
Mongo.DB({
presenter: “Tooraj Taraz”,
course: “DataBases Lab”
});

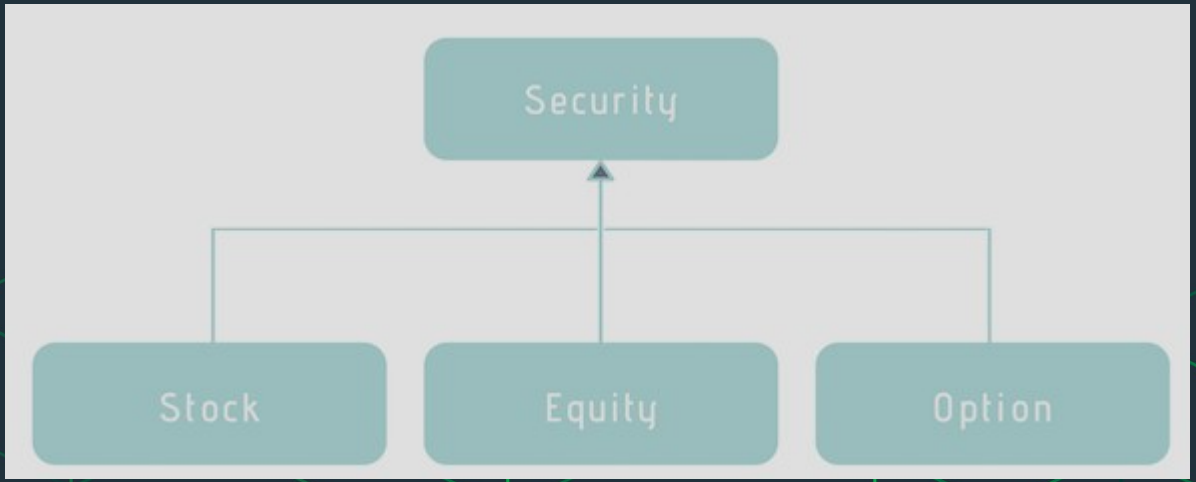
Structured Query Language

```
SELECT CustomerName, City, Country FROM Customers;
SELECT * FROM Customers
WHERE Country='Mexico';
CREATE TABLE Orders (
    OrderID int NOT NULL,
    OrderNumber int NOT NULL,
    PersonID int,
    PRIMARY KEY (OrderID),
    CONSTRAINT FK_PersonOrder FOREIGN KEY (PersonID)
    REFERENCES Persons(PersonID)
);
```

NoSQL - MongoDB

Polymorphic Schema

- The grand schema
- OOP (Inheritance)



To Embed or Not to Embed: That is The Question!

Embedding

- In one-to-one and one-to-few relationships
- When two models belong intrinsically together
- When data is mostly read and rarely updated

Referencing

- In one-to-squillion relationships
- When data is frequently updated
- When data needs to be queried independently
- Array Size cannot grow indefinitely:
  - use Child Referencing for one-to-many relationships
  - and Parent Referencing for one-to-squillion relationships

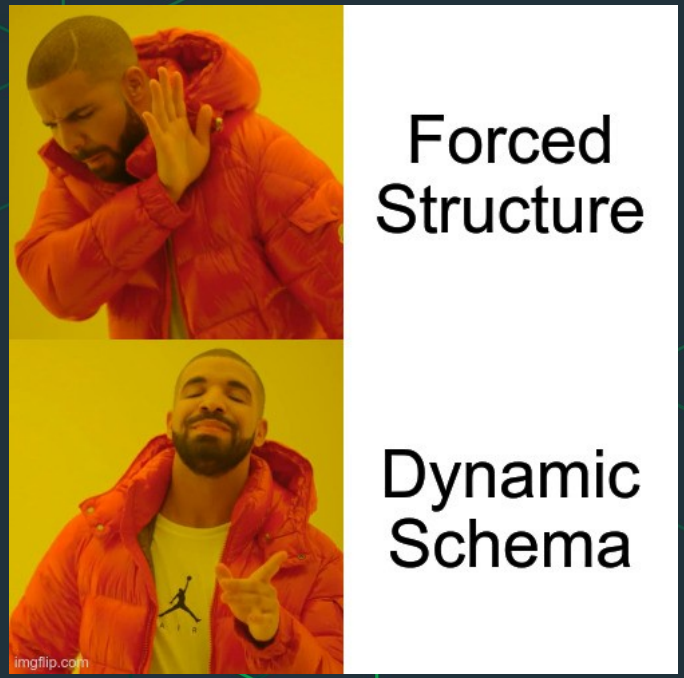
MEMES!


Embedding
Referencing
DEVELOPER
imgflip.com
JAKE-CLARK.TUMBLR

MEMES!


⚠TRADE OFFER⚠
i receive:
REDUNDANCY
you receive:
SPEED
imgflip.com

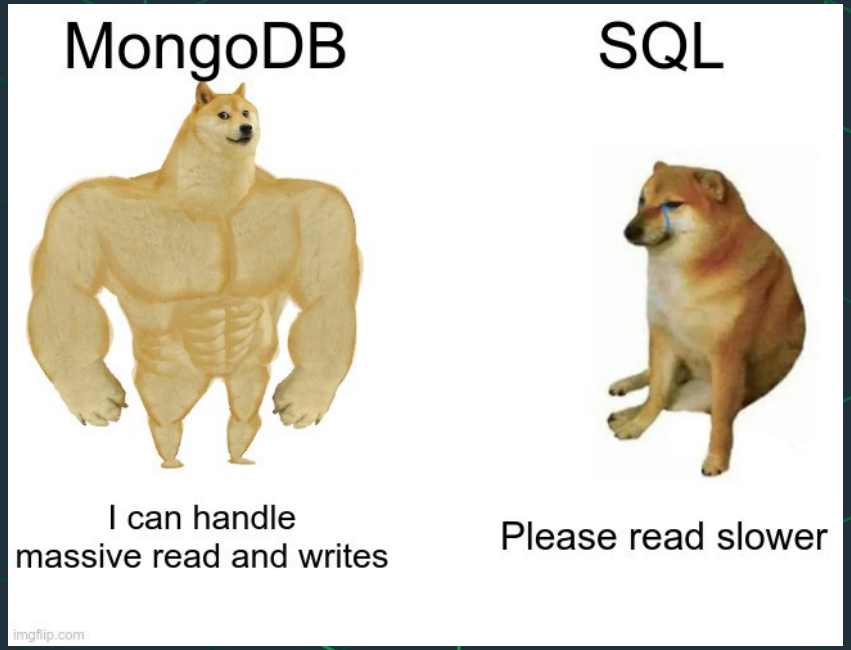
MEMES!



MEMES!


NO REDUNDANCY AND FAST READ AND WRITE
DEVELOPER
NO REDUNDANCY AND FAST READ AND WRITE
REALITY OF SQL AND NOSQL
DEVELOPER
imgflip.com

MEMES!


MongoDB
SQL
I can handle
massive read and writes
Please read slower
imgflip.com

MySQL vs MongoDB

```
CREATE TABLE people (
    id MEDIUMINT NOT NULL
        AUTO_INCREMENT,
    user_id Varchar(30),
    age Number,
    status char(1),
    PRIMARY KEY (id)
)
```

```
INSERT INTO people(user_id,
                   age,
                   status)
VALUES ("bcd001",
        45,
        "A")
```

```
SELECT *
FROM people
WHERE status = "A"
ORDER BY user_id DESC
```

```
db.people.insertOne( {
    user_id: "abc123",
    age: 55,
    status: "A"
} )
```

```
db.people.insertOne({
user_id: "bcd001",
age: 45, status: "A"
})
```

```
db.people.find(
{ status:       "A" } ).sort( {
user_id: -1 }
)
```

MySQL vs MongoDB

- Data uses schemas
- Relations!
- Data is distrebuted across multiple tables
- Vertical scaling
- Speed limitations
- **FREE as in FREEDOM**

- Schema-less
- No/very few relations
- Data is nested in a few collections
- Both horizontal and vertical scaling
- Greate performace for mass read and simple writes
- **OpenSource BUT NOT FREE**

Q&A