SQDATA
CORPORATION

Best Practices Series

Populating Big Data Repositories from IMS

Prepared for the:

Virtual IMS User Group

7 October 2014

Agenda

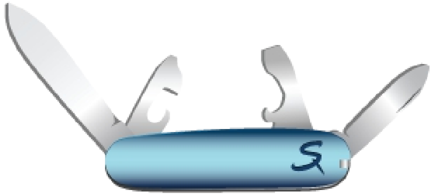
- Introduction
- Big Data Overview
  - ✔ Background
  - ✔ Hadoop
  - ✔ HBase
  - ✔ Cassandra
  - ✔ MongoDB
- IMS to Big Data
  - ✔ Approach
  - ✔ Considerations
- Q & A
- Conclusion

About the Speaker

- **Scott Quillicy**
  - ✔ 30+ Years Database Experience
  - ✔ Commercial Database Software Development
  - ✔ Deployment of Complex Data Integration Solutions
- **Founded SQData to Provide Customers with:**
  - ✔ An Enterprise Class Data Integration / Replication Framework
  - ✔ A Solution that Handles Both Relational and Non-Relational Data
  - ✔ Technology Built Around Best Practices
- **Specialization**
  - ✔ Database Replication
  - ✔ IMS – the More Complex, the Better
  - ✔ Heterogeneous Database Integration
  - ✔ Continuous Availability
  - ✔ Database Performance

About SQData

- **"Swiss Army Knife of Data Integration Tools"**



- **Core Competencies**
  - ✔ High-Performance Changed Data Capture (CDC)
  - ✔ Non-Relational Data → IMS, VSAM, Flat Files
  - ✔ Relational Databases → DB2, Oracle, SQL Server, etc.
  - ✔ Deployment of Complex Data Integration Solutions
  - ✔ Continuous Availability of Critical Applications
  - ✔ Data Conversions / Migrations
- **Customer Usage**
  - ✔ Relational and Non-Relational Data
  - ✔ Data Replication – Relational and Non-Relational
  - ✔ ETL (Bulk Data Extracts/Loads)
  - ✔ Application Integration
  - ✔ Business Event Publishing
  - ✔ Data Conversions / Migrations

What is Big Data?

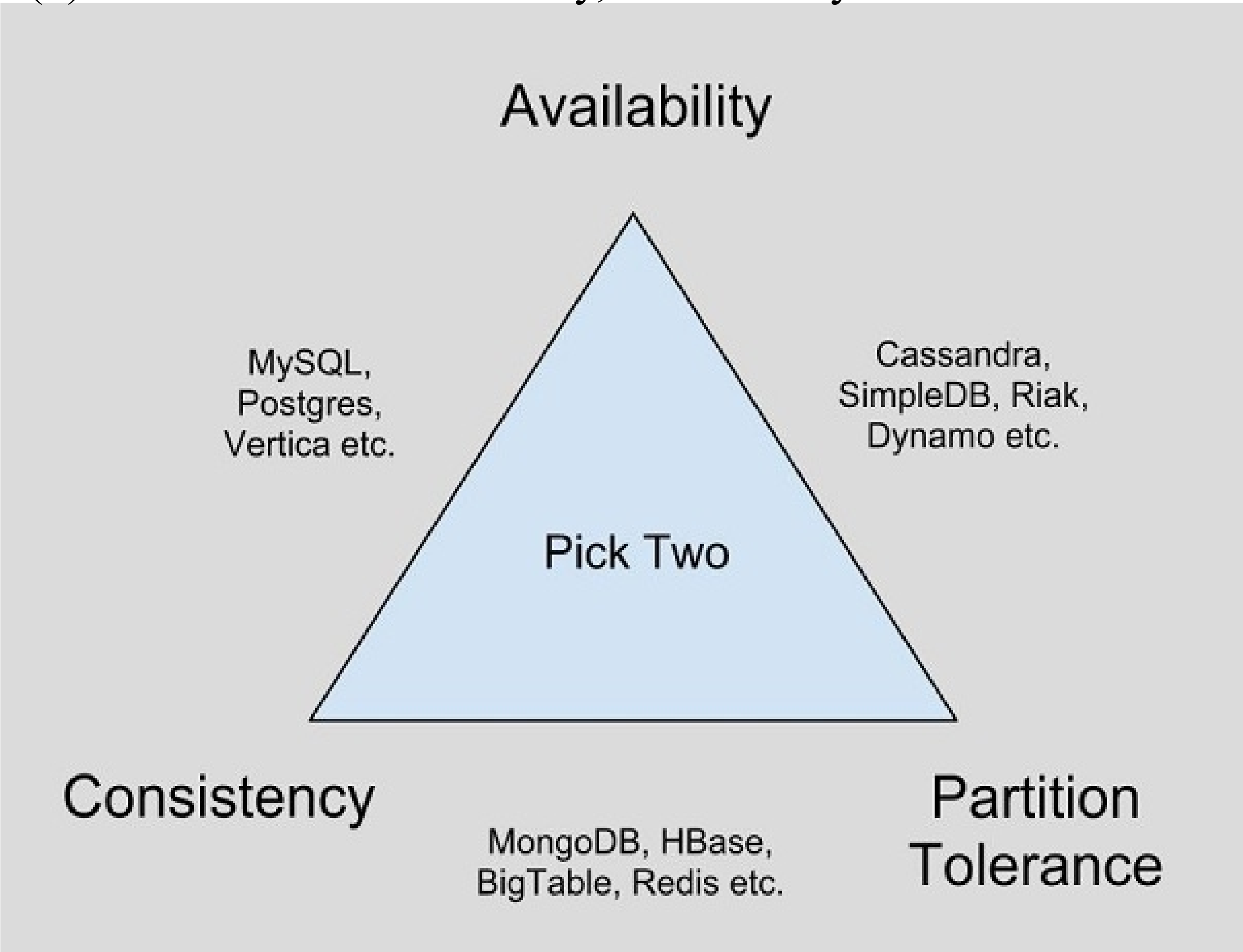
- **What You May Have Heard...**
  - ✔ The 'New Wave' of Technology
  - ✔ Exclusively Hadoop and/or NoSQL Based
  - ✔ Advanced Analytics of Disparate Data
  - ✔ Big Data 'Knows' What You are Doing... ☺
- **A Large Collection of Data → Been Around for 50+ Years**
- **Characteristics**
  - ✔ Significant Amount of Data
  - ✔ Many Different Formats
  - ✔ High Rate of Change
  - ✔ Complex
- **Challenges**
  - ✔ Increasing Data Volumes → Stress Traditional RDBMS
  - ✔ Computing and Infrastructure Costs to Process / Analyze
  - ✔ Most Companies in Early Stages of Adoption

Enter Hadoop and NoSQL

- **Hadoop Family**
  - ✔ HDFS → basic file system
  - ✔ HBase → NoSQL DB built on HDFS
  - ✔ HCatalog → metadata
  - ✔ Hive → SQL interface
  - ✔ Pig → scripting language used for MapReduce for unstructured sources
- **Cassandra**
  - ✔ Wide-Column Store
  - ✔ Handles Very Large Datasets in “Almost” SQL
  - ✔ Ring Architecture
  - ✔ Selectable Replication
- **MongoDB**
  - ✔ Popular Document Store
  - ✔ JSON / BSON Format
  - ✔ Master / Slave Replication

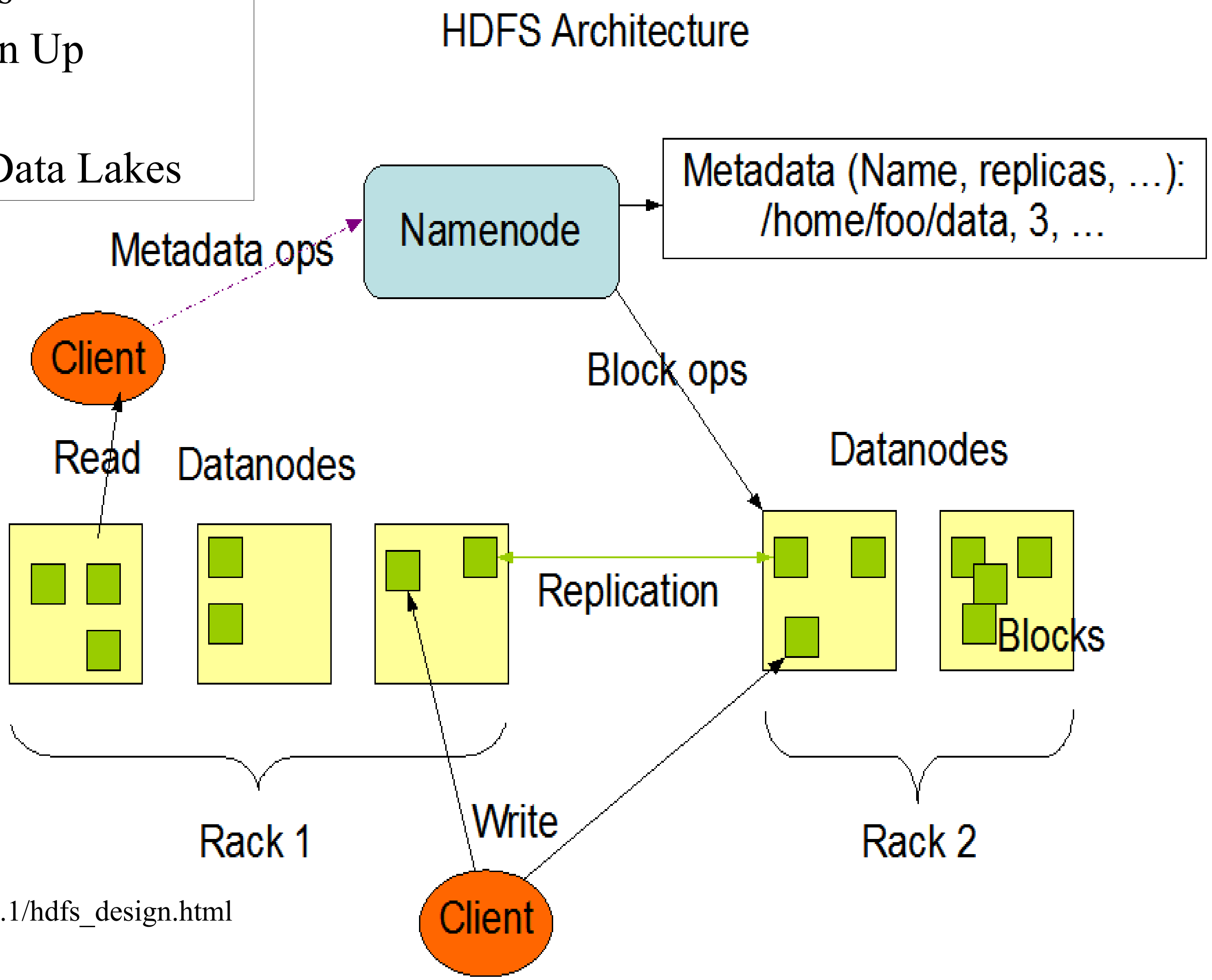
CAP Theorem

- Eric Brewer - 1998 → Impossible for a Distributed System to Provide All Three (3) Guarantees of Availability, Consistency and Partition Tolerance


Availability
MySQL, Postgres, Vertica etc.
Cassandra, SimpleDB, Riak, Dynamo etc.
Pick Two
Consistency
MongoDB, HBase, BigTable, Redis etc.
Partition Tolerance

Hadoop HDFS Architecture

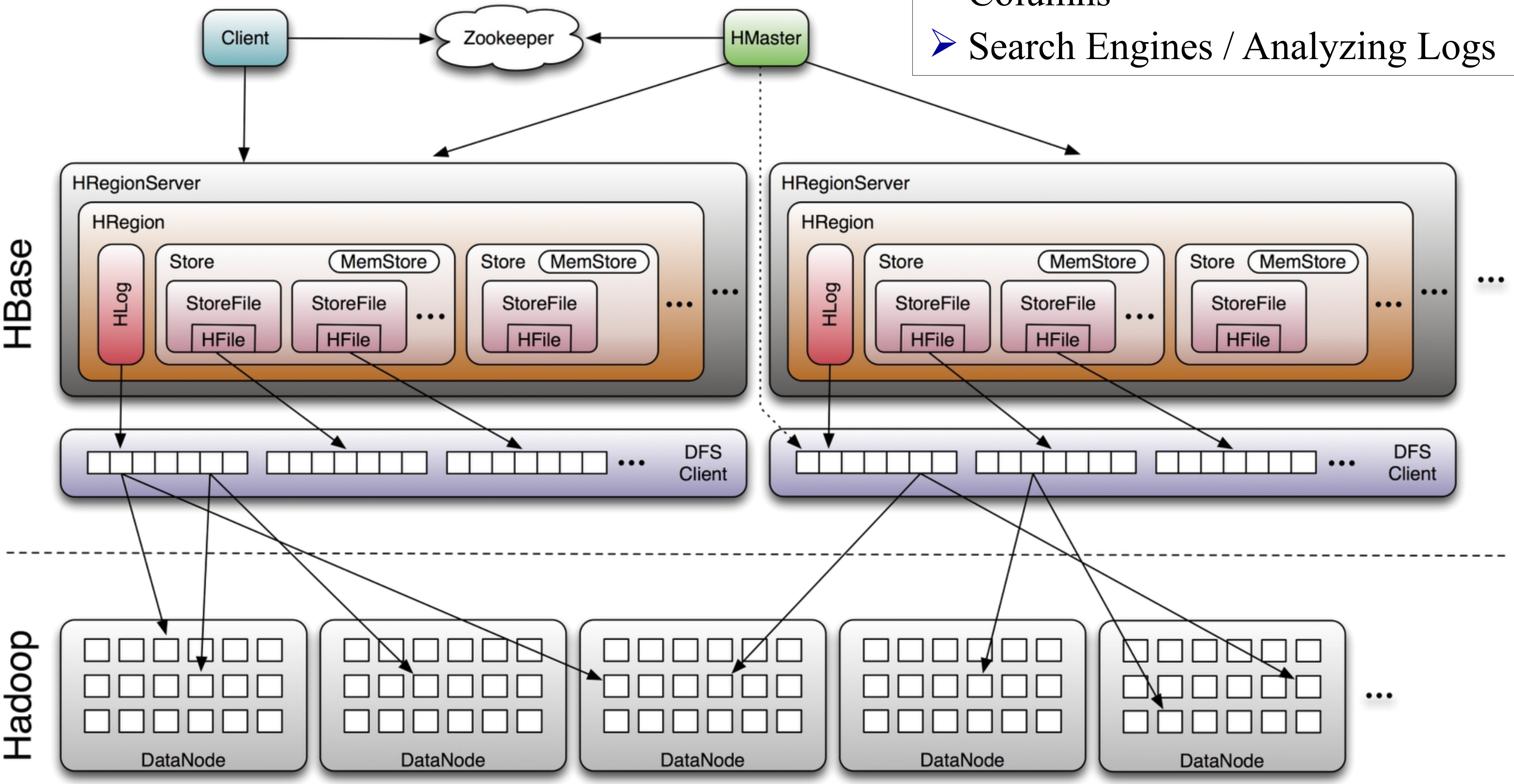
- Basic Distributed File System
- Append-Only Writes
- Compaction to Clean Up
- File Level Locking
- Ideal for Streams / Data Lakes



http://hadoop.apache.org/docs/r1.2.1/hdfs_design.html

Hadoop HBase Architecture

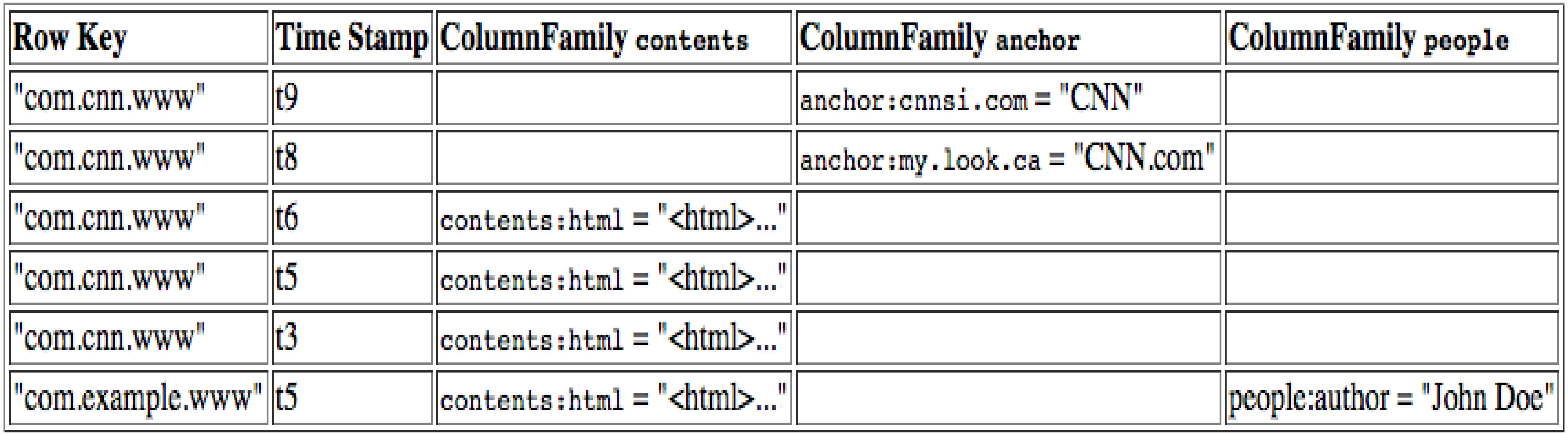
- NoSQL on top of HDFS
- CAP: consistency, part tolerance
- Billions of Rows x Millions of Columns
- Search Engines / Analyzing Logs



HBase Data Model

- Table → Collection of Rows
- Row → Key & Multiple Columns
- Column → Family & Qualifier
- Timestamp → Versioning - Time of Write

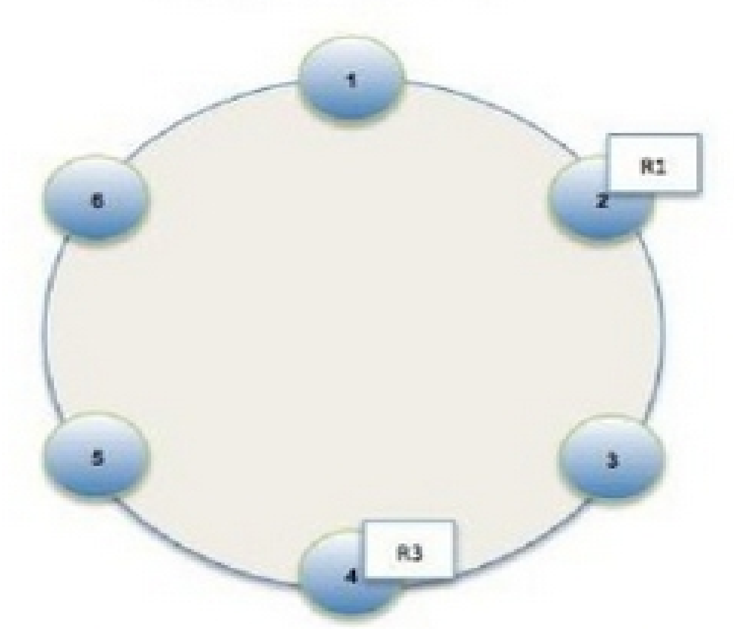
Table 5.1. Table `webtable`

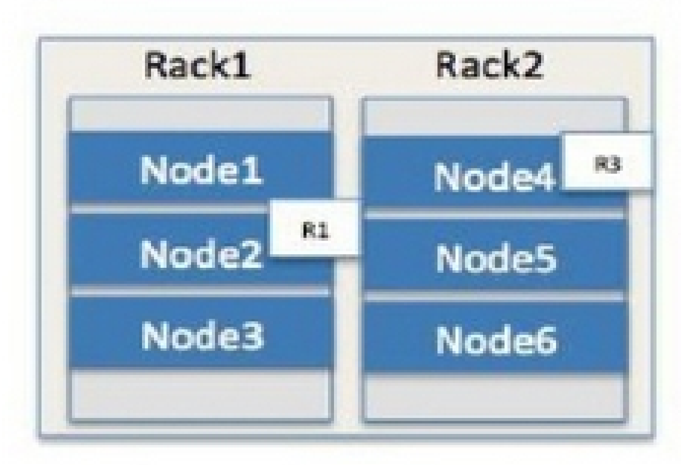


Row Key	Time Stamp	ColumnFamily `contents`	ColumnFamily `anchor`	ColumnFamily `people`
"com.cnn.www"	t9		`anchor:cnnsi.com` = "CNN"	
"com.cnn.www"	t8		`anchor:my.look.ca` = "CNN.com"	
"com.cnn.www"	t6	`contents:html` = "<html>..."		
"com.cnn.www"	t5	`contents:html` = "<html>..."		
"com.cnn.www"	t3	`contents:html` = "<html>..."		
"com.example.www"	t5	`contents:html` = "<html>..."		people:author = "John Doe"

http://hbase.apache.org/book/datamodel.html#conceptual.view

Cassandra Architecture

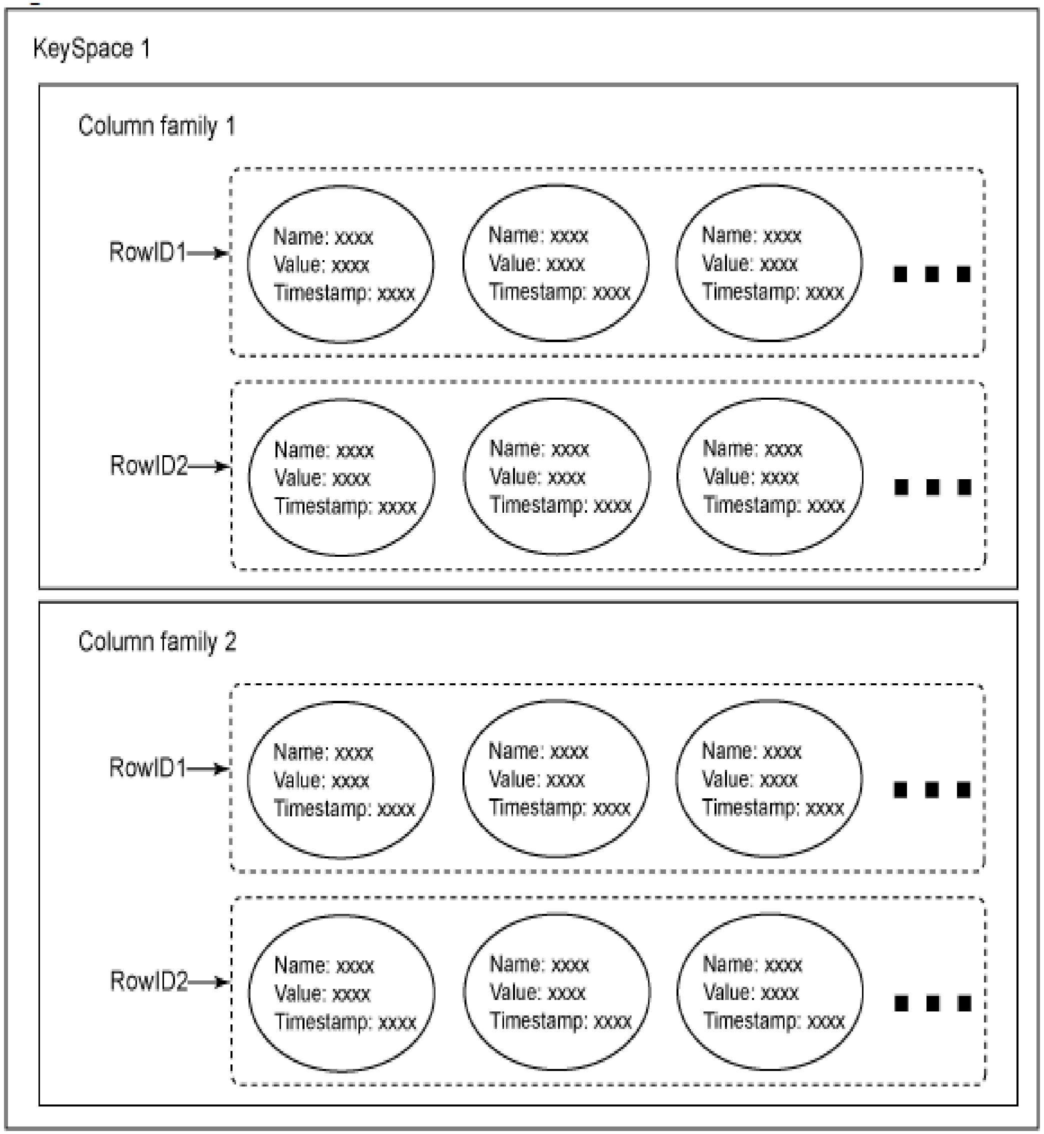




http://www.ibm.com/developerworks/library/os-apache-cassandra/

- NoSQL – Hashed Keys
- Wide-Column Store
- Great Read / Write Performance
- No Transactions / No Joins
- CAP: Availability, Part Tolerance
- Keys Must be Unique

Cassandra Data Model

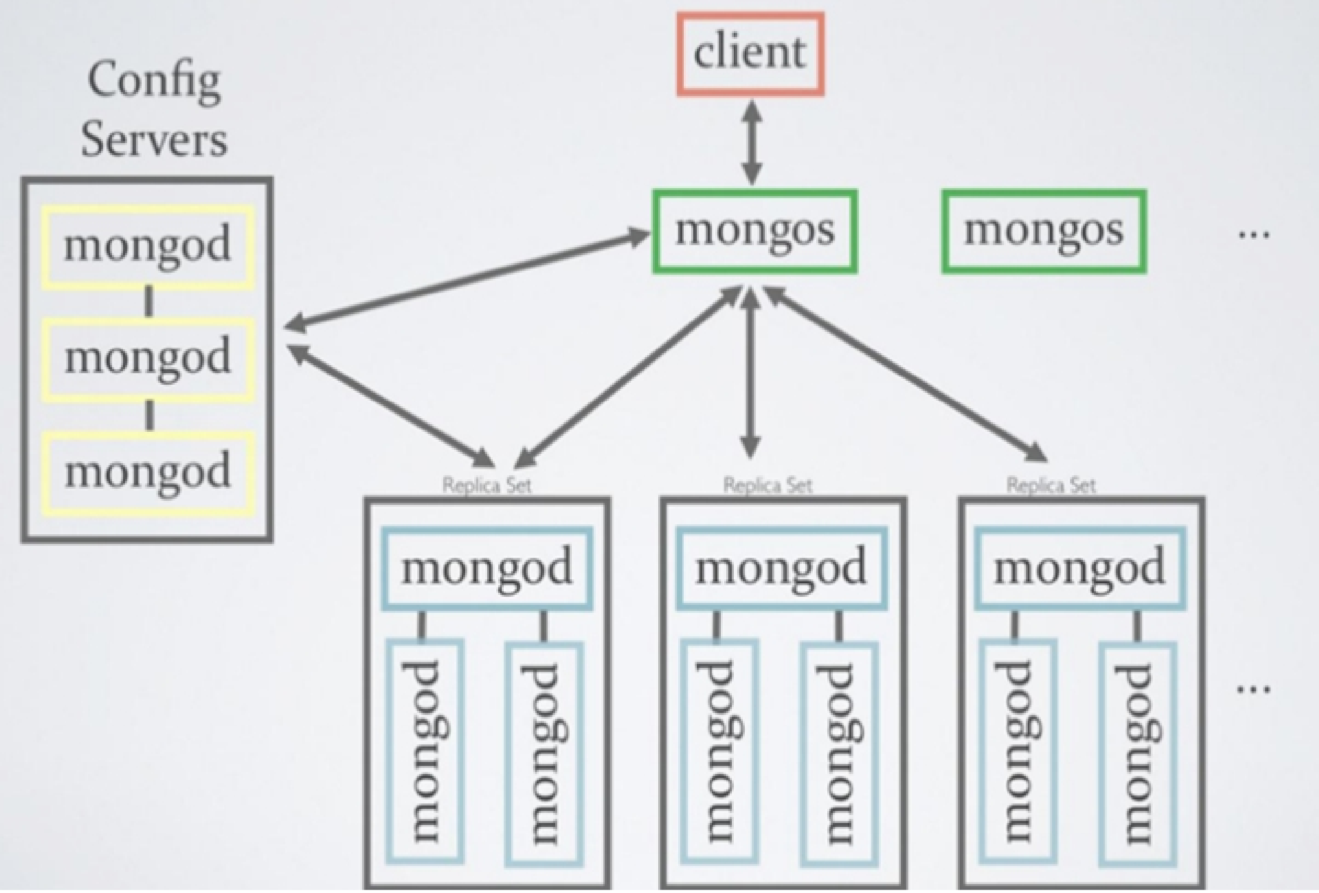


- KeySpace → Database
- Column Family → Table
- Rows → Collection of Columns
- Columns can be Dynamic
- Keys Must be Unique

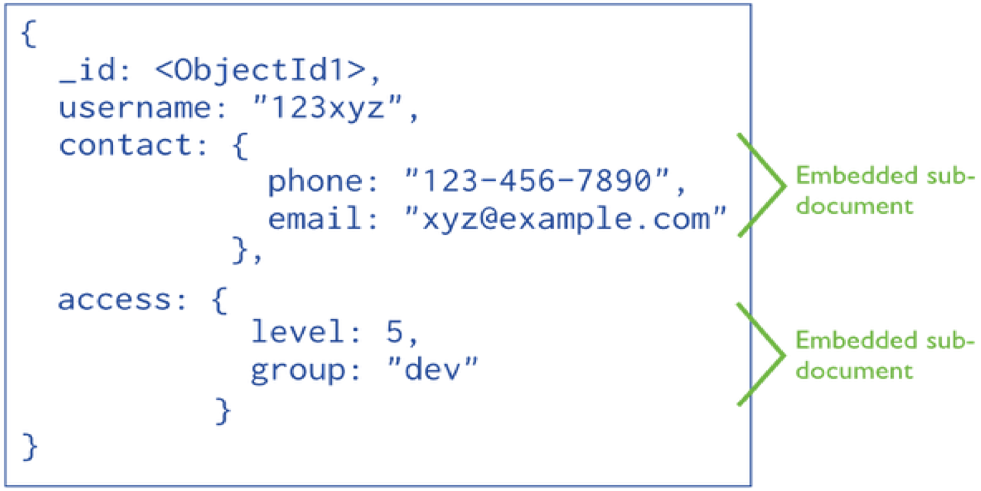
http://www.ibm.com/developerworks/library/os-apache-cassandra/

MongoDB Architecture

- NoSQL – Document Store (JSON/BSON)
- CAP: Consistency / Partition Tolerance
- Keys Not Required to be Unique
- Great for Dynamic Queries

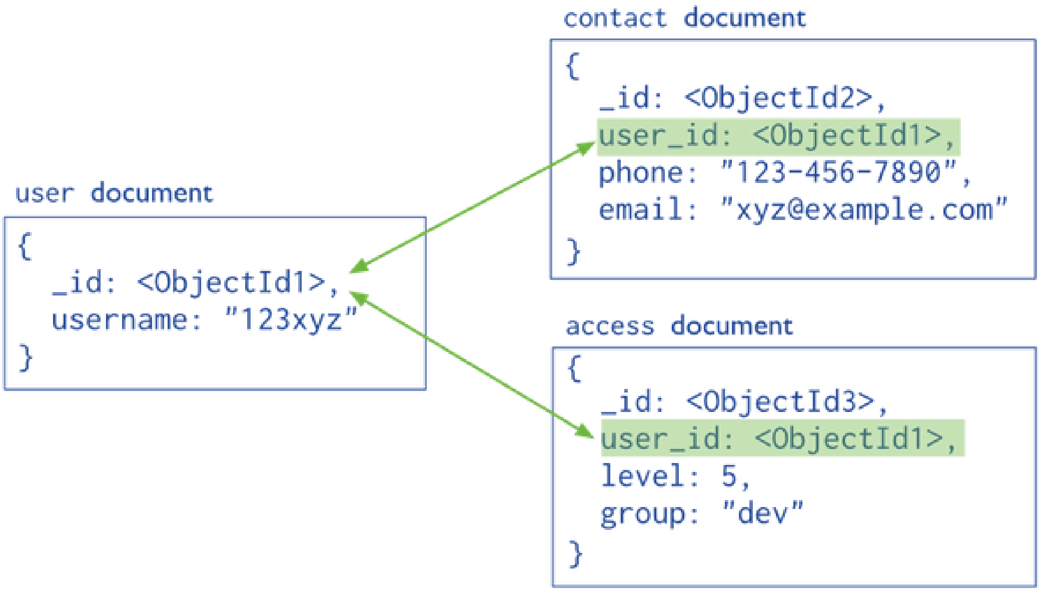


MongoDB Data Model



```
{
  _id: <ObjectId1>,
  username: "123xyz",
  contact: {
              phone: "123-456-7890",
              email: "xyz@example.com"
           },
  access: {
             level: 5,
             group: "dev"
          }
}
```

- Embedded Model
- Denormalized
- Hierarchical Entity Relationships
- One-to-Many Relationships
- Fast Read Performance



user document

```
{
  _id: <ObjectId1>,
  username: "123xyz"
}
```

contact document

```
{
  _id: <ObjectId2>,
  user_id: <ObjectId1>,
  phone: "123-456-7890",
  email: "xyz@example.com"
}
```

access document

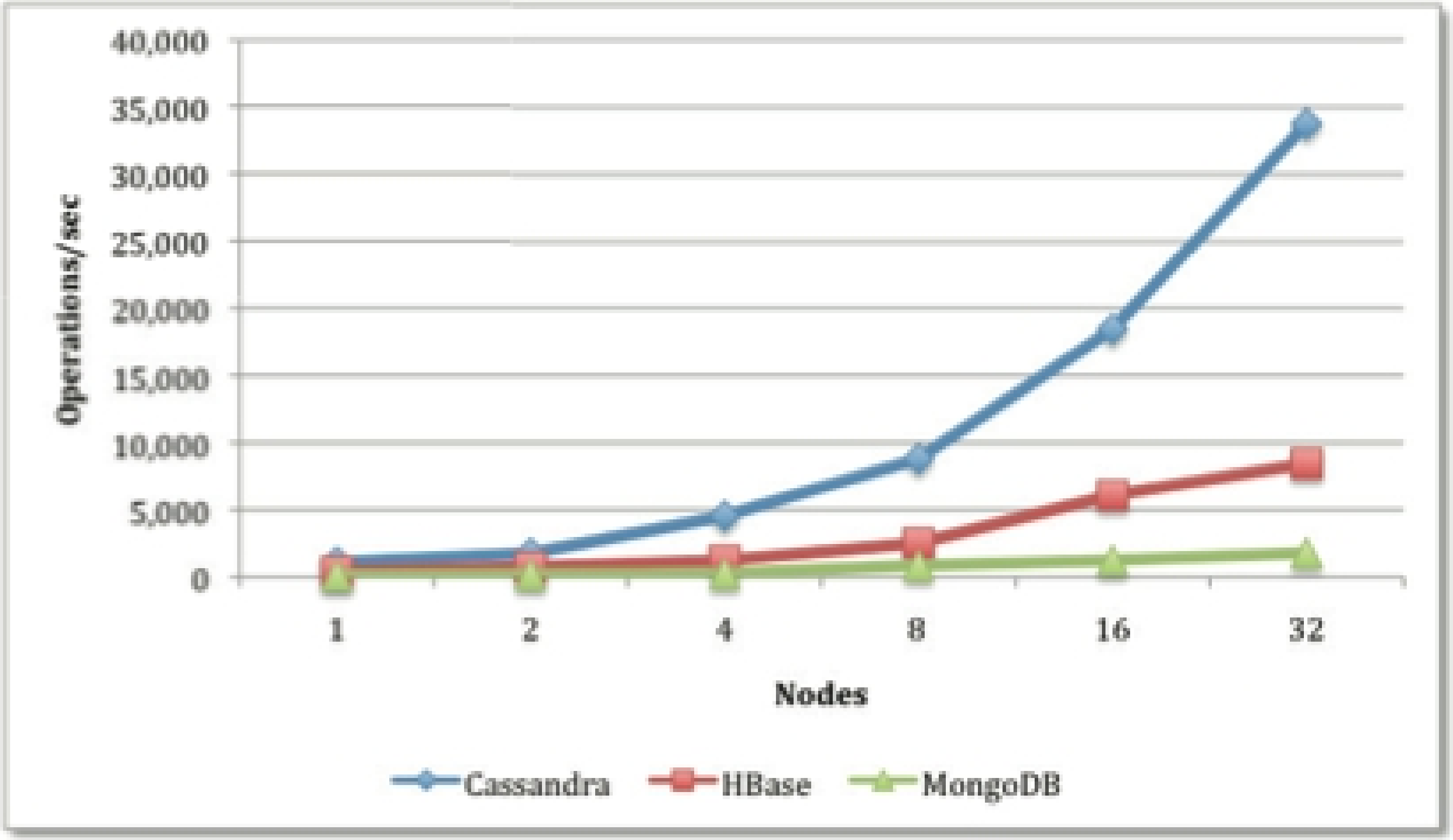
```
{
  _id: <ObjectId3>,
  user_id: <ObjectId1>,
  level: 5,
  group: "dev"
}
```

- Normalized Model
- Higher Degree of Duplication
- Many-to-Many Relationships
- Large, Complex Hierarchies

http://www.ibm.com/developerworks/library/os-apache-cassandra/

Performance

Read/Write Mix Workload



http://planetcassandra.org/nosql-performance-benchmarks/

Agenda

- Introduction
- Big Data Overview
  - Background
  - Hadoop
  - HBase
  - Cassandra
  - MongoDB
- **IMS to Big Data**
  - Approach
  - Considerations
- Q & A
- Conclusion

Why IMS to Big Data?

- Provide a Method of Analyzing Data Outside of IMS
- Business Intelligence / Advanced Analytics
- Combine with Data from other Apps → Structured & Unstructured
- Inexpensive Computing / Storage
- Compliment Established Data Warehouse(s)
- <u>Good News</u> → Less Complicated than IMS to Relational

Best Practices Summary

- **Let the Business Drive the Effort**
  - ✔ Ensures Proper Alignment with Business Goals
  - ✔ Queries Drive the Data Model Design
  - ✔ Avoid I/T Initiated 'Build it and They will Come'
- **Temper the Exuberance**
  - ✔ Inevitable After Successful Implementation for a Given Application
  - ✔ Technology is Rapidly Evolving → What is OK Today may be Obsolete Tomorrow
  - ✔ It is More Expensive than the Hype Leads You to Believe
- **Align with Enterprise Data**
  - ✔ Where I/T Comes Takes a Lead Role
  - ✔ Existing Data Warehouse / Business Intelligence Setups
  - ✔ Infrastructure / Data Integration
- **Use an Iterative Approach for Implementation**
  - ✔ Agile / Agile Like
  - ✔ Set the Relational Mindset Aside
  - ✔ Allows for 'Adjustments' without Major Schedule Impact

Key Considerations

- **Big Data Repository Selection**
  - ✔ Consider Open Source Projects → Large Communities
  - ✔ Beware of Vendor Lock
  - ✔ May Require More than One (1)
- **Data Delivery / Latency**
  - ✔ Business Driven
  - ✔ Full Extracts → Periodic
  - ✔ Near-Real-Time / Scheduled Changes
- **Workload Characteristics**
  - ✔ Read vs Update Ratio
  - ✔ Update Volume → Changes as a Percentage of a Particular Source
  - ✔ Will Effect Big Data Repository Selection
- **Format**
  - ✔ Level of Normalization → Less is Usually Desirable
  - ✔ Privacy / Masking
  - ✔ Level of Transformation

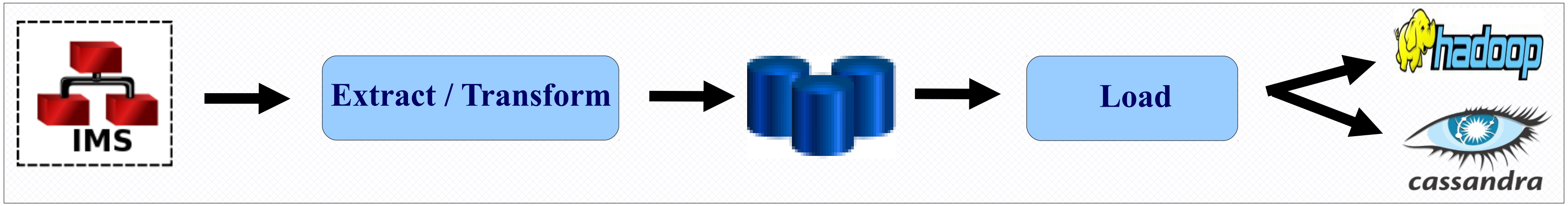
Common IMS Data Challenges

- **Code Page Translation**
- **Invalid Data**
  - ✓ Non-Numeric Data in Numeric Fields
  - ✓ Binary Zeros in Packed Fields (or Any Field)
  - ✓ Invalid Data in Character Fields
- **Dates**
  - ✓ Must be Decoded / Validated if Target Column is DATE or TIMESTAMP
  - ✓ May Require Knowledge of Y2K Implementation
  - ✓ Allow Extra Time for Date Intensive Applications
- **Repeating Groups**
  - ✓ Sparse Arrays
  - ✓ Number of Elements
  - ✓ Will Probably be De-normalized
- **Redefines**
- **Binary / 'Special' Fields**
  - ✓ Common in Older Applications Developed in 1970s / 80s
  - ✓ Generally Requires Application Specific Translation

The Role of ETL and CDC

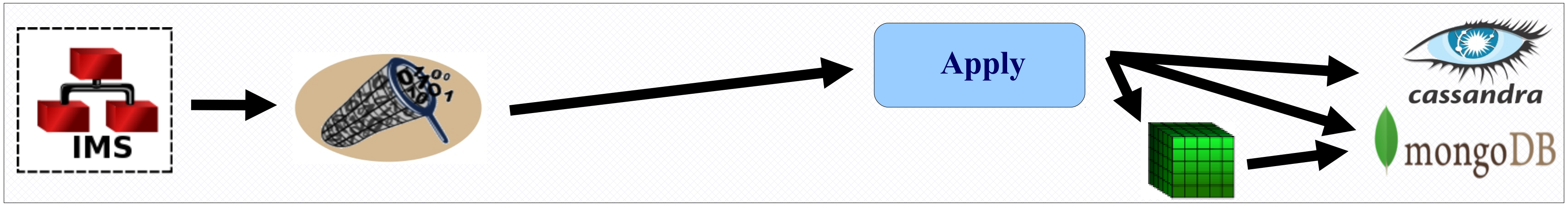
ETL (Extract, Transform, Load):

- ✓ Full Data Extract / Load
- ✓ Data Transformation Logic Defined in this Step
- ✓ Iterative Process – Must be Fast and Efficient
- ✓ Should Minimize Data Landing



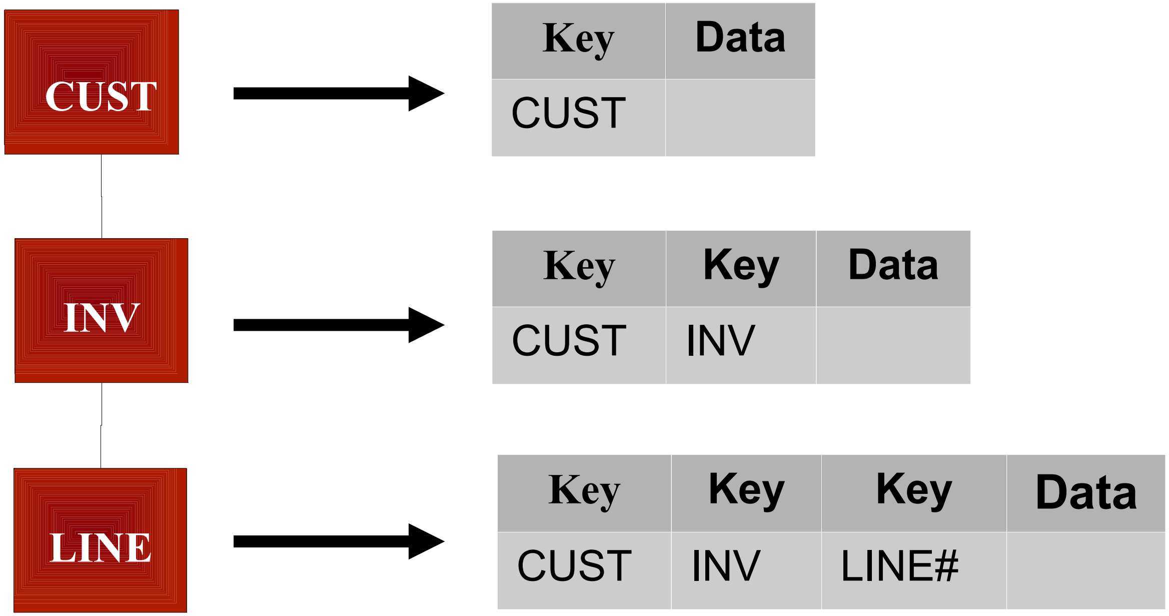
CDC (Changed Data Capture):

- ✓ Move Only Data that has Changed
- ✓ Ideal for Sequence of Events
- ✓ Re-Use Data Transformation Logic from ETL
- ✓ Near-Real-Time / Deferred Latency



IMS to Relational Model

- Normalized → at Least 2nd Normal Form
- Each Segment Typically Maps to One (1) or More Tables



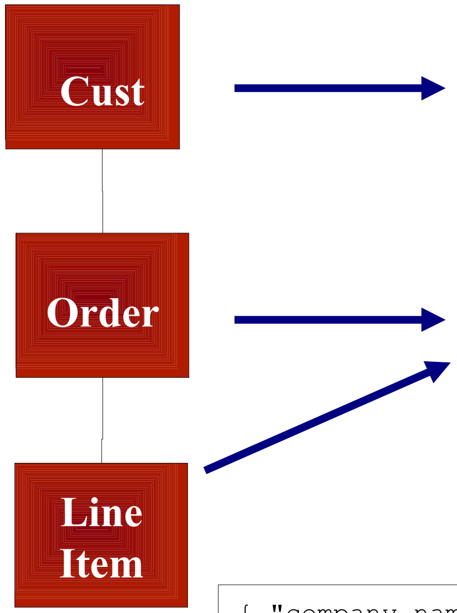
Key	Data
CUST	

Key	Key	Data
CUST	INV	

Key	Key	Key	Data
CUST	INV	LINE#	

IMS to Big Data Model

- De-Normalized / Minimal Normalization
- Degree of Data Redundancy → Trade-Off for Query Performance



Key	Data
Cust#	

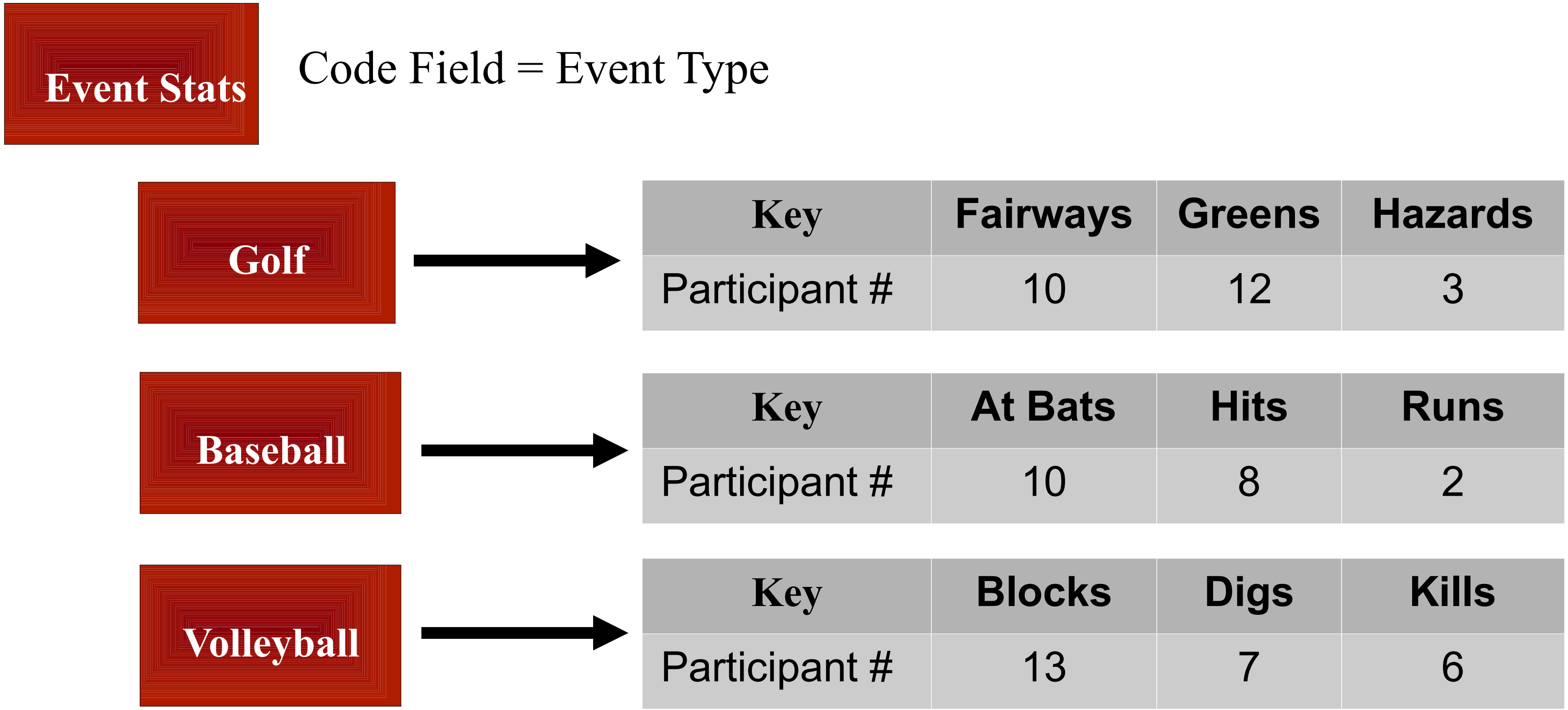
Key	Data	Data	Data	Data	Data	Data
Order#	Cust#		Line #		Line#	

```
{ "company_name" : "Acme",
  "cust_no"       : "20223",
  "contact" :{ "name" : "Jane Smith",
               "address" : "123 Maple Street",
               "city" : "Pretendville",
               "state" : "NY",
               "zip"   : "12345" }
}
```

```
{ "order_no" : "12345",
  "cust_no"  : "20223",
  "price"  : 23.95,
  "Lines" : { "item" : "Widget1",
              "qty"  : "6",
              “cost” : “2.43”
              "item  : “Widge2y"
              "qty"  : "1",
              "cost" : "9.37"
            },
}
```

Redefines: Relational Targets

- Redefine Identified by One (1) or More Code Fields
- Each Redefine Typically Mapped to a Separate Target Table



Event Stats

Code Field = Event Type

Golf

Key	Fairways	Greens	Hazards
Participant #	10	12	3

Baseball

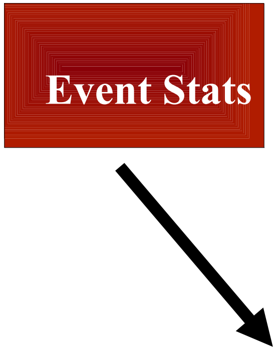
Key	At Bats	Hits	Runs
Participant #	10	8	2

Volleyball

Key	Blocks	Digs	Kills
Participant #	13	7	6

Redefines: NoSQL Targets

- Each Redefine Mapped to Same Target

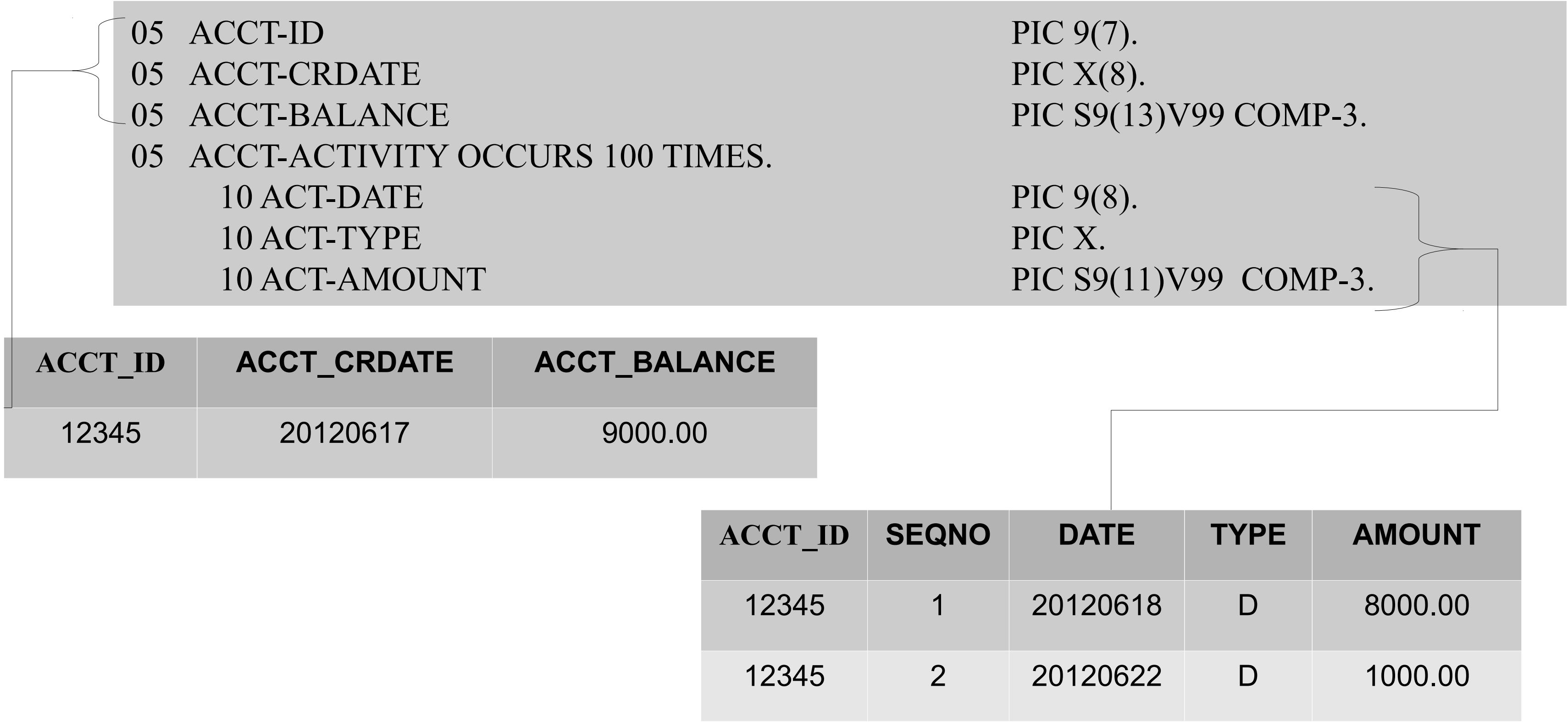


Event Stats

Key	Fairways	Greens	Putts	At Bats	Hits	Runs	Blocks	Digs	Kills
Participant #	10	12	29	10	8	2	13	7	6

Repeating Groups: Relational

- ✓ Typical Candidates for Normalization Based on # Occurs
- ✓ Options:
  - Low # Occurs → Keep in Same Table as Rest of Segment
  - Map to Separate Table – Requires a Sequence Number



```
05   ACCT-ID                                        PIC 9(7).
05   ACCT-CRDATE                                    PIC X(8).
05   ACCT-BALANCE                                   PIC S9(13)V99 COMP-3.
05   ACCT-ACTIVITY OCCURS 100 TIMES.
        10 ACT-DATE                                 PIC 9(8).
        10 ACT-TYPE                                 PIC X.
        10 ACT-AMOUNT                               PIC S9(11)V99  COMP-3.
```

ACCT_ID	ACCT_CRDATE	ACCT_BALANCE
12345	20120617	9000.00

ACCT_ID	SEQNO	DATE	TYPE	AMOUNT
12345	1	20120618	D	8000.00
12345	2	20120622	D	1000.00

Repeating Groups: NoSQL

- ✓ All Occurrences into the Same Target
- ✓ No Need for Sequence Number

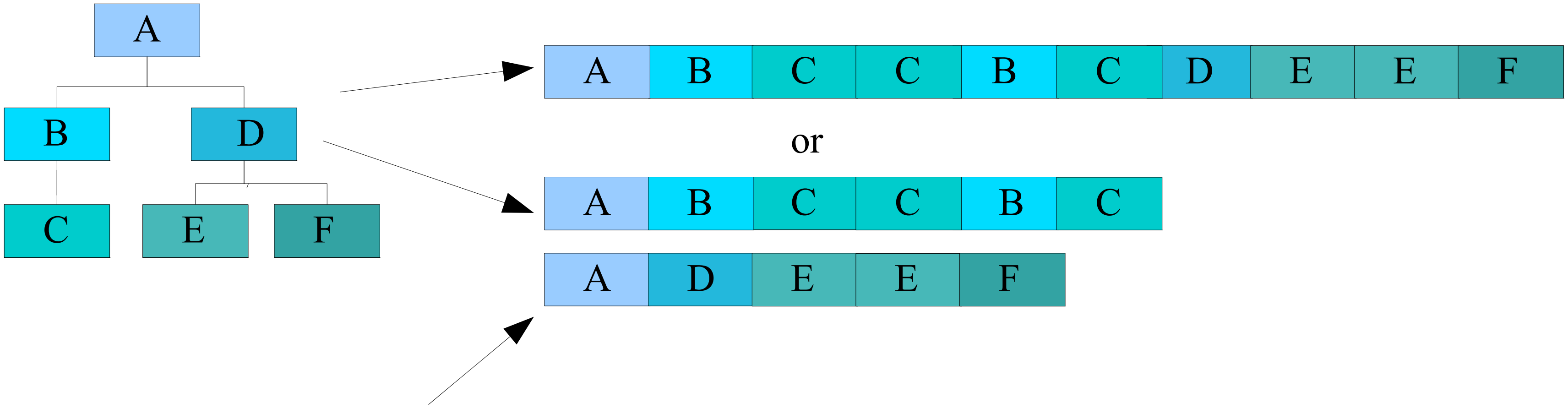
```
05   ACCT-ID                                           PIC 9(7).
05   ACCT-CRDATE                                       PIC X(8).
05   ACCT-BALANCE                                      PIC S9(13)V99 COMP-3.
05   ACCT-ACTIVITY OCCURS 100 TIMES.
        10 ACT-DATE                                    PIC 9(8).
        10 ACT-TYPE                                    PIC X.
        10 ACT-AMOUNT                                  PIC S9(11)V99  COMP-3.
```

ACCT_ID	ACCT_CRDATE	BALANCE	DATE	TYPE	AMOUNT	DATE	TYPE	AMOUNT
12345	20120617	9000.00	20120618	D	8000.00	20120622	D	1000.00

ETL and Changed Data Capture (CDC)

ETL

- ✔ High Level of Control Over Level of De-Normalization
- ✔ Can Combine Many Segments in Target Row / Document
- ✔ Requires that ETL Tool can Handle Consolidation during Extract



A

B D

C E F

A B C C B C D E E F

or

A B C C B C

A D E E F

Changed Data Capture

- ✔ May Dictate that Target not Fully Denormalized
- ✔ Capture Along One (1) Branch of IMS DB Record
- ✔ Path / Lookups *may* be Required

Summary

- Let the Business Drive the Effort
- Temper the Exuberance
- Align with Enterprise Data
- Lose the Relational Model Mentality
- Use an Iterative Approach for Implementation
- Be Ready to Change Direction → Technology Changes
- Select the Correct Tool Vendor
  - ✓ Specializes in Heterogeneous Data Movement
  - ✓ Bulk Data Extract & Changed Data Capture / Replication
  - ✓ Willing to Participate with Design & Deployment

Best Practices Series

Populating Big Data Repositories from IMS

Prepared for the:

Virtual IMS User Group

7 October 2014