

Bibliography

- Plugge, E., Membrey, P., The definitive guide to MongoDB. Ed. Apress
- <http://docs.mongodb.org/>

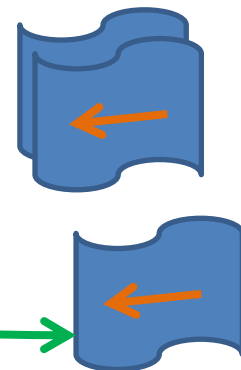
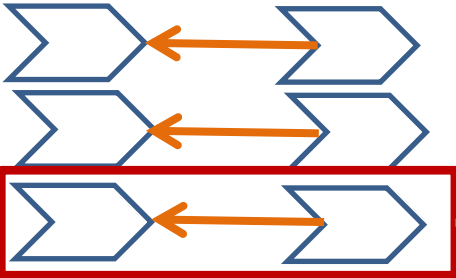
SQL DB
e.g., My
SQL,
Oracle,
Derby

No SQL DB
E.g.,
MongoDB,
DynamoD
B

Table

Colection

Row



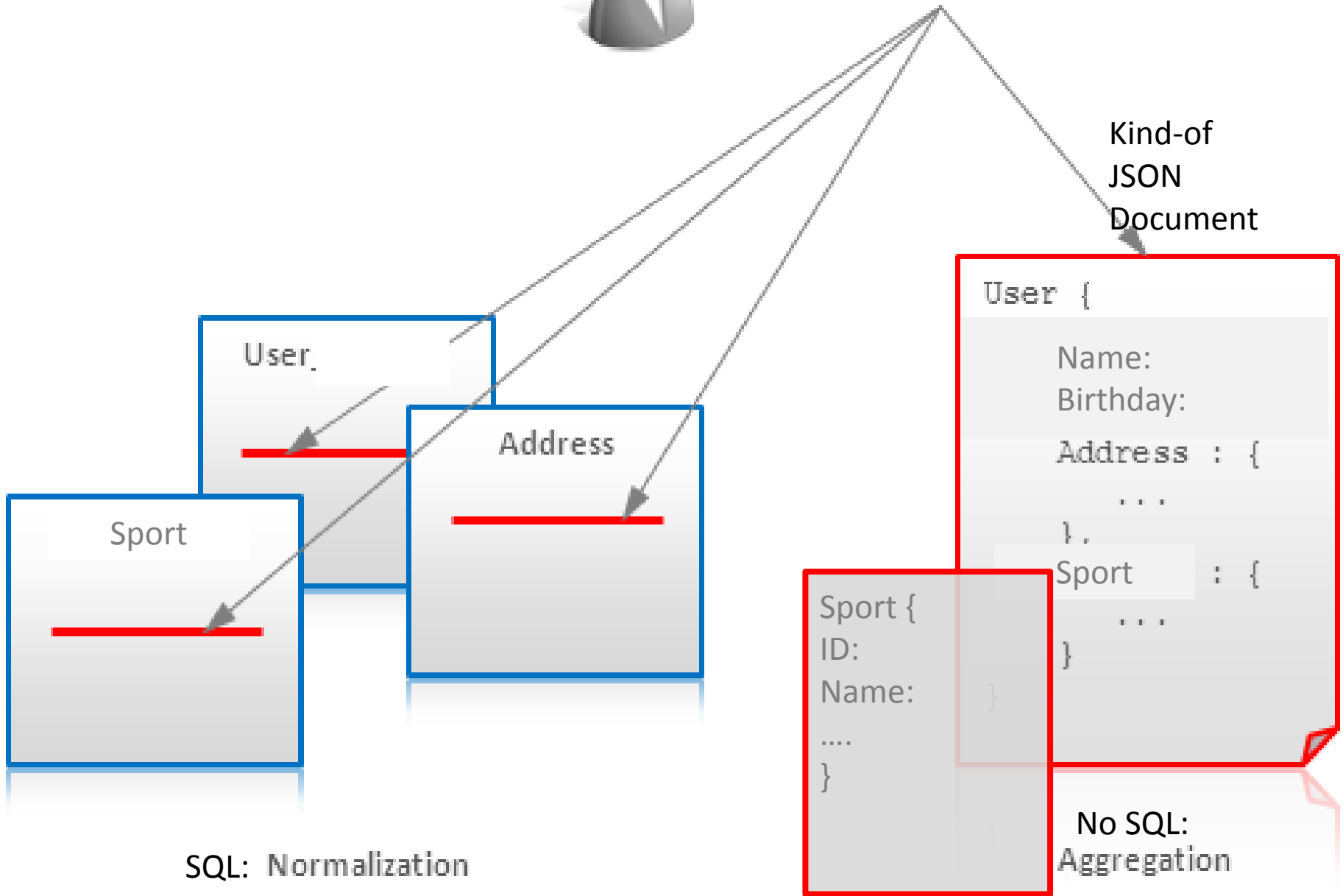
Document

Example: Gym

- A user practices a sport and has multiple addresses
- Functional requirements
 - Get the details of a user
 - Get the list of sports



Update User Business Entity



SQL: Normalization

No SQL:
Aggregation

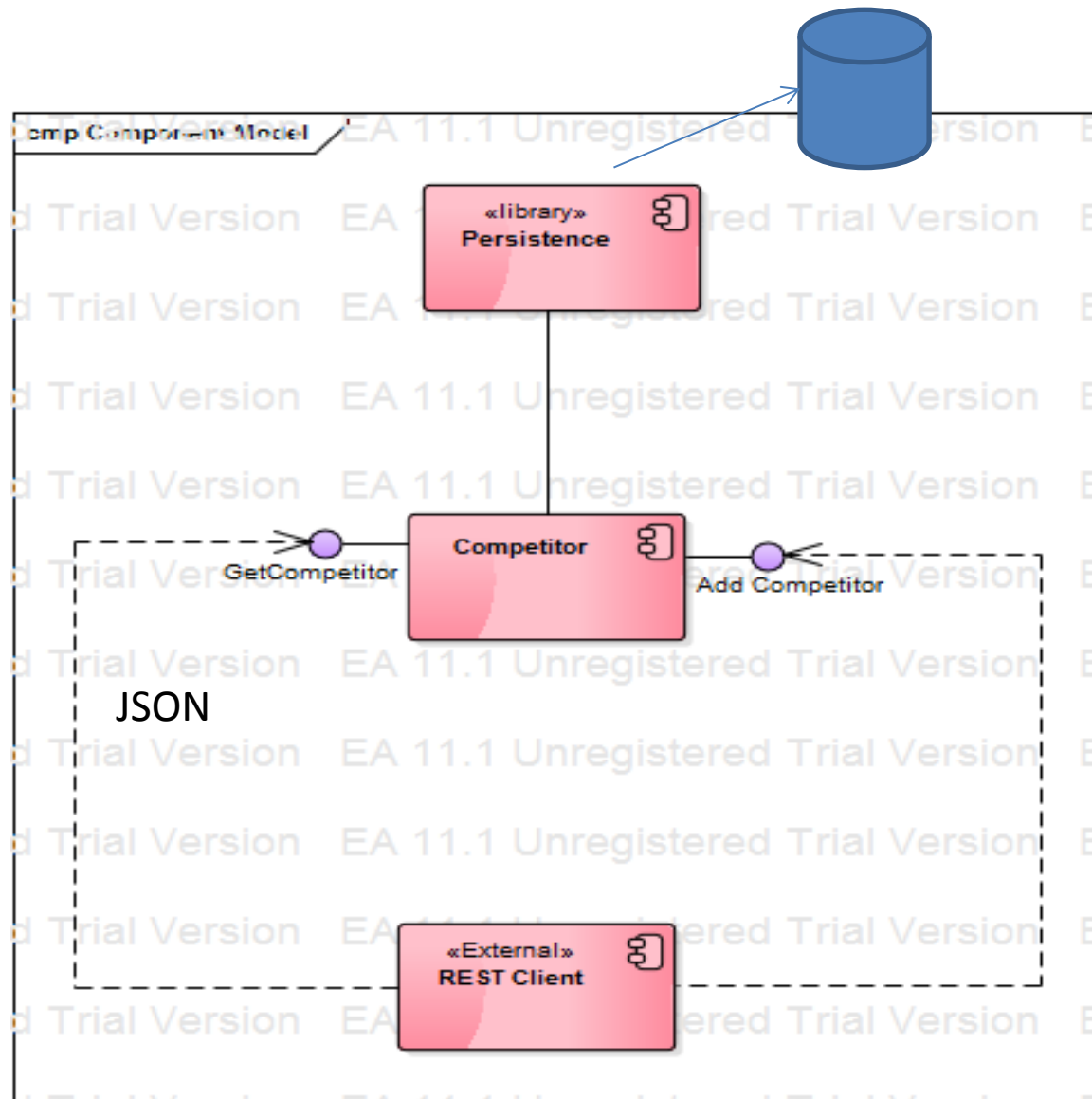
MongoDB supports ACID operations in a single document ,
but not support multi-document transactions

A **C** **I** **D**
ATOMICITY *CONSISTENCY* *ISOLATION* *DURABILITY*

Joins

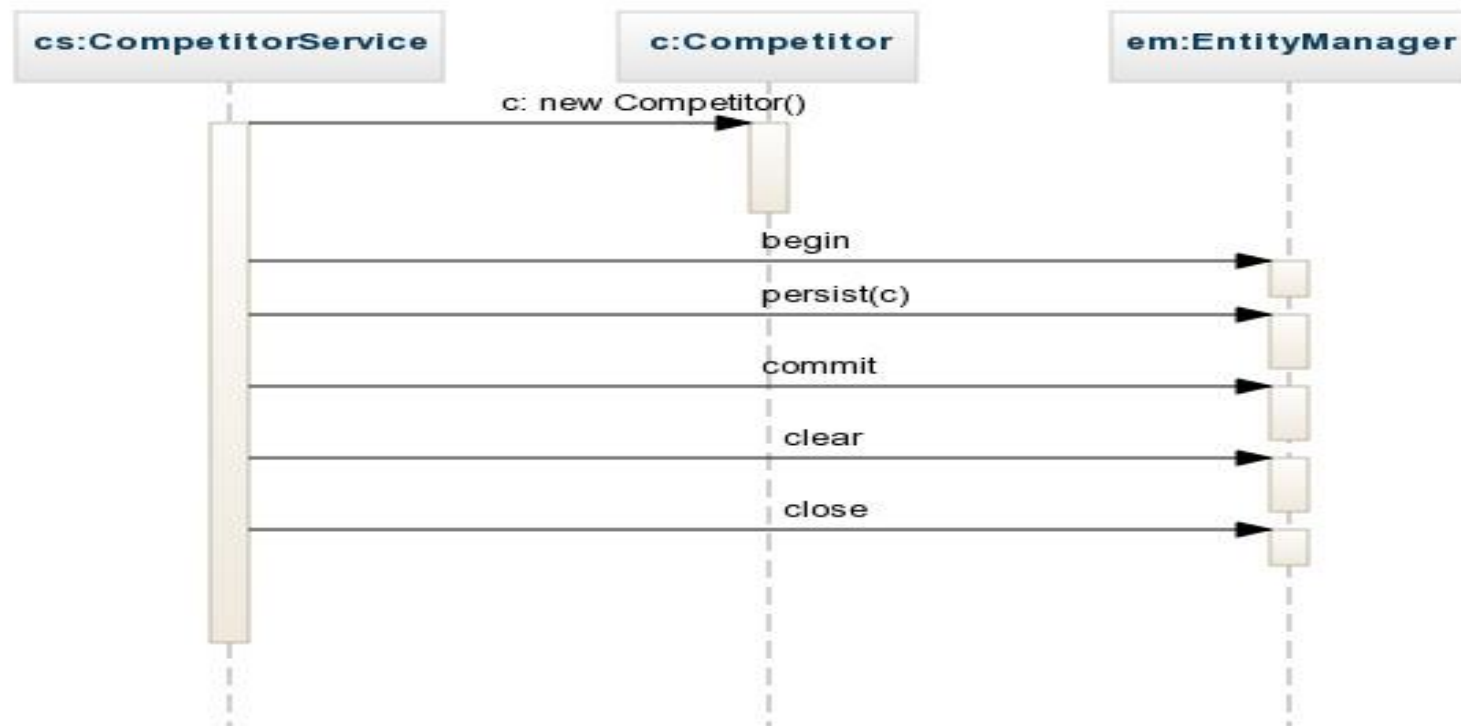
- JDBC and SQL DB
 - `ResultSet rs = stmt.executeQuery("SELECT u.name, u.birthDate, s.name FROM Sport s INNER JOIN User u ON s.id = u.sport_id");`
- JPA and SQL DB
 - `Query q = entityManager.createQuery("select u from User u);`
- JPA and Non SQL DB
 - `Query q = entityManager.createQuery("select u from User u);`

What do I code?



What do I code?

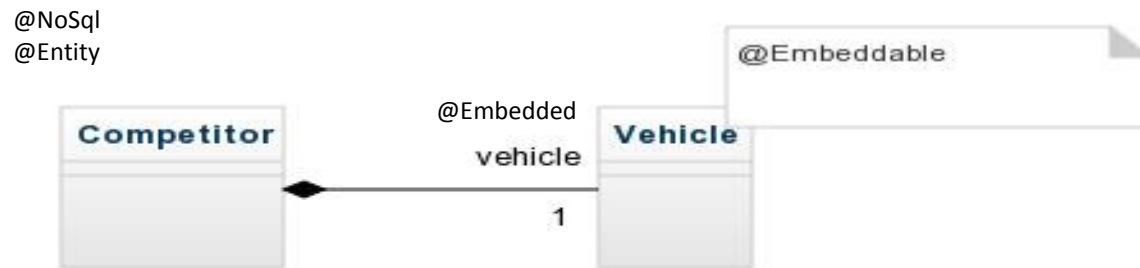
- At the service level, the code is the same for JPA SQL and JPA No SQL



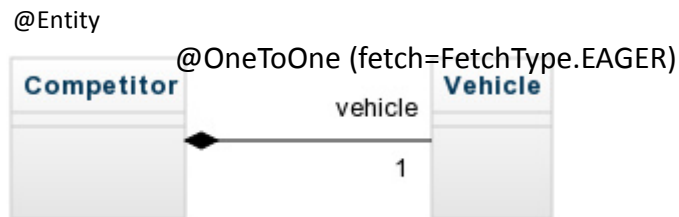
What do I Code?

- There are changes at the persistence level: the persistence unit changes as well as the entities

JPA No
SQL



JPA
SQL



JPA performs an
inner join

What do I code?

- At disk level, the data representation is different

SQL

Competitor

_Id	Winner	Cellphone	Telephone	Surname
54DC369A	False	3002345436	7659675	Gomez

Vehicle

Competitor_Id	Updated_at	Model year	Create_d_at	Plate	brand
54DC369A	Null	345	Null	TRE-465	BMW

No SQL

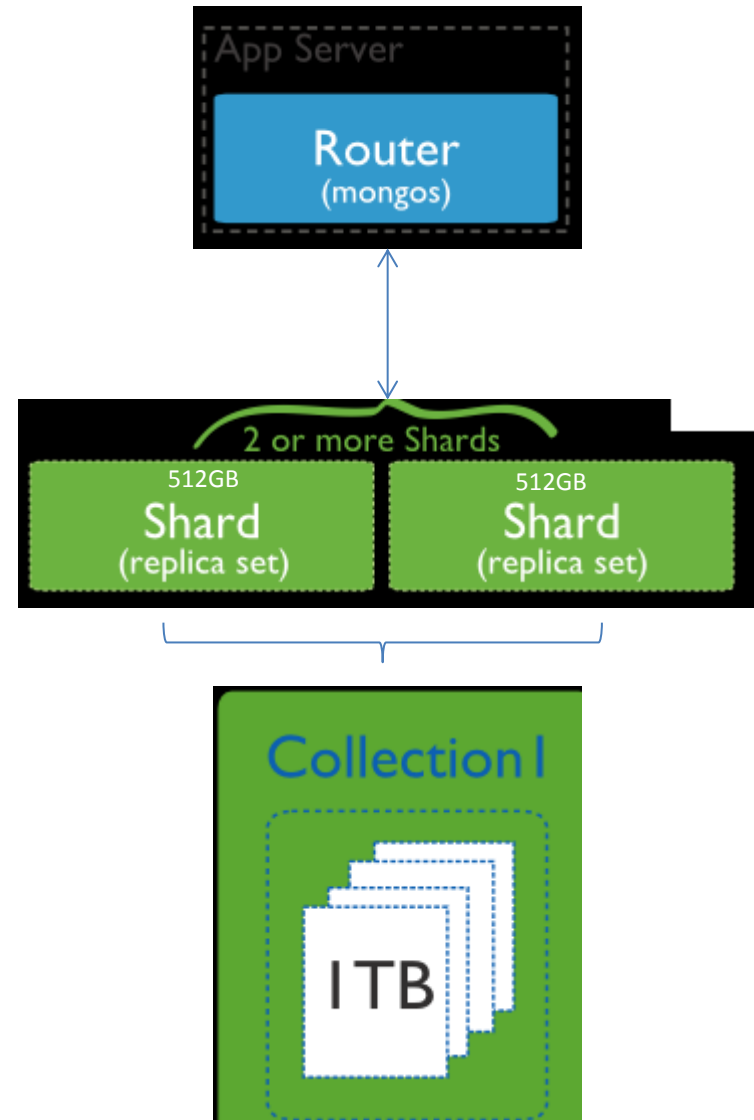
```
{
  "_id": "54DC369A036488ACE3712FEB",
  "WINNER": false,
  "CELLPHONE": "3002345436",
  "TELEPHONE": "7659675",
  "SURNAME": "Gomez",
  "VEHICLE": [
    {
      "updated_at": null,
      "MODELYEAR": 345,
      "create_at": null,
      "PLATE": "TRE-465",
      "BRAND": "BMW"
    }
  ],
}
```

Is it possible to scale out a SQL DB?

- Yes, but it is expensive
 - You buy a bigger, faster server (vertical scaling)
 - If there is not a bigger server to buy, the solution is to spread out to two or more servers (horizontal scaling)
 - buy a active/active application cluster (e.g., Oracle one)

Is it possible to scale out a No SQL DB?

- Yes, it is cheaper than SQL DB scaling
- The documents are separately stored in several servers (horizontal scaling)
- Ideal for Cloud computing



Comparison

No SQL

1. Aggregation (favours performance)
2. Scaling cheaper than SQL
3. Encouraged when requirements change from week to week or day to day (sacrifice data integrity)
 - E.g., Free web apps. that need to scale, social networks

SQL

1. Normalization (Too structured, little flexibility, affects performance)
2. Expensive to scale out
3. Encouraged when you need to be really uptight about data and high availability
 - E.g., bank accounts