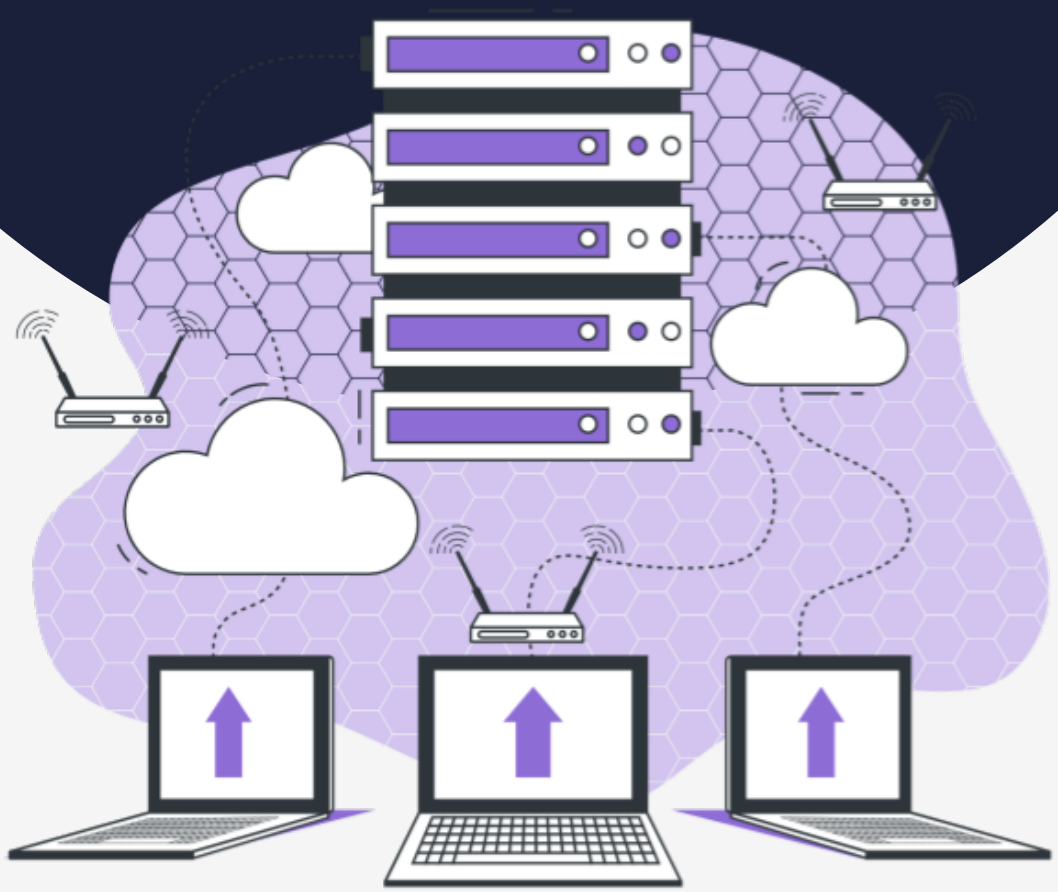
Lesson:

Transactions



PW SKILLS

Lecture Checklist

1. Introduction to Database Transactions.
2. ACID Properties.

In recent days, there has been an increase in the availability and popularity of different databases due to their ability to manage and organize large amounts of data effectively. Despite the emergence of newer technologies like MongoDB and Cassandra, which offer different approaches to data storage and processing, RDBMS systems still have a significant advantage in their support for transactions.

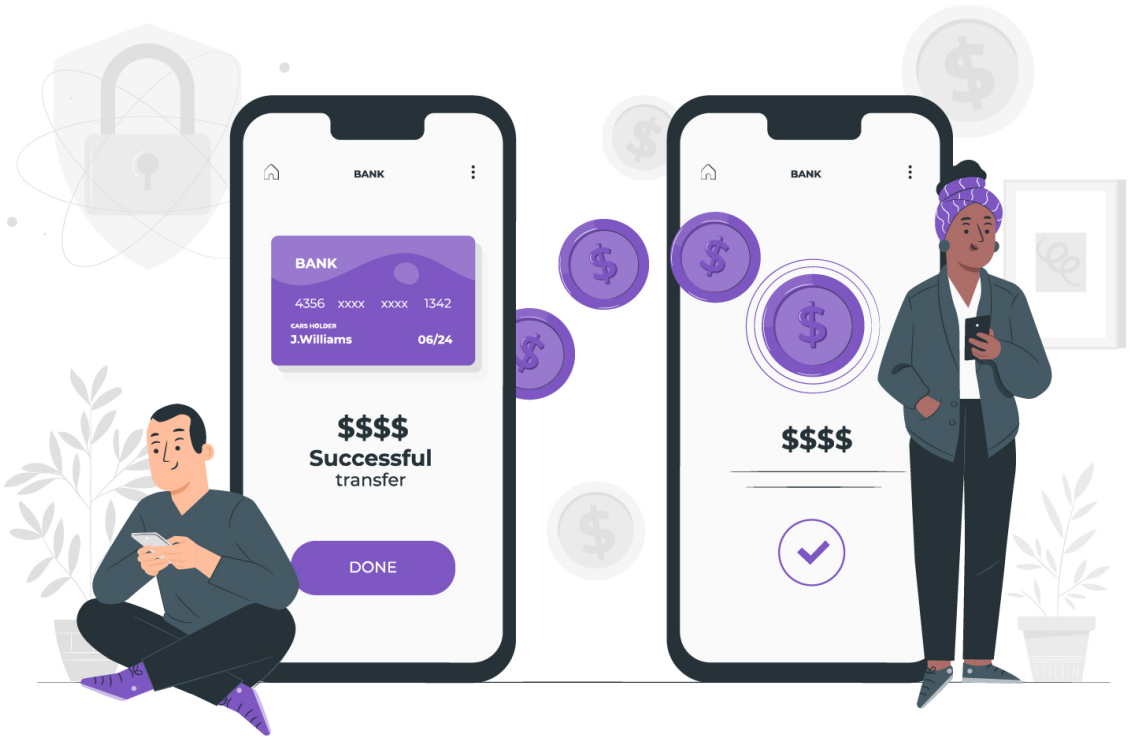
While newer NoSQL databases like MongoDB and Cassandra may offer advantages in terms of scalability, performance, and flexibility, they may lack the robust support for transactions that RDBMS systems provide. This makes RDBMS systems stand out with their USP of transactional support, which continues to be a critical factor for many businesses and applications that require reliable data management.

In this lecture, we will be looking at what transactions are and what determines if a database is transactional or not.

Introduction to Transactions.

Transactions mean either all or nothing.

To understand this let’s have a look at an example. Let's consider a scenario where a student who is extremely happy with Vishwa sir’s work wants to transfer 5 lakh rupees to Vishwa sir, who is known for being an excellent teacher.



This transaction can be broken down into three stages:

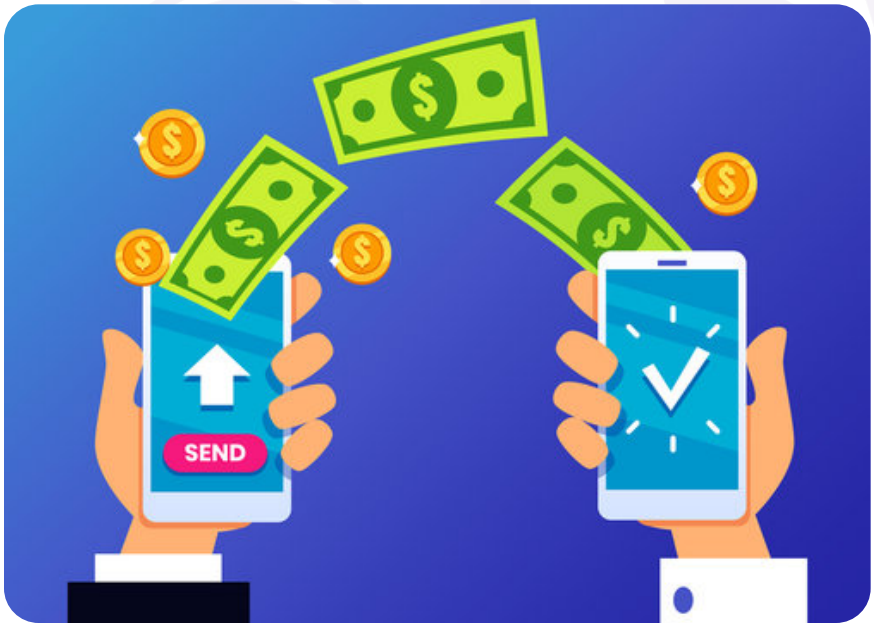
1. Deduction of money from the student's account.
2. Wire transfer of the money.
3. Crediting the money to Vishwa sir's account.

Let’s look into these steps one by one:

In the first stage, the student initiates the transfer by authorizing the deduction of 5 lakh rupees from their own account. This deduction is recorded as a part of the transaction, but the money is not immediately transferred to Vishwa sir's account.

In the next stage, the deducted money is transferred from the student's account to Vishwa sir's account via a wire transfer. This involves multiple steps, such as verifying the account details, checking for sufficient funds, and updating the transaction status. Once the wire transfer is successfully completed, the transaction progresses to the next stage.

In the last stage, the money is credited to Vishwa sir's account, and the transaction is considered successfully completed. However, if this stage fails due to any errors, such as a technical glitch or network issue, the money won't be credited to Vishwa sir's account.



If the third stage fails, the entire transaction will be rolled back, meaning that any changes made in the previous stages, including the deduction of money from the student's account, will be undone, and the money won't be lost. This rollback mechanism ensures that the transaction is either completed entirely, with all stages successfully executed, or not executed at all.

This is why we define transactions as either all or none.

ACID Properties.

For a database to support transactions, it must possess the ACID properties. ACID stands for Atomicity, Consistency, Isolation, and Durability. Transactions must be atomic, meaning they are either fully completed or fully rolled back.

Let's now look at the ACID properties one by one.

Atomicity.

The concept of atomicity in transactions is inspired by the idea of an atom, which is the smallest particle in an element and cannot be further broken down. Similarly, in the context of transactions, atomicity means that a transaction is indivisible and cannot be separated into individual steps. All the steps within a transaction must either be successfully completed or all must fail, ensuring that the transaction is either fully executed or fully rolled back.

Consistent.

Consistency in transactions refers to the state of the system before and after the transaction, which should remain consistent.

To understand this, let’s look at an example. Let's consider ICICI Bank, the bank with multiple customers or account holders with random account balances. Let’s assume that customer A and B plan to do a transaction of 100 rupees. Before the transaction, A has 5000 rupees and B has 4000 rupees. After the transaction is executed, A's balance is reduced to 4900 rupees, and B's balance is increased to 4100 rupees. This ensures that the overall balance of A and B remains consistent, while the account balances of other customers remain unchanged.

Isolation.

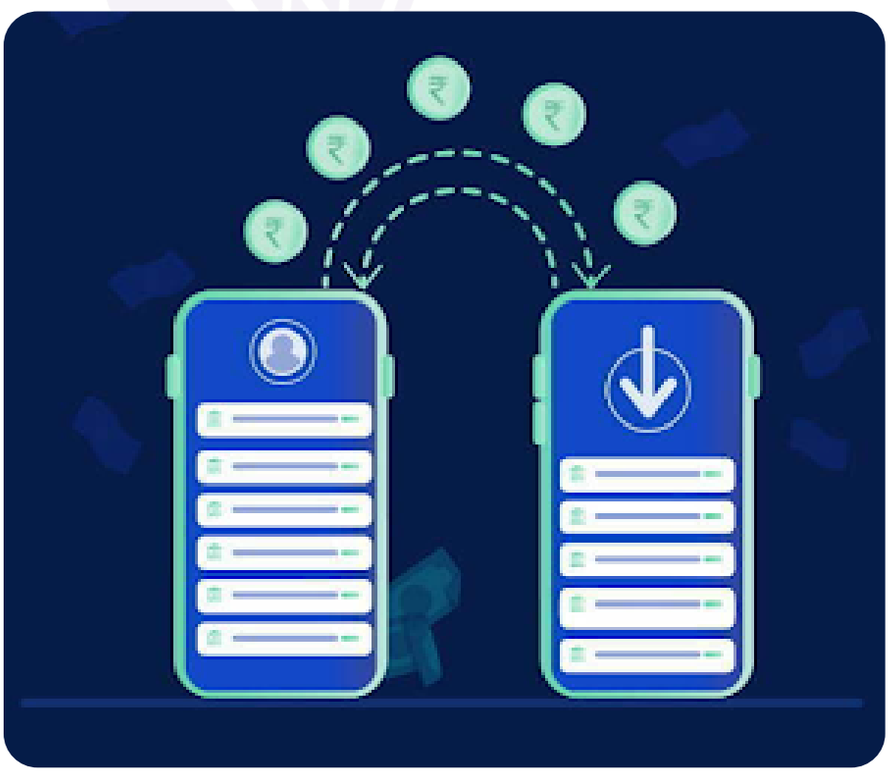
Isolation in transactions refers to the concept that each transaction is independent of another transaction.

For example, when making payments through a QR code at a shop using a payment app like GPay, the shop owner bever asks customers to do payments one after the other. Customers can make payments concurrently. Even though multiple customers are making payments at the same time, their transactions are kept separate and do not affect each other. This ensures that each transaction is processed and completed without any interference from other transactions. This is called transaction isolation.

Durability.

The concept of durability in transactions means that once a transaction is completed, it is considered final and cannot be undone.

For example, once a customer completes a payment transaction using a QR code at a shop, the transaction information is saved by the system and considered final. Even if there is a system failure or restart, the information about the completed transaction will persist and not be lost. This ensures that the effects of the transaction, such as the deduction of the payment amount from the customer's account and the addition of the payment amount to the shop owner's account, are permanent and reliable. Customers and shop owners can have confidence that the completed transactions are durable and will not be undone, providing a sense of trust and integrity in the transactional system.



If a database fully implements these ACID properties, it can be considered transactional, as it provides reliable, consistent, and durable processing of transactions, ensuring the integrity and reliability of the data stored in the database.