

ACID Properties

by Sophia



WHAT'S COVERED

This tutorial explores the ACID properties and how they affect database transactions in two parts:

1. Consistent State
2. ACID properties

1. Consistent State

As we covered in the prior tutorial, a transaction is a single unit of work that has to be fully executed or completely aborted if there are any issues within the transaction. There are no states in between the beginning and end that are acceptable for a database to be in. Recall our prior example, where James made a purchase for \$500 for a computer from a store. All of the SQL statements that we defined in that transaction must be executed entirely:

```
BEGIN;
```

```
UPDATE customer_account  
SET balance = balance - 500  
WHERE account_id = 1000;
```

```
UPDATE store_inventory  
SET quantity = quantity - 1  
WHERE store_id = 5 AND product_name = 'Computer';
```

```
INSERT INTO customer_order(account_id, product_name, store_id, quantity, cost)  
VALUES (1000, 'Computer', 5, 1, 500);
```

```
UPDATE store_account  
SET balance = balance + 500  
WHERE store_id = 5;
```

```
COMMIT;
```

It is not acceptable to only deduct James's account balance or remove inventory from the store. If any of the SQL statements in the transaction fails, the entire transaction is rolled back (i.e, not committed) to the original

state. If the transaction is successful, the changes to the database bring it from one consistent state to another. A consistent state is a state in which all of the data integrity constraints on the database are satisfied.

2. ACID properties

To ensure that we have consistency in the database, every transaction has to begin with the database in a known consistent state. If the database is not in a consistent state, transactions can result in a database that is inconsistent due to violations of integrity or business rules. As such, all of the transactions that occur in the database are controlled and executed to ensure database integrity according to the ACID properties: atomicity, consistency, isolation, and durability.

Atomicity requires that all SQL statements of a transaction be completed. The transaction should be viewed as a single logical unit of work that is indivisible. If any of the SQL statements are not completed, the entire transaction should be aborted. For example, in our transaction above, imagine that the first two statements executed:

```
UPDATE customer_account  
SET balance = balance - 500  
WHERE account_id = 1000;
```

```
UPDATE store_inventory  
SET quantity = quantity - 1  
WHERE store_id = 5 AND product_name = 'Computer';
```

Then imagine that we ran into an error with the data that stopped the transaction. The entire transaction should be reverted to its original state. However, if all four statements in the transaction executed successfully, the entire transaction would be committed to the database.

Consistency ensures that the database is in a consistent state. This means that a transaction takes a database from one consistent state to another. When a transaction starts, the database must be in a consistent state and when the transaction ends, the database must be in a consistent state. If any of parts of the transaction violate one of the integrity constraints, the entire transaction is aborted.

Isolation means that the data that is used during the first transaction cannot be used in another transaction until the first transaction has finished executing. In looking at the example above, Imagine that James and another customer purchased a computer at similar times. If James's transaction has started, the second customer cannot attempt to purchase the same computer until James's transaction is completed. Otherwise, they may have both tried to purchase the single available computer. This is especially important for multiuser databases where you will have many users accessing and updating the database at the same time.

Durability is the last ACID property. It ensures that when the transaction changes are finished and committed, they cannot be undone or removed even if there is a system failure.

We will get into each of these properties in further detail in the upcoming tutorials.



SUMMARY

The ACID properties for a transaction are atomicity, consistency, isolation and durability.

Source: Authored by Vincent Tran