

Database Management Systems | Set 5

Database Management Systems | Set 6

Database Management Systems | Set 7

Database Management Systems | Set 11

Equivalence of Functional Dependencies

Canonical Cover of Functional Dependencies in DBMS

Introduction of Database Normalization

Normal Forms in DBMS

Database Management System | Dependency Preserving Decomposition

Decomposition

Lossless Decomposition in DBMS

Lossless Join and Dependency Preserving Decomposition

How to find the highest normal form of a relation

Concurrency Control in DBMS

ACID Properties in DBMS

Conflict Serializability in DBMS

View Serializability in DBMS Transactions

Recoverability in DBMS

Precedence Graph For Testing Conflict Serializability in DBMS

Indexing in Databases | Set 1

Introduction of B-Tree

Insert Operation in B-Tree

Delete Operation in B-Tree

Most asked Computer Science Subjects Interview Questions in Amazon, Microsoft, Flipkart

Second Normal Form (2NF)

First Normal Form (1NF)

Third Normal Form (3NF)

Boyce-Codd Normal Form (BCNF)

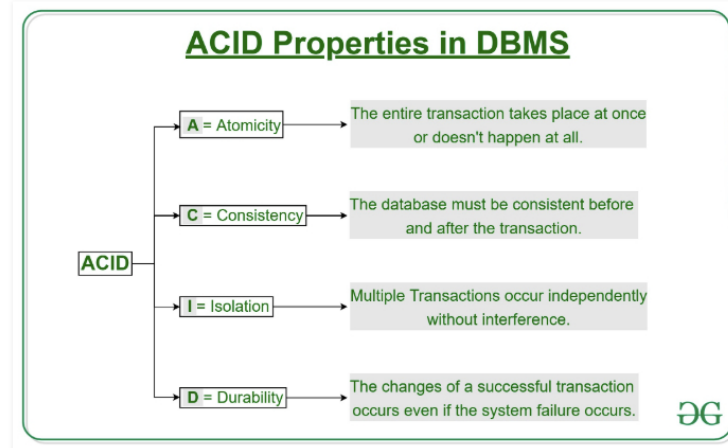
Advanced SQL Interview Questions

Extendible Hashing (Dynamic Hashing)

ACID Properties in DBMS

A **transaction** is a single logical unit of work which accesses and possibly modifies the contents of a database. Transactions access data using read and write operations.

In order to maintain consistency in a database, before and after the transaction, certain properties are followed. These are called **ACID** properties.



Atomicity

By this, we mean that either the entire transaction takes place at once or doesn't happen at all. There is no midway i.e. transactions do not occur partially. Each transaction is considered as one unit and either runs to completion or is not executed at all. It involves the following two operations.

—**Abort**: If a transaction aborts, changes made to database are not visible.

—**Commit**: If a transaction commits, changes made are visible.

Atomicity is also known as the 'All or nothing rule'.

Consider the following transaction **T** consisting of **T1** and **T2**: Transfer of 100 from account **X** to account **Y**.

Before: X : 500	Y: 200
Transaction T	
T1	T2
Read (X)	Read (Y)
X: = X - 100	Y: = Y + 100
Write (X)	Write (Y)
After: X : 400	Y : 300

If the transaction fails after completion of **T1** but before completion of **T2**. (say, after **write(X)** but before **write(Y)**), then amount has been deducted from **X** but not added to **Y**. This results in an inconsistent database state. Therefore, the transaction must be executed in entirety in order to ensure correctness of database state.

Consistency

This means that integrity constraints must be maintained so that the database is consistent before and after the transaction. It refers to the correctness of a database. Referring to the example above,

The total amount before and after the transaction must be maintained.

Total **before T** occurs = **500 + 200 = 700**.

Total **after T** occurs = **400 + 300 = 700**.

Therefore, database is **consistent**. Inconsistency occurs in case **T1** completes but **T2** fails. As a result **T** is incomplete.

Isolation

This property ensures that multiple transactions can occur concurrently without leading to the inconsistency of database state. Transactions occur independently without interference. Changes occurring in a particular transaction will not be visible to any other transaction until that particular change in that transaction is written to memory or has been committed. This property ensures that the execution of transactions concurrently will result in a state that is equivalent to a state achieved these were executed serially in some order.

Let **X** = 500, **Y** = 500.

Starting from
6th January 2019

Geeks Classes

LIVE Sessions

Weekdays

₹17,999
₹9,999

Register Now



Get DSA Online Course
complementary with this course

Most popular in DBMS

Architecture of Apache Cassandra

Properties of Relational Decomposition

MySQL | Database Files

Applications of Commercial Deductive Database Systems

What is Temporary Table in SQL?

Most visited in GATE CS

LOOK Disk Scheduling Algorithm

C-SCAN Disk Scheduling Algorithm

Difference between Fixed and Dynamic Channel Allocations

System Protection in Operating System

Personnel involved in Database Management System

Consider two transactions **T** and **T''**.

T	T''
Read (X)	Read (X)
X: = X*100	Read (Y)
Write (X)	Z: = X + Y
Read (Y)	Write (Z)
Y: = Y - 50	
Write	

Suppose **T** has been executed till **Read (Y)** and then **T''** starts. As a result, interleaving of operations takes place due to which **T''** reads correct value of **X** but incorrect value of **Y** and sum computed by

T'': ($X+Y = 50,000+500=50,500$)

is thus not consistent with the sum at end of transaction:

T: ($X+Y = 50,000 + 450 = 50,450$).

This results in database inconsistency, due to a loss of 50 units. Hence, transactions must take place in isolation and changes should be visible only after they have been made to the main memory.

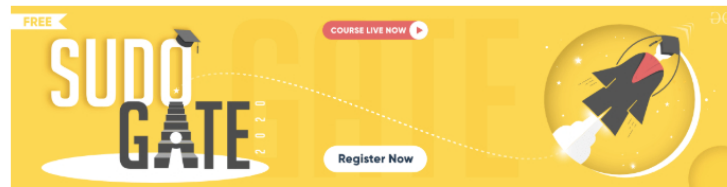
Durability:

This property ensures that once the transaction has completed execution, the updates and modifications to the database are stored in and written to disk and they persist even if a system failure occurs. These updates now become permanent and are stored in non-volatile memory. The effects of the transaction, thus, are never lost.

The **ACID** properties, in totality, provide a mechanism to ensure correctness and consistency of a database in a way such that each transaction is a group of operations that acts a single unit, produces consistent results, acts in isolation from other operations and updates that it makes are durably stored.

This article is contributed by **Avneet Kaur**. If you like GeeksforGeeks and would like to contribute, you can also write an article using contribute.geeksforgeeks.org or mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.



Recommended Posts:

[Properties of Relational Decomposition](#)
[Properties of Determinants of Matrices](#)
[Mathematics | Power Set and its Properties](#)
[Properties of Boolean Algebra](#)
[Properties of Asymptotic Notations](#)
[Need for DBMS](#)
[Starvation in DBMS](#)
[Deadlock in DBMS](#)
[Disadvantages of DBMS](#)
[Interfaces in DBMS](#)
[Recoverability in DBMS](#)
[Difference between DDL and DML in DBMS](#)
[Cascadeless in DBMS](#)
[Database Objects in DBMS](#)
[Lossless Decomposition in DBMS](#)

Improved By : ArchitNangalia

Article Tags : DBMS GATE CS DBMS-Transactions and Concurrency Control

Practice Tags : DBMS



1.9

☐ To-do ☐ Done

Based on 47 vote(s)

Feedback/ Suggest Improvement

Add Notes

Improve Article

Please write to us at contribute@geeksforgeeks.org to report any issue with the above content.

Previous

[GATE 2017 Important dates and links](#)

Next

[DBMS Architecture 2-Level, 3-Level](#)

Writing code in comment? Please use ide.geeksforgeeks.org, generate link and share the link here.

12 Comments

GeeksforGeeks

[Login](#)

[Recommend](#) 11

[Tweet](#)

[Share](#)

Sort by Newest



Join the discussion...

LOG IN WITH

OR SIGN UP WITH DISQUS ?



Name

