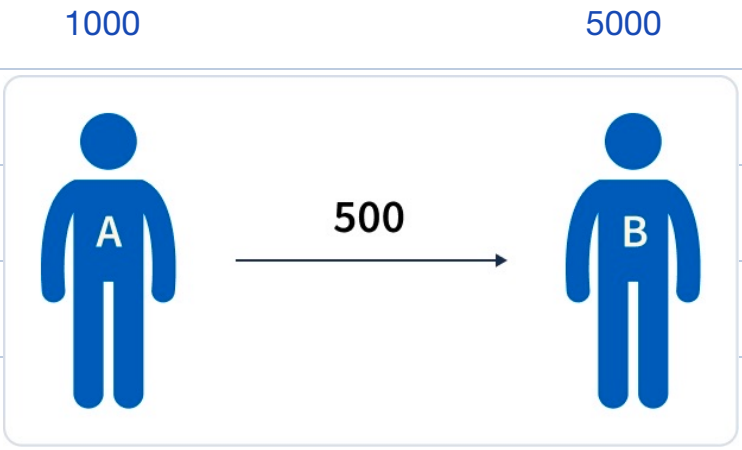
Transactions - 1

TABLE OF CONTENTS

26th Hard day challenge :

1. Assignments + Revision
2. Backlog (Assignments of prev. session)
3. Additional Questions

Transactions



id	name	amount
1	A	1000
2	B	5000

Steps

1. Get balance of A. DB Call
2. Check, if A >= amount.
3. Debit ₹ 500 from A & update table DB Call
4. Update the balance of B. DB Call

```
def bank_transfer(from, to, amount){
    # DB Calls & logical ops
}
```

Initial Amounts :

A = 1000

B = 5000

C = 15000

A $\xrightarrow{500}$ B (T1)

C $\xrightarrow{10000}$ B (T2)

```
bank_transfer(A, B, 500) {
→ Read A → X (1000)
→ if (X ≥ amount):
    write A ← X - 500
      (500)
→ Read B → X (5000)
→ write B ← X + 500
      (5500)
}
```

```
bank_transfer(C, B, 10000) {
→ Read C → X (15000)
→ if (X ≥ 10000):
    C ← X - 10000
    (5000)
→ Read B → X (5000)
→ write B ← 5000 + 10000
      (15000)
}
```

Initial

A = 1000

B = 5000

C = 15000

Total = 21,000

Final

A = 500

B = 15000

C = 5000

Total = 20,500

We lost ₹ 500 during Transaction.

In case of system failure, we might end up in an intermediate state.

→ Deducted from A.

→ But, not received by B.

* **Issues :**

1. Illogical / Incorrect / Inconsistent.
2. Might end up in intermediate state.

* A set of DB operations logically grouped together to perform a task.

→ Transactions

ACID Properties

Expected properties from a transaction

Atomicity

Consistency

Isolation

Durability

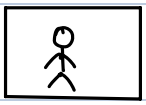
1. Atomicity

A single whole unit.

a) To an outsider, it should feel that either the transaction has happened or hasn't happened at all.

b) Either money is transferred or not transferred.

c) It takes care of intermediate state.



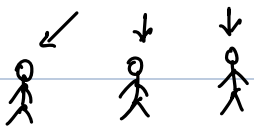
Gmail, YT .....

2. Consistency

- Correct
- Accurate
- Exact
- Logical

1. Numbers before & after a transaction should add up.
2. In our bank transaction, this property was violated.

Jio Cinema :



4.5 cr = 45000000

= 45000001 x

= 45000003 ✓

3. Isolation

Our transaction should not effect other transactions running at same.

4 levels of isolation :

1. Read Uncommitted ✓
2. Read Committed
3. Repeatable Read (Default in MySQL)
4. Serializable

4. Durability

Once a transaction is done, data should persist forever.

---

* **Commit :**

→ It saves the data once transaction is completed.

→ It ensures property of durability.

→ In MySQL, we have autocommit feature by default.

---

* **Rollback :**

→ Rollback undo the changes till last commit.

Isolation Levels

Isolation have different levels (4)

1. **Read Uncommitted** today
2. **Read Committed**
3. **Repeatable Read**
4. **Serialize**

Next session

1. Read Uncommitted

- We can read latest data including uncommitted data as well
- It will lead us to Dirty Read.