

# Transactions

A transaction is a **single unit of work** that has to be done sequentially one after the other.

The Transaction is said to be completed if any only if all the tasks are completed.

## Single Unit of Work



Open Lid

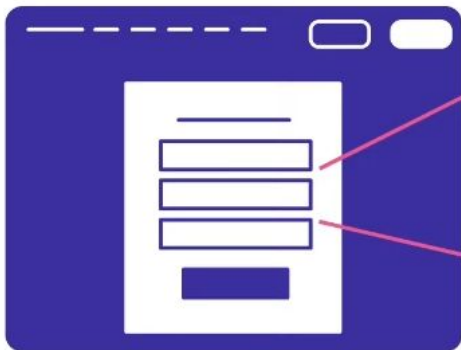


Fill Fuel



Close Lid

## Studentbook



user table

email	password

INSERT

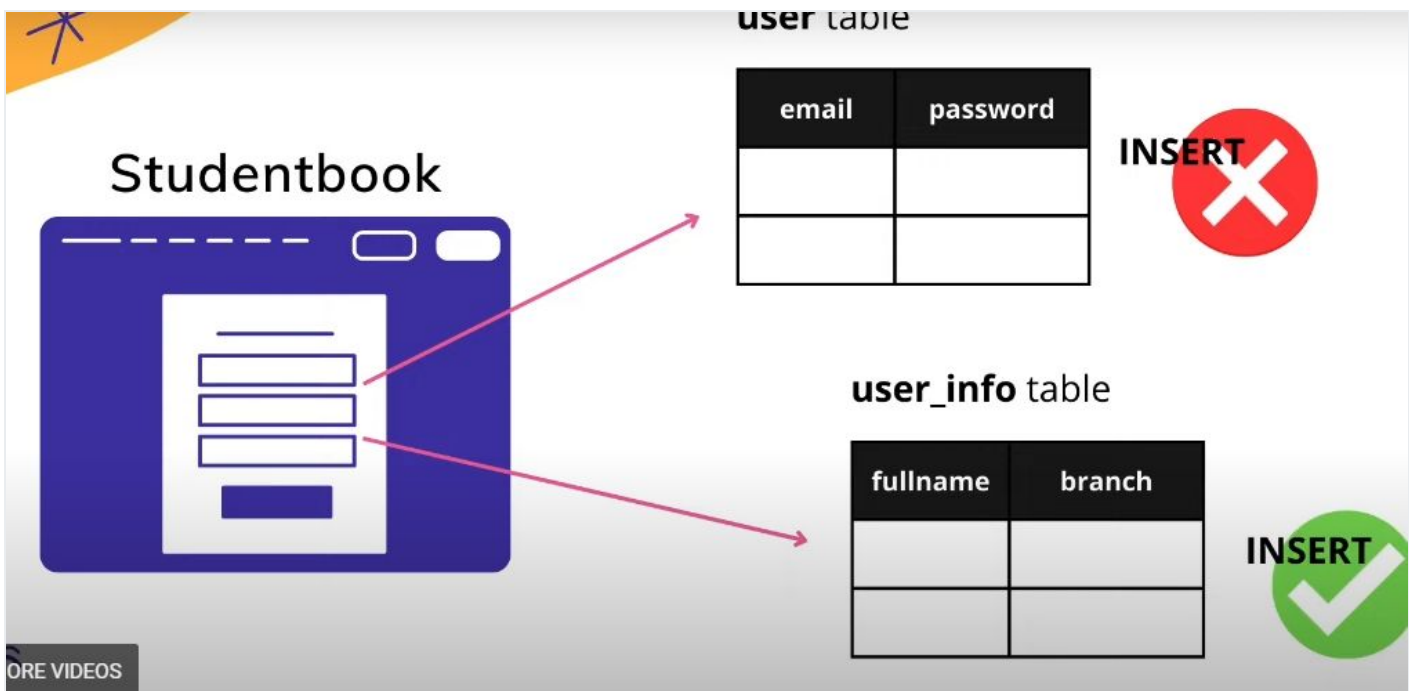


user\_info table

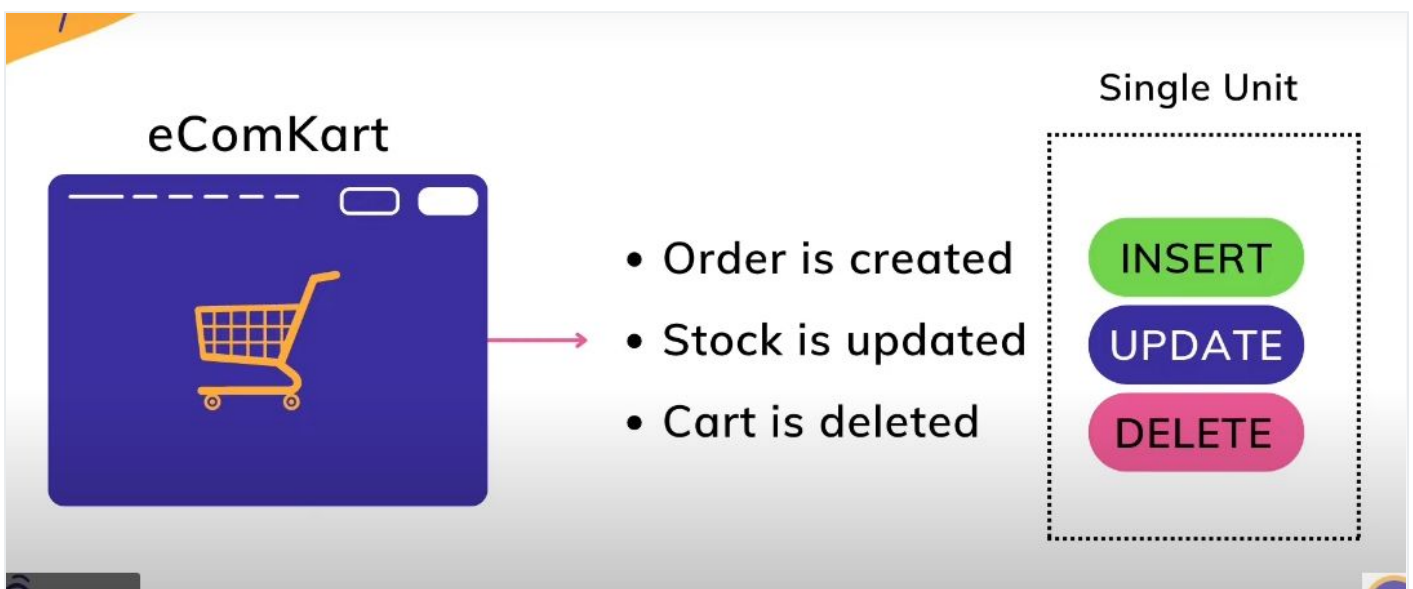
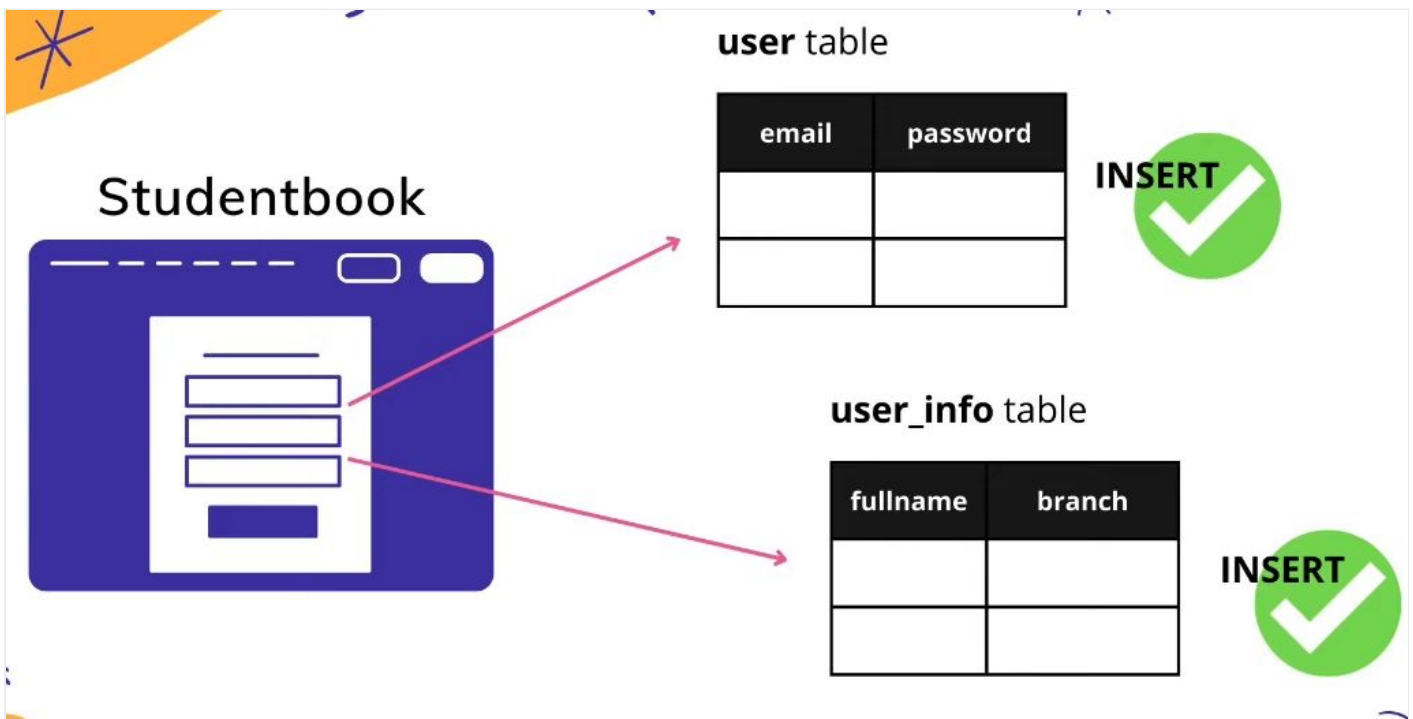
fullname	branch

INSERT





the above 2 examples make the database in-consistent



## Bank Transaction Example



### 3 step process:

- Check if A has Rs. 2000 to transfer?
- Then deduct Rs. 2000 from A's account
- And add Rs. 2000 to B's account.

To enable this successful behavior in the database we need to use transactions.

## Database Transaction or Tx

When in a Database, we have to run multiple SQL queries, that affects the state of data in our database through INSERT, UPDATE or DELETE queries. We must combine them into a single unit of work, so that all of them pass together, or fail together.

VIDEOS

## Database Transaction or Tx

In case even a single query fails, then the whole bundle of query executed in a transaction, have to fail.

The transaction **should** follow **ACID** properties.

## Atomicity

- This is derived from "Atomic", that means 'One' or 'Single'.
- The tasks (SQL queries) executed inside a transactions, should act as a Single Unit of Work.
- Partial success and Partial failure is not allowed.
- In case of Failure, any changes done must be rolledback.

## Consistency

- Just like any other SQL query, a transaction must also follow the Database table constraints.
- If some constraint is failing due to the SQL queries running in the transaction, then the transaction should be failed.

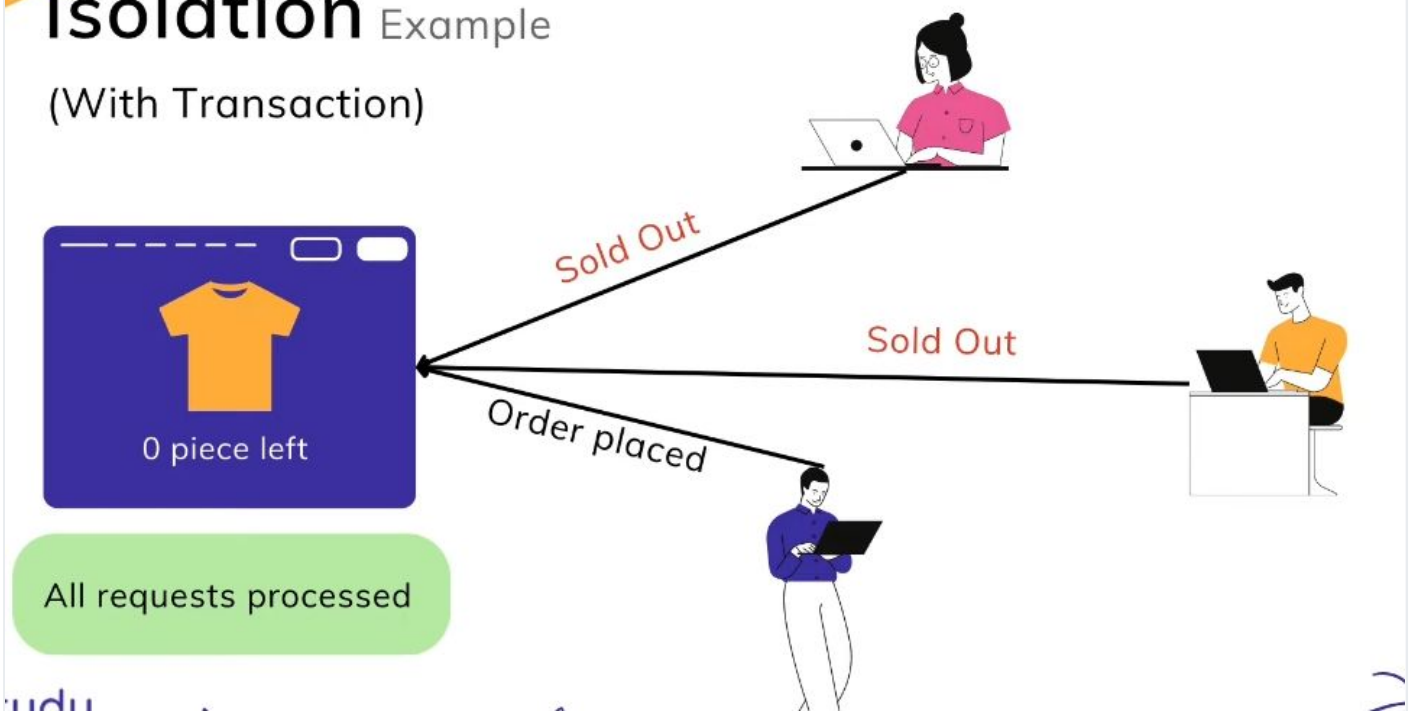
## Consistency

- Just like any other SQL query, a transaction must also follow the Database table constraints.

Isolations: When more than one process acts on the same data then it should be allowed one by one .

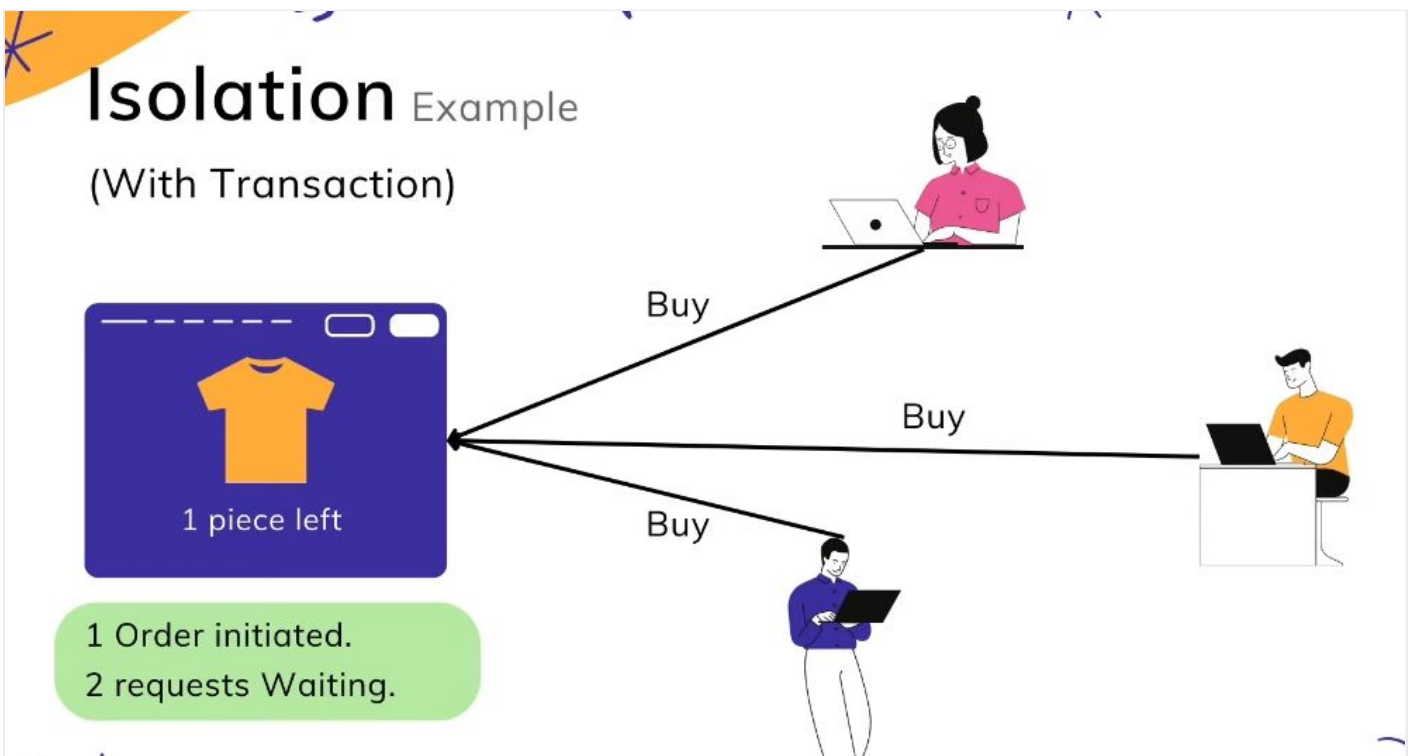
# Isolation Example

(With Transaction)



# Isolation Example

(With Transaction)



# Isolation

- SQL Transactions are executed in isolation.
- Because SQL Transactions are executed in isolation, hence it makes multiple Transaction processing slow.
- No Parallel execution for DB Transaction.



# Durability

- Changes made by SQL queries in DB Transaction should be permanent.

If the transaction does not follow the above properties then the data in the database leads to in-consistency.

point any failure happens it gets rollbacked (all the changes being done are undone.)

## 2. ACID Properties

1. To ensure integrity of the data, we require that the DB system maintain the following properties of the transaction.

### 2. Atomicity

1. Either all operations of transaction are reflected properly in the DB, or none are.

### 3. Consistency

1. Integrity constraints must be maintained before and after transaction.
2. DB must be consistent after transaction happens.

### 4. Isolation

1. Even though multiple transactions may execute concurrently, the system guarantees that, for every pair of transactions  $T_i$  and  $T_j$ , it appears to  $T_i$  that either  $T_j$  finished execution before  $T_i$  started, or  $T_j$  started execution after  $T_i$  finished. Thus, each transaction is unaware of other transactions executing concurrently in the system.
2. Multiple transactions can happen in the system in isolation, without interfering each other.

### 5. Durability

1. After transaction completes successfully, the changes it has made to the database persist, even if there are system failures.

## 3. Transaction states