NoSQL Data stores

Dominic Duggan

Stevens Institute of Technology

Based on materials by A. Haeberlen, Z. Ives, E.Meijer

1

1

DATABASE REVIEW

2

2

- SQL and Natural Join

```
select name, phone
from people
join phonenumbers
on
people.id=phonenumbers.id
```

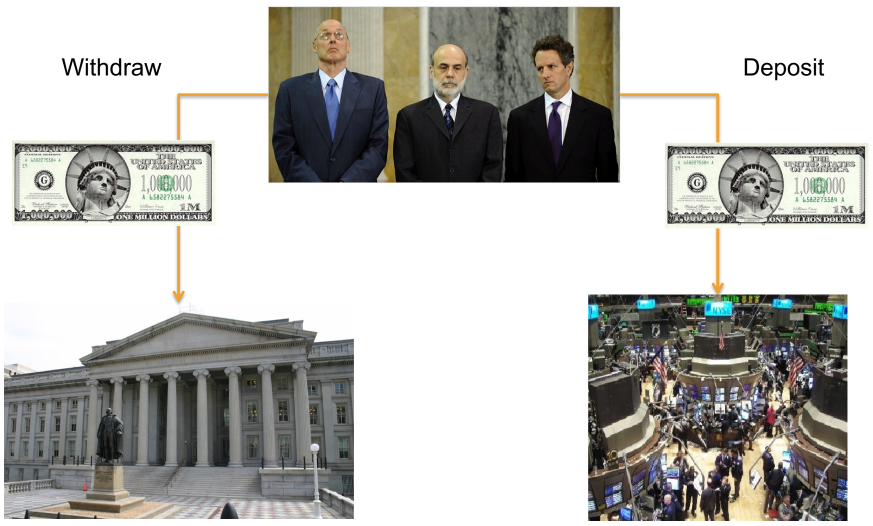
Contacts		
Id	Name	Phone
1	Joe	5532
1	Joe	2234
1	Joe	3211
2	Jane	3421
3	Chris	2341
3	Chris	6655

PhoneNumbers		
PhoneID	Id	Phone
1	1	5532
2	1	2234
3	1	3211
4	2	3421
5	3	2341
6	3	6655

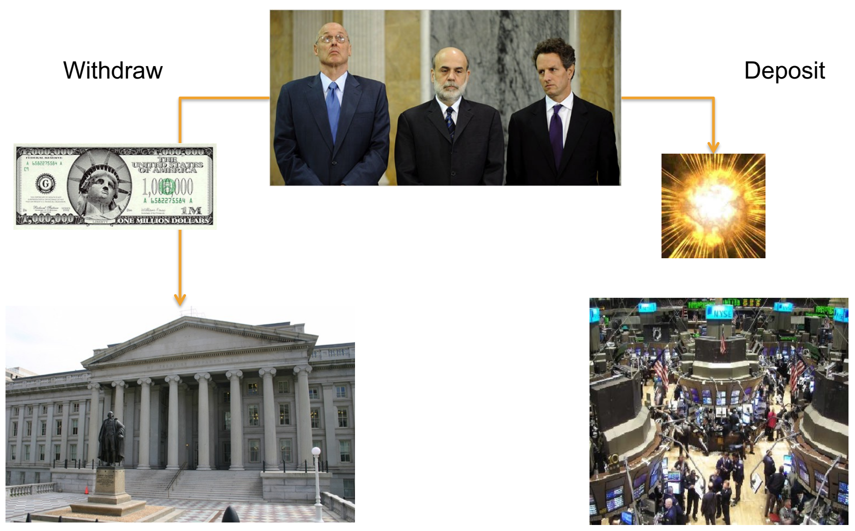
People	
Id	Name
1	Joe
2	Jane
3	Chris

Name	Phone
Joe	5532
Joe	2234
Joe	3211
Jane	3421
Chris	2341
Chris	6655

Transactions



Transactions



5

5

Transactions

Withdraw
Deposit

6

6

Transactional Operations

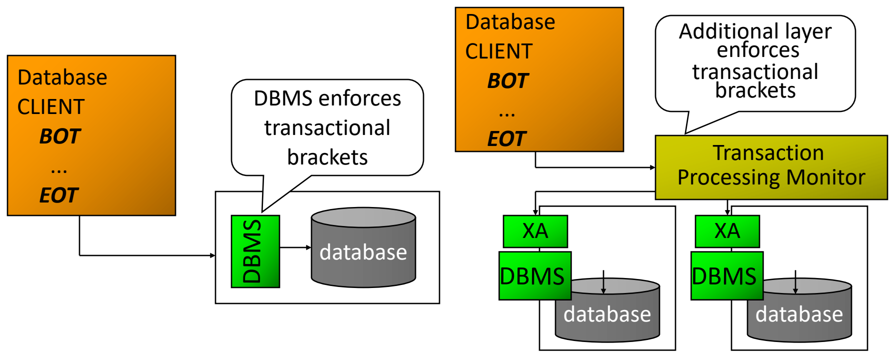
```
Begin Transaction
   AmtOwed = Amazon.Read (CustID);
   Balance = Paypal.Read (AccountID);
   if (AmtOwed ≤ Balance) {
      Paypal.Withdraw (AccountID, AmtOwed);
      Amazon.Deposit (CustID, AmtOwed);
   } else {
      Abort Transaction;
   }
End Transaction
```

7

7

Distributed Databases

- Transactions and atomic commitment



Database CLIENT
BOT
…
EOT
DBMS enforces transactional brackets
DBMS
database
Database CLIENT
BOT
…
EOT
Additional layer enforces transactional brackets
Transaction Processing Monitor
XA
DBMS
database
XA
DBMS
database

8

8

Relational Database Summary

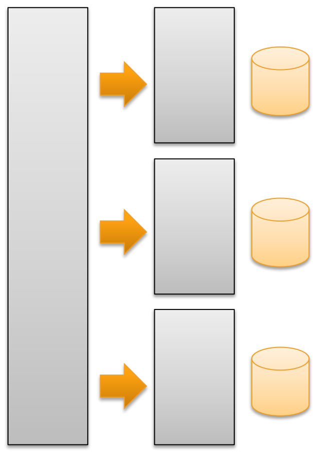
- Database Schema
  - Normalized for efficiency
- SQL for ad-hoc queries
- Transactional updates
  - Atomic
  - Consistent
  - Isolated
  - Durable

9

9

Challenge: Big Data^{TM}

- Historical approach: *vertical scaling*
  - Limited
- Modern approach: *horizontal scaling*
  - *Sharding*
  - Azure: Federated SQL databases
  - Applications see data partitioning
  - No joins across partitions



10

10

SQL vs NoSQL

Relational

- Database Schema
  - Business data model
- SQL for ad-hoc queries
- ACID properties
  - Atomic
  - Consistent
  - Isolated
  - Durable

NoSQL

- Unstructured
  - Web server logs
- Map-Reduce
- BASE properties
  - Basically Available
  - Soft state
  - Eventually consistent

Fault-tolerance

So, how do I query the database?
It's not a database. It's a key-value store!
Ok, it's not a database. How do I query it?
You write a distributed map reduce function in Erlang!
Did you just tell me to go f k myself?
I believe I did, Bob.

NOSQL DATA MODELS

13

13

NoSQL taxonomy

- Key-Value stores (DHT)
- Column stores
- Document stores

14

14

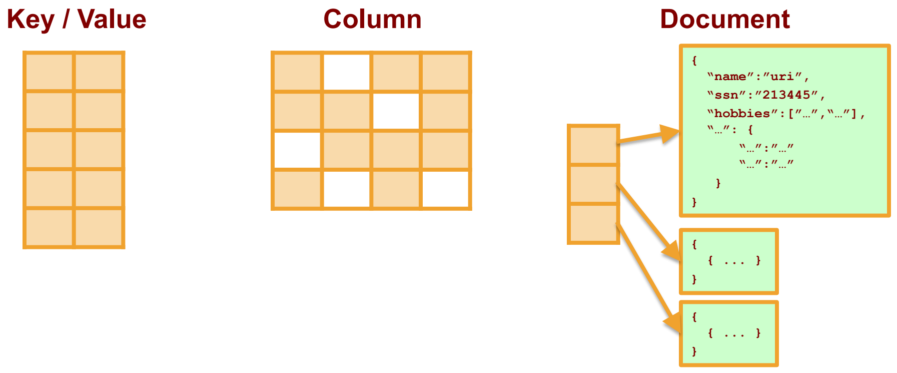
NoSQL taxonomy

- Key-Value stores (DHT) Amazon Dynamo
- Column stores Google Bigtable, Cassandra
- Document stores CouchDB, MongoDB, SimpleDB

15

15

A Universe of Data Models



16

16

Key/Value

- Have the key? Get the value
  - Map/Reduce (sometimes)
  - Good for
    - cache aside (e.g. Hibernate 2nd level cache)
    - Simple, id based interactions (e.g. user profiles)
- In most cases, values are opaque

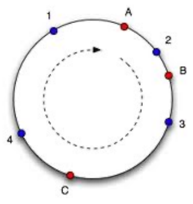
K1	V1
K2	V2
K3	V3
K4	V1

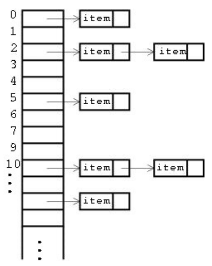
17

17

Key/Value

- Scaling out is relatively easy
  - just hash the keys
- Fixed vs. consistent hashing



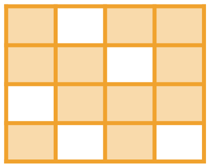


18

18

Column Based

- Mostly derived from Google’s BigTable papers
- One giant table of rows and columns
  - Column == pair (name and a value, sometimes timestamp)
  - Table is sparse:
    (#rows) × (#columns) ≥ (#values)

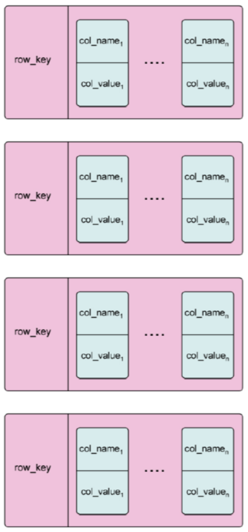


19

19

Column Based

- Query on row key
  - Or column value (aka secondary index)
- Good for a constantly changing, (albeit flat) domain model

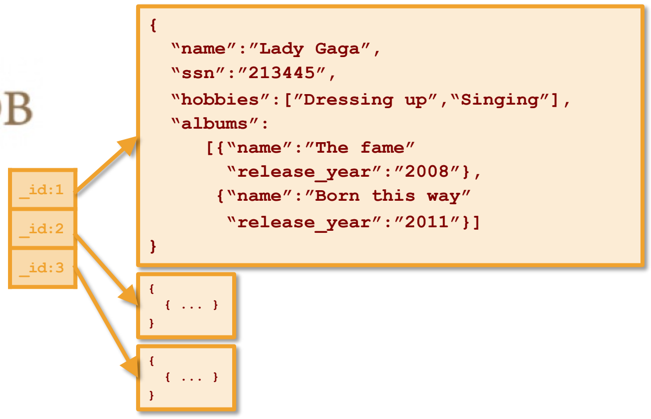


20

20

Document

- Think JSON (or BSON, or XML)



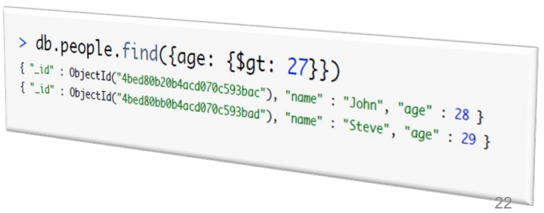
```
{
  "name":"Lady Gaga",
  "ssn":"213445",
  "hobbies":["Dressing up","Singing"],
  "albums":
     [{"name":"The fame"
       "release_year":"2008"},
      {"name":"Born this way"
       "release_year":"2011"}]
}
```

21

21

Document

- Model is not flat, data store is aware of it
  - Arrays, nested documents
- Better support for ad hoc queries
  - MongoDB excels at this
- Very intuitive model
- Flexible schema



```
> db.people.find({age: {$gt: 27}})
{ "_id" : ObjectId("4bed80b20b4acd070c593bac"), "name" : "John", "age" : 28 }
{ "_id" : ObjectId("4bed80bb0b4acd070c593bad"), "name" : "Steve", "age" : 29 }
```

22

22