

1. A Database Management System is a type of _____ software.
- a. **It is a type of system software ✓**
 - b. It is a kind of application software
 - c. It is a kind of general software
 - d. Both A and C

Explanation: The DBMS (or Database Management System) is a kind of system software used for several operations such as creating tables/databases, storing data, managing databases. It also allows modifying the data stored in the database as well.

2. A huge collection of the information or data accumulated from several different sources is known as _____:
- a. Data Management
 - b. Data Mining
 - c. **Data Warehouse ✓**
 - d. Both B and C

Explanation: A huge collection of different types of data/information which collected from many different sources stored in one place is called a data warehouse. It works just like the standard warehouse, which is generally a type of holding area/building to store the different types of different goods in huge quantities until they are further distributed. Therefore the correct answer is Data Warehouse

3. To which of the following the term "DBA" referred?
- a. Data Bank Administrator
 - b. **Database Administrator ✓**
 - c. Data Administrator
 - d. None of the above

Explanation: The term "DBA" refers to the Database Administrator that manages the whole database and updates the database on a regular basis. In short, the database administrator has the responsibility of the whole database and to maintain it in an optimal & stable state.

4. In general, a file is basically a collection of all related _____.
- a. Rows & Columns
 - b. Fields
 - c. Database
 - d. **Records ✓**

Explanation: Whenever we have some related data, information or records, we collect all those related data (or records), put them together, store them in one place, and give that collection a name that is known as a file.

5. The term "Data" refers to:
- a. The electronic representation of the information(or data)
 - b. Basic information
 - c. **Row Facts and figures ✓**
 - d. Both A and C

Explanation: In general, the term "data" refers to the row facts and figure, whereas the information referred to as the data, which is really important for someone or a particular person. Therefore the correct answer is C.

6. Which of the following refers to the number of tuples in a relation?
- a. Entity
 - b. Column
 - c. **Cardinality ✓**
 - d. None of the above

Explanation: Cardinality refers to the number of tuples of relation because cardinality represents the number of tuples in a relation.

To understand it in more detail, consider the following given example:

Suppose we have a relation (or table) that contains 30 tuples (or Rows) and four columns, so the cardinality of our relation will be 30.

7. Which one of the following is a type of Data Manipulation Command?

- a. Create
- b. Alter
- c. Delete ✓**
- d. All of the above

Explanation: In data manipulation language, the command like select, insert, update, and delete is used to manipulate the information (or data, records), for example create a table, update table delete table, etc. Therefore the correct answer is C.

8. Which of the following is a top-down approach in which the entity's higher level can be divided into two lower sub-entities?

- a. Aggregation
- b. Generalization
- c. Specialization ✓**
- d. All of the above

Explanation: In specialization, the top-down approach is used, and it is apposite to the generalization. In the specialization, the higher-level entity can be divided into sub lower entities. It is generally used for identifying the subset of an entity set which share the distinguishing characteristics.

To understand it more clearly, consider the following example:

Suppose you have an entity, e.g., A vehicle. So through the specialization, you can be divided further into sub-entities like two-wheelers and four-wheelers. Therefore the correct answer is C.

9. The term "TCL" stands for_____.

- a. Ternary Control Language
- b. Transmission Control Language
- c. Transaction Central Language
- d. Transaction Control Language ✓**

Explanation: The term "TCL" refers to the Transaction Control Language, which is another language just like the "DDL" and "DML". The commands like commit, save point, rollback come under the TCL used to control the transactions. Therefore the correct answer will be D.

10. What is the relation calculus?

- a. It is a kind of procedural language
- b. It is a non-procedural language ✓**
- c. It is a high-level language
- d. It is Data Definition language

Explanation: In database management systems, relation calculus is a type of non-procedural query language that describes what data needs to be retrieved. However, it does not explain how to retrieve the data.

Q2. Attempt any four

04*5=20

1. What is DBMS? Explain ACID properties also.
2. How many types of database language are? Define each.
3. What do you understand by Data Model? Define types of data model.
4. What is normalization and type of normalization?
5. What is Deadlock in DBMS and define schemes of Deadlock prevention.

1. What is DBMS? Explain ACID properties also.

- Database management system is software which is used to manage the database. For example: MySQL, Oracle, etc are a very popular commercial database which is used in different applications.
- DBMS provides an interface to perform various operations like database creation, storing data in it, updating data, creating a table in the database and a lot more.
- It provides protection and security to the database. In the case of multiple users, it also maintains data consistency.

DBMS allows users the following tasks:

- **Data Definition:** It is used for creation, modification, and removal of definition that defines the organization of data in the database.
- **Data Updation:** It is used for the insertion, modification, and deletion of the actual data in the database.
- **Data Retrieval:** It is used to retrieve the data from the database which can be used by applications for various purposes.
- **User Administration:** It is used for registering and monitoring users, maintain data integrity, enforcing data security, dealing with concurrency control, monitoring performance and recovering information corrupted by unexpected failure.

Characteristics of DBMS

- It uses a digital repository established on a server to store and manage the information.
- It can provide a clear and logical view of the process that manipulates data.
- DBMS contains automatic backup and recovery procedures.
- It contains ACID properties which maintain data in a healthy state in case of failure.
- It can reduce the complex relationship between data.
- It is used to support manipulation and processing of data.
- It is used to provide security of data.
- It can view the database from different viewpoints according to the requirements of the user.

Advantages of DBMS

- **Controls database redundancy:** It can control data redundancy because it stores all the data in one single database file and that recorded data is placed in the database.
- **Data sharing:** In DBMS, the authorized users of an organization can share the data among multiple users.
- **Easily Maintenance:** It can be easily maintainable due to the centralized nature of the database system.
- **Reduce time:** It reduces development time and maintenance need.
- **Backup:** It provides backup and recovery subsystems which create automatic backup of data from hardware and software failures and restores the data if required.
- **multiple user interface:** It provides different types of user interfaces like graphical user interfaces, application program interfaces

Disadvantages of DBMS

- **Cost of Hardware and Software:** It requires a high speed of data processor and large memory size to run DBMS software.
- **Size:** It occupies a large space of disks and large memory to run them efficiently.
- **Complexity:** Database system creates additional complexity and requirements.
- **Higher impact of failure:** Failure is highly impacted the database because in most of the organization, all the data stored in a single database and if the database is damaged due to electric failure or database corruption then the data may be lost forever.

ACID Properties in DBMS

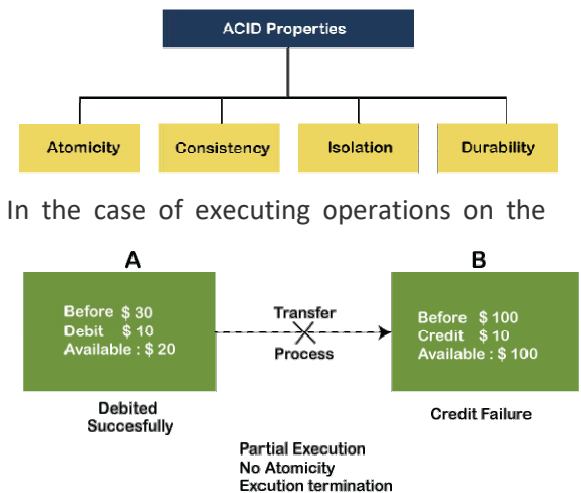
DBMS is the management of data that should remain integrated when any changes are done in it. It is because if the integrity of the data is affected, whole data will get disturbed and corrupted. Therefore, to maintain the integrity of the data, there are four properties described in the database management system, which are known as the **ACID** properties. The ACID properties are meant for the transaction that goes through a different group of tasks, and there we come to see the role of the ACID properties.

ACID Properties

1. **Atomicity:** The term atomicity defines that the data remains atomic. It means if any operation is performed on the data, either it should be performed or executed completely or should not be executed at all. It further means that the operation should not break in between or execute partially. In the case of executing operations on the transaction, the operation should be completely executed and not partially.

Example: If Remo has account A having \$30 in his account from which he wishes to send \$10 to Sheero's account, which is B. In account B, a sum of \$ 100 is already present. When \$10 will be transferred to account B, the sum will become \$110. Now, there will be two operations that will take place. One is the amount of \$10 that Remo wants to transfer will be debited from his account A, and the same amount will get credited to account B, i.e., into Sheero's account. Now, what happens - the first operation of debit executes successfully, but the credit operation, however, fails. Thus, in Remo's account A, the value becomes \$20, and to that of Sheero's account, it remains \$100 as it was previously present.

In the above diagram, it can be seen that after crediting \$10, the amount is still \$100 in account B. So, it is not an atomic transaction.

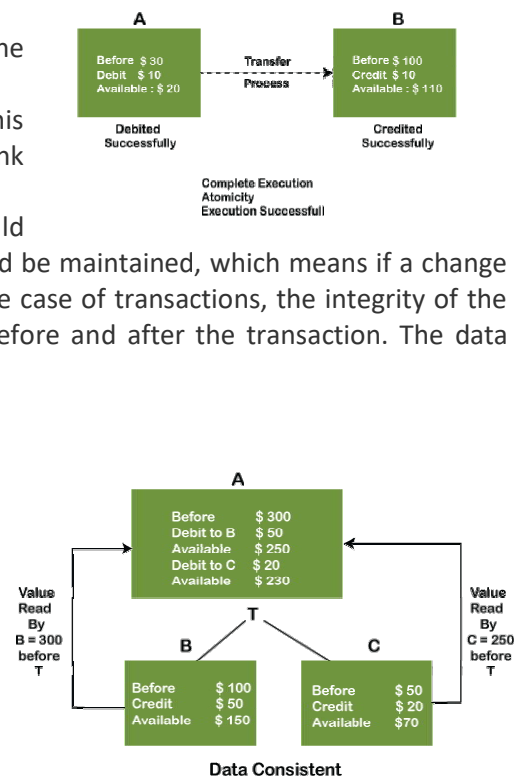


The below image shows that both debit and credit operations are done successfully. Thus the transaction is atomic.

Thus, when the amount loses atomicity, then in the bank systems, this becomes a huge issue, and so the atomicity is the main focus in the bank systems.

2. **Consistency:** The word **consistency** means that the value should remain preserved always. In **DBMS**, the integrity of the data should be maintained, which means if a change in the database is made, it should remain preserved always. In the case of transactions, the integrity of the data is very essential so that the database remains consistent before and after the transaction. The data should always be correct.

Example: In the above figure, there are three accounts, A, B, and C, where A is making a transaction T one by one to both B & C. There are two operations that take place, i.e., Debit and Credit. Account A firstly debits \$50 to account B, and the amount in account A is read \$300 by B before the transaction. After the successful transaction T, the available amount in B becomes \$150. Now, A debits \$20 to account C, and that time, the value read by C is \$250 (that is correct as a debit of \$50 has been successfully done to B). The debit and credit operation from account A to C has been done successfully. We can see that the transaction is done successfully, and the value is also read correctly. Thus, the data is consistent. In case the value

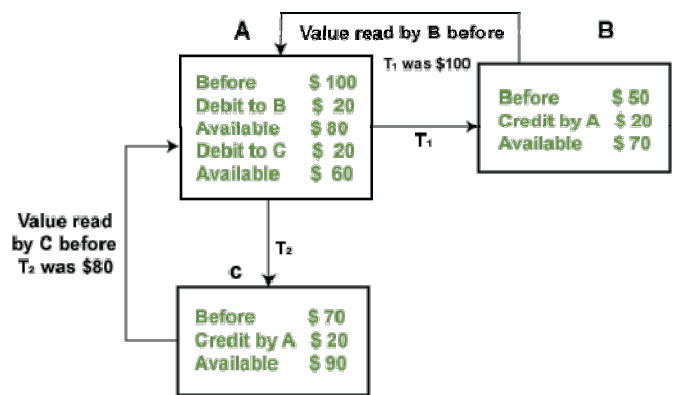


read by B and C is \$300, which means that data is inconsistent because when the debit operation executes, it will not be consistent.

3. **Isolation:** The term 'isolation' means separation. In DBMS, Isolation is the property of a database where no data should affect the other one and may occur concurrently. In short, the operation on one database should begin when the operation on the first database gets complete. It means if two operations are being performed on two different databases, they may not affect the value of one another. In the case of transactions, when two or more transactions occur simultaneously, the consistency should remain maintained. Any changes that occur in any particular transaction will not be seen by other transactions until the change is not committed in the memory.

Example: If two operations are concurrently running on two different accounts, then the value of both accounts should not get affected. The value should remain persistent. As you can see in the below diagram, account A is making T1 and T2 transactions to account B and C, but both are executing independently without affecting each other. It is known as Isolation.

4. **Durability:** Durability ensures the permanency of something. In DBMS, the term durability ensures that the data after the successful execution of the operation becomes permanent in the database. The durability of the data should be so perfect that even if the system fails or leads to a crash, the database still survives. However, if gets lost, it becomes the responsibility of the recovery manager for ensuring the durability of the database. For committing the values, the COMMIT command must be used every time we make changes.



Isolation - Independent execution of T₁ & T₂ by A

Therefore, the ACID property of DBMS plays a vital role in maintaining the consistency and availability of data in the database. Thus, it was a precise introduction of ACID properties in DBMS. We have discussed these properties in the transaction section also.

2. How many types of database language are? Define each.

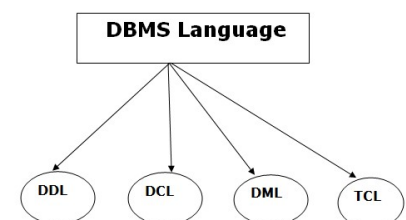
Database Language

- A DBMS has appropriate languages and interfaces to express database queries and updates.
- Database languages can be used to read, store and update the data in the database.

Types of Database Language

1. Data Definition Language

- **DDL** stands for **Data Definition Language**. It is used to define database structure or pattern.
- It is used to create schema, tables, indexes, constraints, etc. in the database.
- Using the DDL statements, you can create the skeleton of the database.



- Data definition language is used to store the information of metadata like the number of tables and schemas, their names, indexes, columns in each table, constraints, etc.

Here are some tasks that come under DDL:

- **Create:** It is used to create objects in the database.
- **Alter:** It is used to alter the structure of the database.
- **Drop:** It is used to delete objects from the database.
- **Truncate:** It is used to remove all records from a table.
- **Rename:** It is used to rename an object.
- **Comment:** It is used to comment on the data dictionary.

These commands are used to update the database schema that's why they come under Data definition language.

2. Data Manipulation Language

DML stands for **Data Manipulation Language**. It is used for accessing and manipulating data in a database. It handles user requests.

Here are some tasks that come under DML:

- **Select:** It is used to retrieve data from a database.
- **Insert:** It is used to insert data into a table.
- **Update:** It is used to update existing data within a table.
- **Delete:** It is used to delete all records from a table.
- **Merge:** It performs UPSERT operation, i.e., insert or update operations.
- **Call:** It is used to call a structured query language or a Java subprogram.
- **Explain Plan:** It has the parameter of explaining data.
- **Lock Table:** It controls concurrency.

3. Data Control Language

- **DCL** stands for **Data Control Language**. It is used to retrieve the stored or saved data.
- The DCL execution is transactional. It also has rollback parameters.
(But in Oracle database, the execution of data control language does not have the feature of rolling back.)

Here are some tasks that come under DCL:

- **Grant:** It is used to give user access privileges to a database.
- **Revoke:** It is used to take back permissions from the user.

There are the following operations which have the authorization of Revoke:
CONNECT, INSERT, USAGE, EXECUTE, DELETE, UPDATE and SELECT.

4. Transaction Control Language

TCL is used to run the changes made by the DML statement. TCL can be grouped into a logical transaction. Here are some tasks that come under TCL:

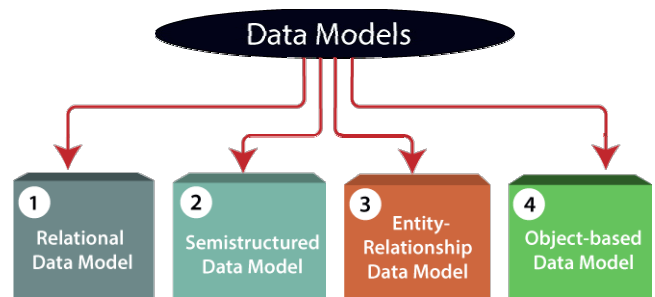
- **Commit:** It is used to save the transaction on the database.

Rollback: It is used to restore the database to original since the last Commit.

3. What do you understand by Data Model? Define types of data model.

Data Model is the modeling of the data description, data semantics, and consistency constraints of the data. It provides the conceptual tools for describing the design of a database at each level of data abstraction. Therefore, there are following four data models used for understanding the structure of the database:

1. Relational Data Model: This type of model designs the data in the form of rows and columns within a table. Thus, a relational model uses tables for representing data and in-between relationships. Tables are also called relations. This model was initially described by Edgar F. Codd, in 1969. The relational data model is the widely used model which is primarily used by commercial data processing applications.



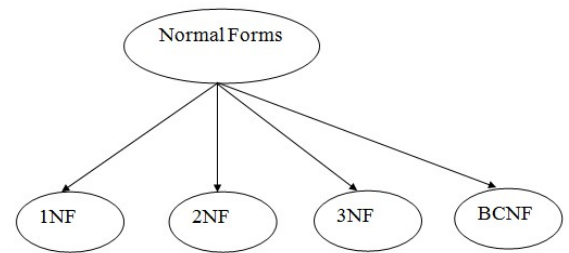
2. Entity-Relationship Data Model: An ER model is the logical representation of data as objects and relationships among them. These objects are known as entities, and relationship is an association among these entities. This model was designed by Peter Chen and published in 1976 papers. It was widely used in database designing. A set of attributes describe the entities. For example, student_name, student_id describes the 'student' entity. A set of the same type of entities is known as an 'Entity set', and the set of the same type of relationships is known as 'relationship set'.

3) Object-based Data Model: An extension of the ER model with notions of functions, encapsulation, and object identity, as well. This model supports a rich type system that includes structured and collection types. Thus, in 1980s, various database systems following the object-oriented approach were developed. Here, the objects are nothing but the data carrying its properties.

4) Semistructured Data Model: This type of data model is different from the other three data models (explained above). The semistructured data model allows the data specifications at places where the individual data items of the same type may have different attributes sets. The Extensible Markup Language, also known as XML, is widely used for representing the semistructured data. Although XML was initially designed for including the markup information to the text document, it gains importance because of its application in the exchange of data.

4. What is normalization and type of normalization?

- Normalization is the process of organizing the data in the database.
- Normalization is used to minimize the redundancy from a relation or set of relations. It is also used to eliminate the undesirable characteristics like Insertion, Update and Deletion Anomalies.
- Normalization divides the larger table into the smaller table and links them using relationship.
- The normal form is used to reduce redundancy from the database table.



Types of Normal Forms

There are the four types of normal forms:

Normal Form	Description
<u>1NF</u>	A relation is in 1NF if it contains an atomic value.
<u>2NF</u>	A relation will be in 2NF if it is in 1NF and all non-key attributes are fully functional dependent on the primary key.
<u>3NF</u>	A relation will be in 3NF if it is in 2NF and no transition dependency exists.
<u>4NF</u>	A relation will be in 4NF if it is in Boyce Codd normal form and has no multi-valued dependency.
<u>5NF</u>	A relation is in 5NF if it is in 4NF and not contains any join dependency and joining should be lossless.

5. What is Deadlock in DBMS and define schemes of Deadlock prevention.

A deadlock is a condition where two or more transactions are waiting indefinitely for one another to give up locks. Deadlock is said to be one of the most feared complications in DBMS as no task ever gets finished and is in waiting state forever.

For example: In the student table, transaction T1 holds a lock on some rows and needs to update some rows in the grade table. Simultaneously, transaction T2 holds locks on some rows in the grade table and needs to update the rows in the Student table held by Transaction T1.

Now, the main problem arises. Now Transaction T1 is waiting for T2 to release its lock and similarly, transaction T2 is waiting for T1 to release its lock. All activities come to a halt state and remain at a standstill. It will remain in a standstill until the DBMS detects the deadlock and aborts one of the transactions.

Deadlock Avoidance

- When a database is stuck in a deadlock state, then it is better to avoid the database rather than aborting or restating the database. This is a waste of time and resource.
- Deadlock avoidance mechanism is used to detect any deadlock situation in advance. A method like "wait for graph" is used for detecting the deadlock situation but this method is suitable only for the smaller database. For the larger database, deadlock prevention method can be used.

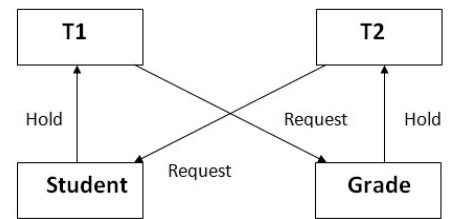


Figure: Deadlock in DBMS

Deadlock Detection

In a database, when a transaction waits indefinitely to obtain a lock, then the DBMS should detect whether the transaction is involved in a deadlock or not. The lock manager maintains a Wait for the graph to detect the deadlock cycle in the database.

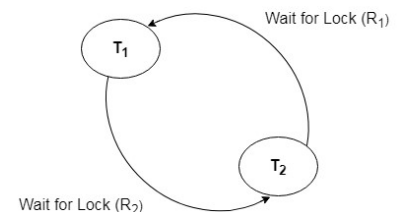
Wait for Graph

- This is the suitable method for deadlock detection. In this method, a graph is created based on the transaction and their lock. If the created graph has a cycle or closed loop, then there is a deadlock.
- The wait for the graph is maintained by the system for every transaction which is waiting for some data held by the others. The system keeps checking the graph if there is any cycle in the graph.

The wait for a graph for the above scenario is shown below:

Deadlock Prevention

- Deadlock prevention method is suitable for a large database. If the resources are allocated in such a way that deadlock never occurs, then the deadlock can be prevented.
- The Database management system analyzes the operations of the transaction whether they can create a deadlock situation or not. If they do, then the DBMS never allowed that transaction to be executed.



Wait-Die scheme

In this scheme, if a transaction requests for a resource which is already held with a conflicting lock by another transaction then the DBMS simply checks the timestamp of both transactions. It allows the older transaction to wait until the resource is available for execution.

Let's assume there are two transactions T_i and T_j and let $TS(T)$ is a timestamp of any transaction T . If T_2 holds a lock by some other transaction and T_1 is requesting for resources held by T_2 then the following actions are performed by DBMS:

1. Check if $TS(T_i) < TS(T_j)$ - If T_i is the older transaction and T_j has held some resource, then T_i is allowed to wait until the data-item is available for execution. That means if the older transaction is waiting for a resource which

is locked by the younger transaction, then the older transaction is allowed to wait for resource until it is available.

2. Check if $TS(T_i) < TS(T_j)$ - If T_i is older transaction and has held some resource and if T_j is waiting for it, then T_j is killed and restarted later with the random delay but with the same timestamp.

Wound wait scheme

- In wound wait scheme, if the older transaction requests for a resource which is held by the younger transaction, then older transaction forces younger one to kill the transaction and release the resource. After the minute delay, the younger transaction is restarted but with the same timestamp.
- If the older transaction has held a resource which is requested by the Younger transaction, then the younger transaction is asked to wait until older releases it.