CORNELL NYCTECH

Databases and Storage

cs5356
Daniel Doubrovkine
@dblockdotorg

DBMS = Data Management System

Software

Hardware

Data

It's a bit special …

Persistent

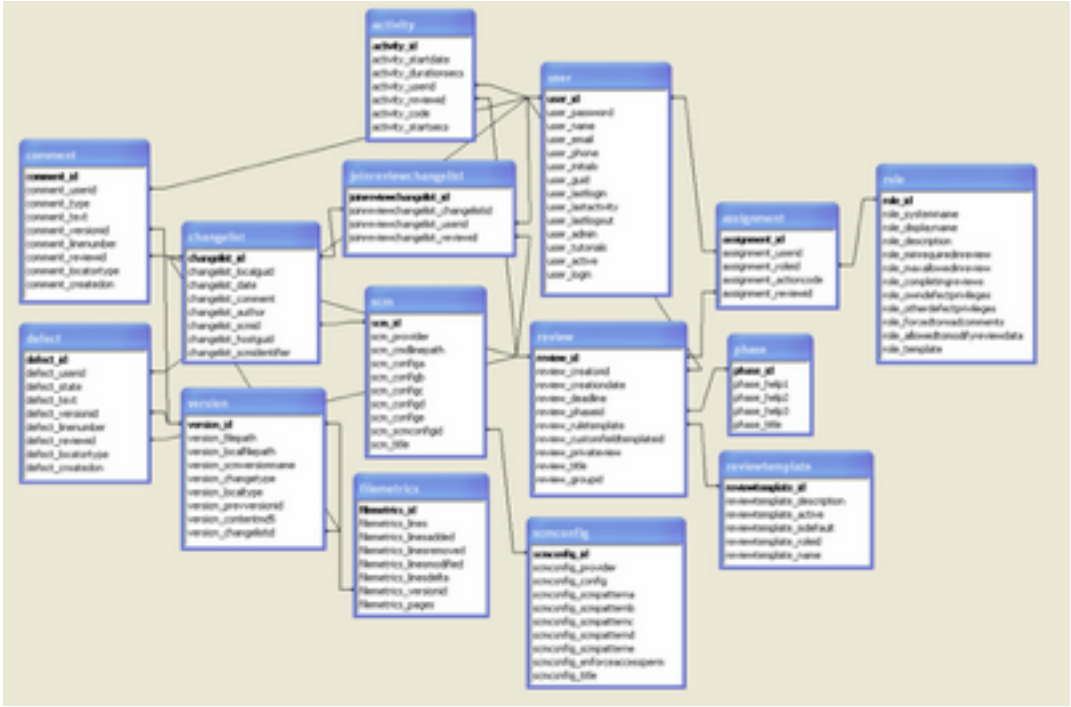
Available

Manipulation

Create
Retrieve
Update
Delete

CORNELL NYCTECH

Data Definition



Storage

Capacity
Access Speed
Cost

Memory, Solid State Disks, Spinning Disks, Magnetic Tapes, …

In-Memory Databases

Memory Snapshots to Disk

Spinning Disk / SSD Databases

Buffer

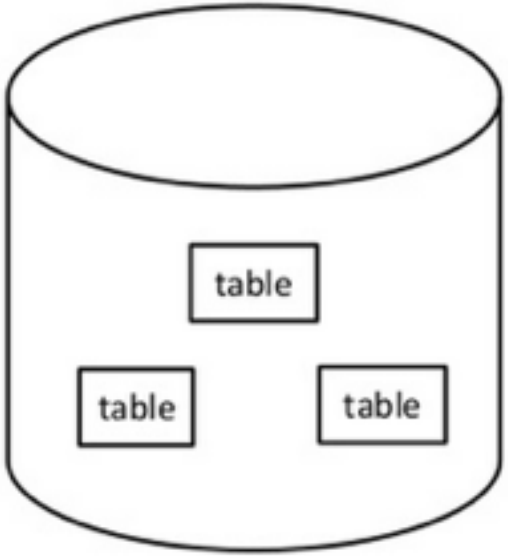
Storage

RAID or Replication

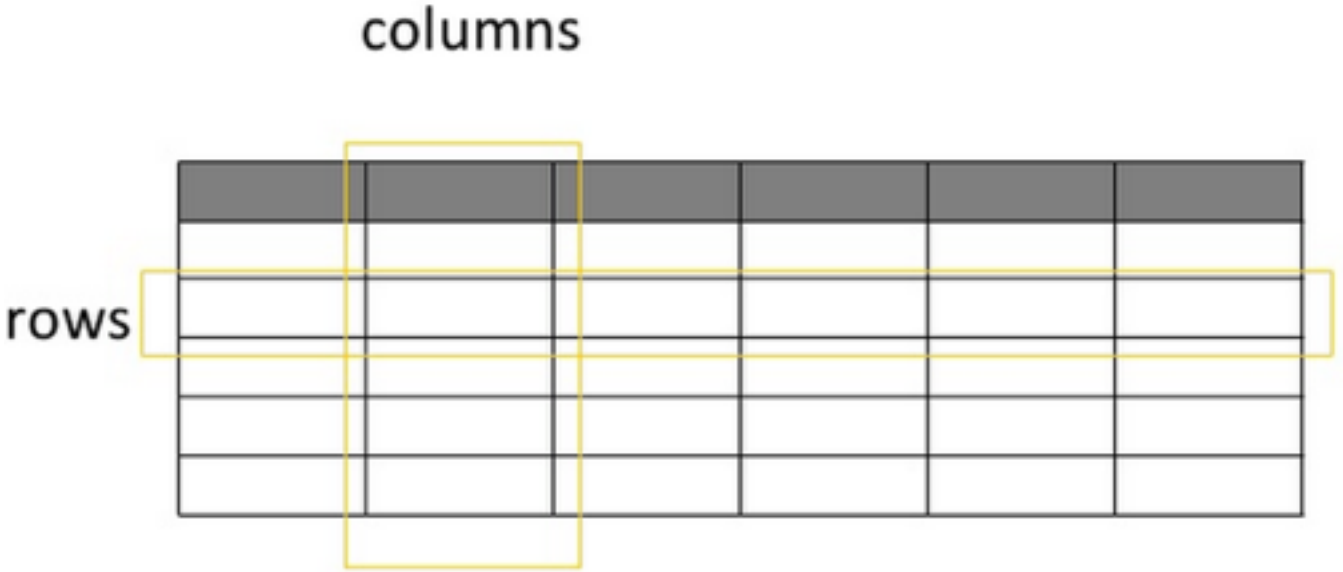
Offline Storage

Tape

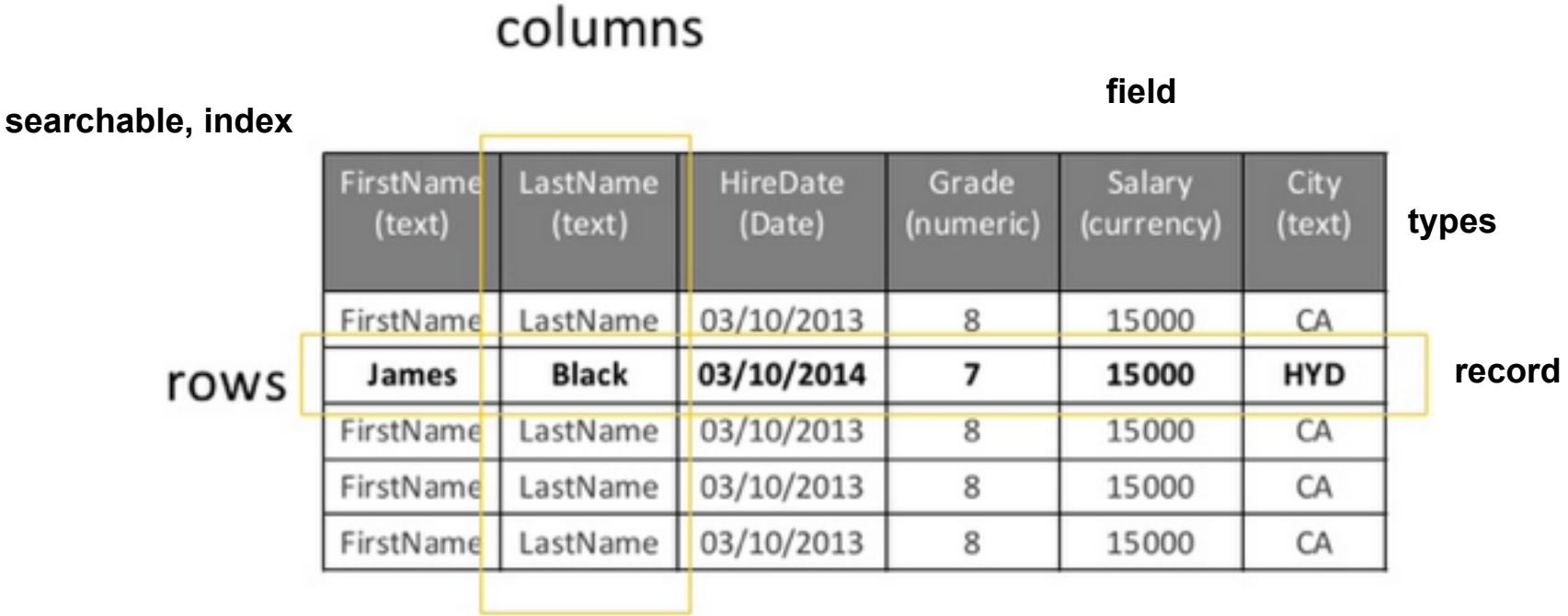
Type: Relational Databases



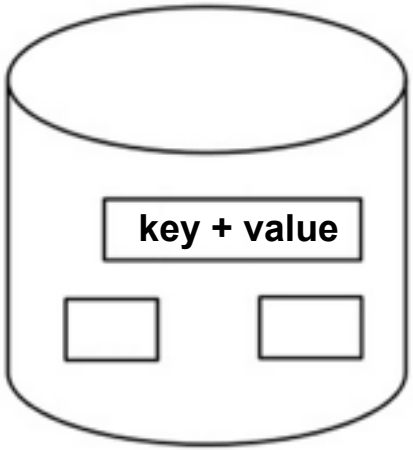
RDBMS: Table

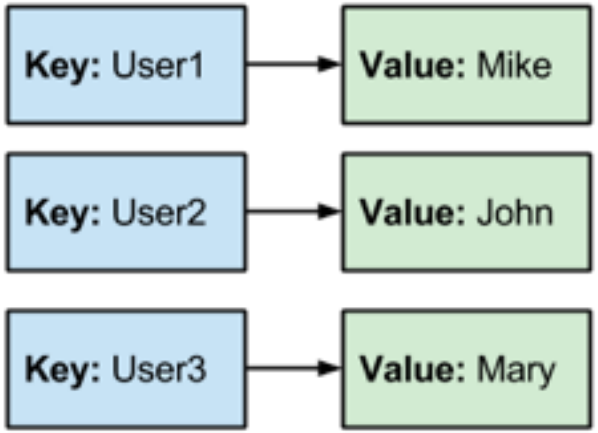


RDBMS: Fields and Types



Type: Key/Value





redis

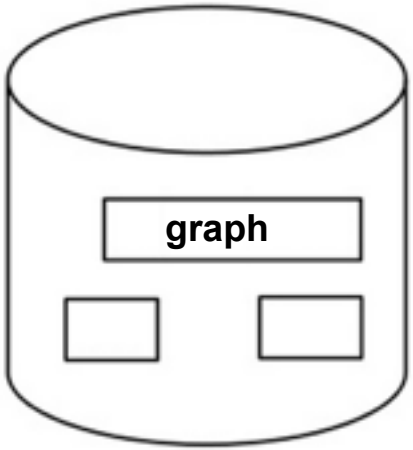
Type: Document Databases


```
<Books>
    <Book ISBN="0553212419">
        <title>Sherlock Holmes: Complete Novels...
        <author>Sir Arthur Conan Doyle</author>
    </Book>
    <Book ISBN="0743273567">
        <title>The Great Gatsby</title>
        <author>F. Scott Fitzgerald</author>
    </Book>
    <Book ISBN="0684826976">
        <title>Undaunted Courage</title>
        <author>Stephen E. Ambrose</author>
    </Book>
    <Book ISBN="0743203178">
        <title>Nothing Like It In the World</title>
        <author>Stephen E. Ambrose</author>
    </Book>
</Books>
```

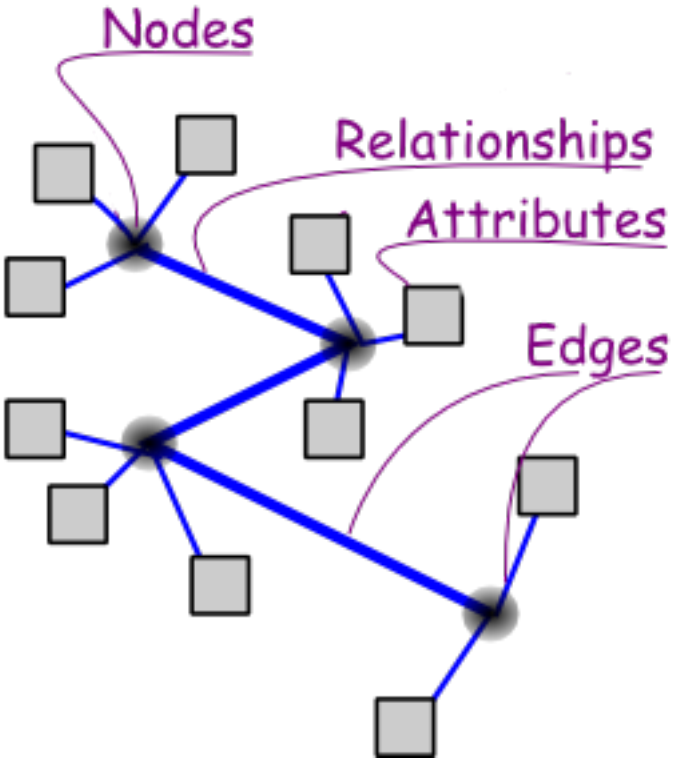
```
{
    hey: "guy",
    anumber: 243,
  - anobject: {
        whoa: "nuts",
      - anarray: [
            1,
            2,
            "thr<h1>ee"
        ],
        more: "stuff"
    },
    awesome: true,
    bogus: false,
    meaning: null,
    japanese: "明日がある。",
    link: http://jsonview.com,
    notLink: "http://jsonview.com is great"
}
```

mongoDB

Type: Graph Databases



neo4j



Type: Column Databases

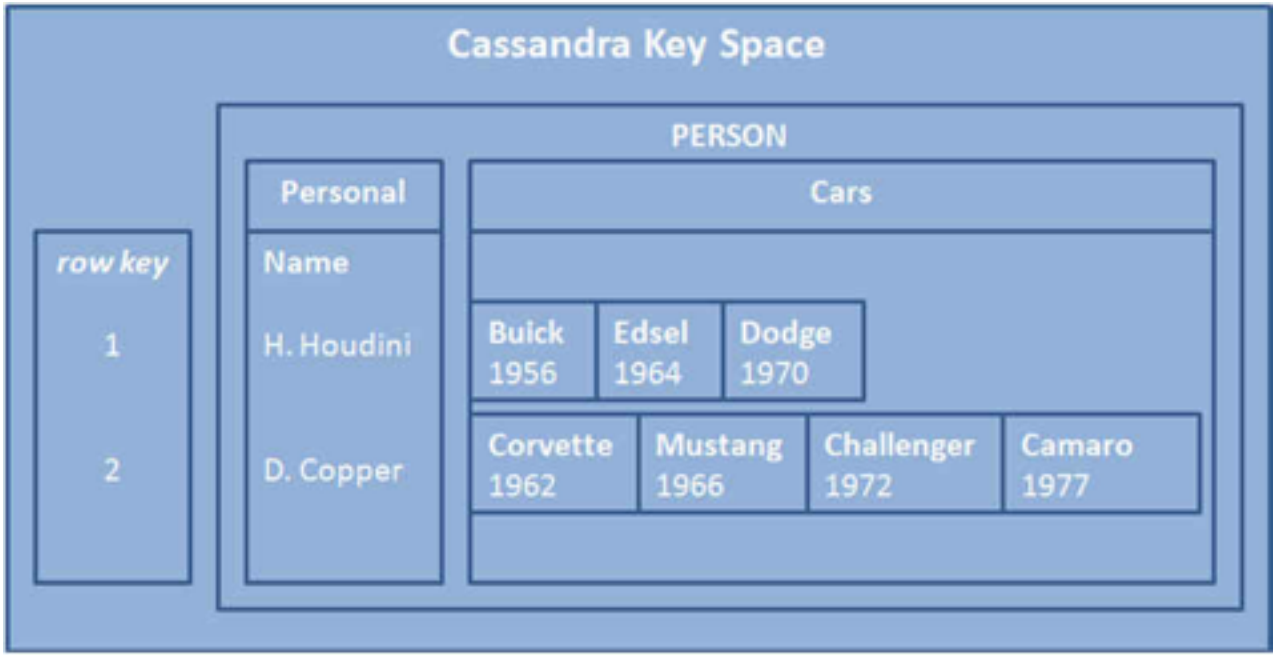


Figure 6 - Static/Dynamic Data

cassandra

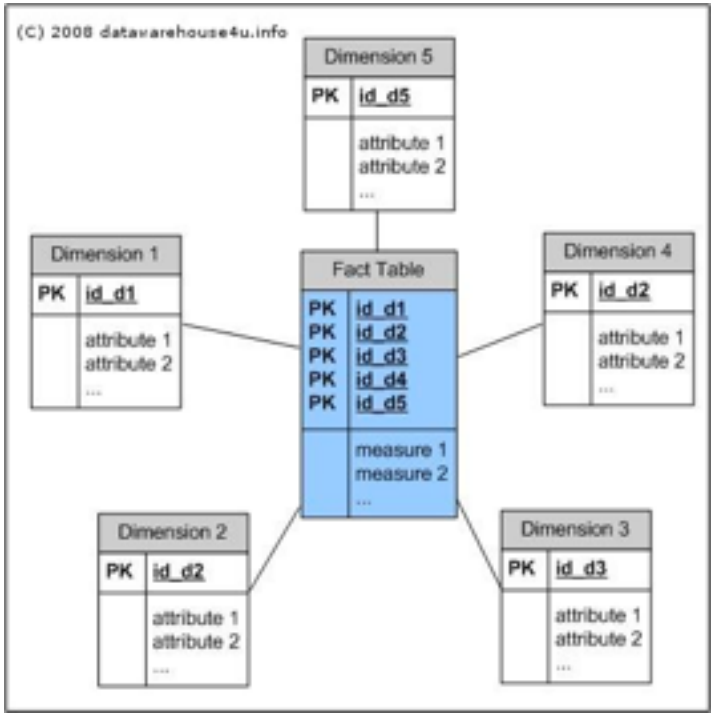
APACHE HBASE

Type: Datom Databases

Fact = Datom

Entity	Attribute	Value	Tx	op
21005	:name	"Stuart"	1000	assert
21005	:likes	tea	1000	assert

RDBMS: Star Schema



Query: SQL

```
SELECT e.ID, e.LastName, e.FirstName, pn.Number
FROM Employee e
LEFT OUTER JOIN PhoneNumber pn
ON e.ID = pn.ID
```

Results | Messages

	ID	LastName	FirstName	Number
1	1	Johnson	Joe	555-2323
2	2	Lewis	Larry	NULL
3	3	Thompson	Thomas	555-9876
4	4	Patterson	Patricia	NULL

Query: ?QL

db.users.find({ name: ”Bob” });

```
db.users.update(                          <-- collection
    { age: { $gt: 18 } },                 <-- update criteria
    { $set: { status: "A" } },            <-- update action
    { multi: true }                       <-- update option
)
```

Transactions

single logical operation

Atomicity — **all or nothing**

Consistency — **valid state, including constraints, triggers and cascades**

Isolation — **in parallel = serial**

Durability — **committed transactions persist in error**

BASE

Basic Availability
Soft State
Evenutal Consistency

Atomicity
~~Consistency~~
Isolation
Durability

BASE vs. ACID & CAP

Consistency — **all nodes see the same data at the same time**
note: not C in ACID (!), which is constrained data

Availability — **every request receives a response**

Partition Tolerance — **the system operates under network partition**

NoSQL

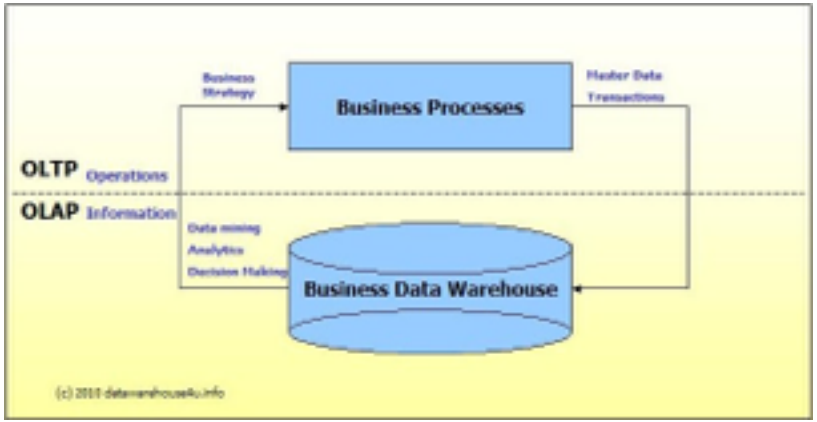
- Doesn’t use SQL
- BASE rather than ACID
- Schemaless

NoSQL Implementation

- Nodes Exchange Writes
- Deal with Conflicts

Transactional vs. Analytical

OnLine
Transaction
Analytical
Processing



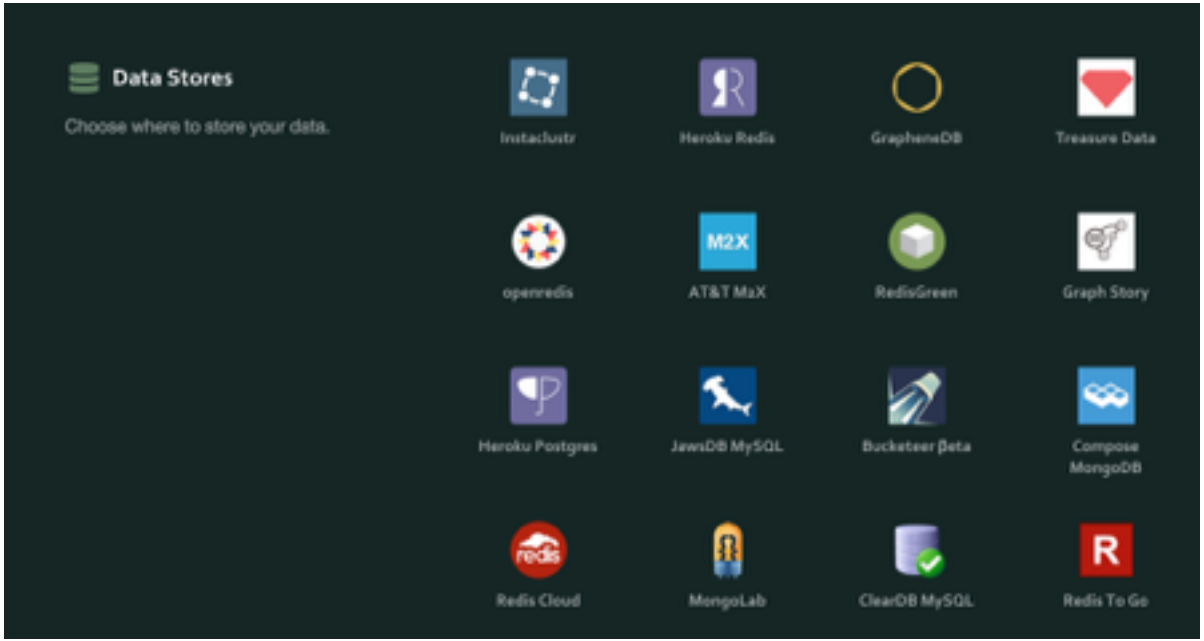
More?

Relational OLAP

Hybrid

Multidimensional

I just want a database …



I just want to store files …

Simple
Storage
Service

amazon web services | S3

I just want to …

… write some code!