NoSQL Data Modeling

Allen Wang

March 18, 2015

BD02

MARCH 18-19 ERworld 2015

ca technologies

Presenter Bio

- **Allen Wang**
- CA ERwin® Development Manager, joined CA Technologies in 2006. Currently, he is in charge of the CA ERwin engineering organization. Allen has been involved with CA ERwin since r8.0, just delivered r9.6 and the next release.
- Passion for Big Data innovation. Driving Big Data research project in cooperation with Tsinghua University. The project focuses on big data modeling, managing the schema and migrating the database between relational database and NoSQL.
- Teaching a Master Degree level Big Data and Cloud Computing course for FuDan University.
- CA CTE (Council for Technical Excellence) member

Agenda

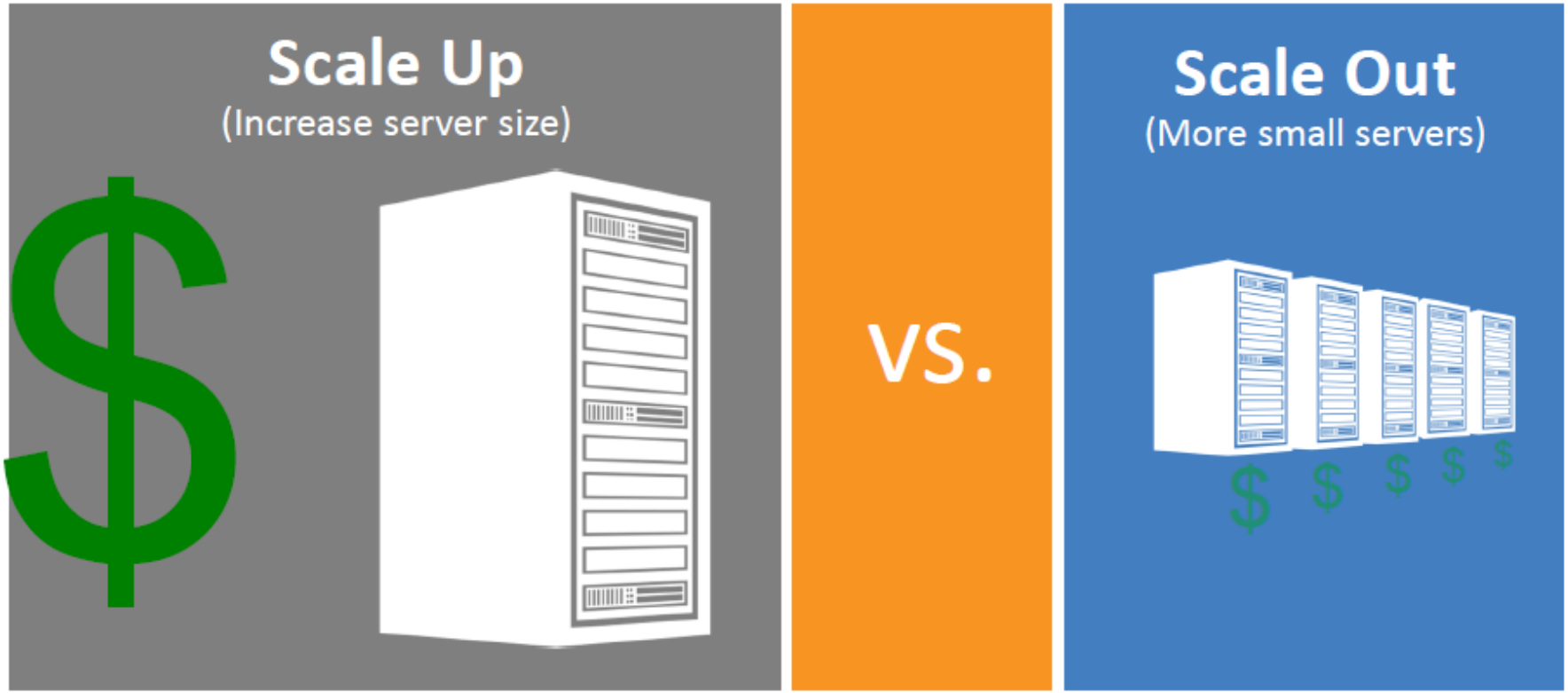
- NoSQL wave (Why)
- NoSQL modeling (How)
- CA ERwin Big Data management (What)
- A Real project
- Demo

NoSQL Wave

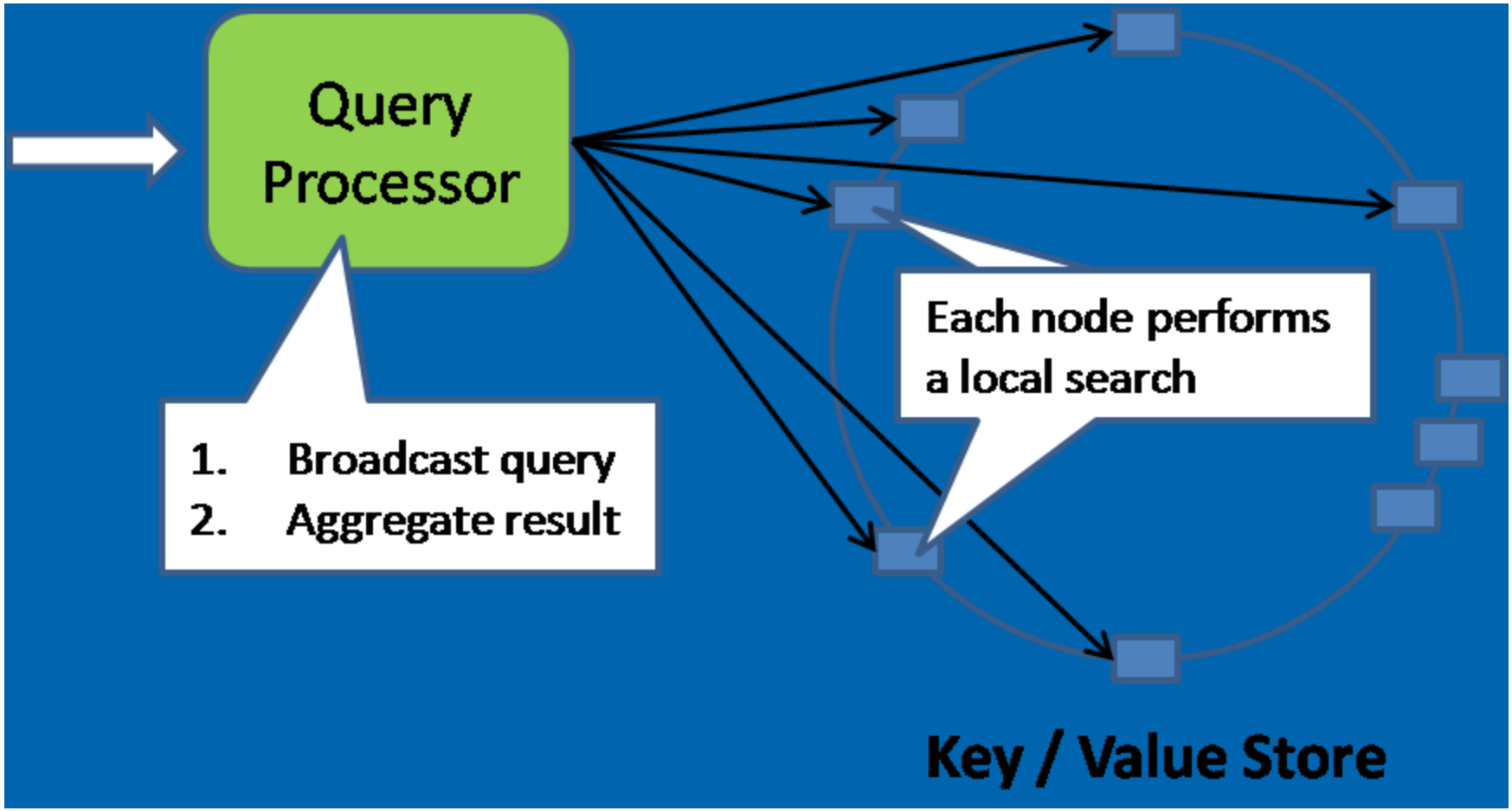
- Companies like Google, Amazon and LinkedIn need greater scalability & schema flexibility
- New databases are developed by developers, not database people
- Provided scale-out, but lost SQL
- Worked well at web startups because:
  - In some cases, use cases did not need ACID
    - Atomicity Consistency
    - Isolation Durability
  - Willing to handle exceptions at app level

Scale-Out: The Future of Databases

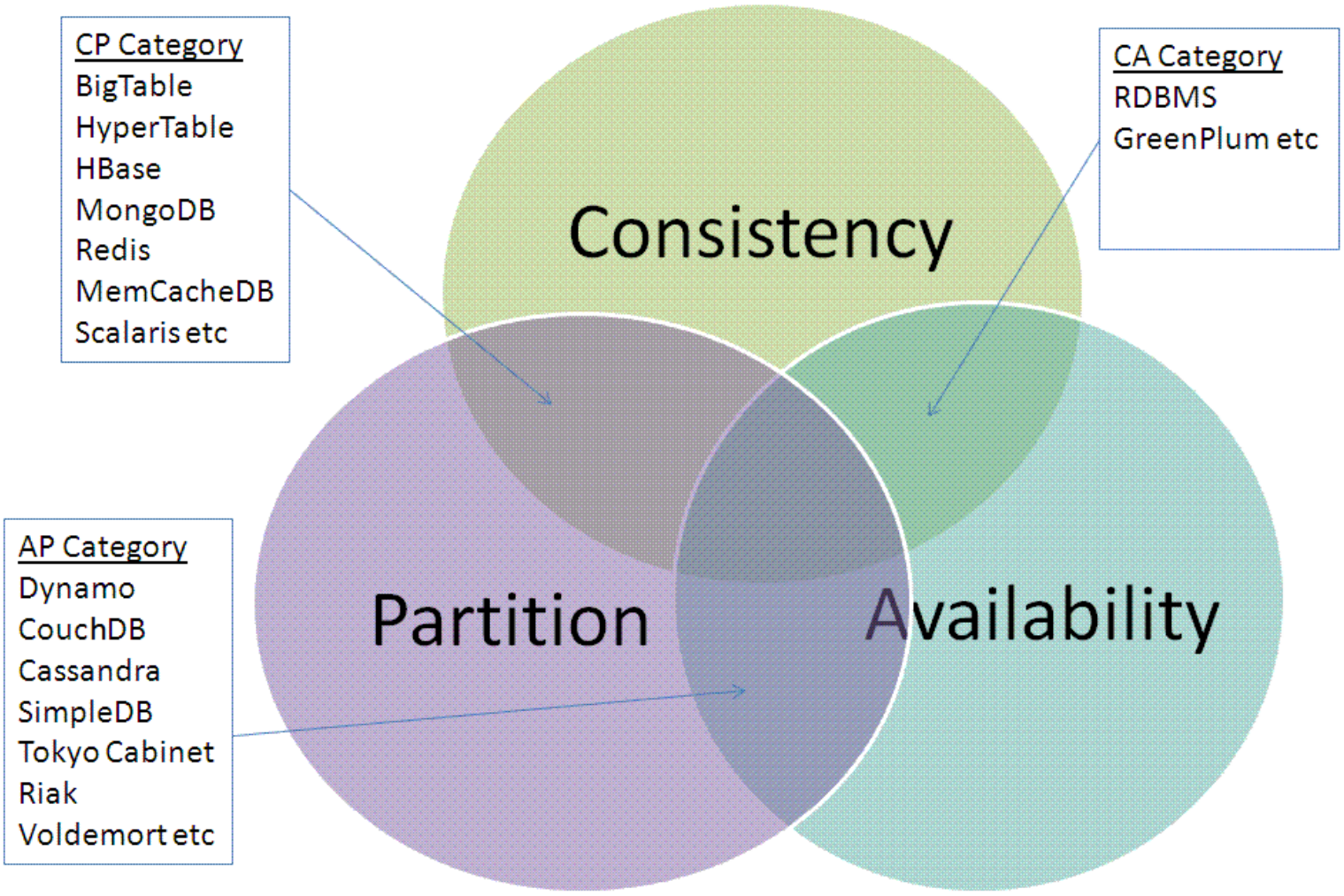
Dramatic improvement in price/performance



Horizontally Scalable



CAP
Consistency – Availability - Partition



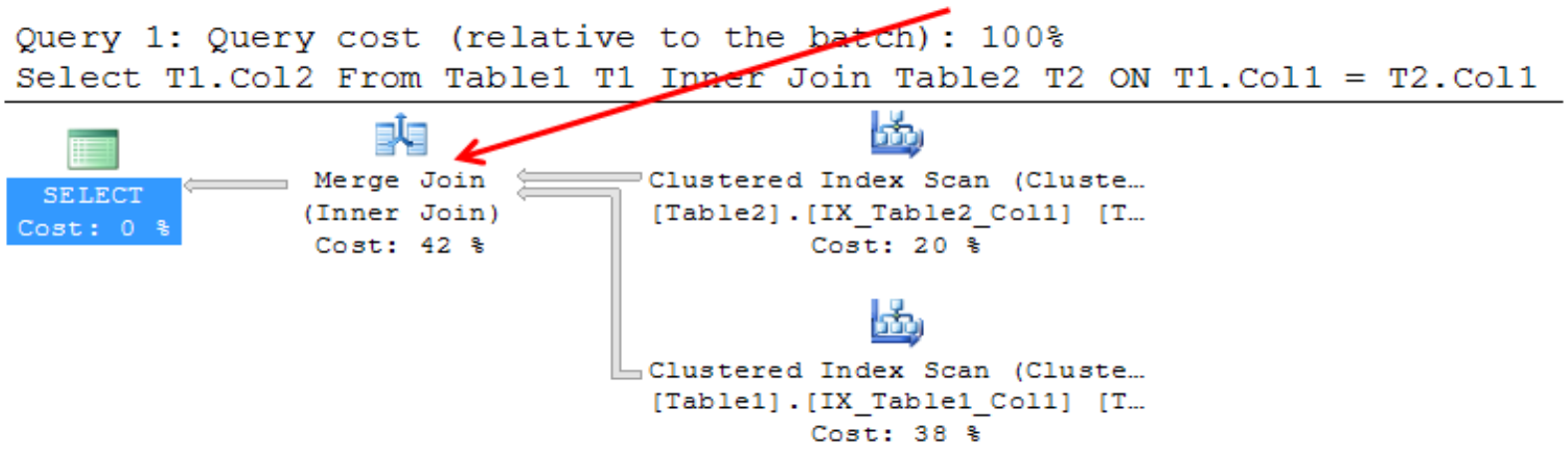
Traditional Database

- Normalization: 3NF
- ACID
  - Atomicity
  - Consistency
  - Isolation
  - Durability

PubID	Publisher	PubAddress
03-4472822	Random House	123 4th Street, New York
04-7733903	Wiley and Sons	45 Lincoln Blvd, Chicago
03-4859223	O'Reilly Press	77 Boston Ave, Cambridge
03-3920886	City Lights Books	99 Market, San Francisco

AuthorID	AuthorName	AuthorBDay
345-28-2938	Haile Selassie	14-Aug-92
392-48-9965	Joe Blow	14-Mar-15
454-22-4012	Sally Hemmings	12-Sept-70
663-59-1254	Hannah Arendt	12-Mar-06

ISBN	AuthorID	PubID	Date	Title
1-34532-482-1	345-28-2938	03-4472822	1990	Cold Fusion for Dummies
1-38482-995-1	392-48-9965	04-7733903	1985	Macrame and Straw Tying
2-35921-499-4	454-22-4012	03-4859223	1952	Fluid Dynamics of Aquaducts
1-38278-293-4	663-59-1254	03-3920886	1967	Beads, Baskets & Revolution



ca technologies

NoSQL Modeling

RDBMS VS NoSQL in Data Modeling

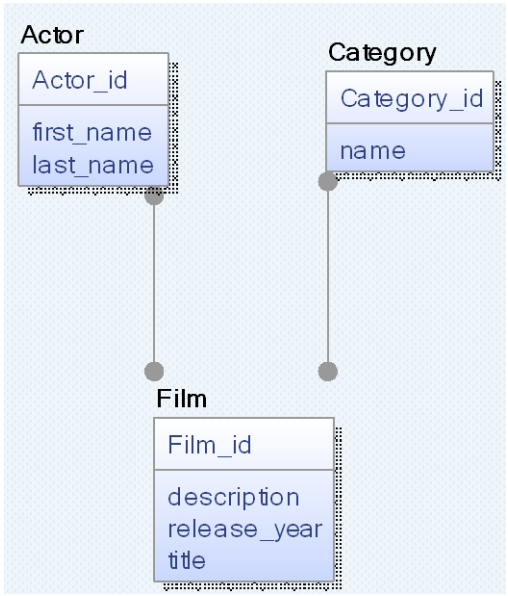
RDBMS	NoSQL
ER (Entity Relationship)	Four main types: document, column-oriented, key-value, and graph
Schema predefined	Schema-less
Vertical scaling	Horizontal scaling
ACID	Performance prioritized based on CAP principle

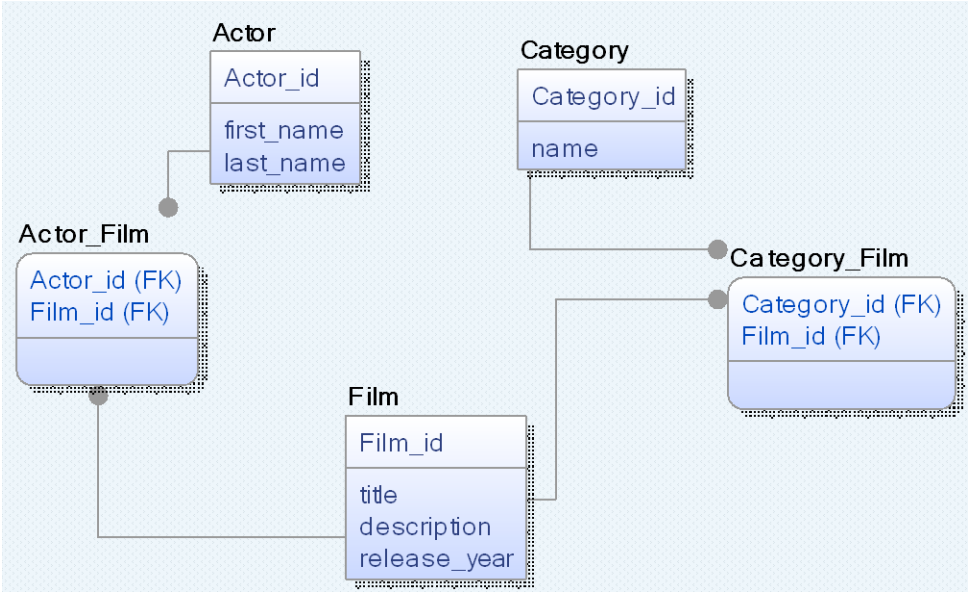
RDBMS vs. NoSQL in Data Modeling Cont.

Logical Model	RDBMS Model (Physical)	NoSQL Model (Physical)
Entity	Table	Document / Column Family / Graphic
Entity instance	Row	Collection / Row
Attribute Value	Column Value	Field Value
Domain	Data type	Data type
Relationship	Constraint	Reference, Embedded, Additional table,
Key Group	Index / constraint	Index, Additional table, Reference

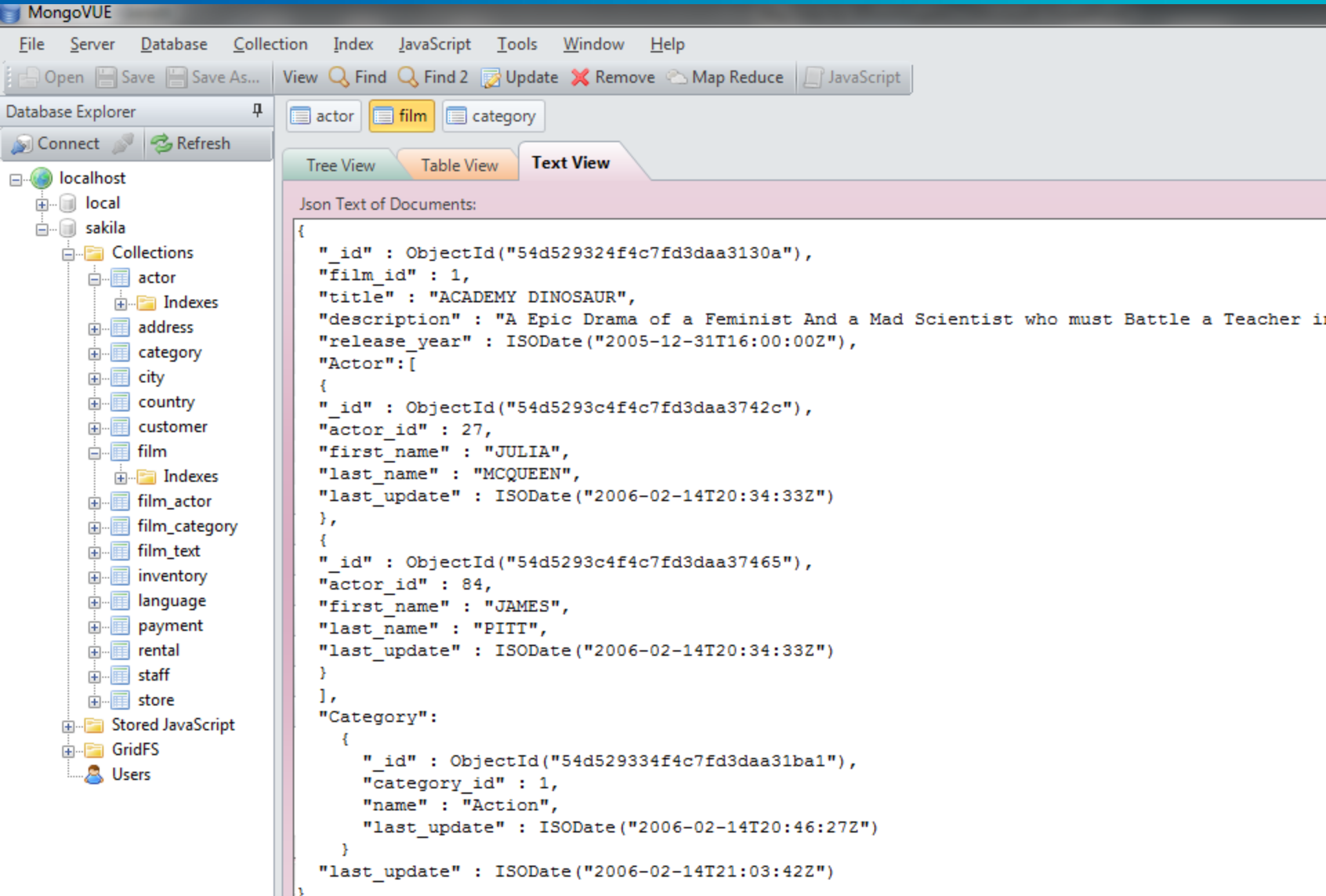
Example of NoSQL Four Types – ER Diagram

- **ER Diagram**
- Document oriented
- Column oriented
- Key-value pair
- Graphic





Example of NoSQL Four Types – Document



Example of NoSQL Four Types – Column Oriented

- ER Diagram
- Document oriented
- **Column oriented**
- Key-value pair
- Graphic

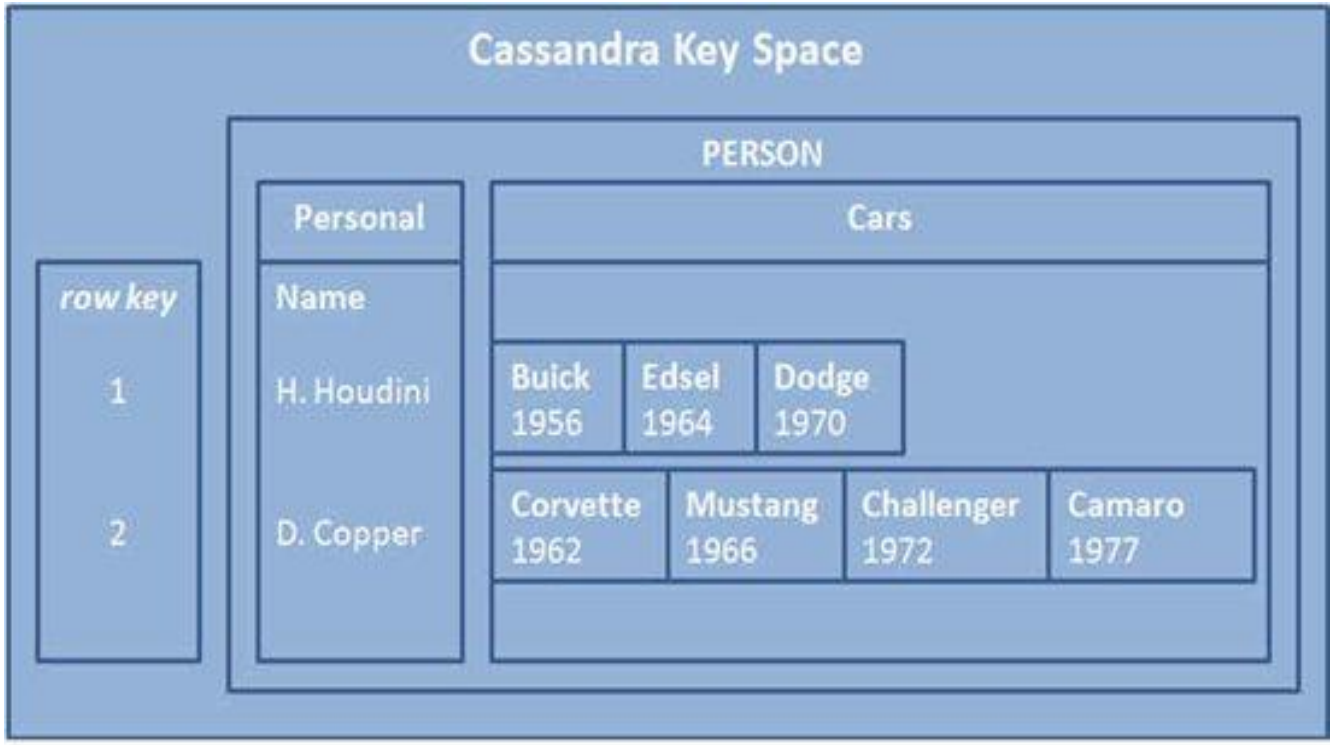


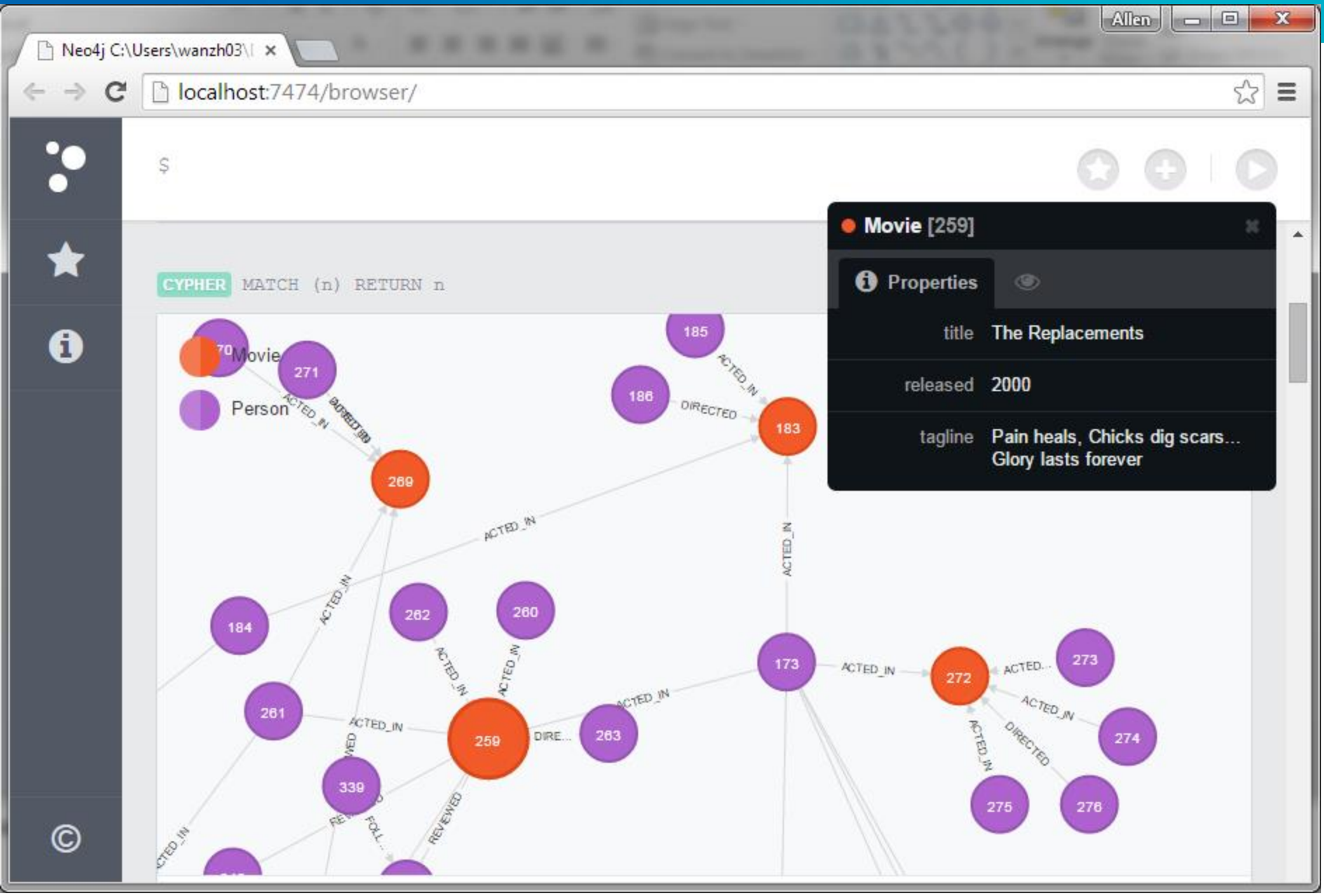
Figure 6 - Static/Dynamic Data

Example of NoSQL Four Types – Key-Value

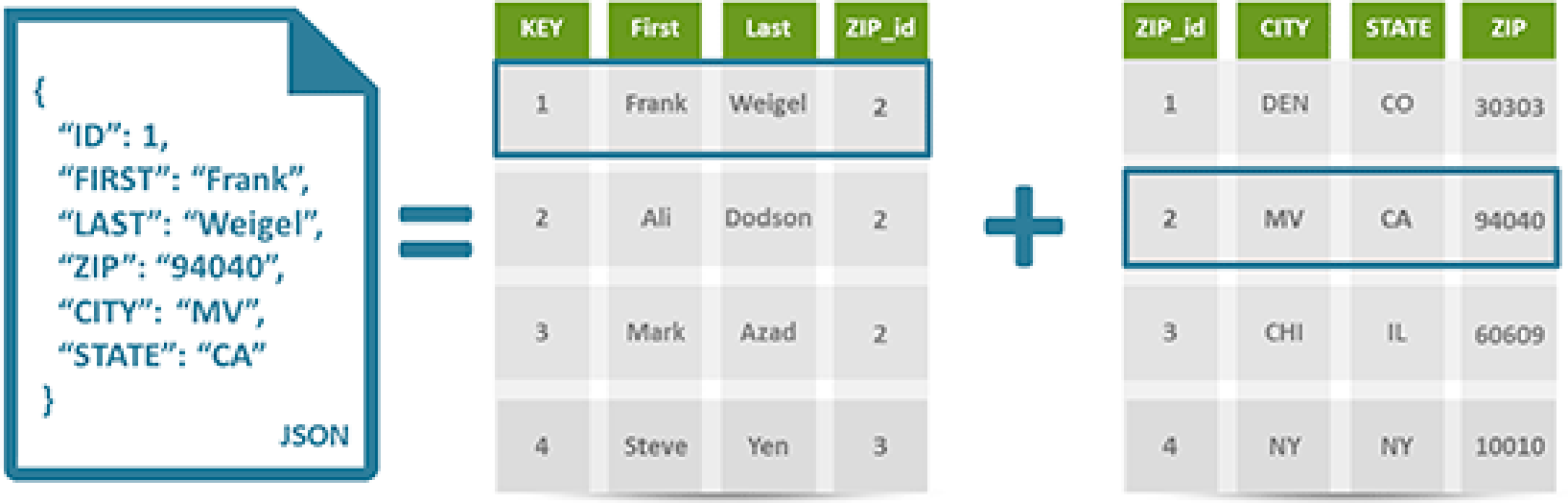
- ER diagram
- Document oriented
- Column oriented
- **Key-value pair**
- Graphic

Key	Value
“Film_id:1:title”	"ACADEMY DINOSAUR"
“Film_id:1:description”	"A Epic Drama of a Feminist And a Mad Scientist who must Battle a Teacher in The Canadian Rockies"
“Film_id:1:release_year”	"2005-12-31”
“Actor:27:First_name”	"JULIA"
“Actor:27:Last_name”	"MCQUEEN"
“Film_id:1:actor”	“27,31,66”

Example of NoSQL Four Types – Graphic



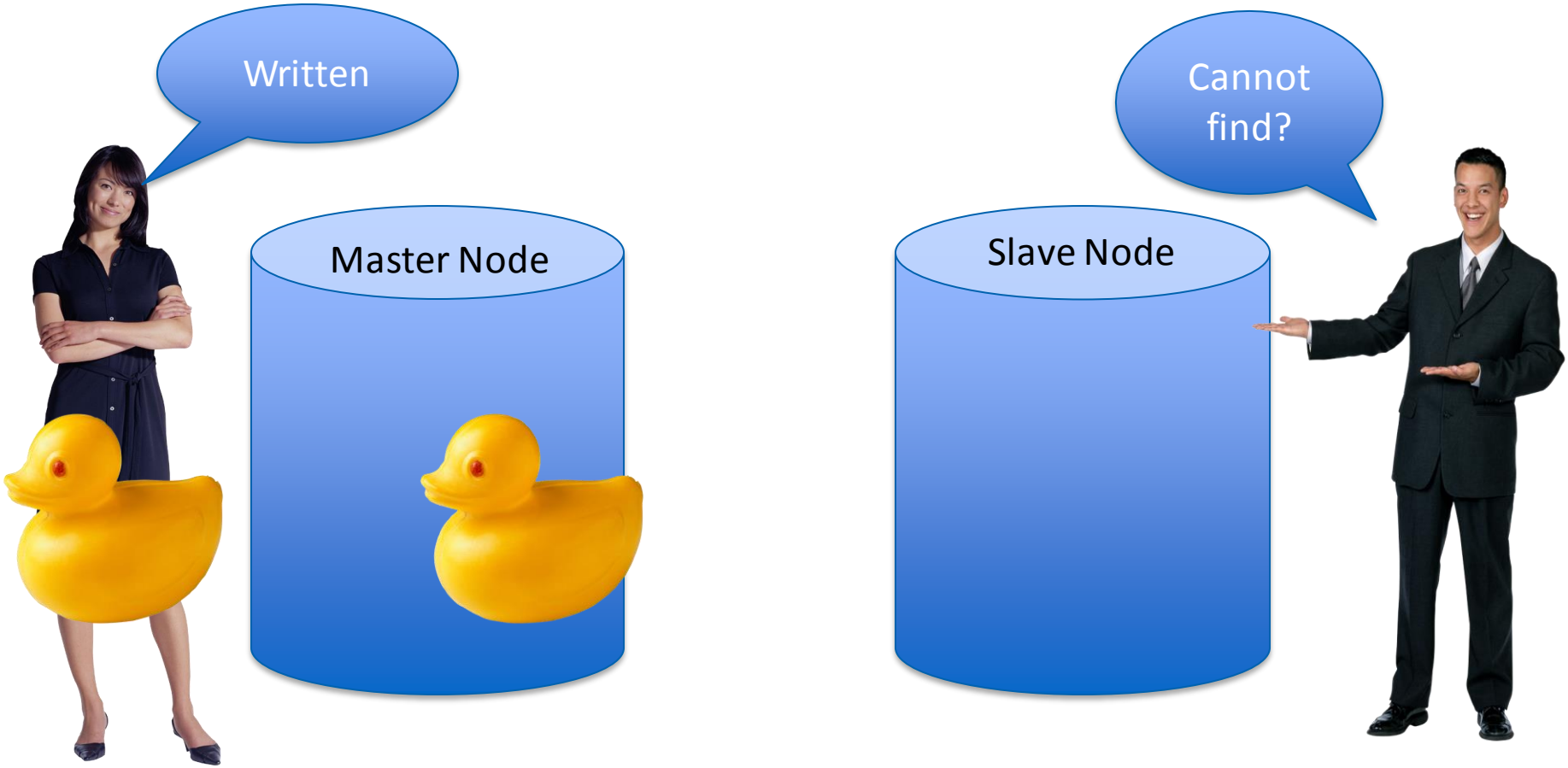
Normalize & De-normalize



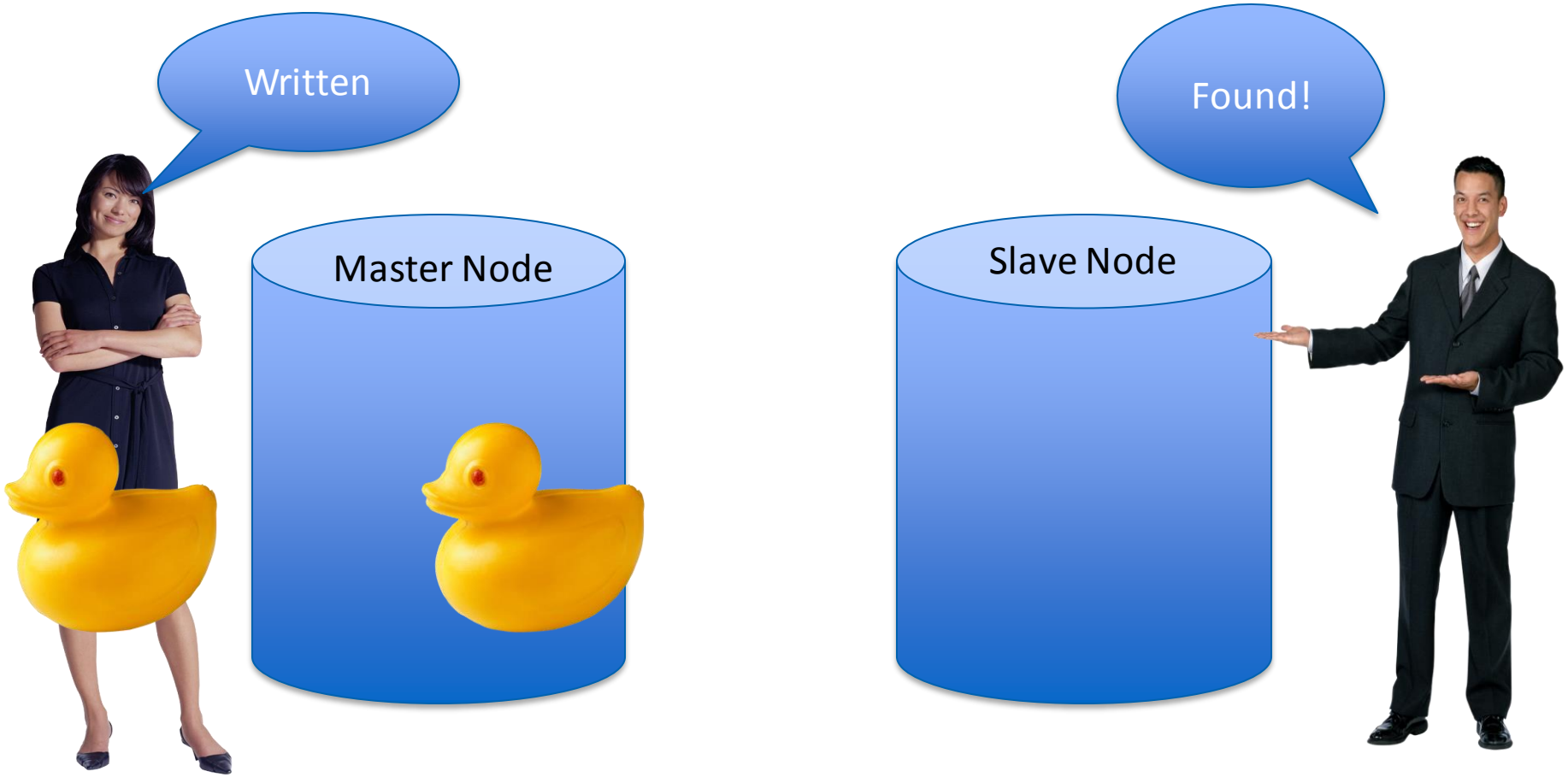
NoSQL Modeling – as an Intelligent Strategy in CA ERwin Big Data Management Solution

- How is the data used (query)?
- Embed, reference or add a table?
  - Strong dependency (composite)
  - One to One
  - Update / Append / Read Only
- Embed, identified aggregated entity
  - Big volume
  - High frequently
- Performance (Index) ?

More Thoughts on NoSQL Modeling – Eventual Consistency



More Thought Before NoSQL Modeling – Strong Consistency



More Thought Before NoSQL Modeling - Consistency

Eventual consistency

- Advantage
  - Scalable
  - Read/Write decouple
- Scenario
  - Allow minor data loss
  - High speed performance
  - Network not stable

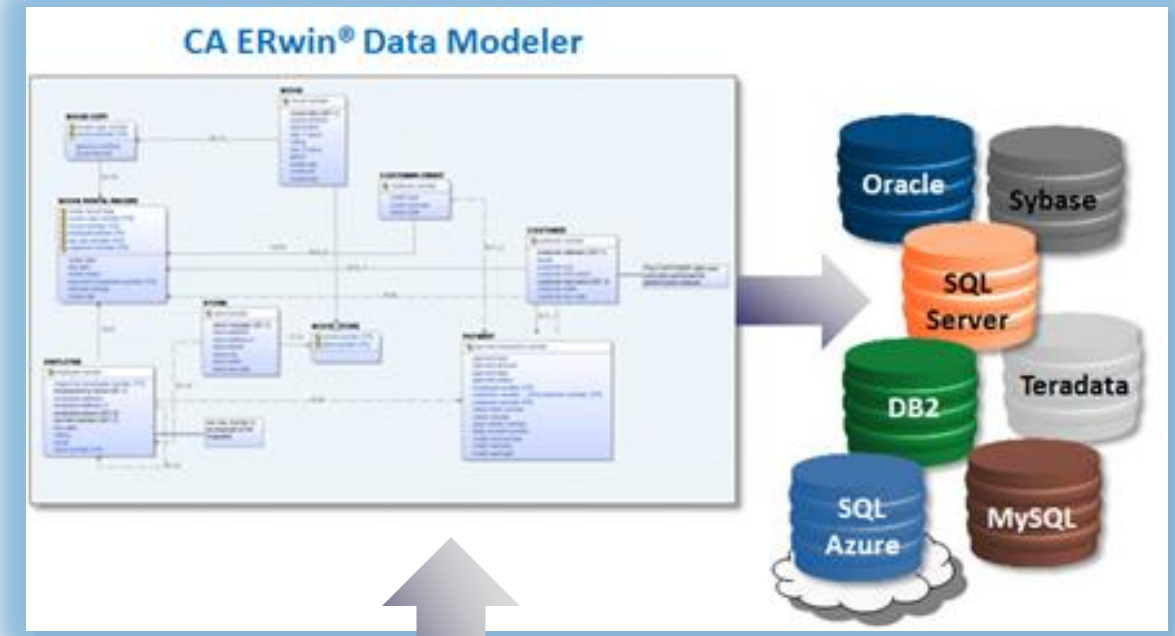
Strong consistency

- Advantage
  - Data reliable
- Scenario
  - Highly accurate
  - No data loss

CA ERwin Big Data Management

MARCH 18-19
ERworld 2015

CA ERwin Big Data Management Solution



CA ERwin Big Data Management

Redis
HBASE
Cassandra
mongoDB
riak
CouchDB
relax
Tokyo Cabinet
Scalaris
Project Voldemort
Neo4j the graph database
membase

MARCH 18-19 ERworld 2015

CA ERwin Big Data Management Solution

- Data Schema transformation between RDBMS and NoSQL (Normalize & De-normalize)
- Data Migrate between RDBMS and NoSQL
- Reverse Engineering from NoSQL
- NoSQL reporting

Core Technical – Sample Data Transformation

– RDBMS Data ➔ MongoDB JSON

Customer

```
"User0","M","20","IN","400000","high school"
"User1","F","50","MX","400000","PhD degree"
"User2","F","50","GR","400000","master degree"
"User3","F","20","DE","300000","PhD degree"
"User4","F","50","AU","100000","master degree"
"User5","F","30","ES","300000","PhD degree"
⋮
```

Aggregate Entity

Product

```
"Product0","electronics","SubCategory25","glass","14900","8046"
"Product1","sporting-goods","SubCategory98","wood","19900","14129"
"Product2","furniture","SubCategory33","steel","15900","12402"
"Product3","toys-and-games","SubCategory110","glass","13800","7590"
"Product4","toys-and-games","SubCategory118","wood","8600","5504"
"Product5","furniture","SubCategory31","aluminium","15300","12087"
⋮
```

sales

```
"User5754","Product3225","2013-05-19","48"
"User5897","Product7228","2013-04-01","58"
"User4851","Product2693","2014-07-21","58"
"User7175","Product8690","2013-08-03","57"
"User2731","Product188","2014-09-07","45"
⋮
```

Transformation strategy

```
{
"_id" : ObjectId("52cb7034e4b03046d744cc90"),
"income" : "400000",
"age" : "30",
"gender" : "M",
"education" : "high school",
"customerID" : "User0",
"country" : "JP",
"Product" : [
        {
        "productID" : "Product8588",
        "category" : "tools-products",
        "subcategory" : "SubCategory100",
        "material" : "plastic",
        “price" : "600",
                ⋮
        }
                ⋮

            ]
}
```

Core Technical – Transformed Sample Data

One document in collection "Customer"


```
{
"_id" : ObjectId("52cb7034e4b03046d744cc90"),
"income" : "400000",
"age" : "30",
"gender" : "M",
"education" : "high school",
"customerID" : "User0",
"country" : "JP",
"Product" : [
        {
        "productID" : "Product8588",
        "category" : "tools-products",
        "subcategory" : "SubCategory100",
        "material" : "plastic",
        "price" : "600",
        "cost" : "360",
        "date" : "2013/08/21",
        "sales" : "19"
        },
        {
        "productID" : "Product5620",
        "category" : "furniture",
        "subcategory" : "SubCategory33",
                ⋮
        }
                ⋮

        ]
}
```


Sub- document "Product"

Core Technical - NoSQL Schema & Data

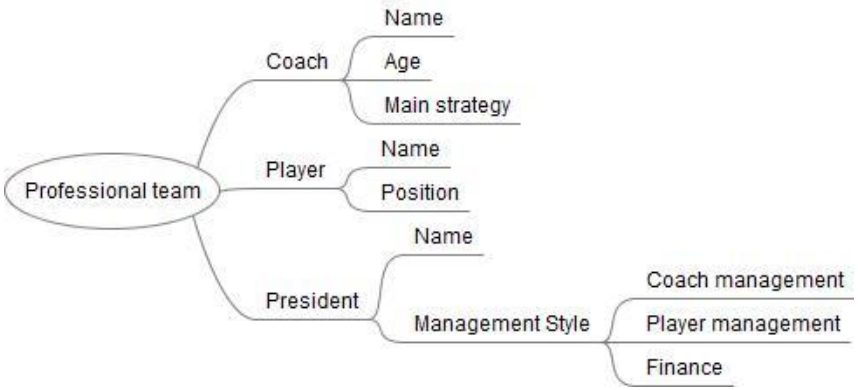
- NoSQL is schema-less.
- Data schema reflects data.
- NoSQL forward engineering means data movement with schema.
- NoSQL reverse engineering means going through data and extracting schema
- Must touch data to manage NoSQL schema.

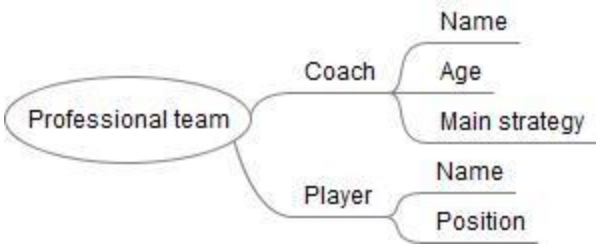
Demo
Migrate RDBMS to MongoDB

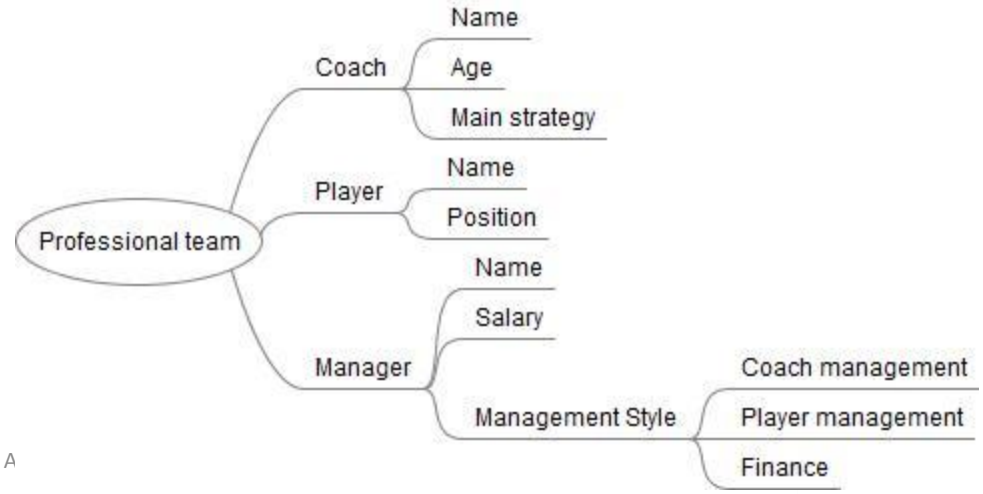
Core Technical – Reverse Engineering of NoSQL

"Professional team" example: Data collection in NoSQL may have different structures.

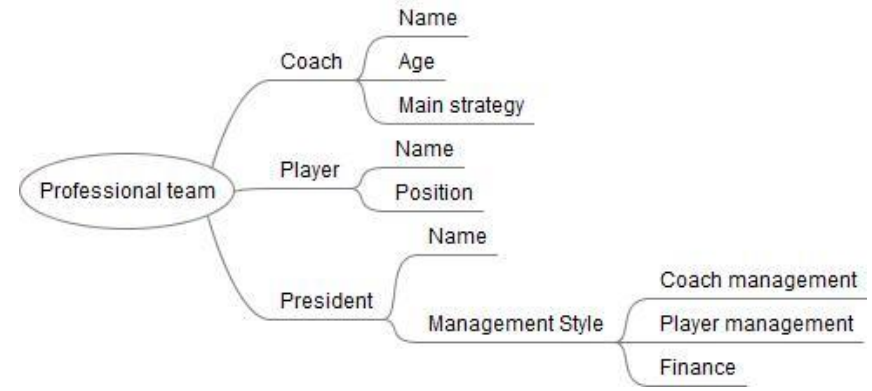
How can we extract the proper schema from NoSQL data?

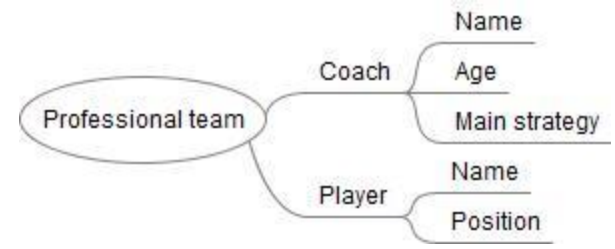


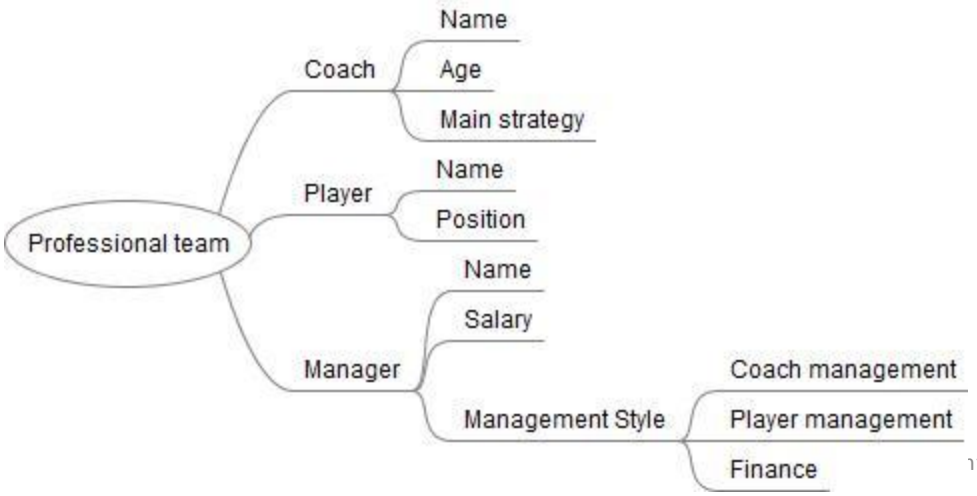


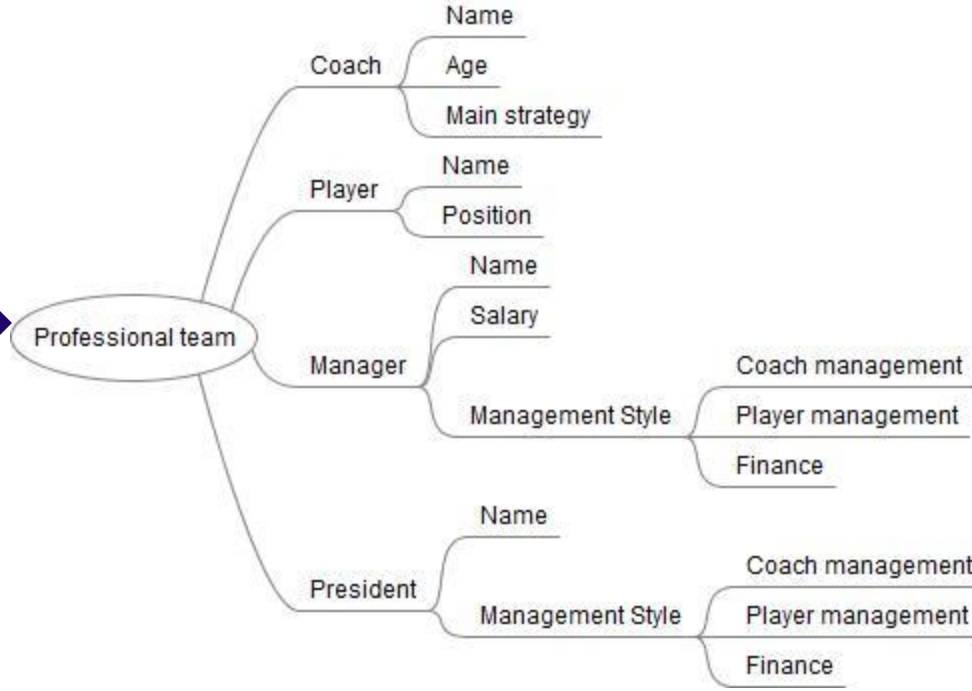


Core Technical – Reverse Engineering of NoSQL





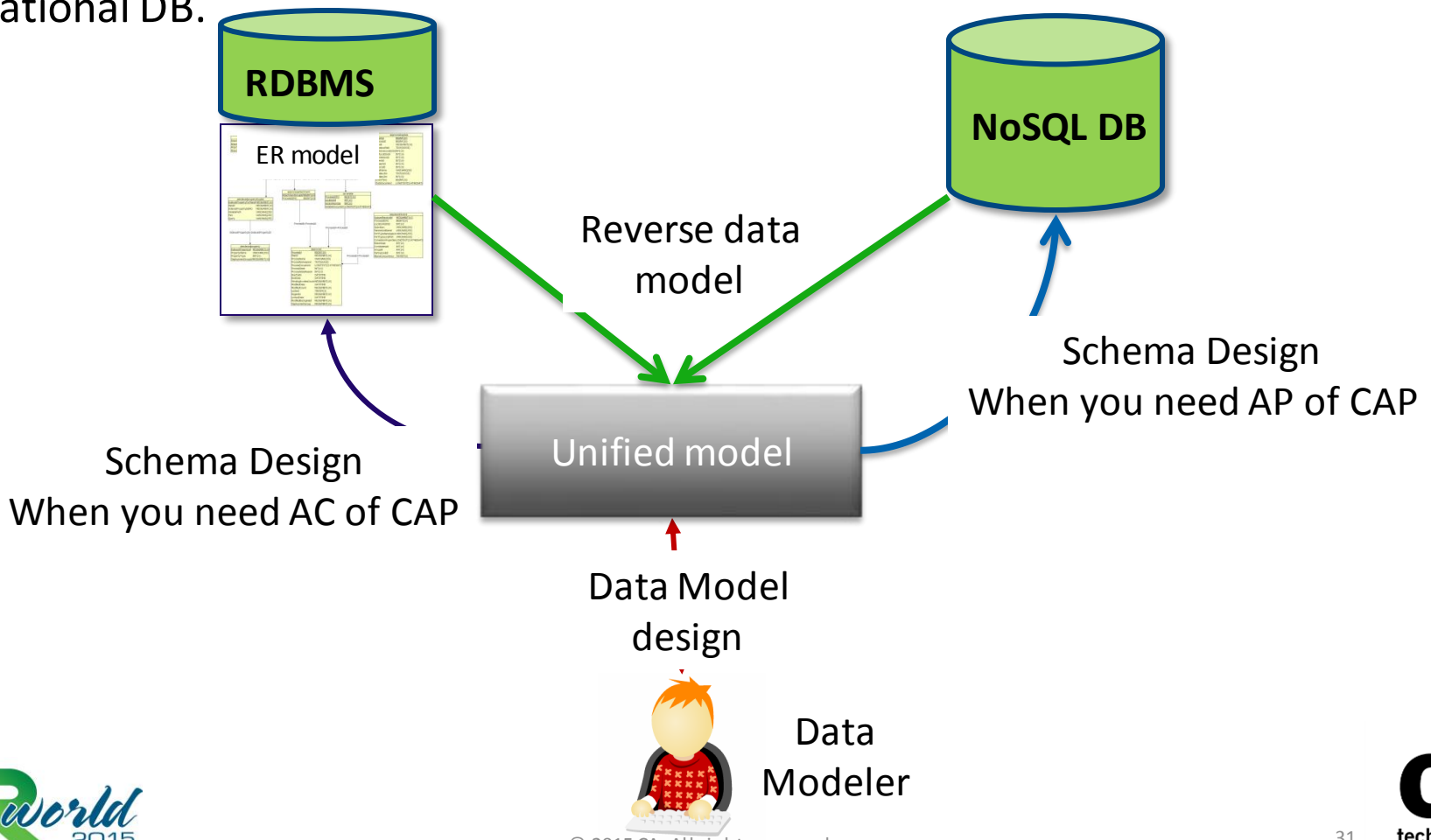




ca technologies

Core Technical - Unified Data Model

Innovative unified model can describe structured and non-structured data, and bi-directional communications transforms the data schema between Big Data and Relational DB.



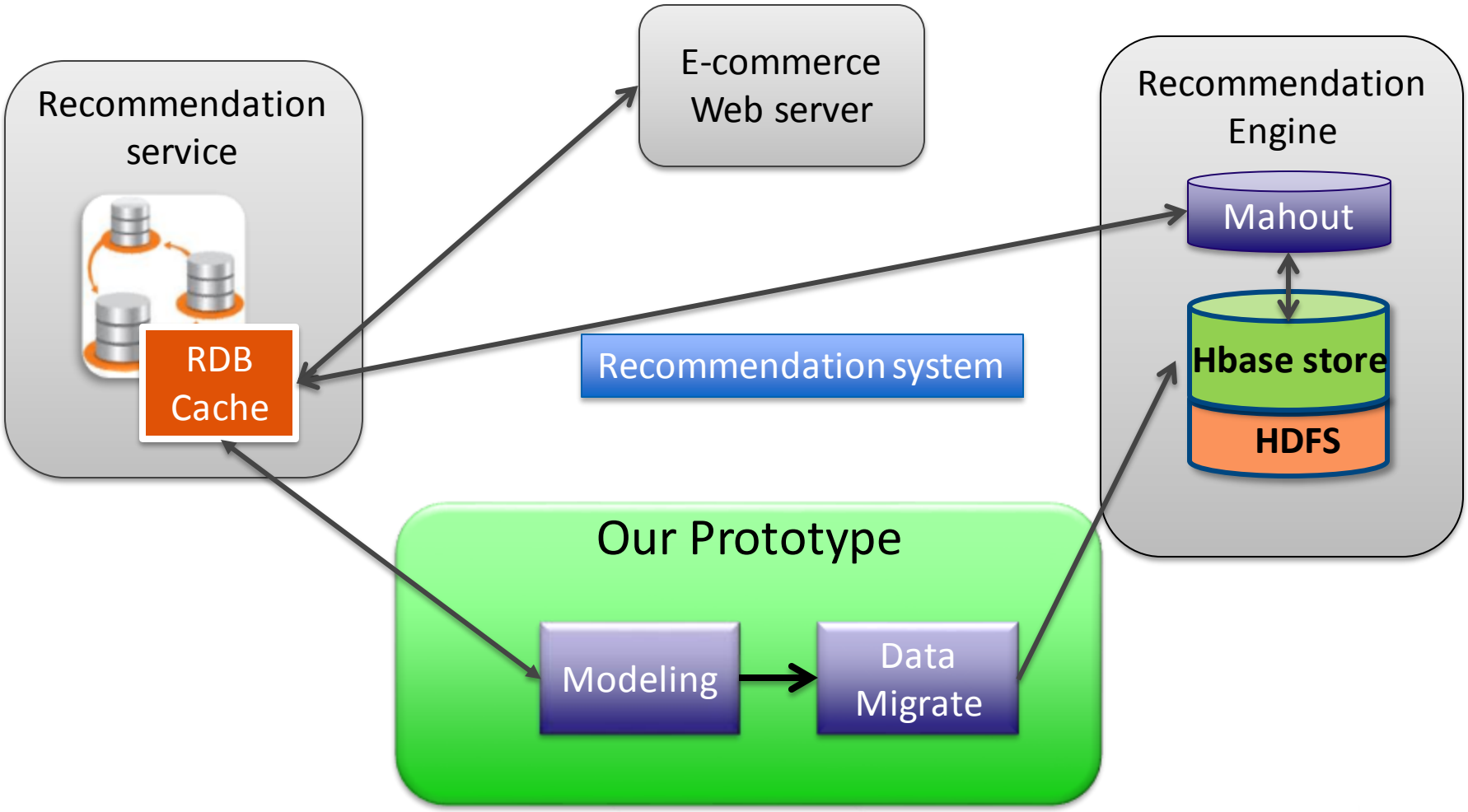
Demos:
Reverse Engineering from NoSQL
& NoSQL BI (Reporting)

A Real Project

Real Project - Context

- **Customer**
  - An online video service vendor in China, provides online TV and films.
- **Requirement**
  - The customer wants to upgrade their video recommendation service. The new system based on Mahout, needs to migrate the database from MySQL to HBase. Hbase can be natively accessed by Mahout and HBase can support the possibility of large data in future.
- **Functionality**
  - The project is to move data with its structure from MySQL to HBase. The original MySQL DB has two databases. One is simple with about 20 tables, and the other is more complex with more than 100 tables. These two DBs mainly store about 1GB of video and related user information.

Real Project - Architect



Next Step

- Recruit more real project trials & enhance on demand
- allen.wang@ca.com or neil.buchwalter@ca.com

Thank You for Attending!

For any further questions, feel free to join the Chat Session following this presentation, or contact me outside of ERworld.

Allen Wang

Allen.Wang@ca.com

Please enjoy the rest of your time at ERworld 2015!

Legal Notice

THIS PRESENTATION IS FOR YOUR INFORMATIONAL PURPOSES ONLY. CA assumes no responsibility for the accuracy or completeness of the information. TO THE EXTENT PERMITTED BY APPLICABLE LAW, CA PROVIDES THIS DOCUMENT "AS IS" WITHOUT WARRANTY OF ANY KIND, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. In no event will CA be liable for any loss or damage, direct or indirect, in connection with this presentation, including, without limitation, lost profits, lost investment, business interruption, goodwill, or lost data, even if CA is expressly advised of the possibility of such damages.