

# **Introduction au NoSQL**

Guillaume BILLEY



**PHARMAGEST**

WELCOOP SOLUTION

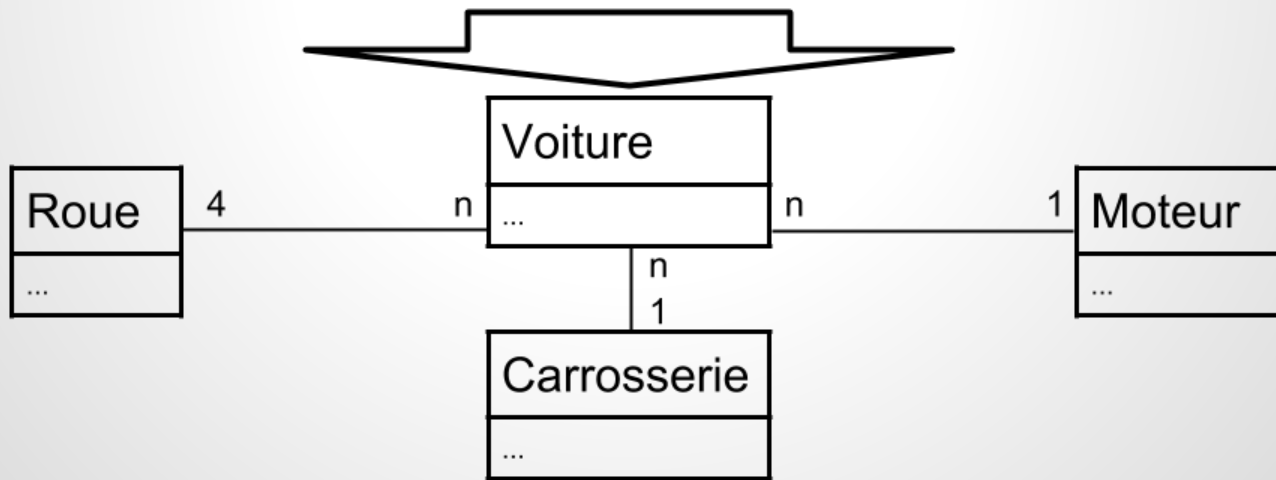
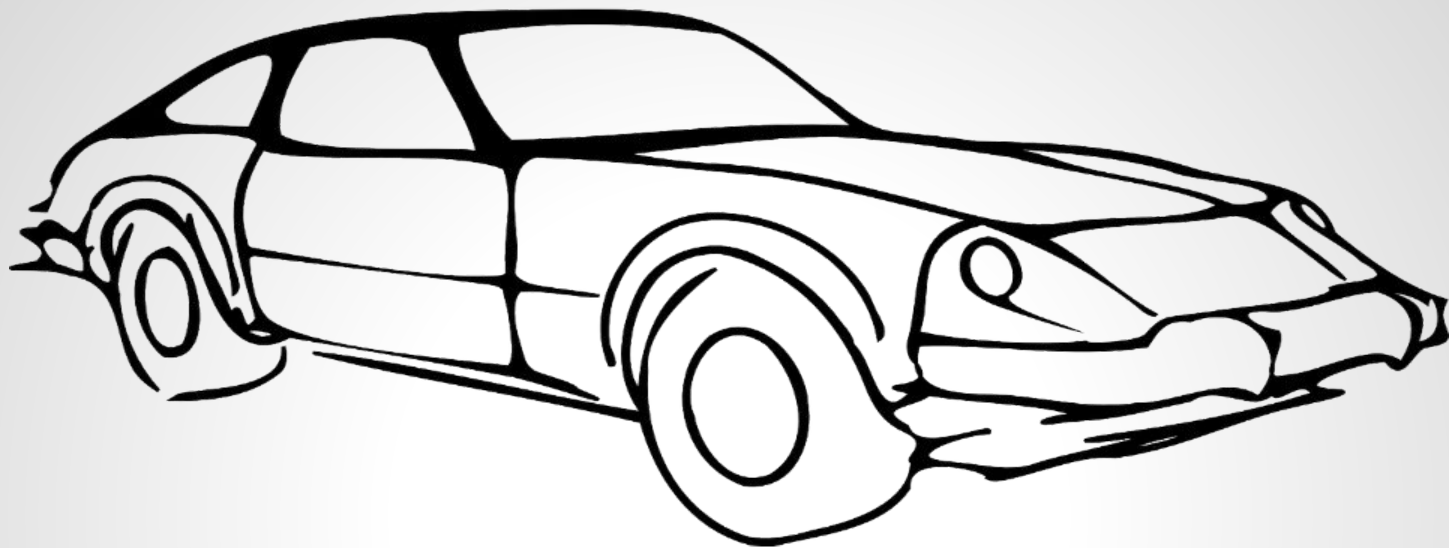


**LOR'n'JUG**

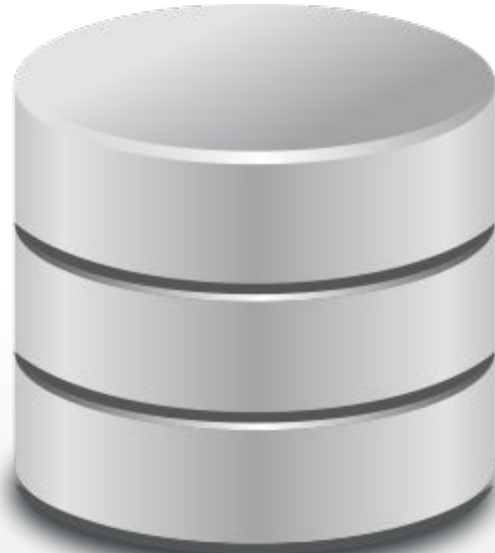


# Edgar Frank Codd

“Toute l'information dans la base de données est représentée d'une et une seule manière, à savoir par des valeurs dans des champs de colonnes de tables.”



# **Bases de données relationnelles**





**A**tomicity

**C**onsistency

**I**solation

**D**urability

# **A**tomicity

TOUT OU RIEN





Compte A



$x$



Compte B

1.  $\text{Compte A} - x$
2.  $\text{Compte B} + x$



Compte A



$x$



Compte B

1.  $\text{Compte A} - x$
2.  $\text{Compte B} + x$





Compte A



$x$



Compte B

1. Compte A -  $x$
2. Compte B +  $x$





Compte A



$x$



Compte B

TRANSACTION

1.  $\text{Compte A} - x$
2.  $\text{Compte B} + x$



**C**onsistency

TOUJOURS VALIDE

Nom	Age
Bobby	34
34	Bobby



**I**solation

**TRANSACTIONS  
INDEPENDANTES**

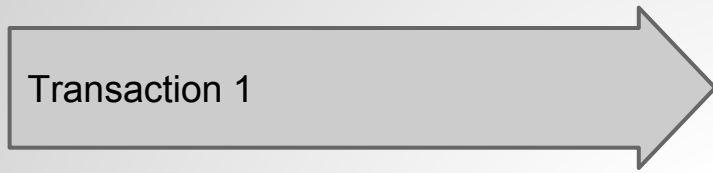
Transaction 1 (A vire 10 à B)

Transaction 2 (B vire 5 à A)

Temps

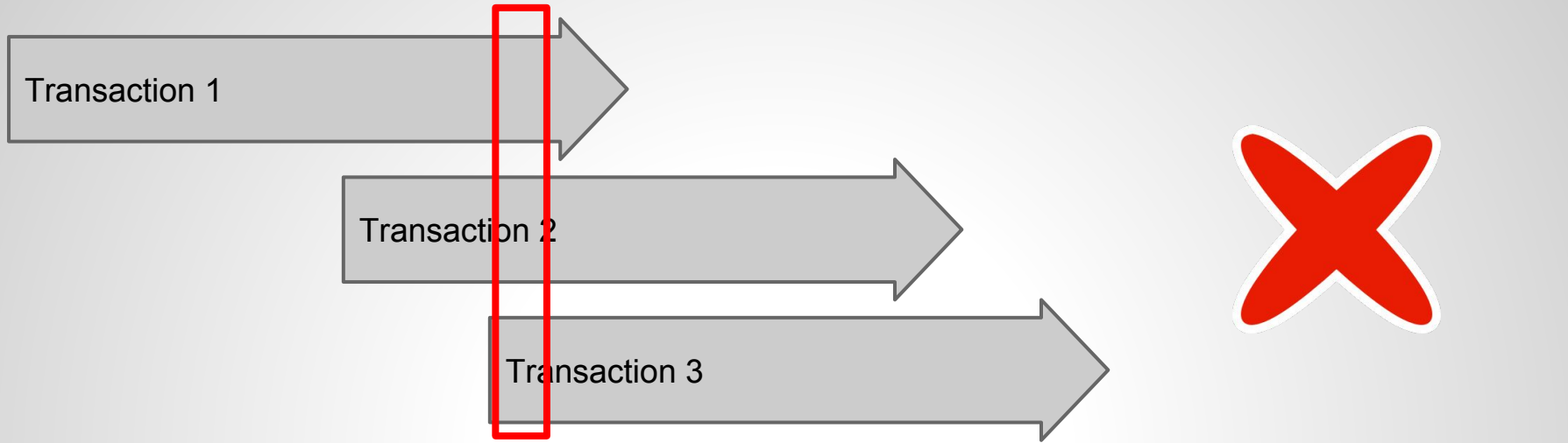
		account A	account B
		10	10
T1	A - 10	0	10
T2	B - 5	0	5
T2	A + 5	5	5
T1	B + 10	5	15



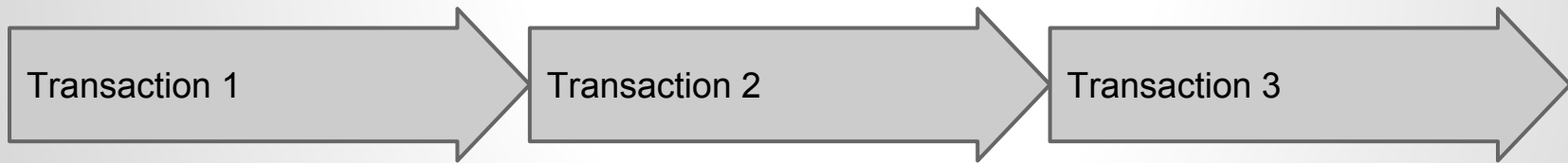


Temps 

		account A	account B
		10	10
T1	A - 10	0	10
T2	B - 5	0	5
T2	A + 5	5	5
T1	<del>B - 10</del>	5	5
T1	Rollback	10	5



Temps



# Durability



# LA VIE ETERNELLE



Valeur = 1234



Valeur = 1234



**A**tomicity

**C**onsistency

**I**solation

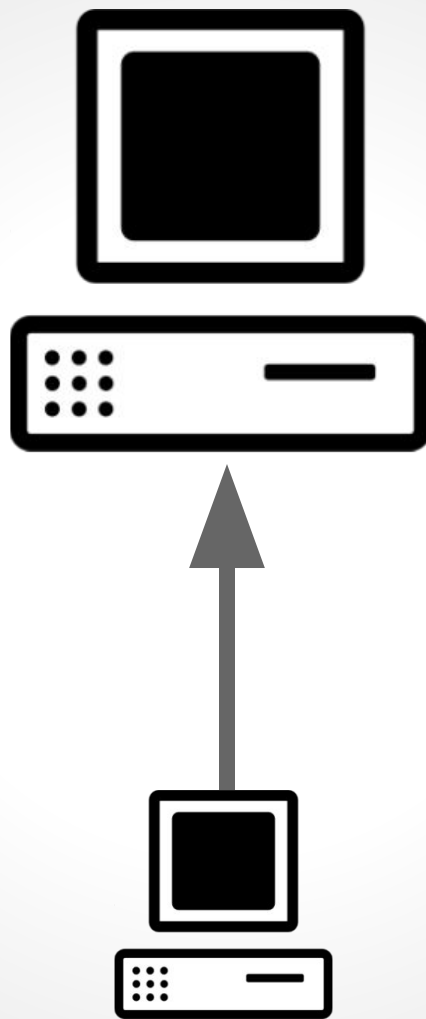
**D**urability

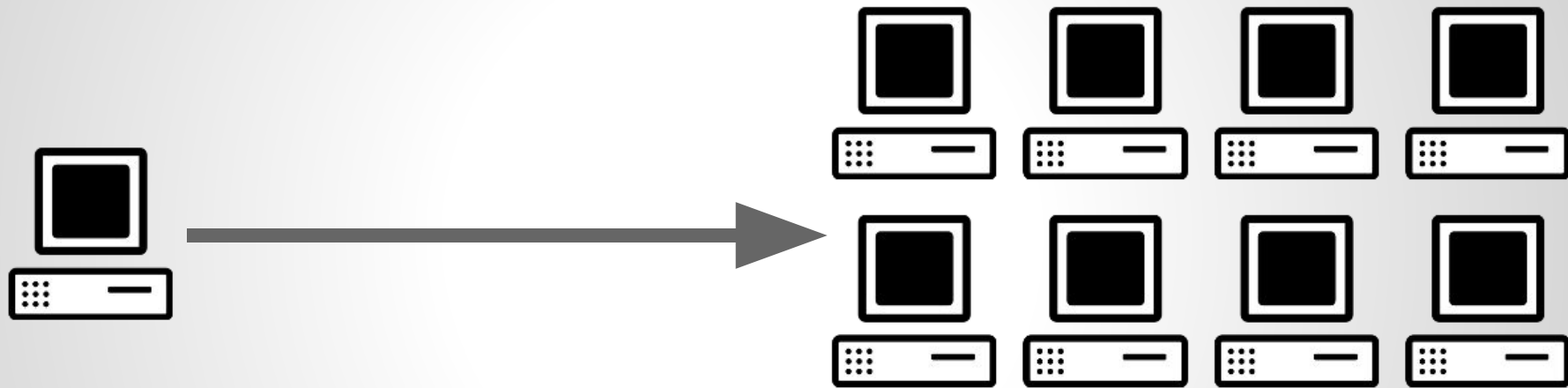
# SQL

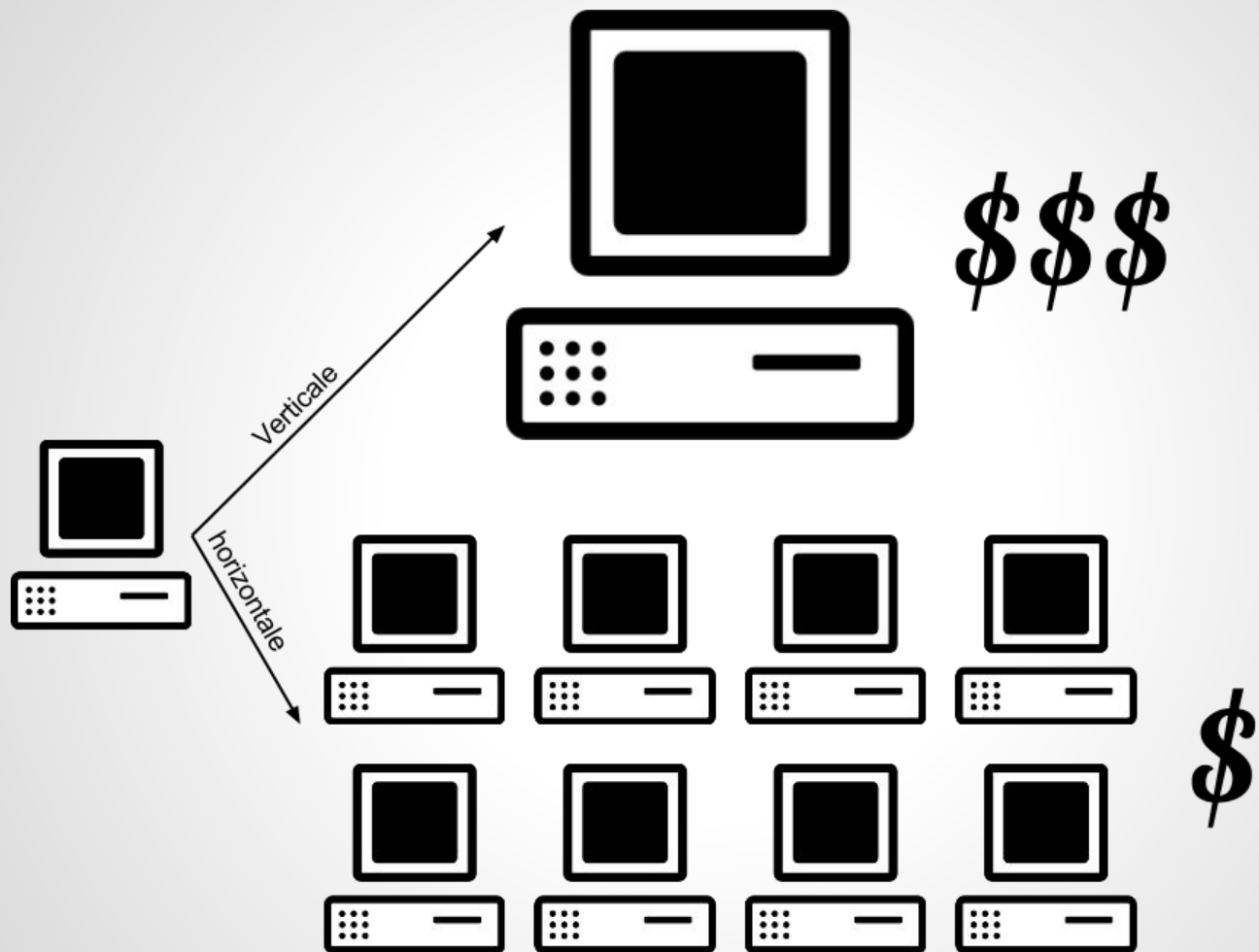


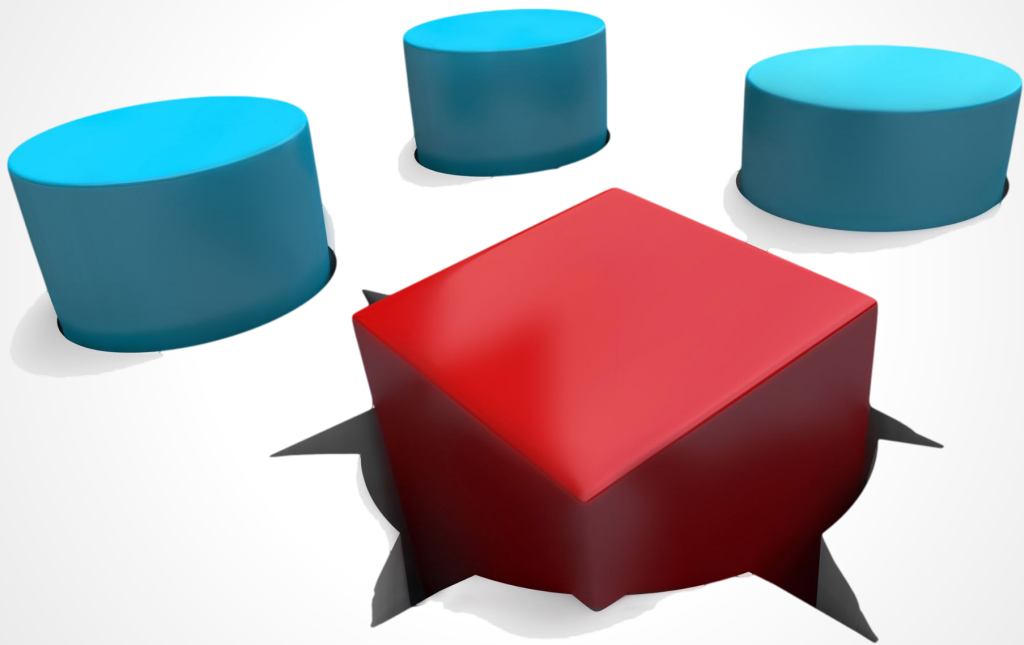






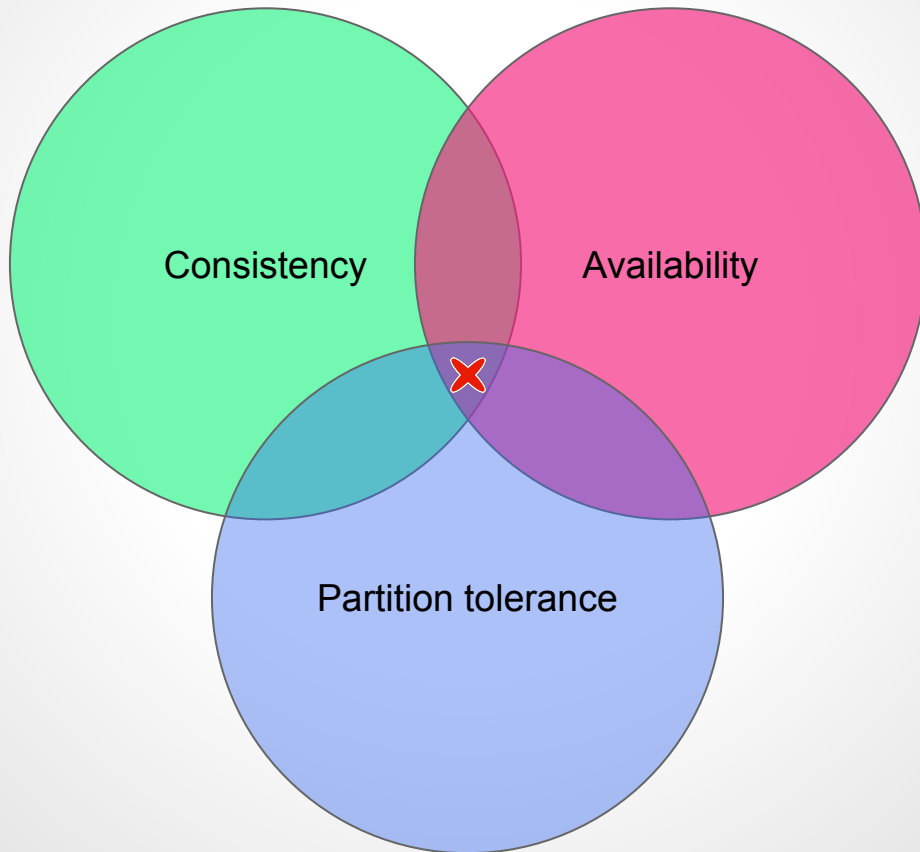






# Théorème de CAP

(Eric Brewer)



Google

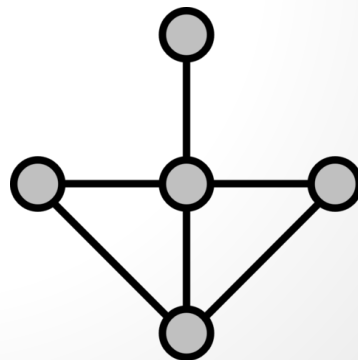
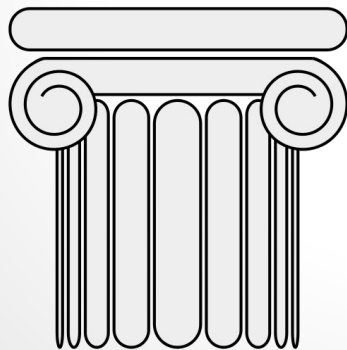
amazon

**#NoSQL**

# #NoSQL

Open-source  
Non relationnel  
Distribué  
Sans schéma





## Bon de commande

M. Bobby  
3 rue de Lorraine  
57000 Metz

Un truc très cher : 1 000 000€

Un truc moins cher : 500 €

Un truc : 5 €



### Personne

Civilité	Nom
M.	Bobby

### Adresse

Adress	Code postal	Ville
3 rue de Lorraine	57000	Metz

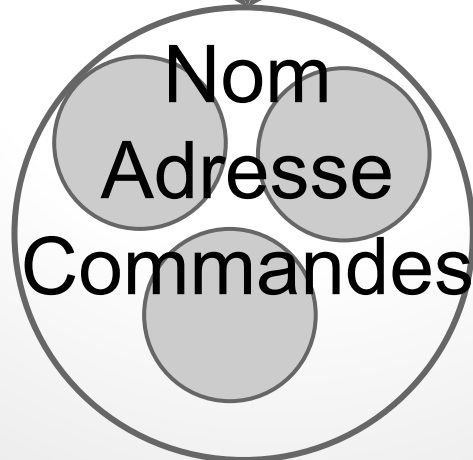
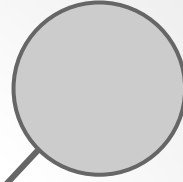
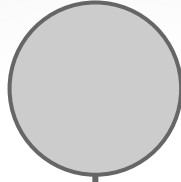
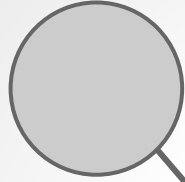
### Commandes

Libellé	Prix
Un truc très cher	1 000 000 €
Un truc moins cher	500 €
un truc	5 €

Nom

Adresse

Commandes



# Cas d'utilisation :

## Page de profil utilisateur

Bon de commande

M. Bobby  
3 rue de Lorraine  
57000 Metz

Un truc très cher : 1 000 000€

Un truc moins cher : 500 €

Un truc : 5 €



```
1.  {  
2.    name : Bobby,  
3.    adress : {  
4.      city : "Paris"  
5.    }  
6.    orders : [  
7.      {  
8.        items : [{ product : "banana"}]  
9.      }  
10.   ]  
11. }
```





# Cas d'utilisation : Administration des commandes

Bon de commande

M. Bobby  
3 rue de Lorraine  
57000 Metz

Un truc très cher : 1 000 000€

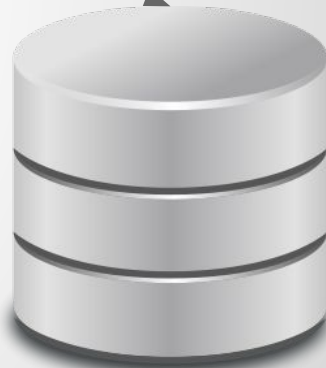
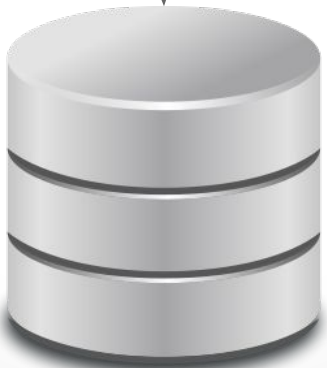
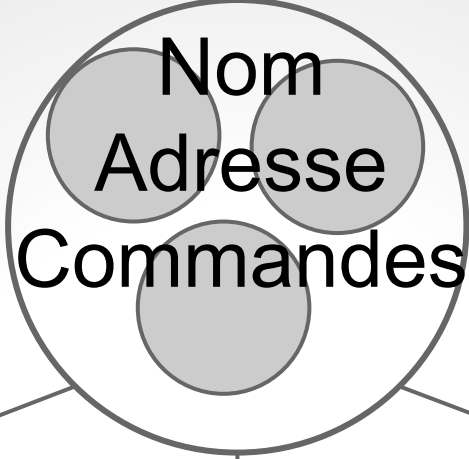
Un truc moins cher : 500 €

Un truc : 5 €

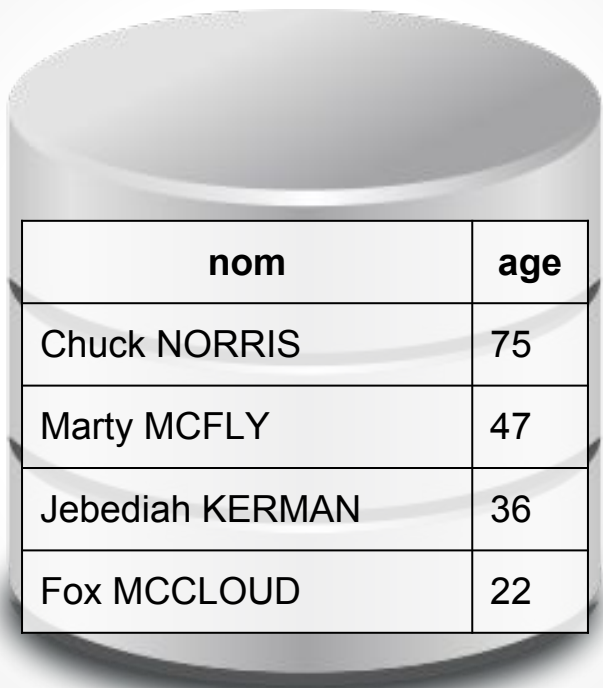


```
1.  orders : [  
2.    {  
3.      items : [{ product : "banana"}]  
4.      user : {  
5.        name : Bobby,  
6.        adress : {  
7.          city : "Paris"  
8.        }  
9.      }  
10.    }  
11.  ]
```

Exemple : [Twitter API](#)



# Sharding



nom	age
Chuck NORRIS	75
Marty MCFLY	47
Jebediah KERMAN	36
Fox MCCLOUD	22

# Sharding

**30 ans**

nom	age
Fox MCCLOUD	22

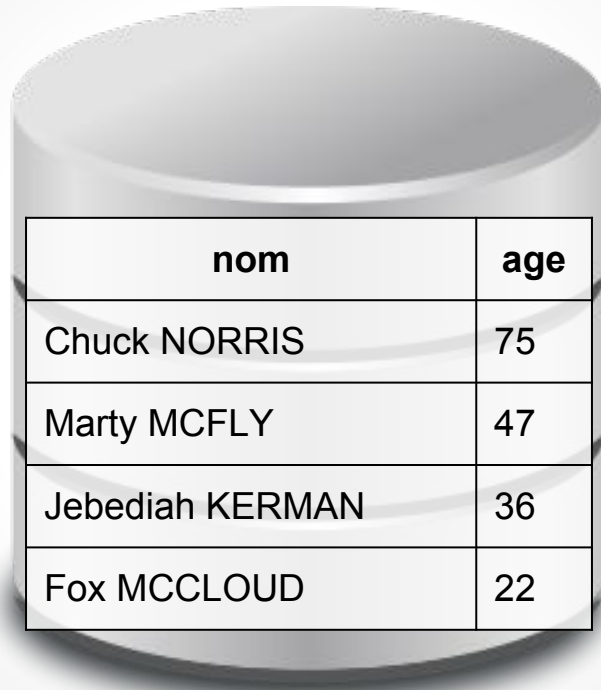
**30 ans --> 50 ans**

nom	age
Marty MCFLY	47
Jebediah KERMAN	36

**50 ans**

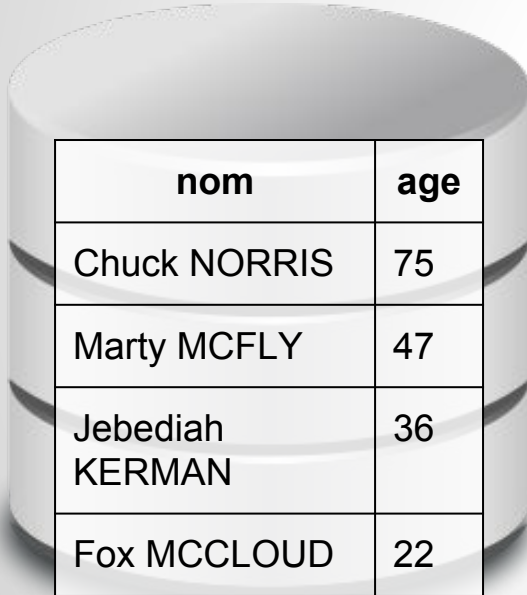
nom	age
Chuck NORRIS	75

# Replication

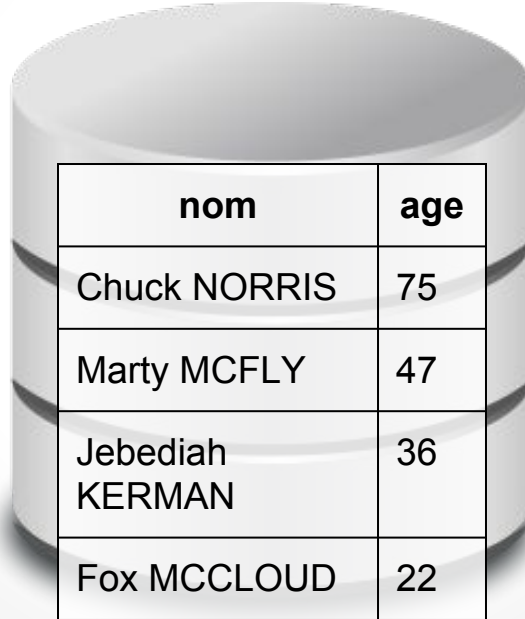


nom	age
Chuck NORRIS	75
Marty MCFLY	47
Jebediah KERMAN	36
Fox MCCLOUD	22

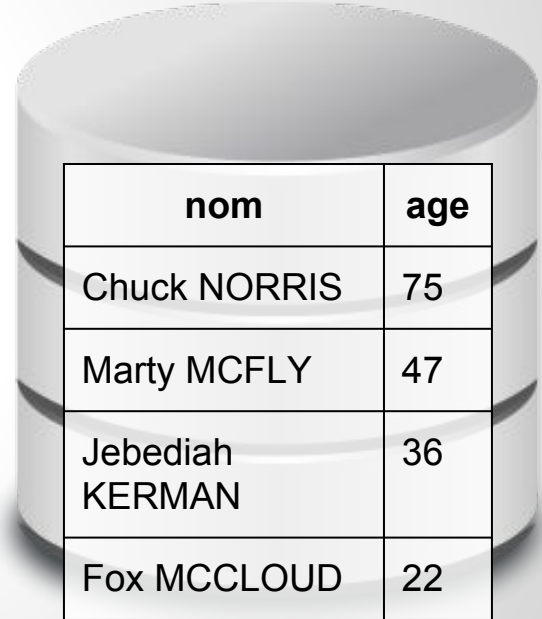
# Réplication



nom	age
Chuck NORRIS	75
Marty MCFLY	47
Jebediah KERMAN	36
Fox MCCLOUD	22

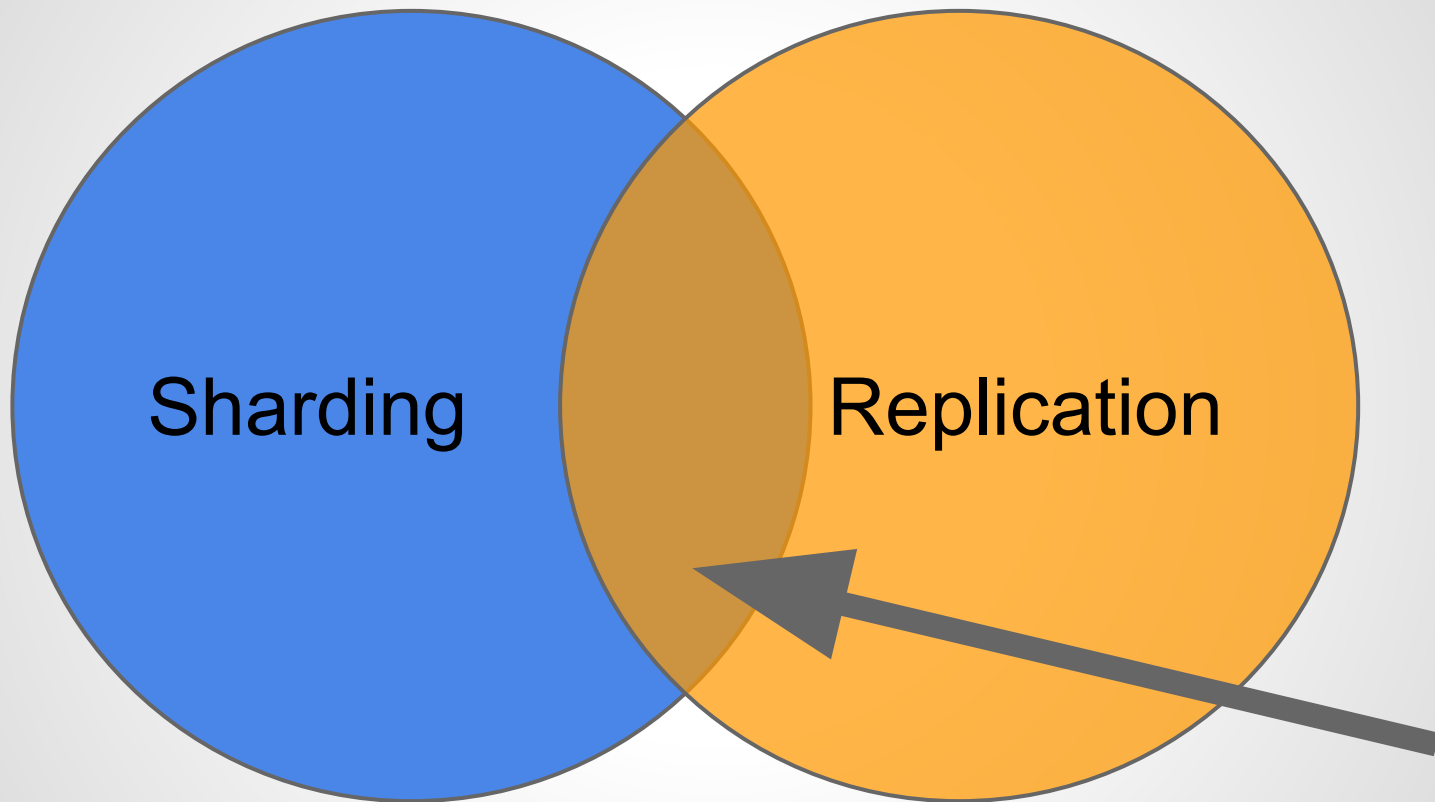


nom	age
Chuck NORRIS	75
Marty MCFLY	47
Jebediah KERMAN	36
Fox MCCLOUD	22

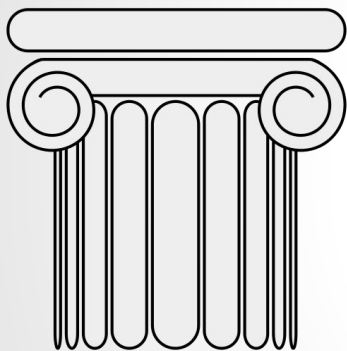


nom	age
Chuck NORRIS	75
Marty MCFLY	47
Jebediah KERMAN	36
Fox MCCLOUD	22















*Whatever*

...



?

12345



67890



98765

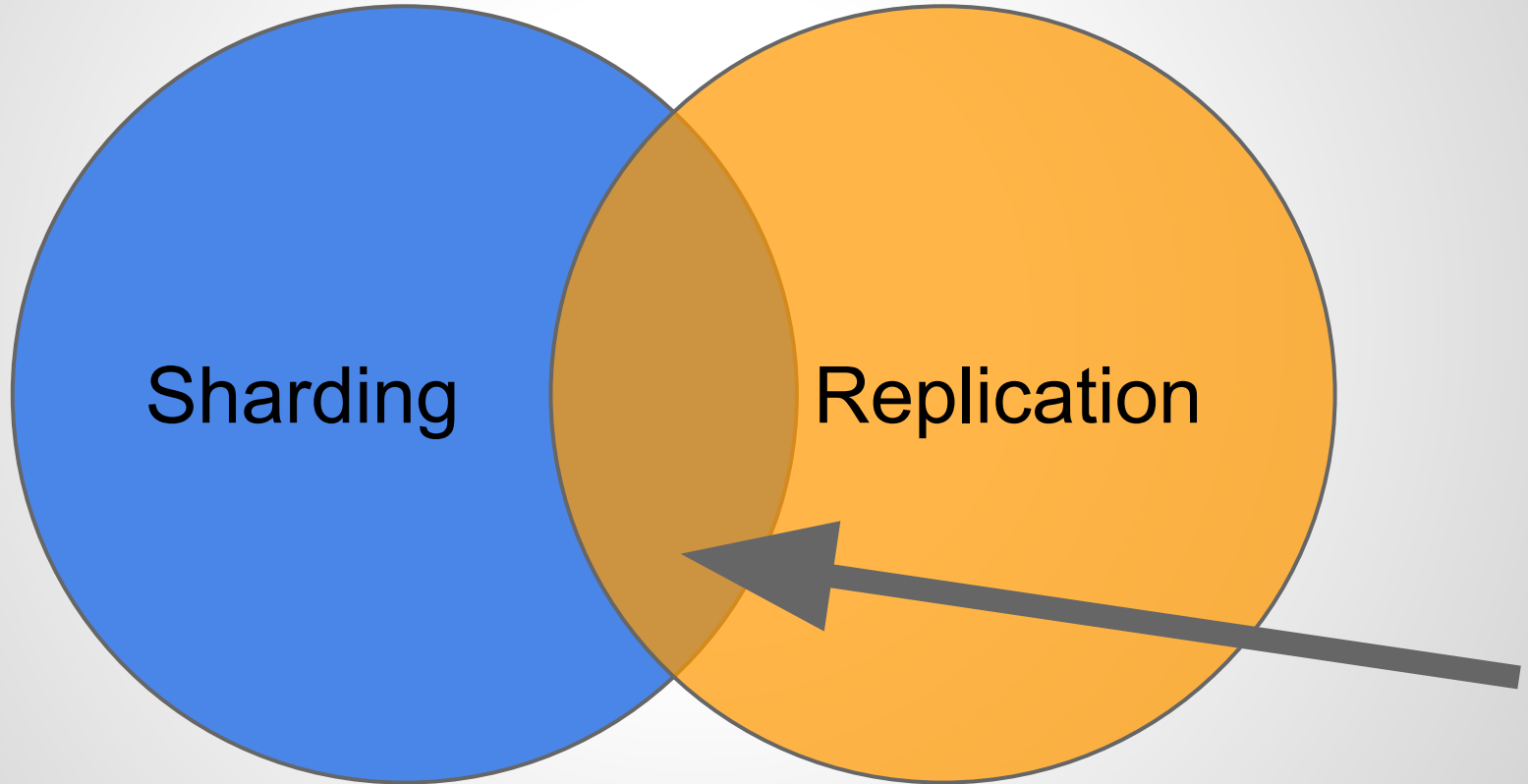


43210

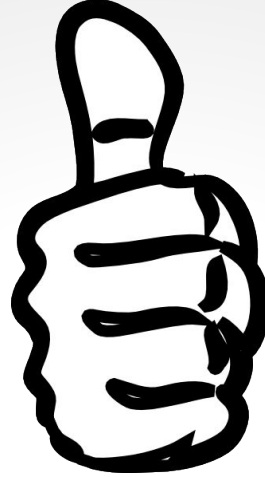




# Mise à l'échelle





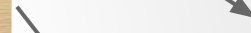
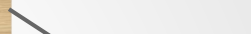


Information  
de session

Profil  
utilisateur

Panier  
shopping





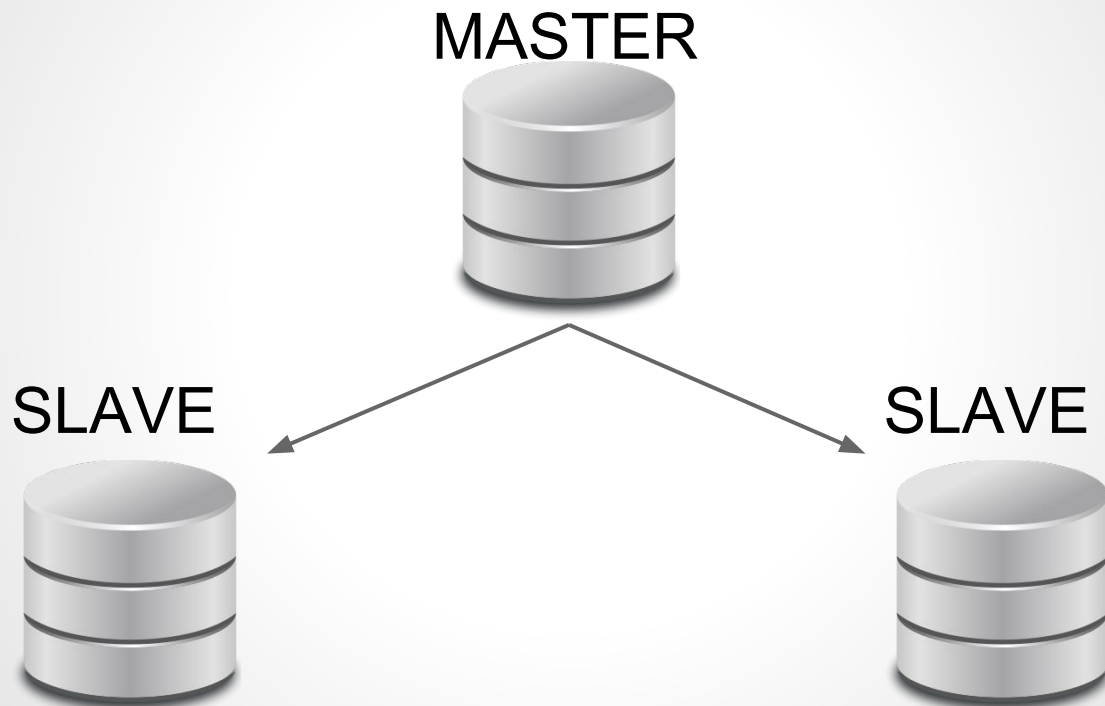


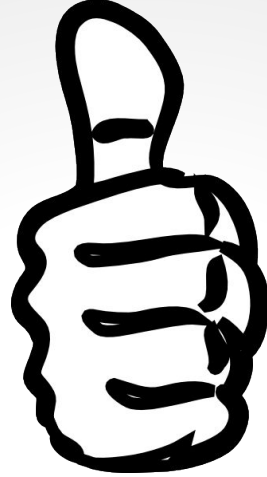
# Requêtes



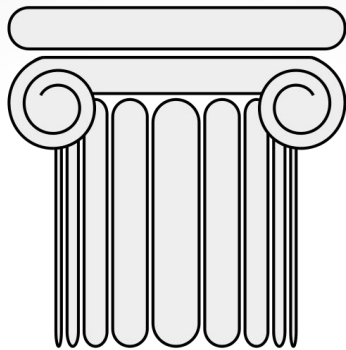


# Mise à l'échelle

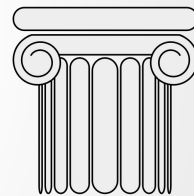
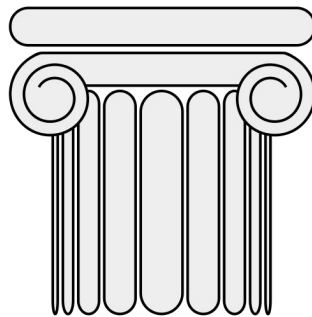
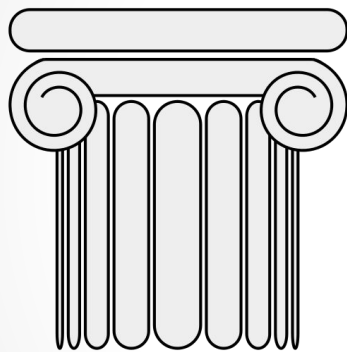
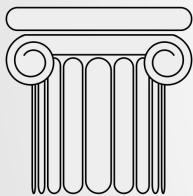
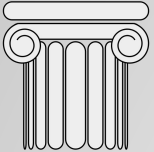




Logger



APACHE  
**HBASE**





Id	Nom	Prénom	Ville	Age	Hobby
1	McFerrin	Bobby	New York	67	Music
2	Johnson	Jack	California	42	Music
3	McFly	Marty	Hill Valley	30	Hoverboard

Id	Nom	Prénom	Ville	Age	Hobby	Profession
1	McFerrin	Bobby	New York	67	Music	Musicien
2	Johnson	Jack	California	42	Music	
3	McFly	Marty	Hill Valley	30	Hoverboard	

Id	Nom	Prénom	Ville	Age	Hobby	Profession
1	McFerrin	Bobby	New York	67	Music	Musicien
2	Johnson	Jack	California	42	Music	
3	McFly	Marty	Hill Valley	30	Hoverboard	

Trouvez les utilisateurs qui sont de la ville Hill Valley qui font du hoverboard

Id	Nom	Prénom	Ville	Age	Hobby	Profession
1	McFerrin	Bobby	New York	67	Music	Musicien
2	Johnson	Jack	California	42	Music	
3	McFly	Marty	Hill Valley	30	Hoverboard	

Id	Nom	Prénom	Ville	Age	Hobby	Profession
1	McFerrin	Bobby	New York	67	Music	Musicien
2	Johnson	Jack	California	42	Music	
3	McFly	Marty	Hill Valley	30	Hoverboard	

Id	Nom	Prénom	Ville	Age	Hobby	Profession
1	McFerrin	Bobby	New York	67	Music	Musicien
2	Johnson	Jack	California	42	Music	
3	McFly	Marty	Hill Valley	30	Hoverboard	

1:McFerrin;2:Johnson;3:McFly

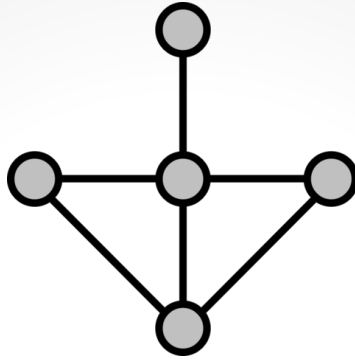
1:Bobby;2:Jack;3:Marty

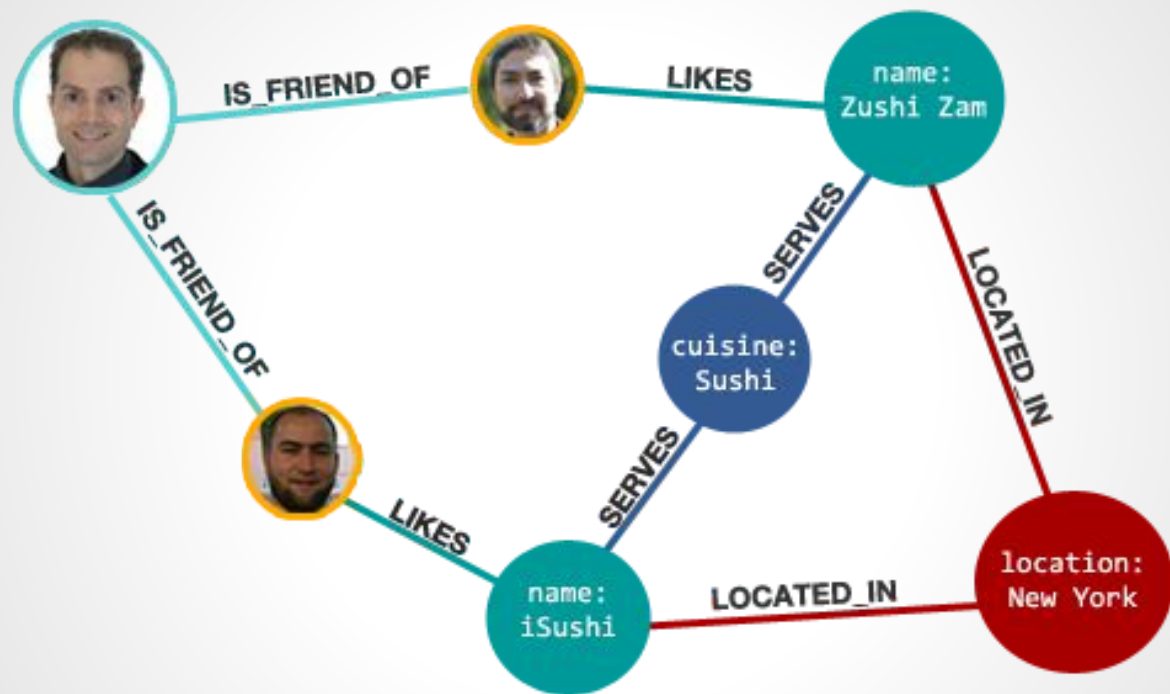
1:New York;2:California;3:Hill Valley

1:67;2:42;3:30

1:Music;2:Music;3:Hoverboard

1:Musicien





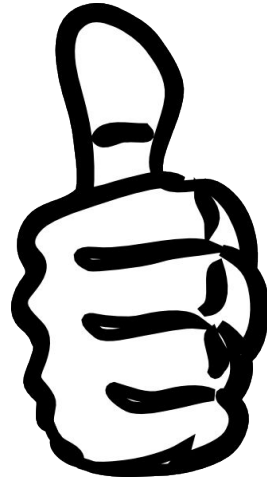


```
1. Node martin = graphDb.createNode();  
2. martin.setProperty("name","Martin");  
3.  
4. Node iSushi = graphDb.createNode();  
5. iSushi.setProperty("name","iSushi");  
6.  
7. martin.createRelationshipTo(iSushi,"LIKE");
```

# Cohérence



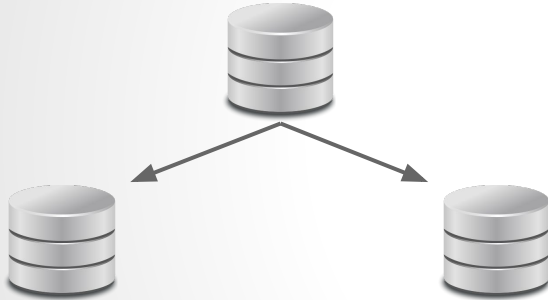
# Transaction



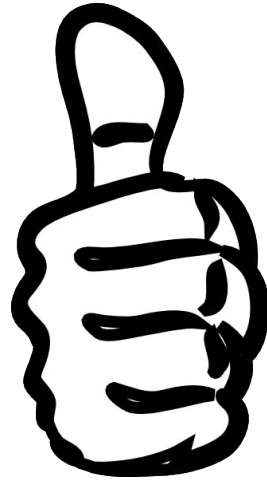
```
1. Transaction transaction =  
   database.beginTx();  
2. try {  
3.     Node node = database.createNode();  
4.     node.setProperty("name", "Martin");  
5.     node.setProperty("type", "Geek");  
6.     transaction.success();  
7. } finally {  
8.     transaction.finish()  
9. }
```



# Availability (Disponibilité)



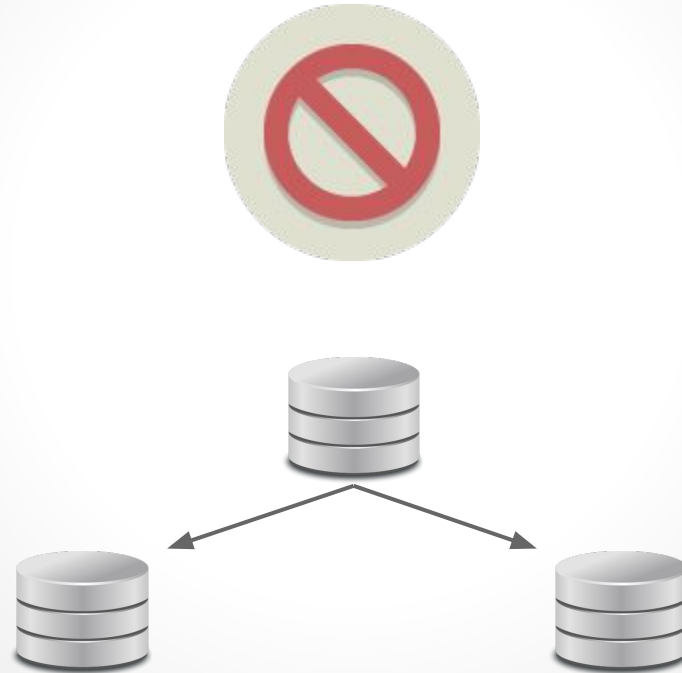
# Requêtes



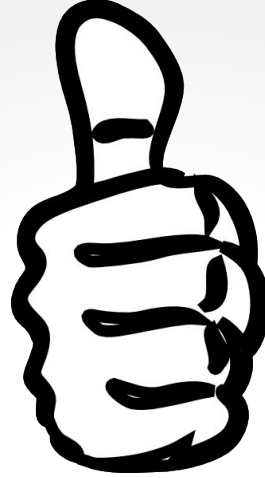
```
1. Index<Node> nodeIndex =  
2. graphDb.index().forNodes("nodes");  
3.  
4. Node node =nodeIndex.get("name", "Martin");  
5. relation = node.getRelationship(LIKES);  
6.  
7. [...]
```



# Mise à l'échelle







Données  
connectées

Recommandation



Productivité des développeurs

Performance

Polyglotte

Encapsuler

?

TP : MongoDB, Redis

Projet

Cours