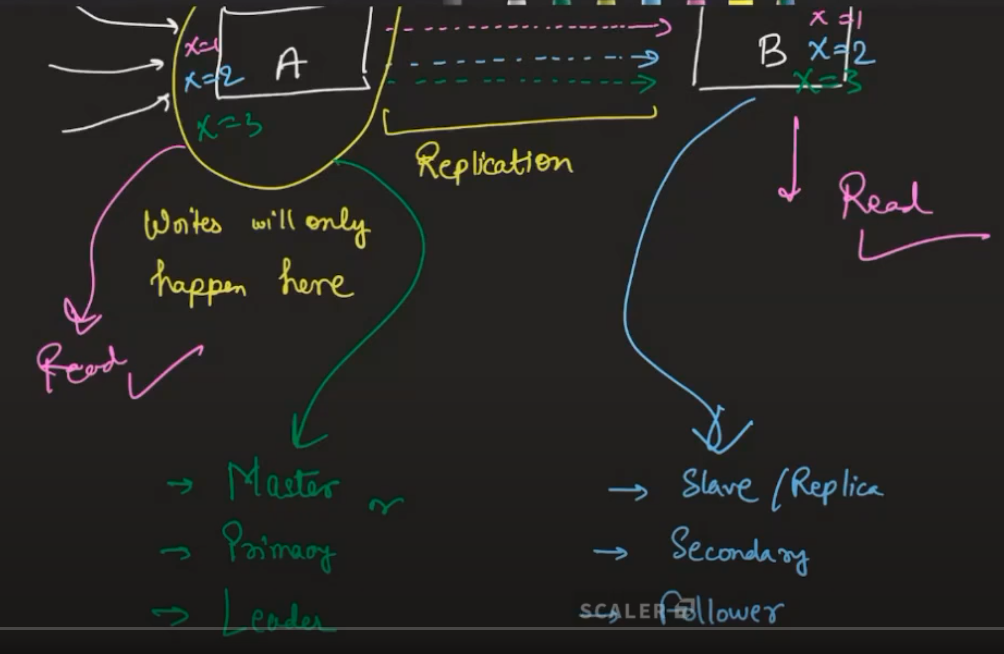
**Part 5-**

**Agenda**

1. Replication
2. Master Slave Architecture



A is master and all the writes are happening on this server and then propagate or copy the written data from A to B server and this is called Replication.

Write action is only done to A server.

Read action can be done from both A and B servers.

Main server can be called Master, Primary, Leader

Replication server can be called Slave/Replca, Secondary, Follower

There can be a little bit of delay in reading just-written-data from Slave server as it might take some time to replicate the same Master to Slave server. This is called Replication Lag.

Example- E-Commerce (Order Management System - OMS)

New Order details should be written to the Master database and then it should propagate the respective data to the Slave server.

Then, other functionalities/services like Notification, Invoice Generation, Analytics etc. should read the data from the Slave server.

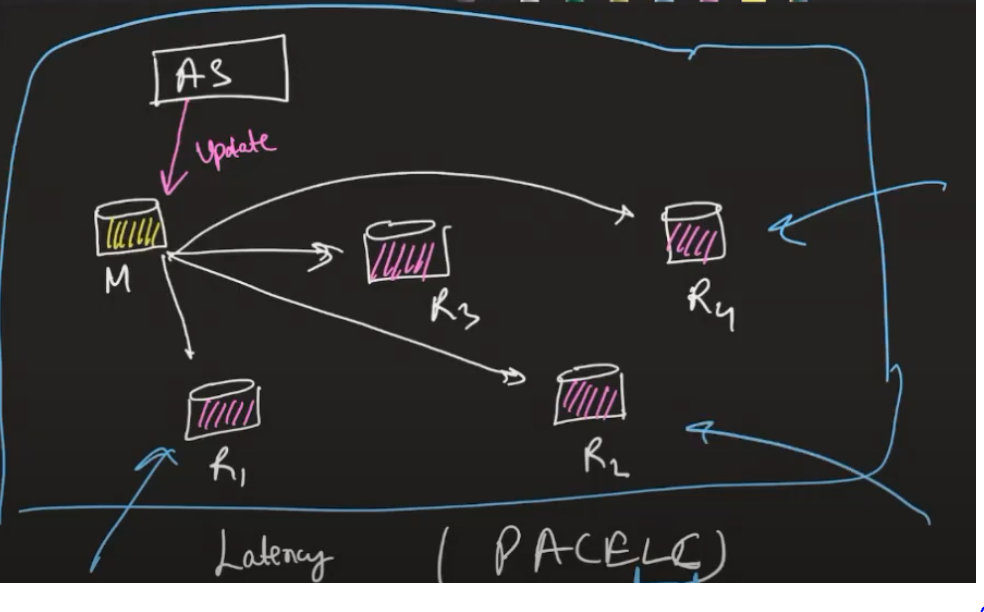
Benefits of Master-Slave Architecture-

1. Reads are highly available OR Faster reads from Slave
2. Switch from Master to Slave in case of Master failure

**Advantages of Replications -**

1. Fault Tolerance
2. Reducing read latency
3. Secondary reads(Notifications, Analytics) can happen at a faster rate

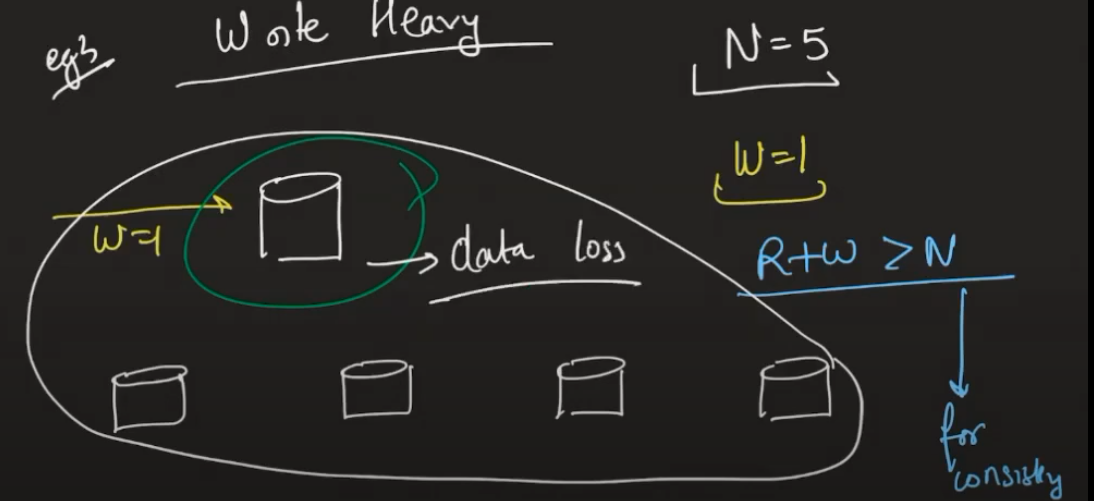
**Bank Applications**

****

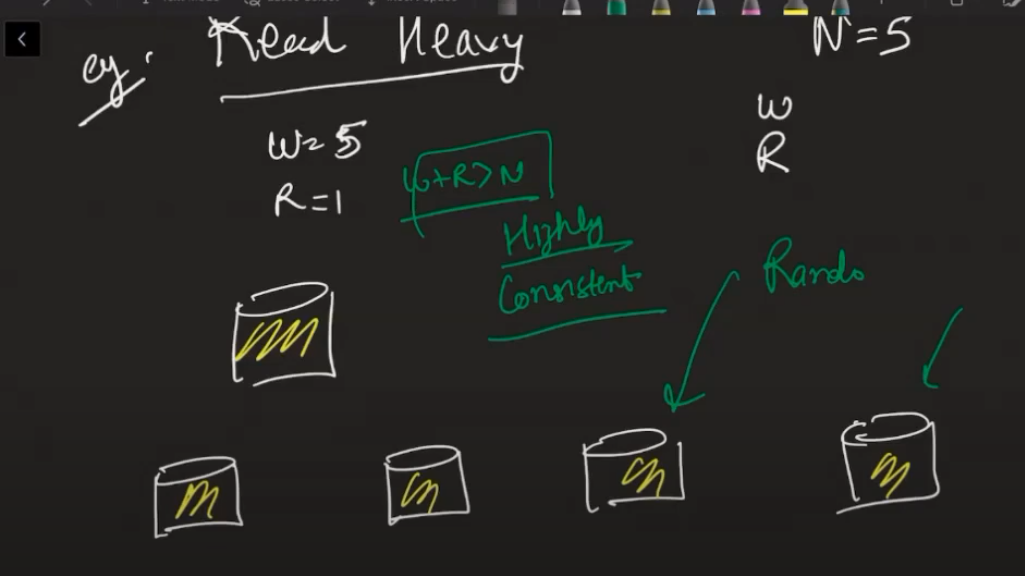
In banking applications, Application server writes to one Master server and then propagates to other slaves at same time to maintain consistency across all the databases.

The banking applications are examples of Highly Consistent Systems.

**Write Heavy Systems -**

****

**Read Heavy Systems-**

****

**Whenever W** (Write) **+ R** (Read) **> N**(number of servers), then it is highly consistent system

1.17

