ACID Properties in DBMS

The ACID properties in DBMS are one of the most important aspects of the database. These are essential as per the GATE CSE syllabus as well.

The ACID properties in DBMS stand for

- A → Atomicity
- C → Consistency
- I → Isolation
- D → Durability

The database is called an ACID transaction if the database operation has these ACID properties, and data storage systems that apply these operations are called transactional systems. ACID transactions guarantee that each table's write, read, or modification has these properties.

Atomicity

One of the ACID properties in DBMS is Atomicity. The transaction should execute all operations, including committing or executing none of them. Users should not worry about incomplete transactions(when a system crash occurs). In recent years, various questions in the GATE question paper have been seen from atomicity.

Let A and B be two bank balances with Rs100 and Rs200, respectively. Transaction T transferred Rs50 from A to B

```
T: read_item(A)
A= A-50
Write_item(A)
read_item(B)
B = B + 50
Write_item(B)
COMMIT
```

If transaction T fails anywhere before COMMIT, undo all the database modifications, and the recovery management component does a rollback.

Consistency

After the atomicity ACID properties in DBMS, the next one is consistency. Users are responsible for ensuring transaction consistency. The user who submits a transaction must ensure that, when to run to completion by itself against a 'consistent database instance, the transaction will leave the database in a consistent state.

For example, the user may (naturally) have the consistency criterion that fund transfers between bank accounts should not change the total amount of money in the accounts. However, to transfer money from one account to another, a transaction must debit one account, temporarily leaving the database inconsistent globally, even though the new account balance may satisfy any integrity constraints concerning the range of acceptable account balances. The user's notion of a consistent database is presented when the second account is credited with the transferred amount.

Suppose a faulty transfer program always credits the second account with one rupee less than the amount debited from the first account. In that case, the DBMS cannot be expected to detect inconsistencies due to such errors in the user program's logic.

Isolation

After the consistency of ACID properties in DBMS, the next one is in isolation. This property is ensured by guaranteeing that, even though actions of several transactions might be interleaved, the net effect is identical to executing all transactions one after the other in some serial order. Isolation property plays an important role in the GATE exam.

For example, if two transactions, T_1 and T_2, are executed concurrently, the net effect is guaranteed to be equivalent to executing (all of) T_1 followed by executing T_2 or executing T_2 followed by executing T_1. (The DBMS provides no guarantees about which of these orders is effectively chosen.)

Furthermore, if each transaction maps a consistent database instance to another consistent database instance, executing several transactions one after the other(on a consistent initial database instance) results in a consistent final database instance.

Durability

The last ACID property in DBMS is durability. The changes that are applied to the database by a committed transaction must be preserved in the database. These changes should not be lost because of any failure that is transaction should be able to recover under any case failure.

For example, transaction failure can be because of the following:

- Power failure
- Software failure
- Hardware crash