Durability

by Sophia Tutorial

1. Introduction

Durability is the last ACID property, and one of the easiest to understand. This property focuses on ensuring that once the data from a transaction has been saved/committed to the database, it will stay in place and will not be affected by a system failure. This means that any completed transactions should be recorded and saved in memory.

2. Durability Example

Once again, consider our banking 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. Here's what the transaction looks like:

1. Jennifer's (10) account would be deducted by $100.
2. The banking location where Jennifer has her account would have their location's account deducted by $100.
3. The banking location where Randall(50) has his account would be increased by $100.
4. Randall's account would be increased by $100.

Let us look 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;

Once the transaction has successfully completed execution, the updates and changes to the database are stored in and written to the database. These changes will still persist even if the system fails, as those updates are now permanent. The effects of the account transfer will never be lost. Note that durability is only applied after the transaction has occurred and the COMMIT has been successfully executed. Anything that occurs prior to that is part of another ACID property.

☑ SUMMARY

The durability property ensures that once a transaction is completed and committed, the data is permanent and never lost, even with a system failure.

Source: Authored by Vincent Tran