

*Introduction to*

**MongoDB**

**Jayant Gupta**  
**CSE VI - B**  
**2100290109006**

*Database*

```
graph TD; Database[Database] --> RDBMS([RDBMS]); Database --> NoSQL([NoSQL]);
```

A hand-drawn diagram in brown ink. At the top, the word "Database" is written in a cursive font and enclosed in a rectangular box with a double border. A vertical line descends from the bottom center of this box. This line then branches into two diagonal lines, each ending in an arrowhead pointing towards an oval below. The left oval contains the text "RDBMS" in a serif font. The right oval contains the text "NoSQL" in a serif font.

RDBMS

NoSQL

# *MySQL vs MongoDB*

Table Format

Requires a schema

Uses FK

Supports Joins

Can scale vertically

No sharding

JSON Format

No prior schema

Does not use FK

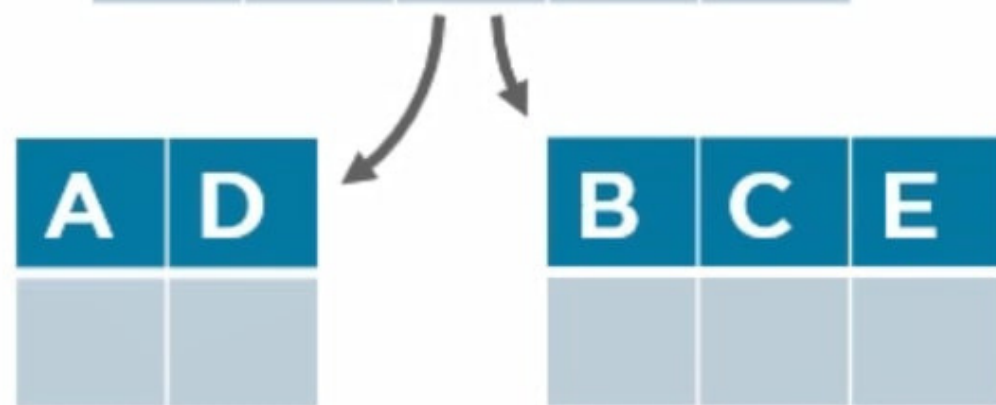
Does not support joins

Both vertically and  
horizontally

Support sharding

## Orders

A	B	C	D	E



## Vertical Partitioning

ID	Name	Email
1	Rahul	Rahul@gmail.com
2	Sonu	
3	Jay	
4	Monu	

Range Based Sharding



ID	Name	Email
1		
3		

ID	Name	Email
2		
4		

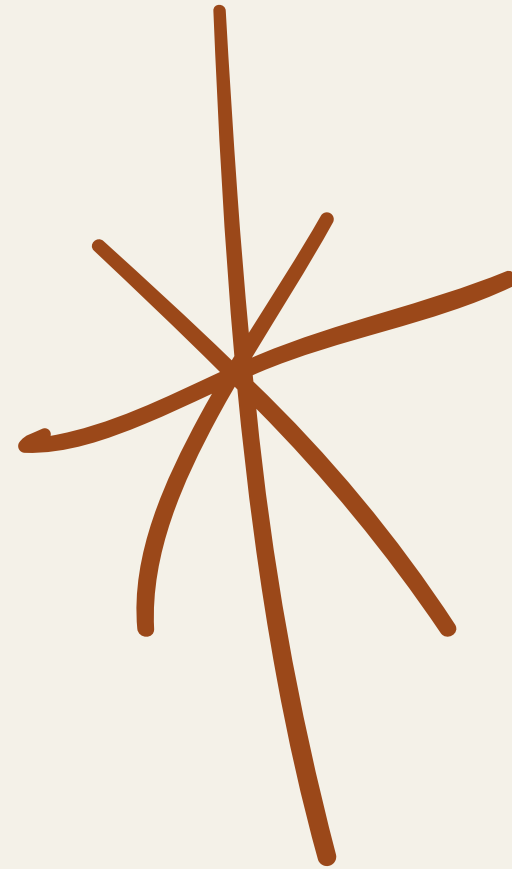
# Conclusion

*If requires the multi-row transaction and have structured data use **RDBMS***

*For real-time analytics, IoT, and unstructured data use **NoSQL.***

# *Some examples of RDBMS and NoSQL*

*Oracle*  
*MySQL*  
*PostgreSQL*  
*SQLite*



*MongoDB*  
*Dynamo DB*  
*Oracle NoSQL*  
*Cassandra DB*  
*Redis*

# *Data Types in MongoDB*

String  
Integer  
Double  
Boolean  
Array  
Null  
Object  
Undefined



```
{  
  "date": {  
    "$date": 936144000000  
  },  
  "amount": 955,  
  "transaction_code": "buy",  
  "symbol": "csco",  
  "tags": ["arm", "cde", "dli"],  
}
```

# *Some terminologies in MongoDB*

**Collection**

**Document**

**Object**

**Key-value pair**





# Mostly used operations in MongoDB

## *Create*

`db.createCollection()`

`db.collection.insertOne()`

`db.collection.insertMany()`

## *Read*

`db.collection.find()`

`db.collection.findOne()`

## *Update*

db.collection.updateOne()

db.collection.updateMany()

db.collection.replaceOne()

## *Delete*

db.collection.deleteOne()

db.collection.deleteMany()



*Thank*

*You*