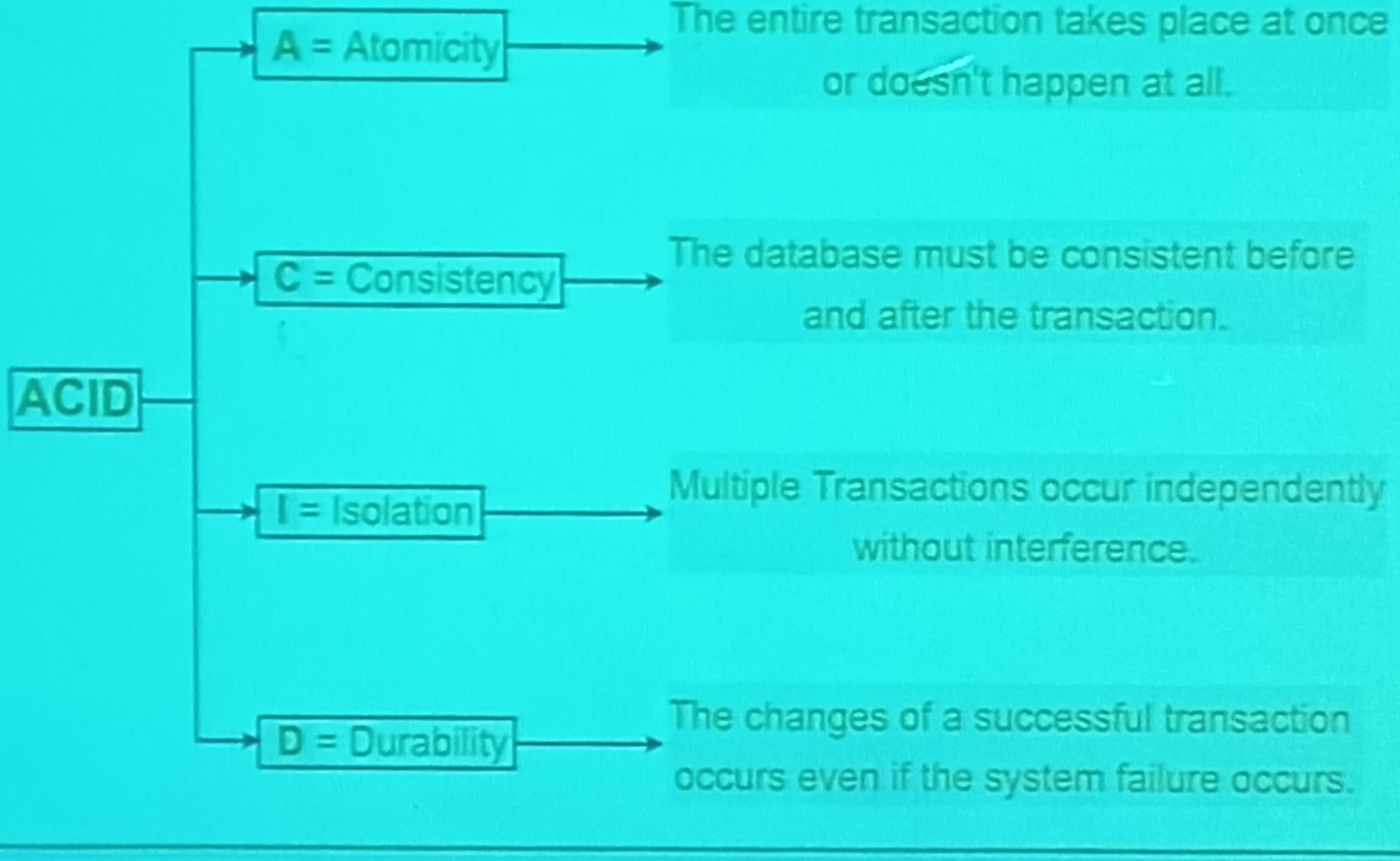
Even if many transaction may be done at the same time but isolation ensures that if transaction A and B are executing concurrently, then either A must execute first then B is executed or, B must execute first then A is executed.

4. Durability:

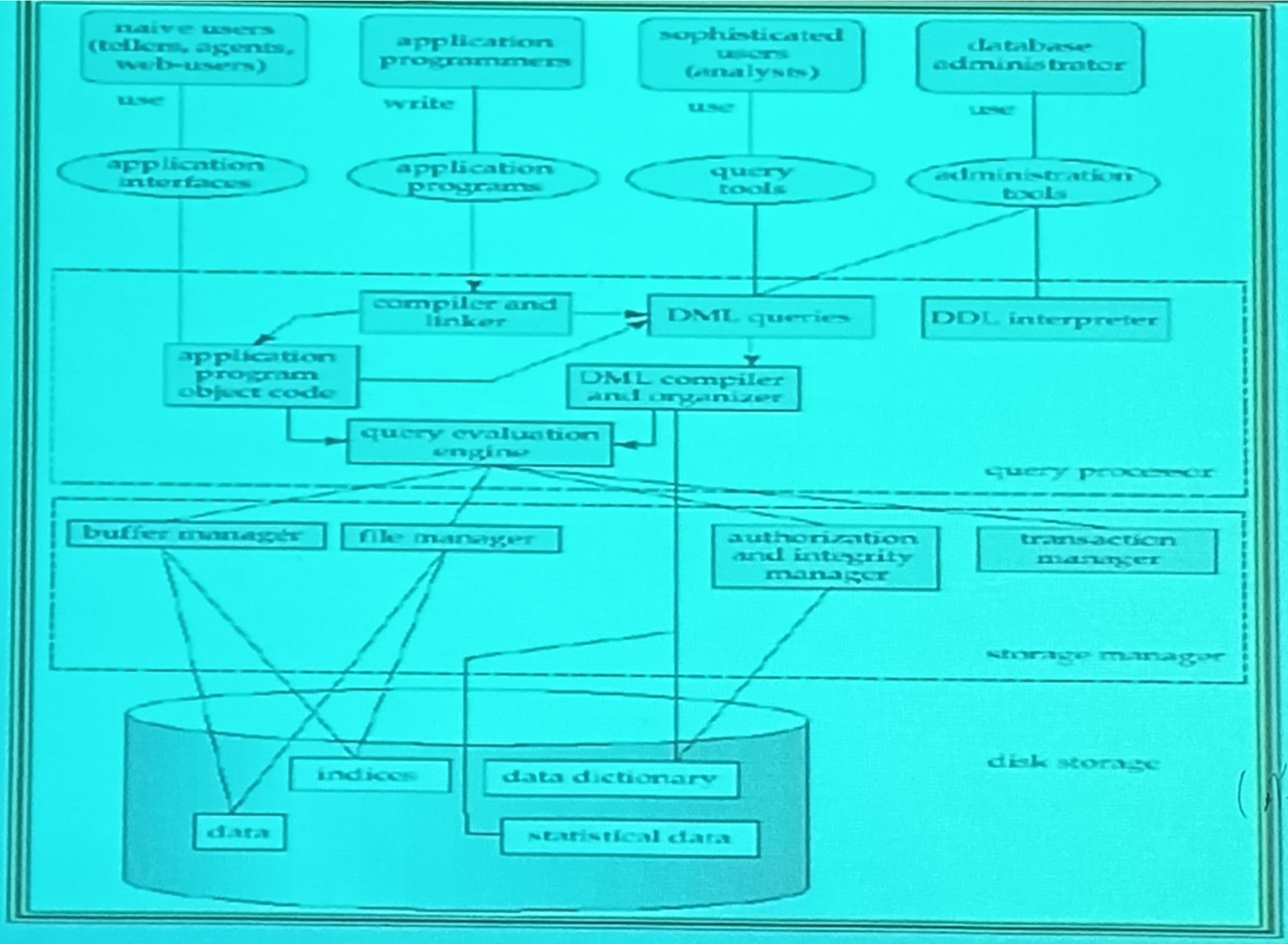
Once a transaction is successfully completed, its effect should persist even if the system crashes before all changes are reflected in the disk. This effect should never be lost even in subsequent system failure.

Properties of Transaction Management (ACID property)

ACID Properties in DBMS



$(A'+B')'$



Database system structure

1. Storage Manager

- Storage manager stores, retrieves, and updates data in the database.
- Its components are:
  - **Authorization and integrity manager**: It checks for the correctness of the constraints specified and also checks the authenticity of the users while entering the database.
  - **Transaction manager**: It ensures that the database follows the ACID property.
  - **File manager**: It is responsible to manage allocation of space on the disk storage.
  - **Buffer manager**: It is responsible for fetching data from the disk storage and to decide what data to cache in the main memory. It enables to handle data sizes which are larger than the size of the disk.

- It is responsible for allocating the disk space and also for maintaining the data in the database.
- Its components are:
  - **Data files**: It is the place where database is stored.
  - **Data dictionary**: It stores metadata about the schema of the database.
  - **Indices**: It provides quick access to the data items that hold particular values.

$(A'+B')'$

2. Disk Storage

- It simplifies and facilitates easy access to the data.
- It translates queries written in non-procedural language, and operates the statement.
- Its components are:
  - **DDL interpreter**: It interprets DDL statements into low-level language understood by the system.
  - **DML compiler**: It translates DML statements into an evaluation plan consisting low level instructions that the query evaluation engine understands.
  - **Query evaluation engine**: It executes low level statements generated by the DML compiler.

$(A'+B')'$

3. Query Processor

- In two tier architecture, user interface and application programs are on the client side and query and transaction facility are on the server side
- When DBMS access is required, the application program establishes connection with client and server
- Server provides response to the authenticated client request/queries.
- There is direct interaction of client with the server
- The business logic coupled with either client side or the server side.

$(A'+B')'$

Two-tier architecture