

1. we need to persist / store / retain our data

- between n app starts
- share it across the globe.

formats : text file X
 : binary file
 • SQL DB

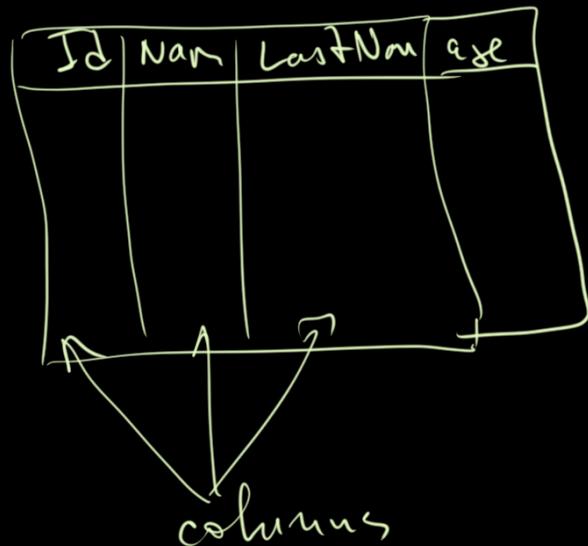
problems?

- app not load
- simultaneous access

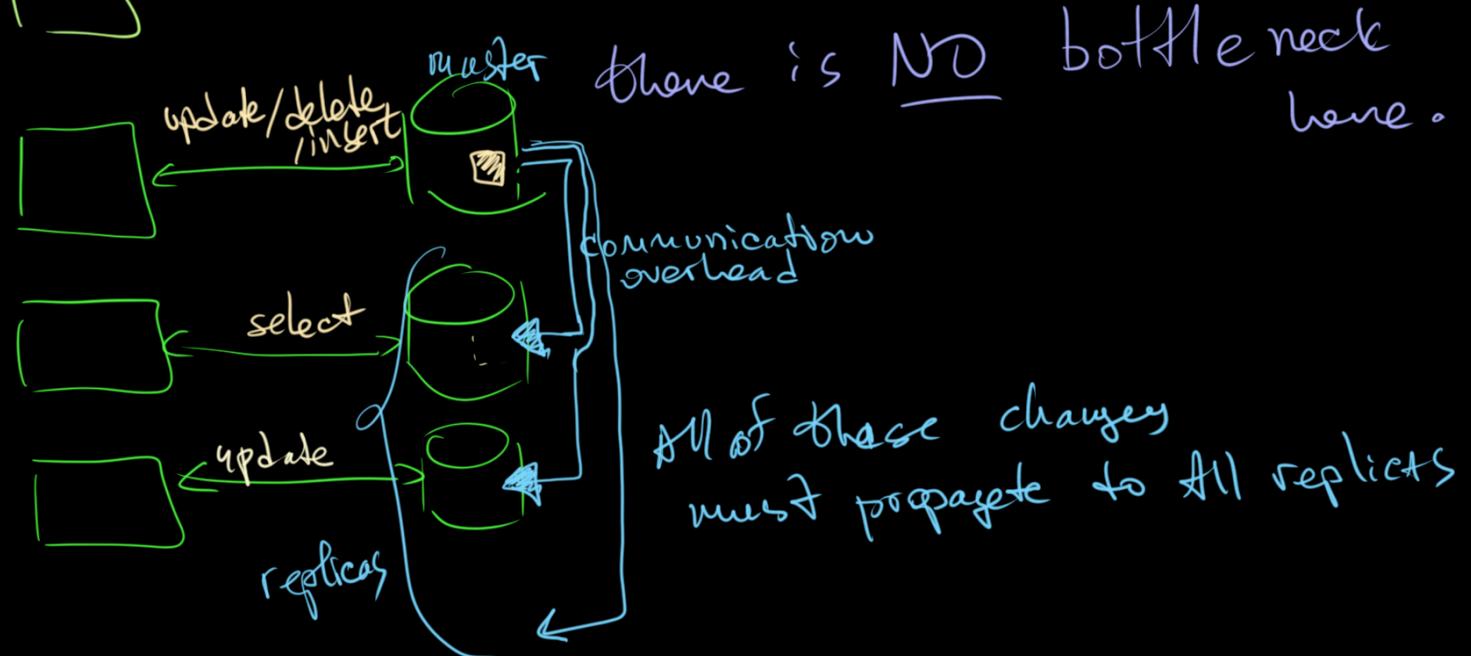
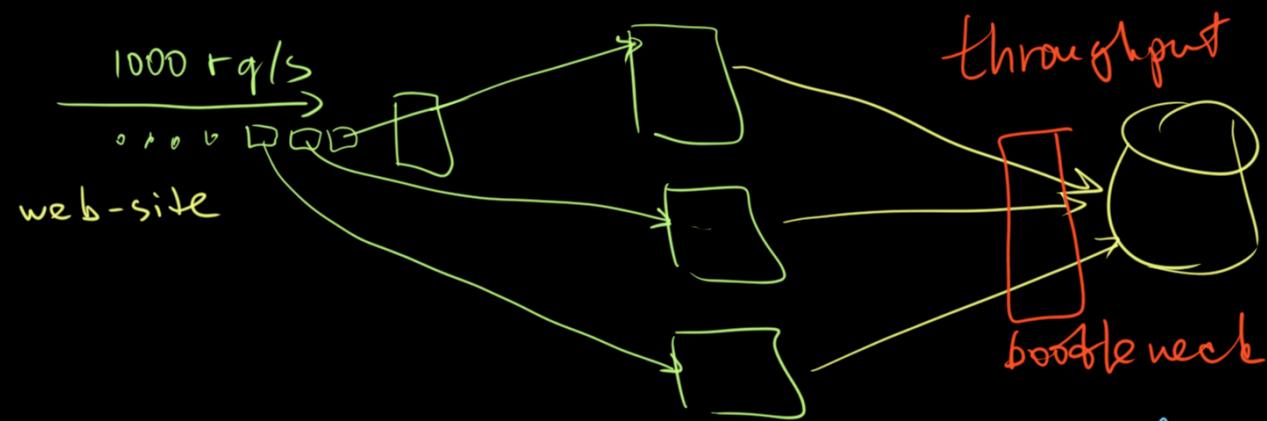
Table: User

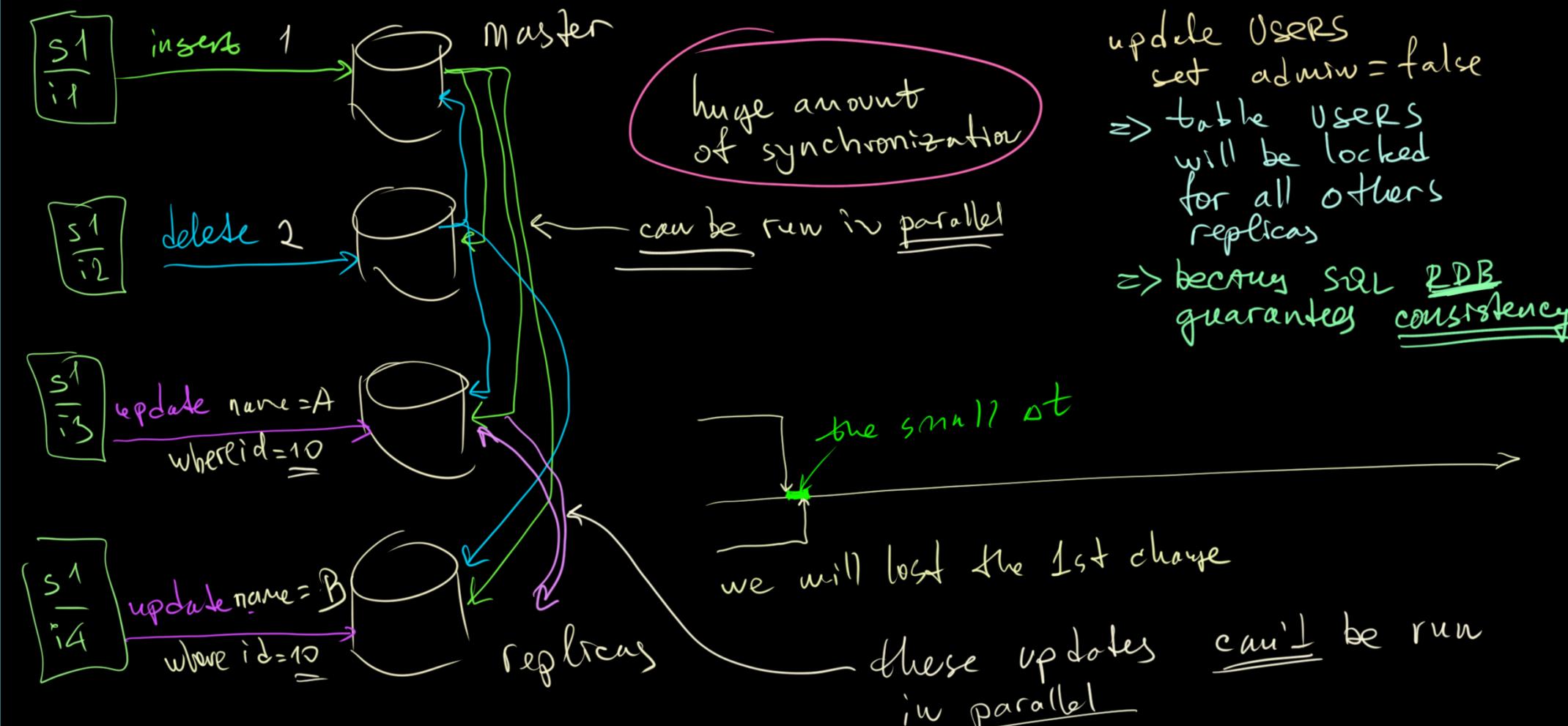
ID	Name	Last Name	Age
1	John	Doe	30

columns



10 connections - OK
 50 ~ OK
 100+ problem
 1000+





update USERS
set admin=false
=> table USERS
will be locked
for all others
replicas
=> because SQL PDB
guarantees consistency

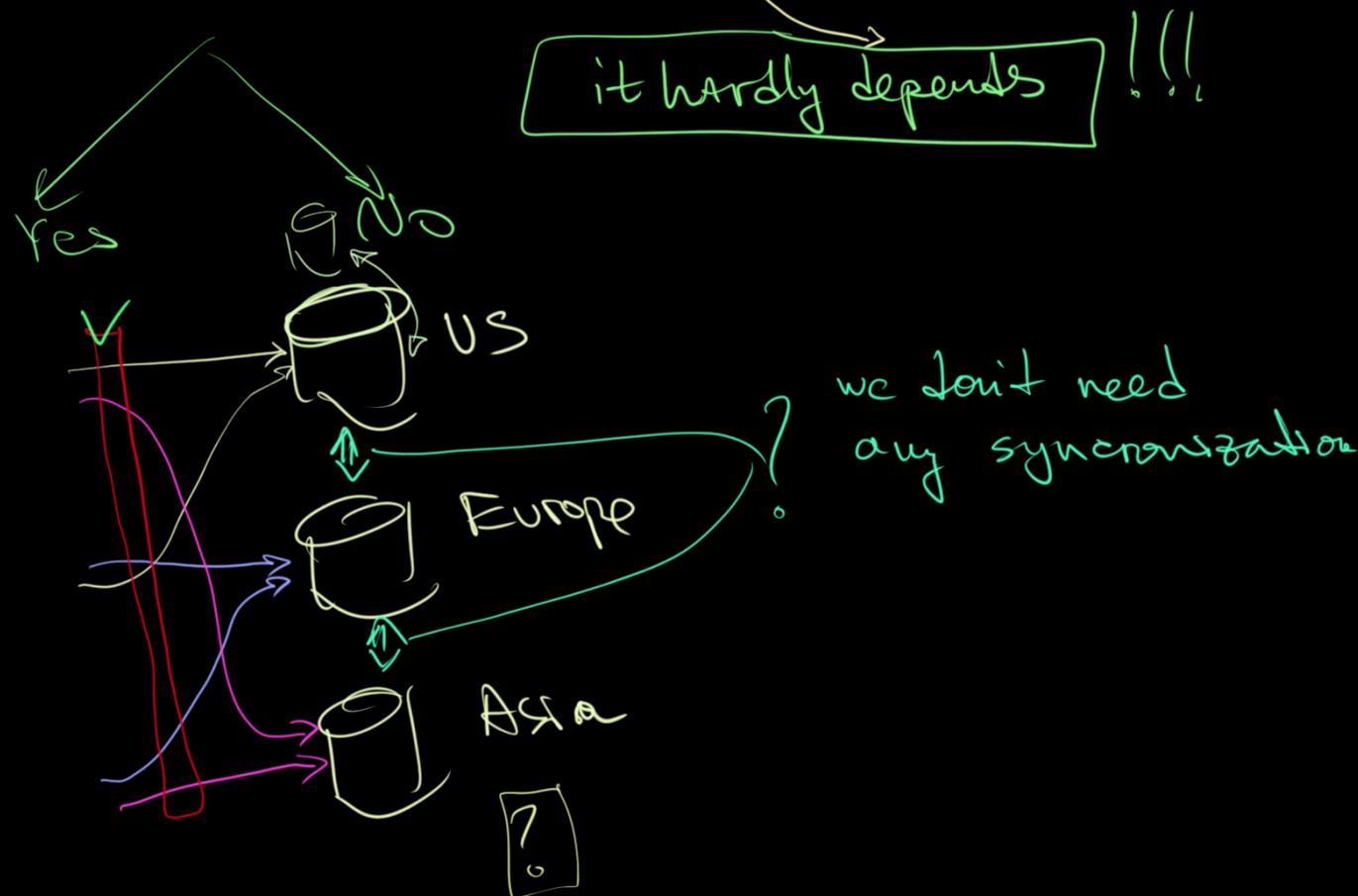
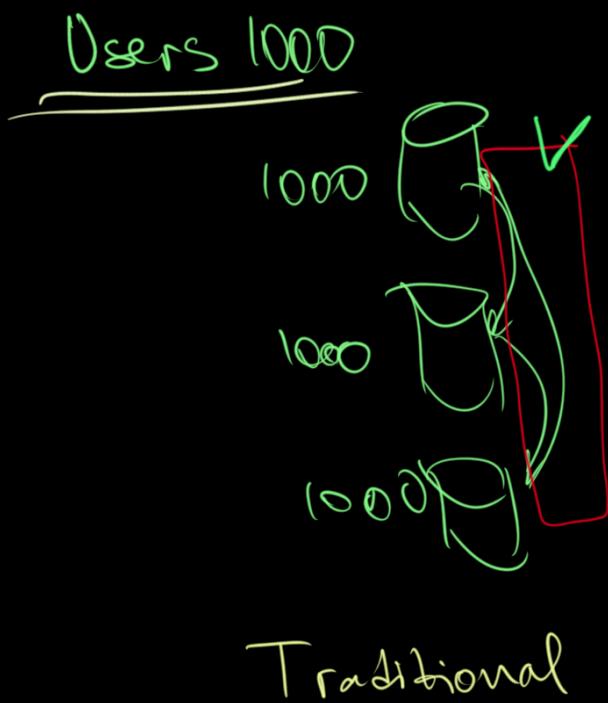
• what id connection problem

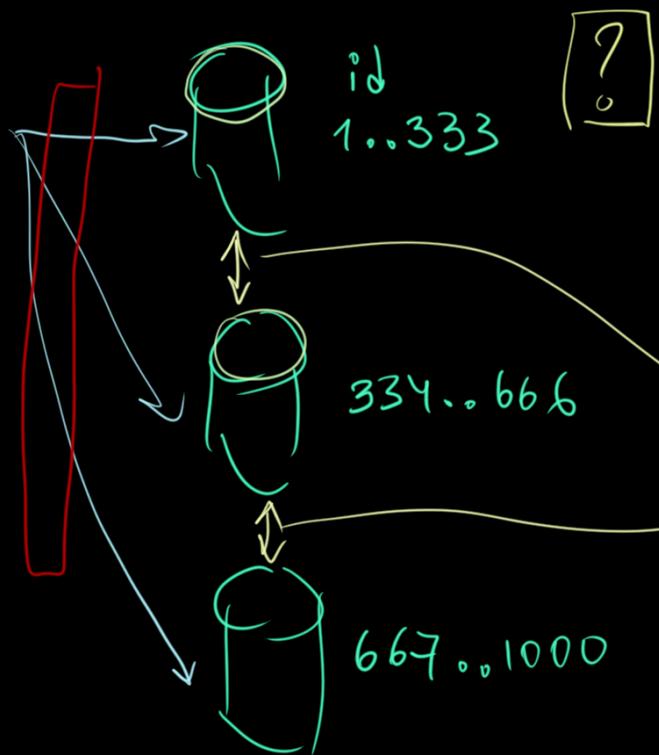
SOLUTIONS

+1. we can sync. NOT immediately (during non-business hours)

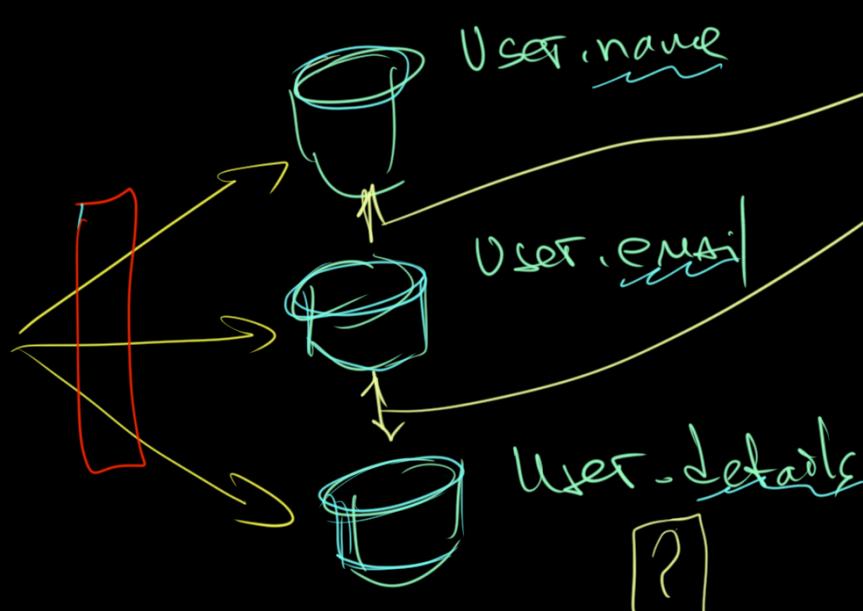
=> some instances will have the OLD data - bad

it hardly depends !!!

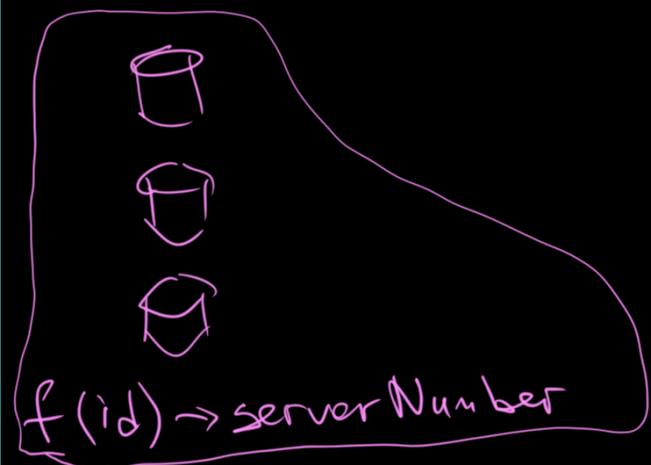




? we don't need
any sync



? we don't
need
sync

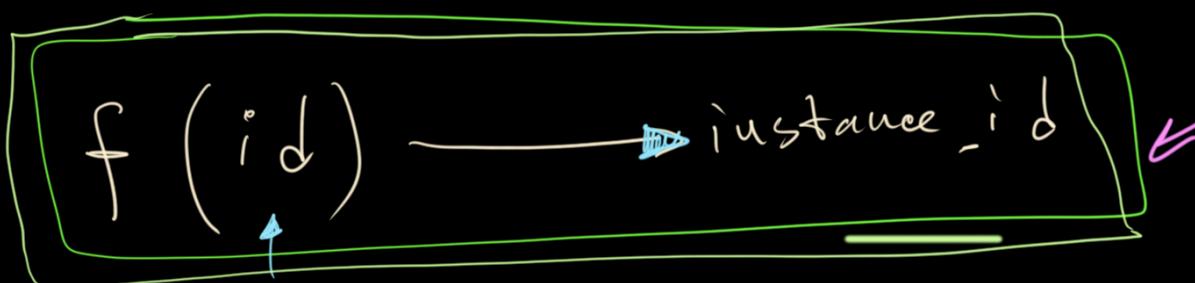


↓↓ we solve the problem

~~we still have
a problem~~

without you don't know where your data is #6

- ✓ This problem
▼ is much simpler!!!
▼ can be delegated!!!



sharding

This sharding can be dynamic

$$333 \% 3 = 0$$

1. N servers +1

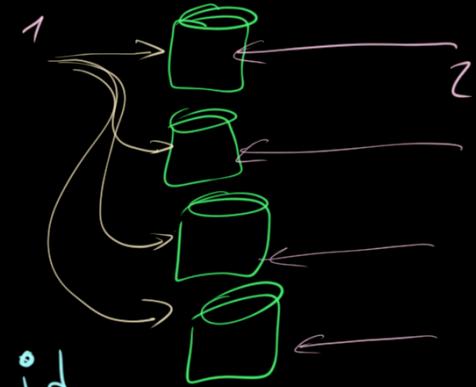


this sharding approach

1 huge amount of data (doesn't fit 1 machine)

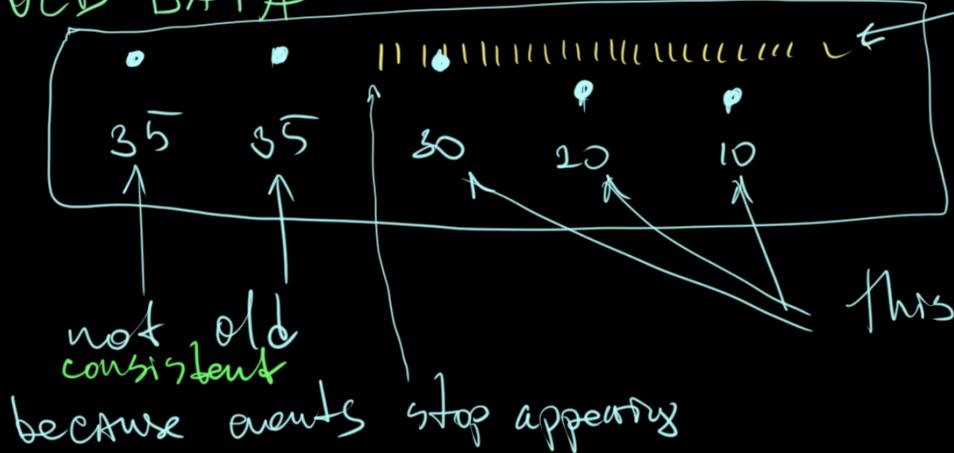
2 huge amount of requests to the same data

+ shard function $f(\text{property}) \rightarrow \text{shard-id}$



#7

OLD DATA



events

EVENTUAL
CONSISTENCY

this data "can be old"
aren't consistent

~~NoSQL~~ databases

NoSQL - wrong term

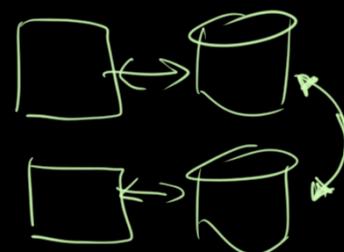
- it doesn't reflect the name

Non-Relational Database

- we don't have any references (foreign key, join)
- we don't have any transactions

why do we have/need such kind of weak ACID?

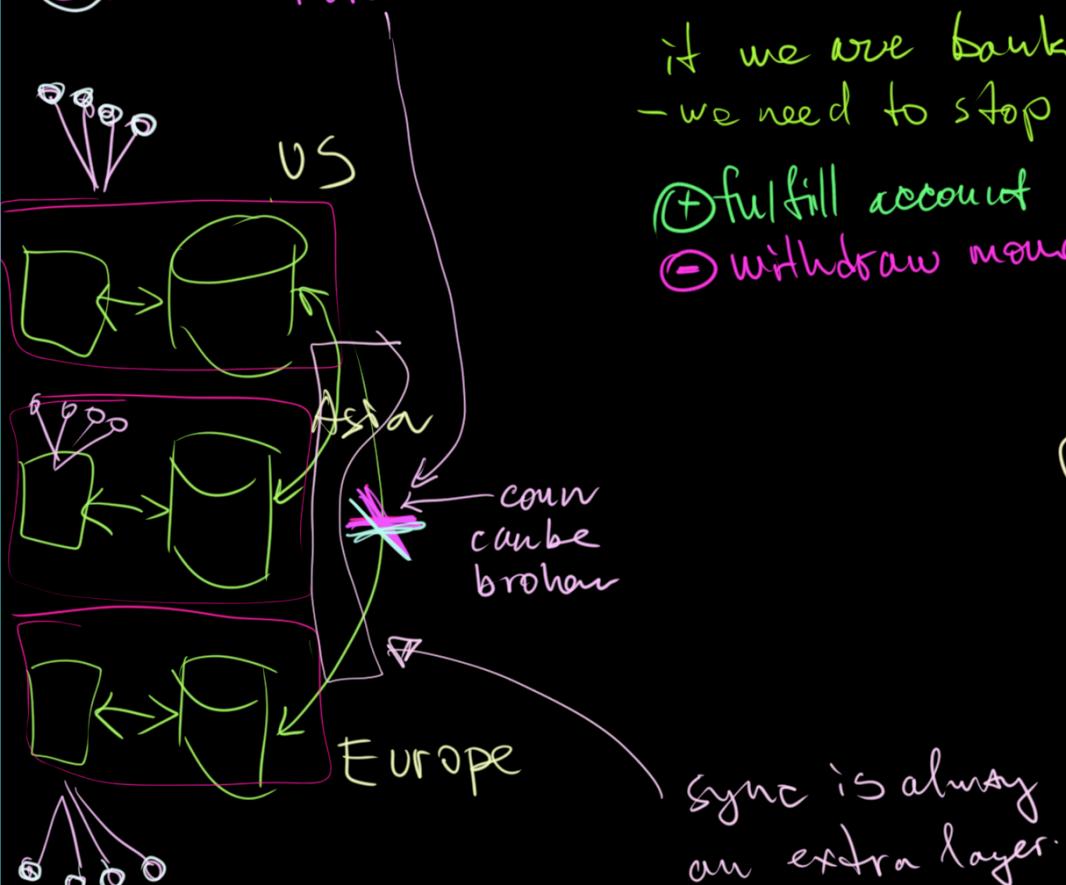
when we have # requests > 1 machine



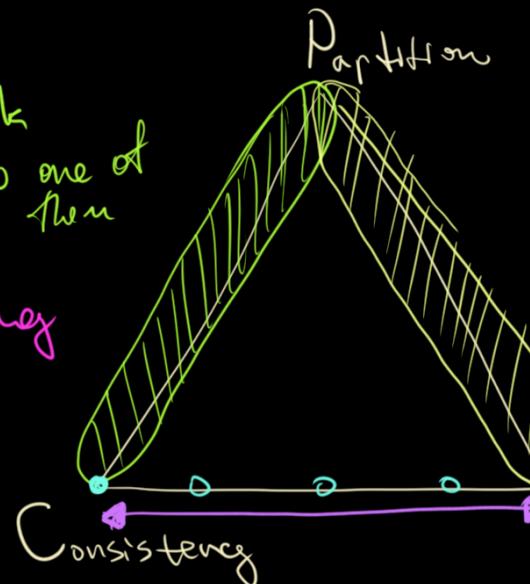
we can't guarantee consistency

once we start sync
 \Rightarrow no way to have the same throughput

1. Consistency
2. Availability
3. Partition



100% synced
many requests / many data



CAP Theorem

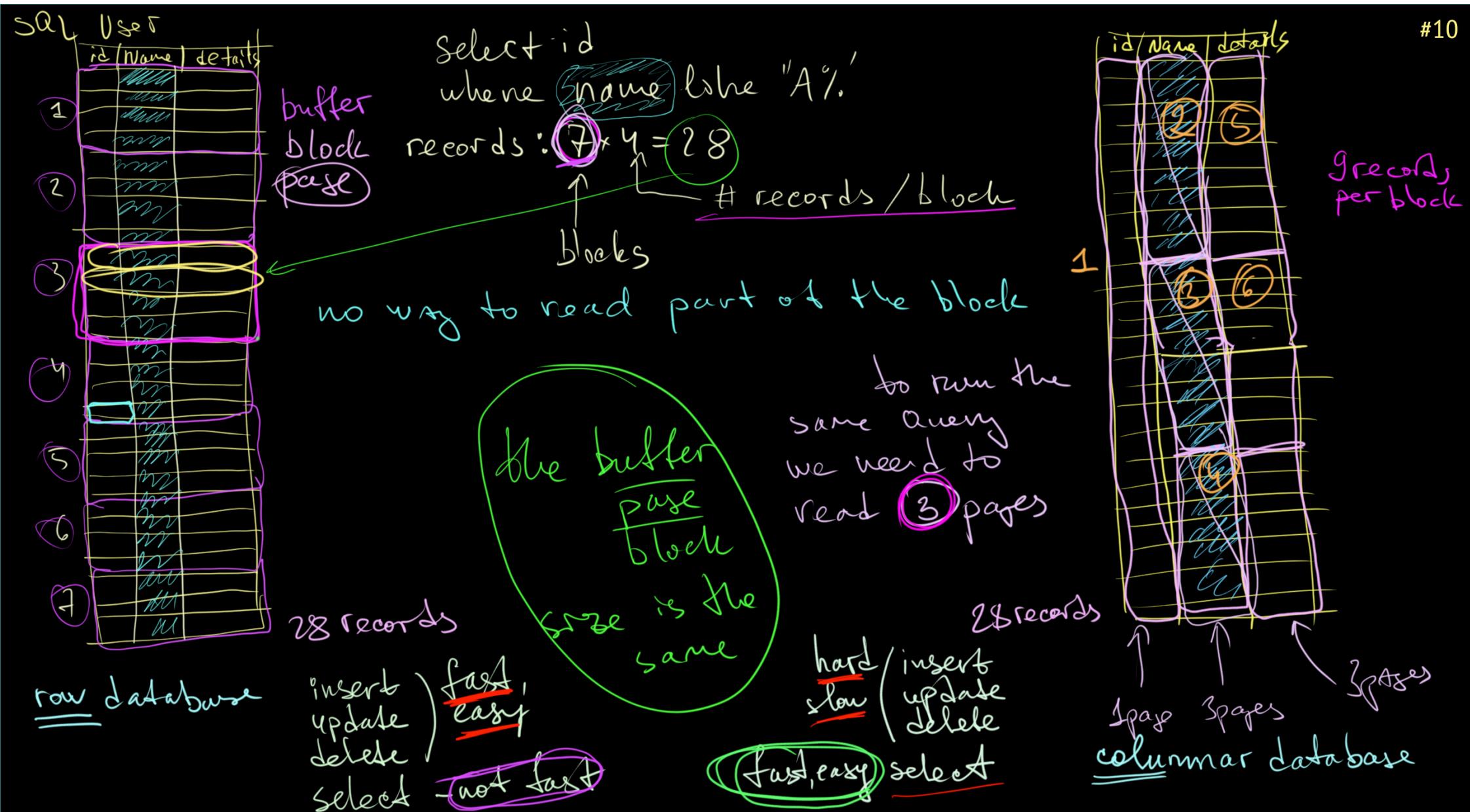
in case of Partition

decide → Consistent
Available

that's the business task

developer's task
is to explain

NO WAY to Be consistent and available at the same moment



```

select *
from users
where name like 'A'
and id > 100
and details like '%smart%';

```

row by row
 we access many columns
 in one query



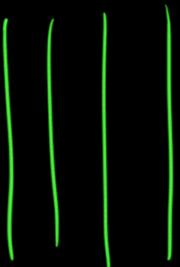
ORACLE, Postgres, MySQL
row databases

```

select *
from users
where name like 'A'

```

- do we need to all rows
- we are interested in one column ONLY
- not all(many)



Cassandra, Vertica
columnar databases