Consistency

by Sophia

WHAT'S COVERED

This lesson explores the consistency property in a transaction and how it affects the database, in three parts. Specifically, this lesson will cover:

1. Ensuring Consistency

Consistency within the ACID properties focuses on ensuring that the data in the database moves from one valid state to another valid state. This ensures that any data that has been modified in the database is uncorrupted and correct at the end of the transaction.

The consistency property follows these criteria:

- If the transaction has been completed successfully, the changes will be applied to the database.
- If there was an error in the transaction, all of the changes should be reverted/rolled back automatically. This means that the database should restore the pre-transaction state.
- If there was a system failure or external issue while the transaction was executing, all of the changes that were made in the transaction up to that point should automatically be reverted/rolled back.

TERM TO KNOW

Consistency

The quality of being in the same state. In database terms, consistency refers to a transaction starting and ending in the same state.

2. Consistency Example

Let's look at our banking example again.

EXAMPLE Jennifer would like to make a $100 payment to Randall through an account transfer. This transaction is a balance transfer between two accounts at two different branches of the same bank; Jennifer's account is at branch 10, and Randall's account is at branch 50. Let us review the transaction:

1. $100 would be deducted from Jennifer’s account at branch 10.
2. Branch 10's balance would be decreased by $100.
3. Branch 50's account would be increased by $100.
4. Randall’s account would be increased by $100.

The consistency property ensures that the total amount of money is the same at the start and finish. One account and branch are decreased, and another account and branch are increased by the same amount, so the transactions balance one another. Let us look back at the transaction in SQL:

```
BEGIN;
UPDATE customer_account
SET balance = balance - 100
WHERE account_id = 10;
UPDATE branch_account
SET balance = balance - 100
WHERE branch_id = (SELECT branch_id FROM customer_account where account_id = 10);
UPDATE branch_account
SET balance = balance + 100
WHERE branch_id = (SELECT branch_id FROM customer_account where account_id = 50);
UPDATE customer_account
SET balance = balance +100
WHERE account_id = 50;
COMMIT;
```

Imagine that during the second UPDATE statement, the system had a failure, and when it recovered, the transaction had only partially executed. There would be an inconsistent state because the total balances would not match up. In this situation, the system would roll back those UPDATE statements to the consistent state before the transaction started.

If both Jennifer’s and Randall's account balances started at $1000, the end result should have the appropriate expected balances. Jennifer’s account balance should be $900, and Randall’s balance should be $1100. If the end values were not what was expected for any reason, the transaction would also be rolled back.

SUMMARY

In this lesson, you learned how consistency ensures that a database maintains a valid and expected state before and after a transaction. In the context of ACID, **ensuring consistency** requires all database operations to abide by predefined rules, constraints, and business logic. As a result, the database's states before the transaction and after it are consistent.

The **consistency example** showed a banking transaction that transferred money between two accounts. The total amount of money was the same before and after the transaction, reflecting consistency. If the transaction were to violate that constraint, it would be rolled back, with any changes undone, to maintain data integrity.

Source: THIS TUTORIAL WAS AUTHORED BY DR. VINCENT TRAN, PHD (2020) AND FAITHE WEMPEN (2024) FOR SOPHIA LEARNING. PLEASE SEE OUR **TERMS OF USE**.

TERMS TO KNOW

Consistency

The quality of being in the same state. In database terms, consistency refers to a transaction starting and ending in the same state.