

---

---

---

---

---



## SQL (Global term)

### What is a Database? (Global term)

Central system where data will be stored in some logical related manner.

MySQL      PostgreSQL      Oracle

Software Applications

(DBMS)

- ↳ helps to store the data
- ↳ Manipulate the data

### Types of Databases

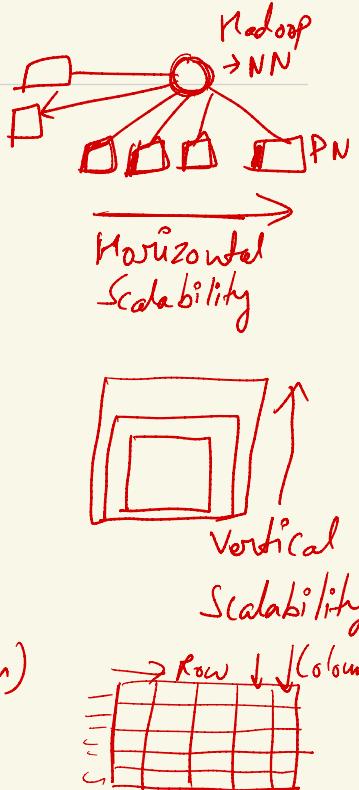
→ Transactional Databases (RDBMS)

→ NoSQL Databases

↳ MongoDB  
↳ Cassandra  
↳ HBase

↳ MySQL  
↳ PostgreSQL  
↳ Oracle

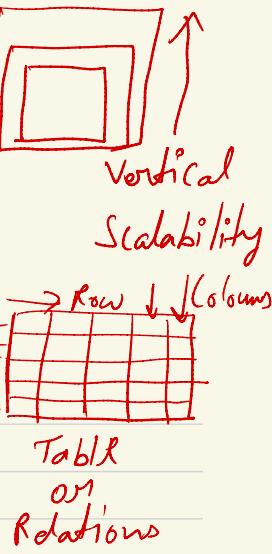
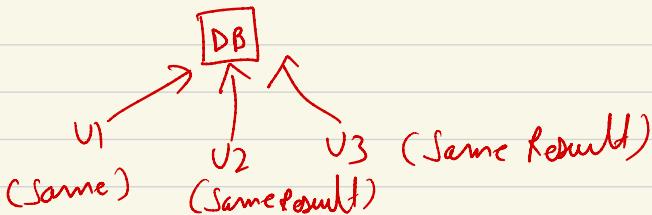
## ↳ CouchDB



## Transactional Databases

- Vertical Scalable
- stores data in Row based form
- Consistent
  - Data will be stored in structured form like Row-Column (known as Table or Relation)
- Highly available

## Consistency



## NoSQL database

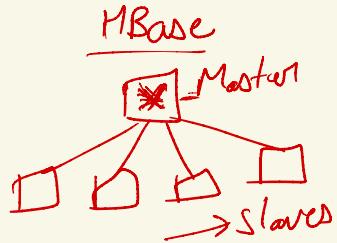
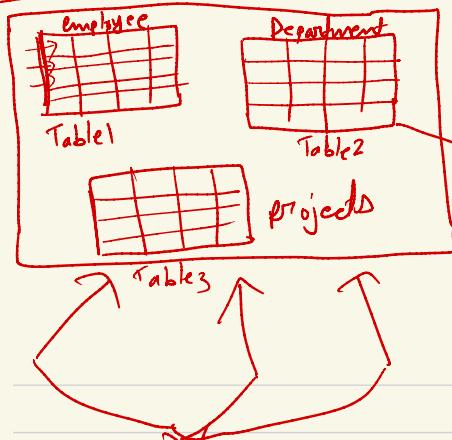
- Horizontal Scalability
- Distributed
- stores data in different form
  - Key-Value
  - Columnar
  - Graph

↳ Document

↳ Not always Consistent

↳ Highly available if no Master-Slave Architecture

Database

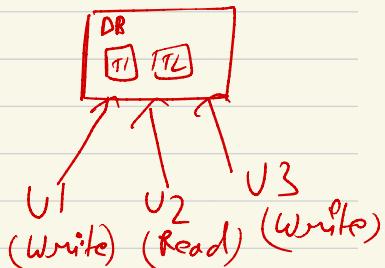


SQL (Structured Query language)

RDBMS will help us  
↳ MySQL to maintain it  
↳ Oracle

CRUD operations

- ↳ Insert
- ↳ Update } write
- ↳ Delete
- ↳ Select ( Read )

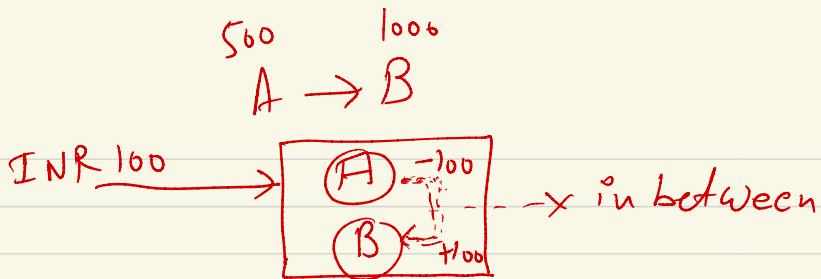


## Transaction

→ Smallest unit of execution will be known as transaction

Transactional Databases follows ACID properties.

- A → Atomicity ( each operation will be atomic )  
i.e; it will be a complete transaction)
- C → Consistency ( After success or failure data should be consistent )
- I → Isolation
- D → Durability



Inconsistent       $-100$       No update  
on A side      on B side

on failure → Rollback will happen

on success → Commit will happen

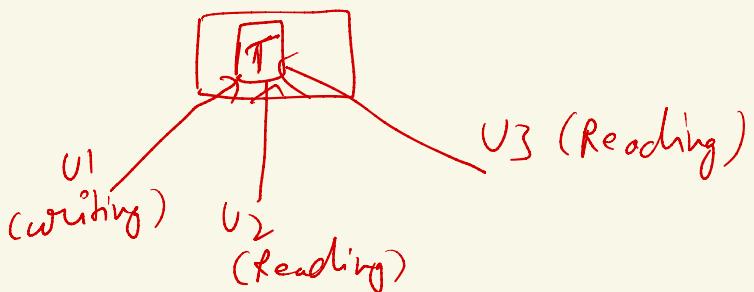
Rollback → Take our DB to previous state ( $A \rightarrow 500, B \rightarrow 1000$ )

Commit → Save the changes w/ Next Consistent State

( A → 400, B → 1100 )

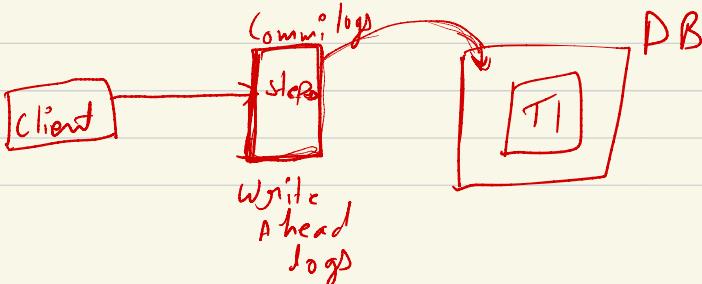
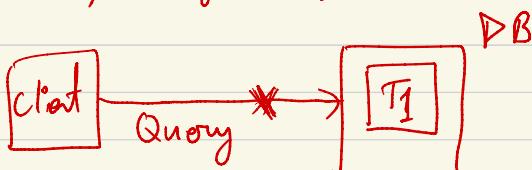
## Isolation

↳ each transaction will execute without impacting each other



## Durability

↳ holding data in the logs  
↳ helps us from failure & recovery



Transfer INR 100 from A to B

Steps

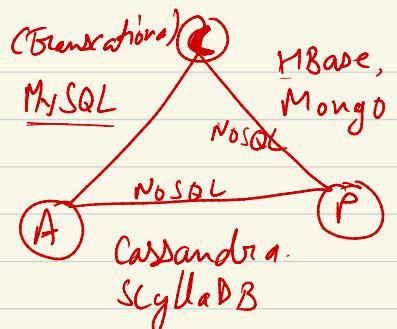
- ↳ Bring total Amount from A
- ↳ minus 100 from A
- ↳ Bring total Amount from B
- ↳ add 100 to B

CAP Theorem

- ↳ There is no Database as such which can provide all three properties at once
- ↳ Only two can be achieved

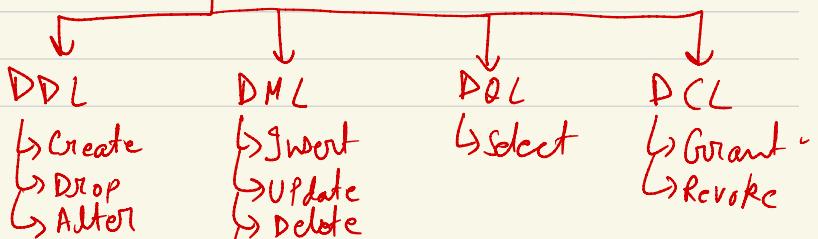
CAP

- ↳ C → Consistency
- ↳ A → Availability
- ↳ P → Partition tolerance

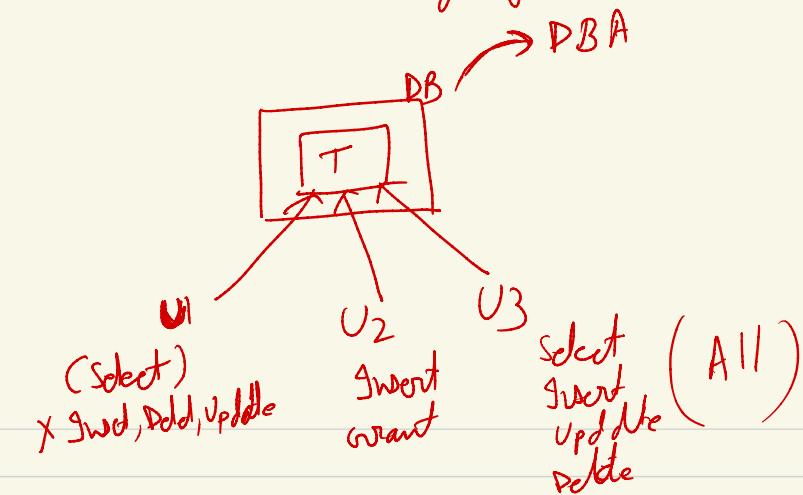


SQL → To Query the data

SQL Commands



- ① DDL → Data definition Language
- ② DML → Data Manipulation language
- ③ DQL → Data Query language
- ④ DCL → Data Control language



Revoke Select

## Data Types in SQL

### ① String Data Type

↳ CHAR(n) → all characters whose ASCII value is b/w 0 - 255  
 ↳ VARCHAR(n) → all unicode characters as well  
 0 to 65535

### ② Numeric Data Type

→ INT(n)	<u>(-2147483648 to</u>
→ BIGINT(n)	<u>2147483647</u>
→ Decimal(size,d)	864.532 size = 6 d = 3

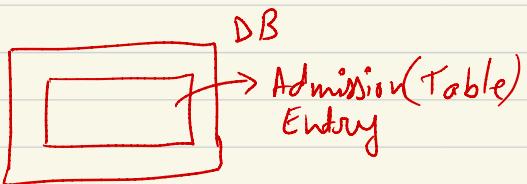
### ③ Date and Time Data Types

- Date ( YYYY-MM-DD )
- Timestamp ( YYYY-MM-DD hh:mm:ss )

### Integrity Constraints

↳ Why? → To keep data consistent with respect to Data Quality

Rules to be followed



- ↳ Name (Not Null)
- ↳ Address (Not Null)
- ↳ Attachment (<5MB)
- ↳ CGPA (>55%)
- ↳ Country (Default - INDIA)

### Integrity Constraints in MySQL

① NOT NULL

② UNIQUE

③ PRIMARY KEY

④ FOREIGN KEY

⑤ CHECK

⑥ DEFAULT

⑦ CREATE INDEX