2WaysToScale

September 9, 2025

[ ]:

```
```

Query optimization: read about prepared statements

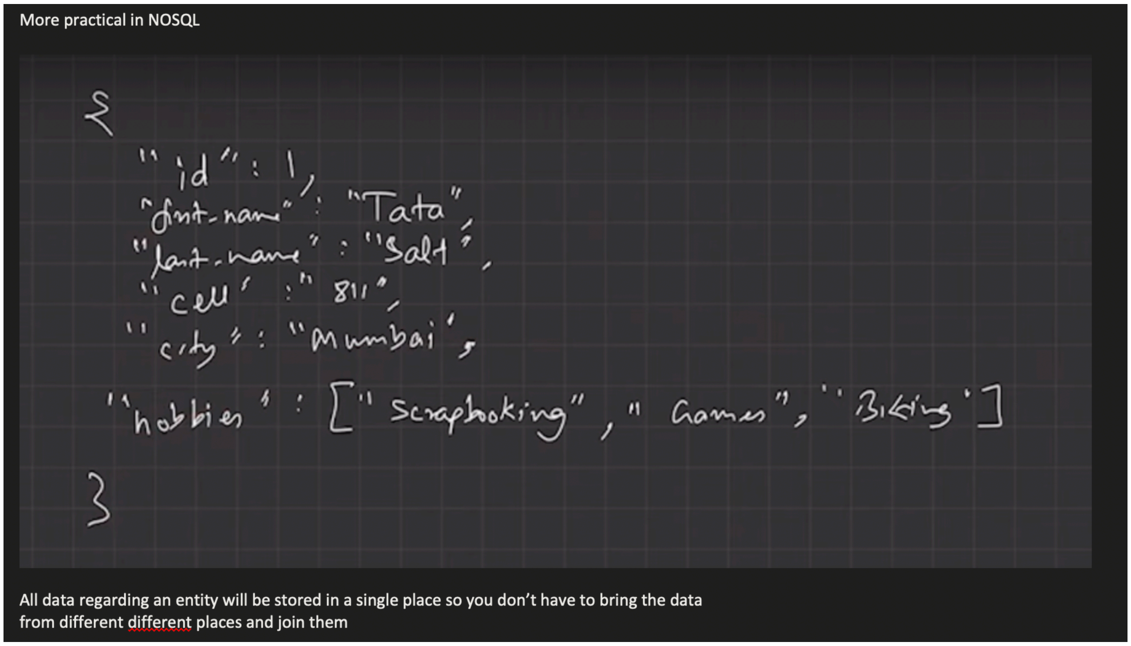
connection pool

cache non dynamic data
booking history
payment history
user profile

Database replication
helps in disaster recovery
related to clustering

Command Query Responsibility Segregation Cqrs

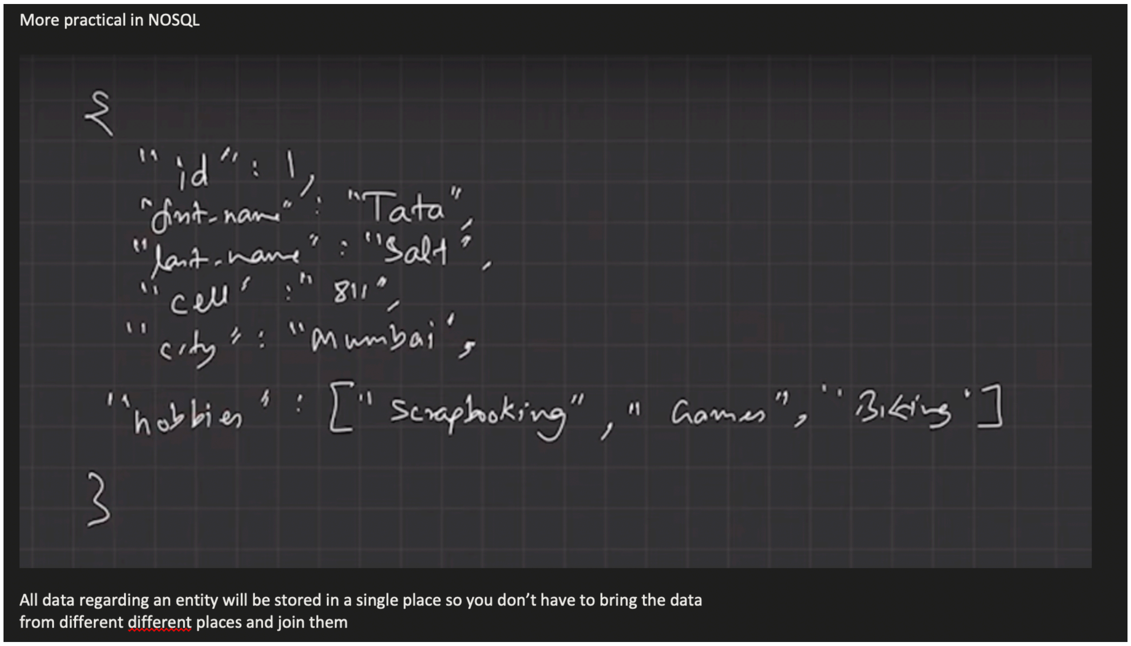
doubt



More practical in NOSQL

```
{
   "id" : 1,
   "first-name" : "Tata",
   "last-name" : "Salt",
   "cell" : "811",
   "city" : "Mumbai",
 "hobbies" : ["Scrapbooking", "Games", "Biking"]
}
```

All data regarding an entity will be stored in a single place so you don’t have to bring the data
from different different places and join them

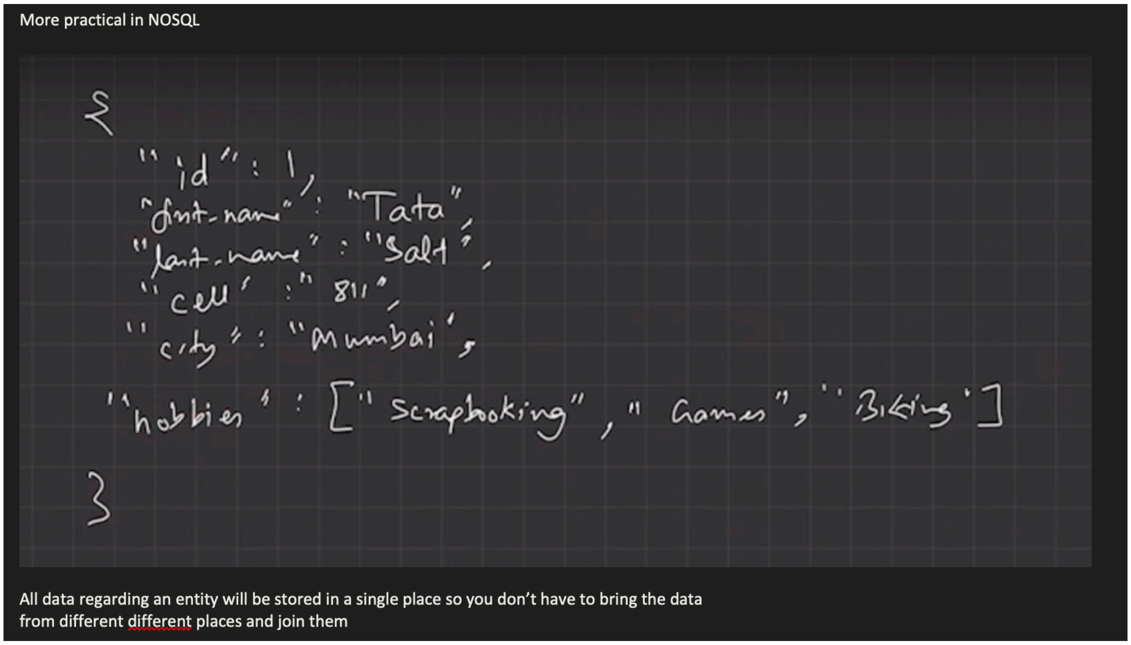


More practical in NOSQL

```
{
  "id": 1,
  "first-name": "Tata",
  "last-name": "Salt",
  "cell": "811",
  "city": "Mumbai",
 "hobbies": ["Scrapbooking", "Games", "Biking"]
}
```

All data regarding an entity will be stored in a single place so you don't have to bring the data from different different places and join them

gaurav sen gyaan



More practical in NOSQL

```
{
  "id": 1,
  "first-name": "Tata",
  "last-name": "Salt",
  "cell": "811",
  "city": "Mumbai",
 "hobbies": ["Scrapbooking", "Games", "Biking"]
}
```

All data regarding an entity will be stored in a single place so you don't have to bring the data from different different places and join them

Partitioning
availability
performance
parallelism
manageability
reduce cost (horizontal scaling cheaper then vertical scaling)

Sharding

Whats the difference from partitioning?

technique to do horizontal partitioning ?
we also have to introduce a routing layer

bad when your data is spread over different db instances (analytical type of queries, scatter-gather problem)