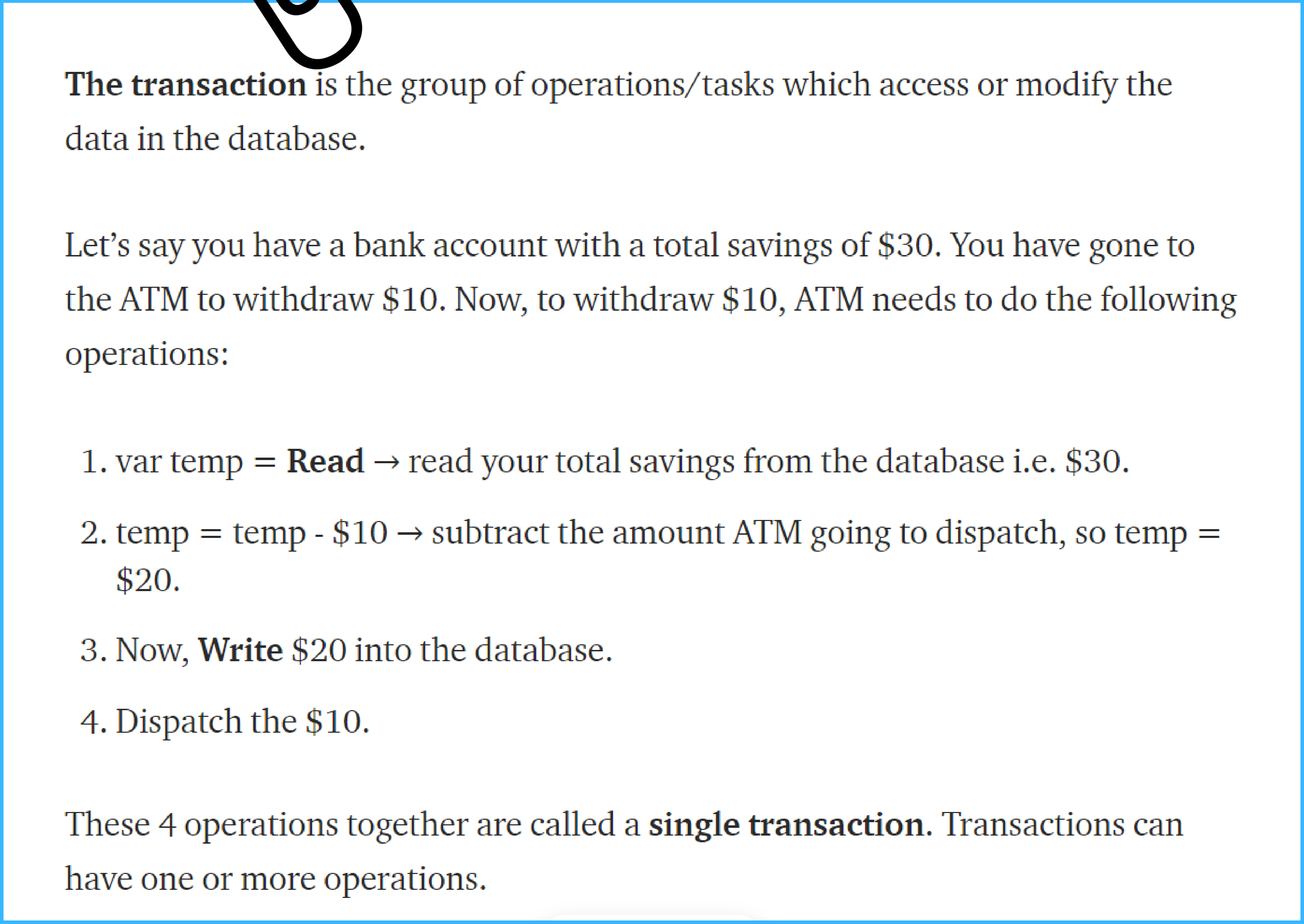
ACID Properties

(Database Management System)

By

Ashay Nayak,
Software
Developer



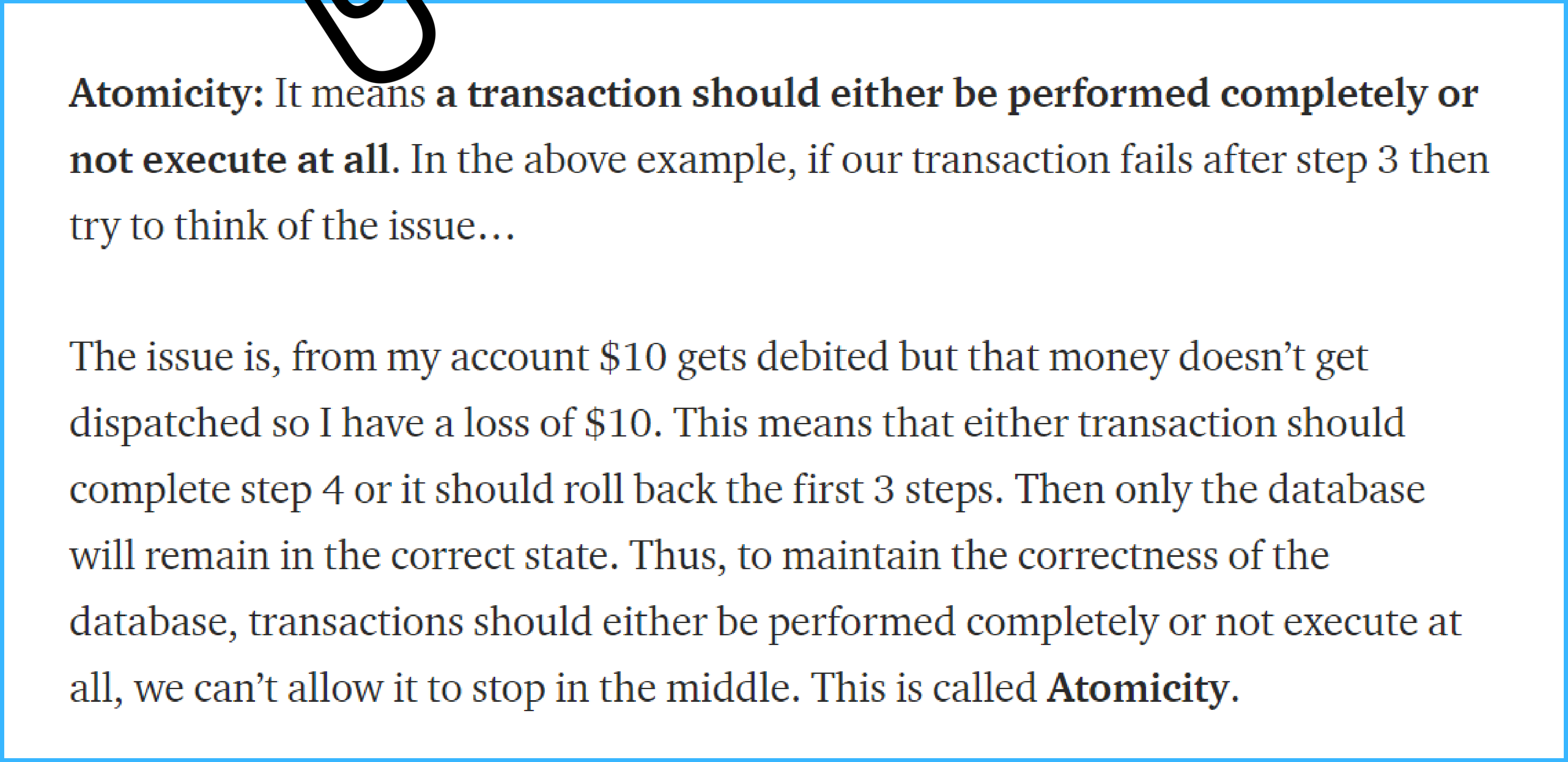
The transaction is the group of operations/tasks which access or modify the data in the database.

Let's say you have a bank account with a total savings of $30. You have gone to the ATM to withdraw $10. Now, to withdraw $10, ATM needs to do the following operations:

1. var temp = **Read** → read your total savings from the database i.e. $30.
2. temp = temp - $10 → subtract the amount ATM going to dispatch, so temp = $20.
3. Now, **Write** $20 into the database.
4. Dispatch the $10.

These 4 operations together are called a **single transaction.** Transactions can have one or more operations.

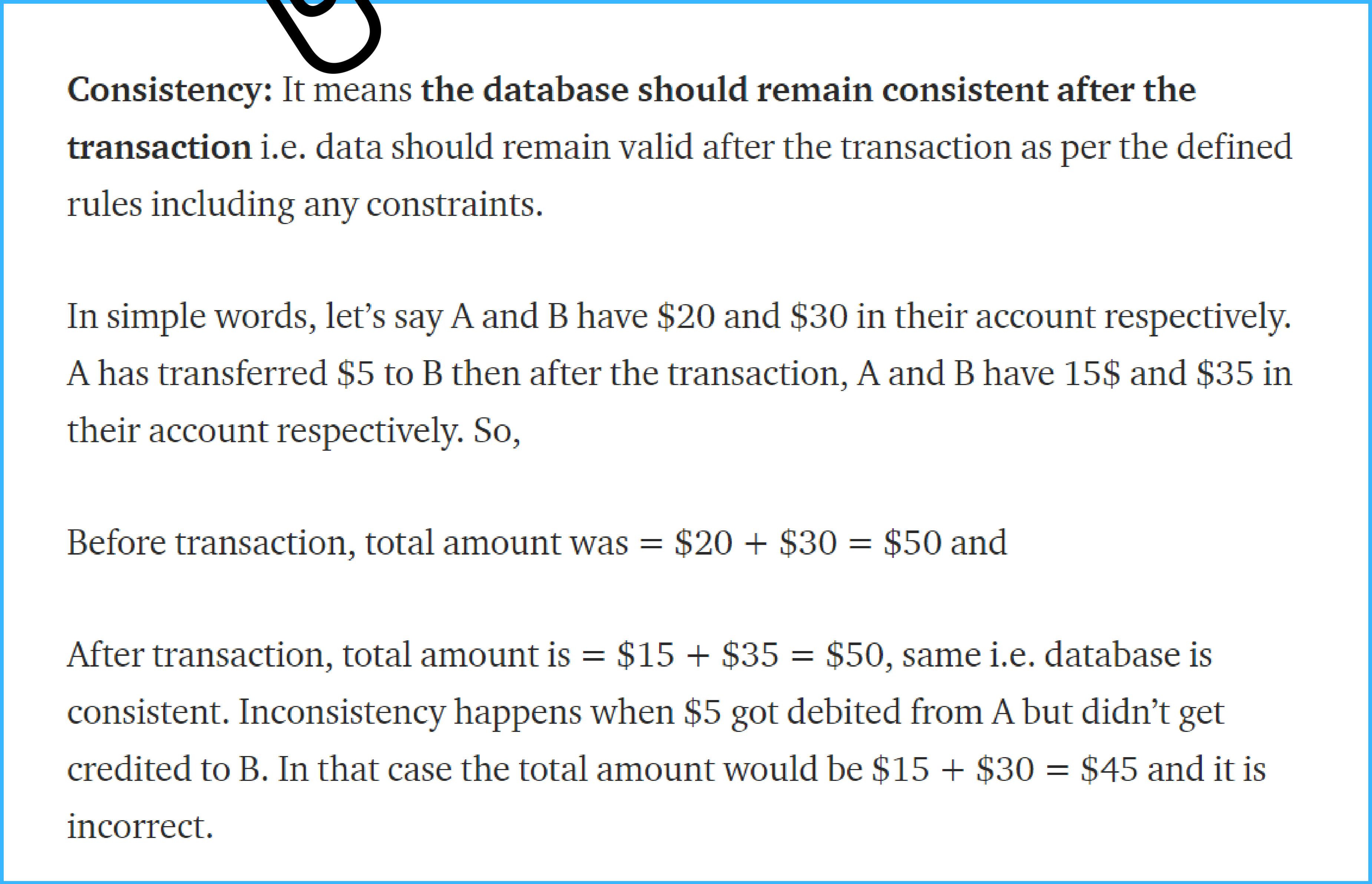
Source: My Medium Article
Made By: Ashay Nayak



Atomicity: It means **a transaction should either be performed completely or not execute at all**. In the above example, if our transaction fails after step 3 then try to think of the issue…

The issue is, from my account $10 gets debited but that money doesn't get dispatched so I have a loss of $10. This means that either transaction should complete step 4 or it should roll back the first 3 steps. Then only the database will remain in the correct state. Thus, to maintain the correctness of the database, transactions should either be performed completely or not execute at all, we can't allow it to stop in the middle. This is called **Atomicity**.

Source: My Medium Article
Made By: Ashay Nayak



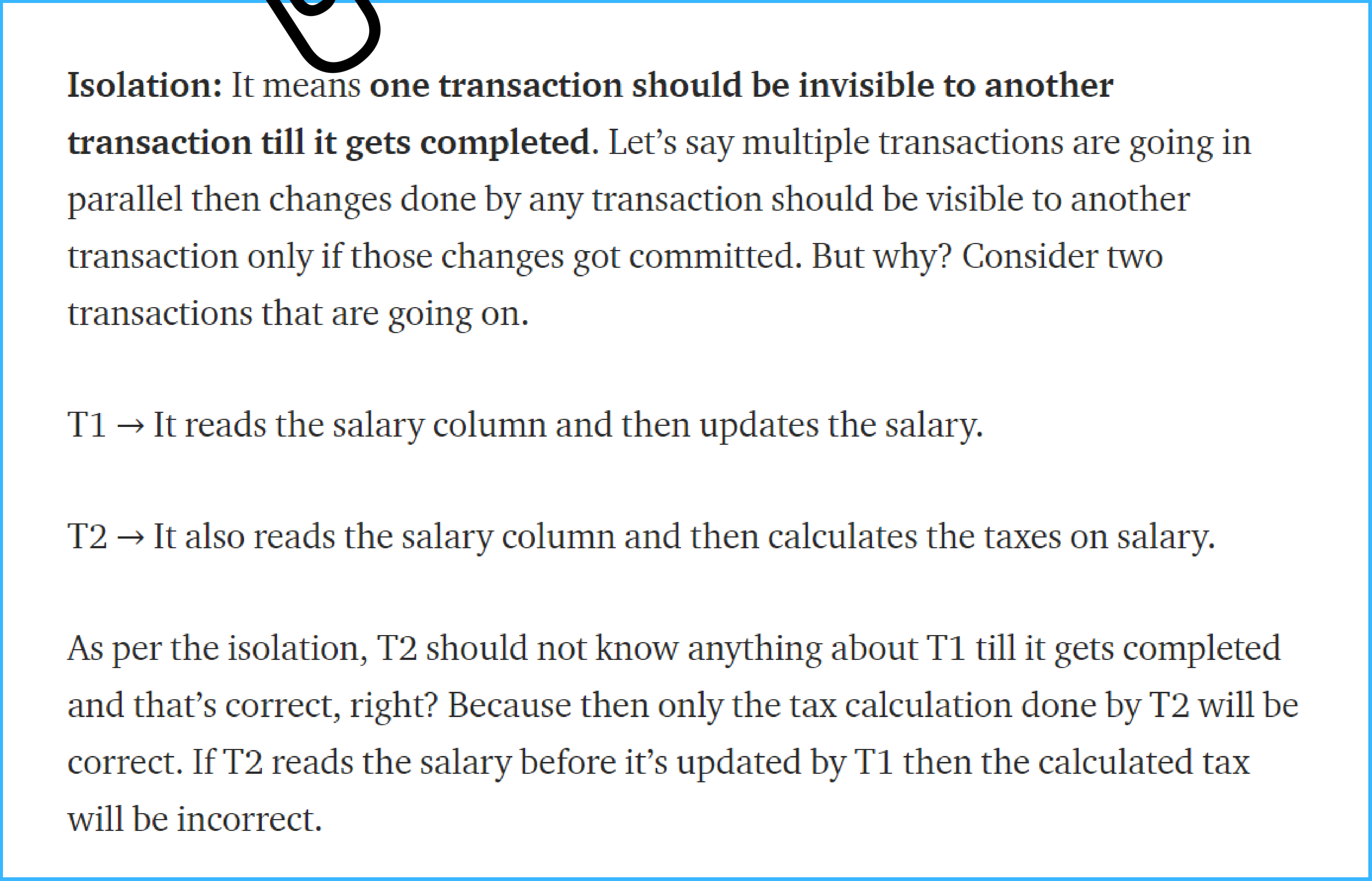
Consistency: It means **the database should remain consistent after the transaction** i.e. data should remain valid after the transaction as per the defined rules including any constraints.

In simple words, let’s say A and B have $20 and $30 in their account respectively. A has transferred $5 to B then after the transaction, A and B have 15$ and $35 in their account respectively. So,

Before transaction, total amount was = $20 + $30 = $50 and

After transaction, total amount is = $15 + $35 = $50, same i.e. database is consistent. Inconsistency happens when $5 got debited from A but didn’t get credited to B. In that case the total amount would be $15 + $30 = $45 and it is incorrect.

Source: My Medium Article
Made By: Ashay Nayak



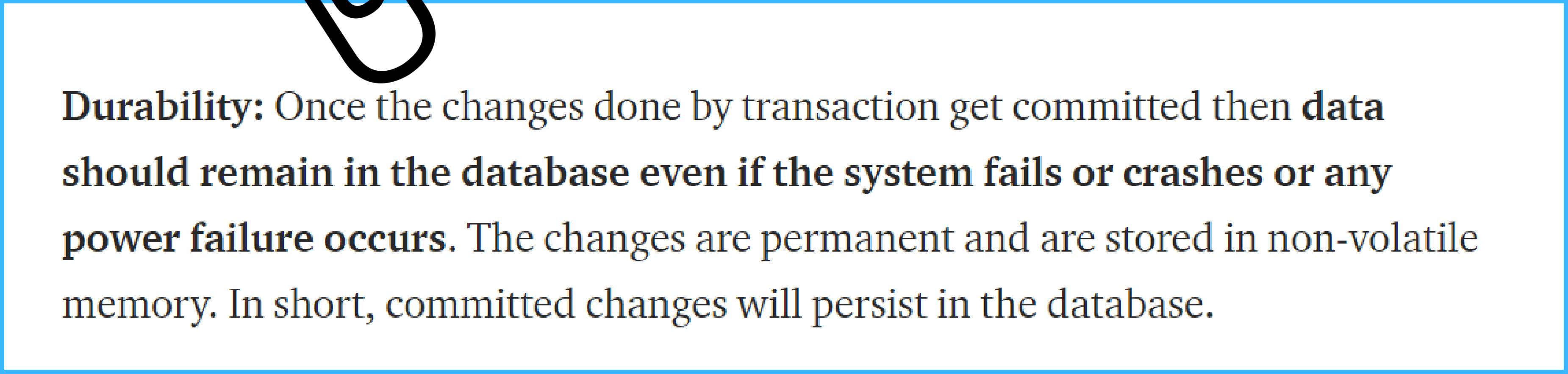
Isolation: It means **one transaction should be invisible to another transaction till it gets completed**. Let's say multiple transactions are going in parallel then changes done by any transaction should be visible to another transaction only if those changes got committed. But why? Consider two transactions that are going on.

T1 → It reads the salary column and then updates the salary.

T2 → It also reads the salary column and then calculates the taxes on salary.

As per the isolation, T2 should not know anything about T1 till it gets completed and that's correct, right? Because then only the tax calculation done by T2 will be correct. If T2 reads the salary before it's updated by T1 then the calculated tax will be incorrect.

Source: My Medium Article
Made By: Ashay Nayak



Durability: Once the changes done by transaction get committed then **data should remain in the database even if the system fails or crashes or any power failure occurs.** The changes are permanent and are stored in non-volatile memory. In short, committed changes will persist in the database.

Source: My Medium Article
Made By: Ashay Nayak

Checkout Interview Oriented Articles on DBMS

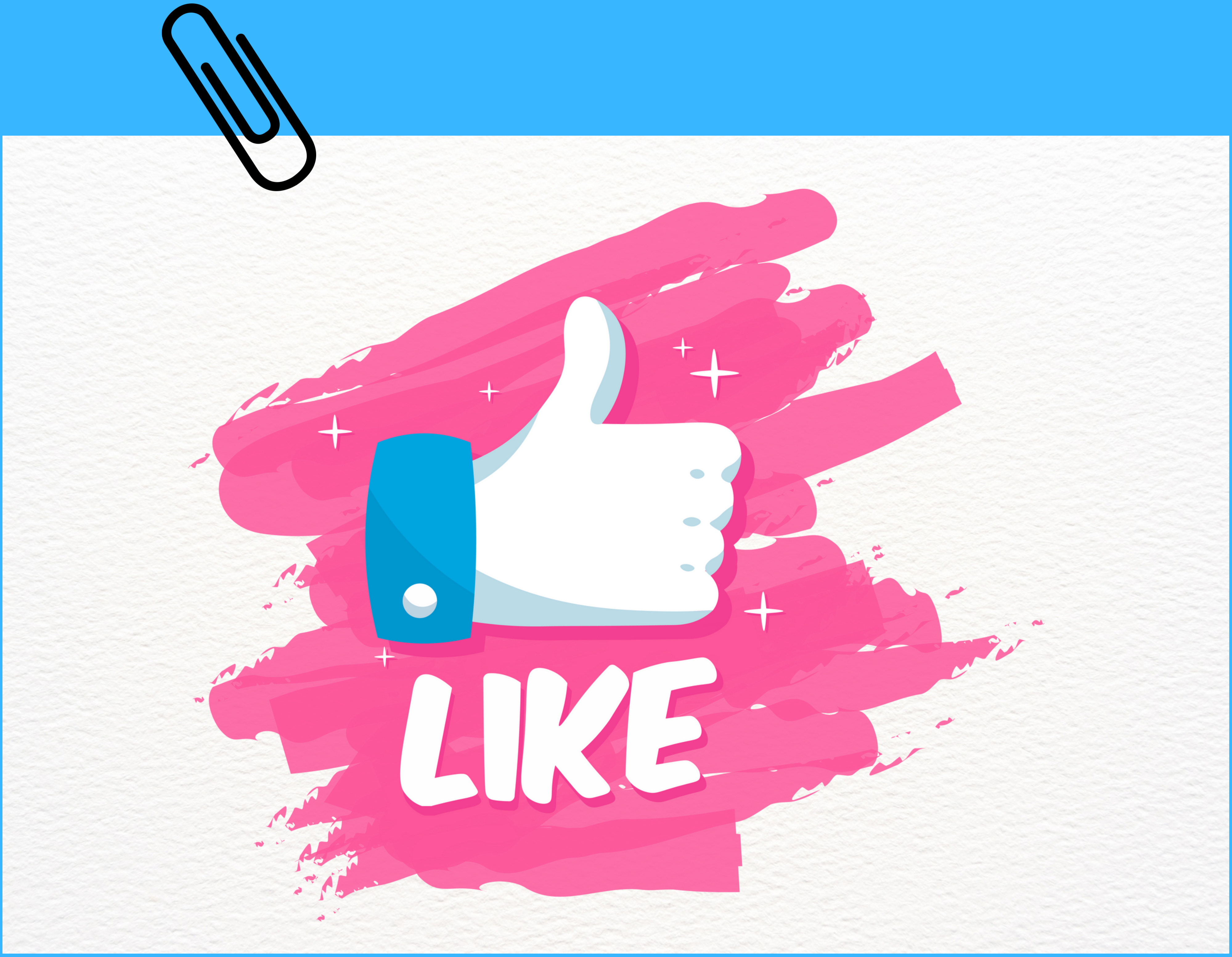
1.)ACID Properties in DBMS - Interview Question

2.) Indexing in DBMS - Everything for Interview Preparation

3.) Normal Forms in Database Management System

Links in the comment

Made By: Ashay Nayak



Made By: Ashay Nayak