**What is ACID (atomicity, consistency, isolation, and durability)?**

In [transaction](https://www.techtarget.com/searchcio/definition/transaction) processing, ACID (atomicity, consistency, isolation, and durability) is an acronym and mnemonic device used to refer to the four essential properties a transaction should possess to ensure the [integrity](https://www.techtarget.com/searchdatacenter/definition/integrity) and reliability of the [data](https://www.techtarget.com/searchdatamanagement/definition/data) involved in the transaction. The acronym is commonly associated with relational database management systems ([RDBMSs](https://www.techtarget.com/searchdatamanagement/definition/RDBMS-relational-database-management-system)) such as [MySQL](https://www.techtarget.com/searchoracle/definition/MySQL) or [SQL Server](https://www.techtarget.com/searchdatamanagement/definition/SQL-Server), although it can apply to any system or [application](https://www.techtarget.com/searchsoftwarequality/definition/application) that processes transactions.

In computing, a transaction is a set of related tasks treated as a single action. Together, the tasks form a logical unit of work. For example, a bank customer might use a [mobile app](https://www.techtarget.com/whatis/definition/mobile-app) to transfer $200 from a savings account to a checking account. The customer [logs into](https://www.techtarget.com/searchsecurity/definition/logon) the app, enters the necessary information and clicks a button, which launches a "transfer" transaction.

The transaction consists of multiple steps, all invisible to the user. The steps might include verifying availability of funds in the savings account, deducting $200 from that account, adding $200 to the checking account, and verifying the final balance in each account to ensure the funds were transferred properly. All these tasks together represent the entire transaction; either they all succeed or none succeed.

If all the steps ran successfully, the transaction is considered complete and is committed to the managing system. The customer is notified that the funds were transferred from one account to the other. If any steps had failed, all completed steps would have been rolled back, and the data returned to its original state. In this case, the user would have received an error message.

Applying the ACID properties determines whether a transaction should be committed or rolled back, given the success or failure of each step in the transaction. If a transaction does not adhere to these properties, the integrity and reliability of the data can be called into question. For example, if the transfer transaction had failed after deducting the $200 from the savings account but before adding the funds to the checking account, the customer's accounts would be short $200.

**What are the ACID properties?**

When data integrity and reliability are top considerations in a [transaction processing system](https://www.techtarget.com/searchdatacenter/definition/OLTP), the system will typically apply four properties to those transactions for ACID-compliance:

* **Atomicity.** A transaction is treated as a single atomic unit. All steps that make up the transaction must succeed or the entire transaction rolls back. If they all succeed, the changes made by the transaction are permanently committed to the managing system. Consider the transfer transaction example. For the transaction to be committed to the [database](https://www.techtarget.com/searchdatamanagement/definition/database), the $200 must be successfully deducted from the savings account and added to the checking account, and the funds in both accounts must be verified to ensure their accuracy. If any of these tasks fail, all changes roll back and none are committed.
* **Consistency.** A transaction must preserve the consistency of the underlying data. The transaction should make no changes that violate the rules or constraints placed on the data. For instance, a database that supports banking transactions might include a rule stating that a customer's account balance can never be a negative number. If a transaction attempts to withdraw more money from an account than what is available, the transaction will fail, and any changes made to the data will roll back.
* **Isolation.** A transaction is isolated from all other transactions. Transactions can run concurrently only if they don't interfere with each other. Returning to the transfer transaction example, if another transaction were to attempt to withdraw funds from the same savings account, isolation would prevent the second transaction from firing. Without isolation, it might be possible for the second transaction to withdraw more funds than are available in the account after the first transaction was completed.
* **Durability.** A transaction that is committed is guaranteed to remain committed -- that is, all changes are made permanent and will not be lost if an event such as a power failure should occur. This typically means persisting the changes to nonvolatile [storage](https://www.techtarget.com/searchstorage/definition/storage). If durability were not guaranteed, it would be possible for some or all changes to be lost, affecting the data's reliability.

Systems that support transaction processing employ various methods to ensure that each transaction is ACID-compliant. For example, SQL Server includes integrity constraints, such as primary key, foreign key, unique and check, to achieve data consistency. The system also supports isolation levels, such as read committed, read uncommitted and snapshot, to provide the necessary isolation for each transaction. These features are built into the database system so they can be incorporated easily into an application's backend database, although they need to be carefully implemented to ensure maximum ACID compliance.

Achieving ACID compliance for distributed transactions is more complicated than those based on a single location.

An example of how networks, servers and computers are structured in distributed computing.

One common approach to distribution is to use the [two-phase commit](https://www.techtarget.com/searchapparchitecture/definition/two-phase-commit-2PC) protocol, which coordinates each transaction across the participating [nodes](https://www.techtarget.com/searchnetworking/definition/node) and ensures that either all nodes commit to the transaction's completion or none do.