CHULA ENGINEERING
Foundation toward Innovation
COMPUTER

INTRODUCTION TO NOSQL

Peerapon Vateekul, Ph.D.

Department of Computer Engineering, Faculty of Engineering, Chulalongkorn University

Peerapon.v@chula.ac.th

OUTLINES

- What is NoSQL?
- Why NoSQL?
- Types of NoSQL
- Key-Value databases
- Document databases
- Column-Store databases
- Graph databases

WHAT IS NOSQL ?

- NoSQL means Not Only SQL, implying that when designing a software solution or product, there are more than one storage mechanism that could be used based on the needs.
- NoSQL does not have a prescriptive definition, but we can make a set of common observations, such as:
  - Not using the relational model (ACID)
  - Scalability (Distributed System)
  - Schema-less (Variety, Flexibility)

WHY NOSQL ?

Dynamic Schemas (Variety, Flexibility)

NoSQL databases are built to allow the insertion of data without predefined schema

Auto-sharding (Scalability)

NoSQL databases usually support auto-sharding, meaning that they natively and automatically spread data across an arbitrary number of servers

Replication (Tolerance)

Most NoSQL databases also support automatic database replication to maintain availability in the event of outages

Integrated Caching (Availability)

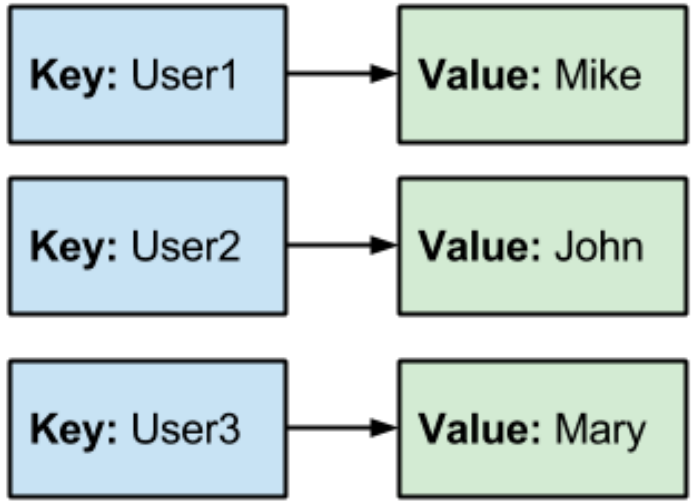
Many NoSQL database technologies have excellent integrated caching capabilities, keeping frequently-used data in system memory as much as possible

TYPES OF NOSQL

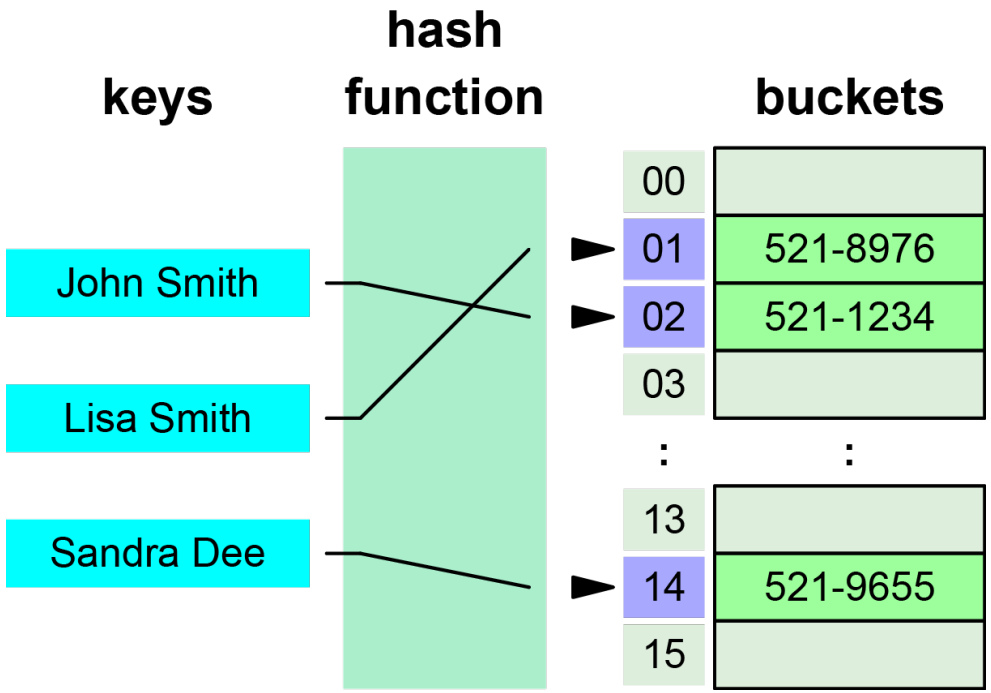
1. Key-Value databases
2. Document databases
3. Column-Store databases
4. Graph databases

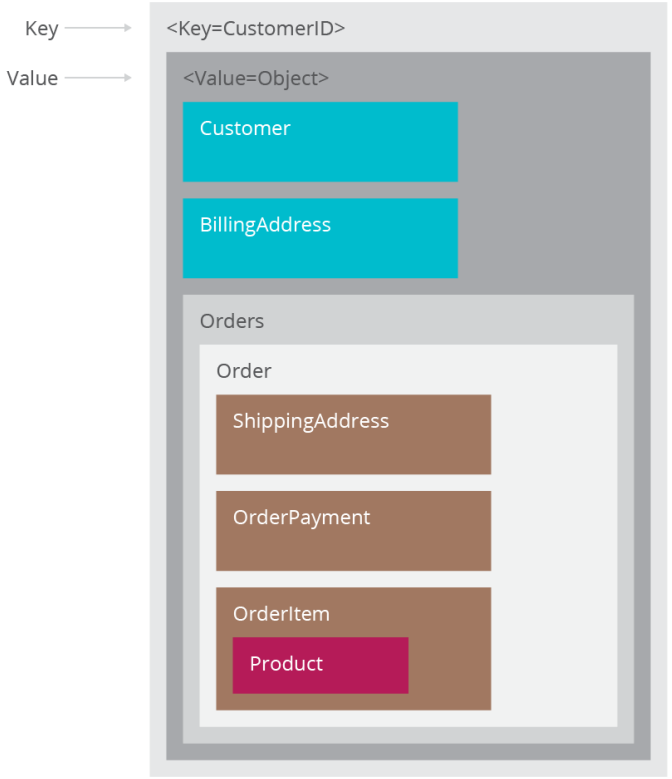
1) KEY-VALUE DATABASES

- Data is stored in a key-value pairs, where attribute is the KEY and content is the VALUE.
- Data can only be queries and retrieved using the keys only.
- We can use tricks, e.g., enumerated keys, to implement range queries.



KEY-VALUE DATABASES (CONT.)





KEY-VALUE DATABASES (CONT.)

Table A

Key	Value
Post:16082015:220030:Bank	{“Im so sleepy”....}
Post:18092015:120010:Lucky	{“This burger is good”....}
Post:18092014:133005:Pang	{“Yummy, good pizza”....}

A key pattern is

“Post:Day:Time:Name”

In the Key-Value database, if we use a proper designed key, data can be queried easier.

In Redis database, to query “all posts in July 2024” from Table A:

```
> KEYS Post:??072024:*
```

KEY-VALUE DATABASES (CONT.)

Criteria

- Key-Value databases are well suited to applications that have **frequent small reads** and writes along with **simple data models**.
- Key-Value stores provide **high scalability**.

Use cases

- **Caching data** from relational databases to improve performance
- Tracking transient attributes in **a web application**, such as a shopping cart
- Storing configuration and user data information for **mobile applications**
- **Storing data form sensors (IoT)**

Caching & IoT

KEY-VALUE DATABASES (CONT.)

Key-value based software

- Redis
- Amazon DynamoDB
- Memcached
- Hazelcast

2) DOCUMENT DATABASES

<Key=CustomerID>

```
{
   “customerid”: “fc986e48ca6”
   “customer”:
   {
   “firstname”: “Pramod”,
   “lastname”: “Sadalage”,
   “company”: “ThoughtWorks”,
   “likes”: [ “Biking”,”Photography” ]
   }
   “billingaddress”:
   { “state”: “AK”,
      “city”: “DILLINGHAM”,
      “type”: “R”
    }
}
```

Key

DOCUMENT DATABASES

- Designed for storing, retrieving, and managing document-oriented information, also known as semi-structured data.
- Can store the data that have the different set of data fields (columns).
- Each document is assigned a unique key, which is used to retrieve the document.
- Most of the databases available under this category use XML, JSON

DOCUMENT DATABASES (CONT.)


```
{
 "created_at": "Tue Mar 21 20:50:14 +0000 2006",
   "text": "just setting up my twttr",
   "user": {
    "name": "Jack Dorsey",
    "screen_name": "jack"
                }
},
 {
   "created_at": "Sun Feb 09 23:25:34 +0000 2014",
   "id": 432656548536401920,
   "text": "POST statuses/update. Great way to start.
https://t.co/9S8YO69xzf (disclaimer, this was not posted via the
API).",
   "user": {
     "name": "TwitterDev",
    "screen_name": "TwitterDev",
    "description": "Developers and Platform Relations @Twitter. We
are developers advocates. We can't answer all your questions, but we
listen to all of them!",
    "url": "https://t.co/66w26cua1O"
                }
}
```


Document 1


```
{
 "id": "1",
 "name": "John Smith",
 "isActive": true,
 "dob": "1964-30-08"
}
```

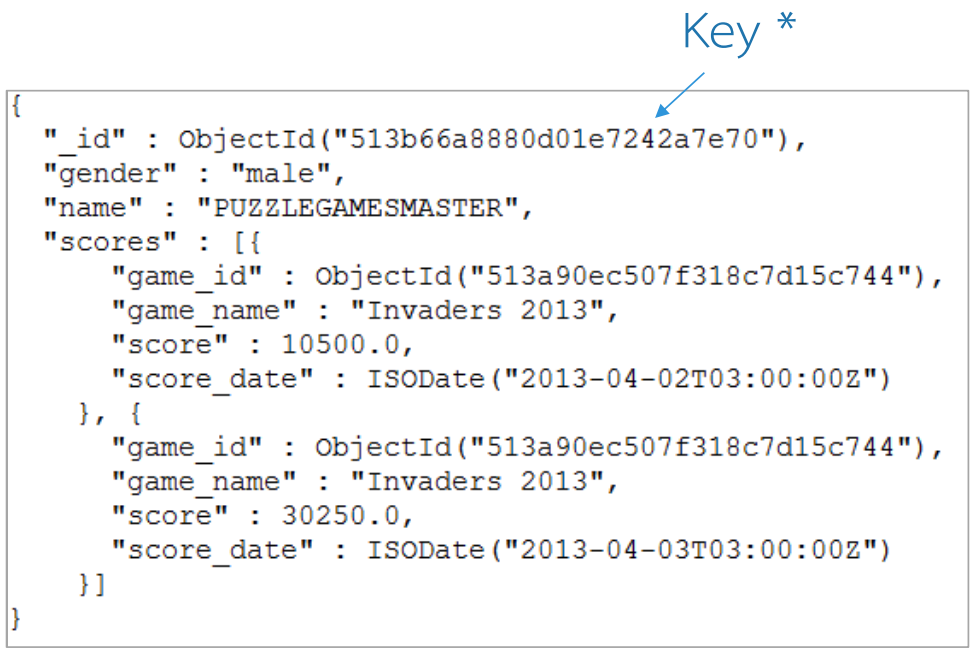

Document 2


```
{
 "id": "2",
 "fullName": "Sarah Jones",
 "isActive": false,
 "dob": "2002-02-18"
}
```


Document 3


```
{
 "id": "3",
 "fullName":
 {
  "first": "Adam",
  "last": "Stark"
 },
 "isActive": true,
 "dob": "2015-04-19"
}
```

DOCUMENT DATABASES (CONT.)



In MongoDB, to query documents that contain specific properties

```
db.collectionName.find ({properties})
```

For example, to query documents (records) , whose gender is male from collection (table) “players”

```
db.players.find ({gender : "male"})
```

* Key in MongoDB is automatically generated, so users won’t know the key , but can use other properties search in value.

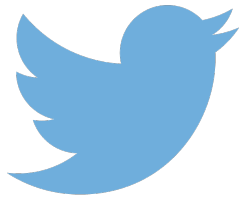
DOCUMENT DATABASES (CONT.)

Criteria

- Application that requires the ability to store varying attributes along with large amounts of data
- Application that stored data in standard formats like JSON, XML.

Use cases

- Back-end support for websites with high volumes of reads and writes
- Applications that use JSON data structures such as twitter data



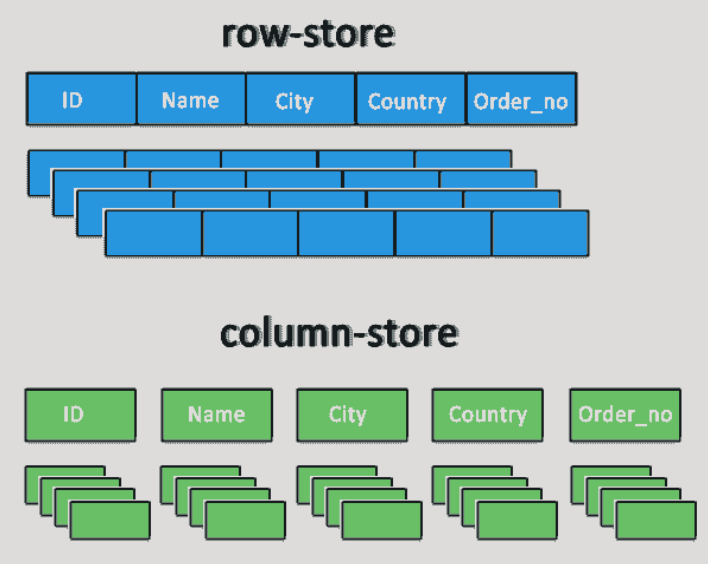
DOCUMENT DATABASES (CONT.)

Document-based software

- MongoDB
- Couchbase
- CouchDB
- IBM Cloudant
- Microsoft Azure DocumentDB

COLUMN-STORE DATABASES

- Designed for storing data tables as sections of columns of data associated with a row key .
- All columns are treaded individually and values of single column are stored together.
- Having stored data in wide-Column-Stores offer very high performance and a highly scalable architecture.

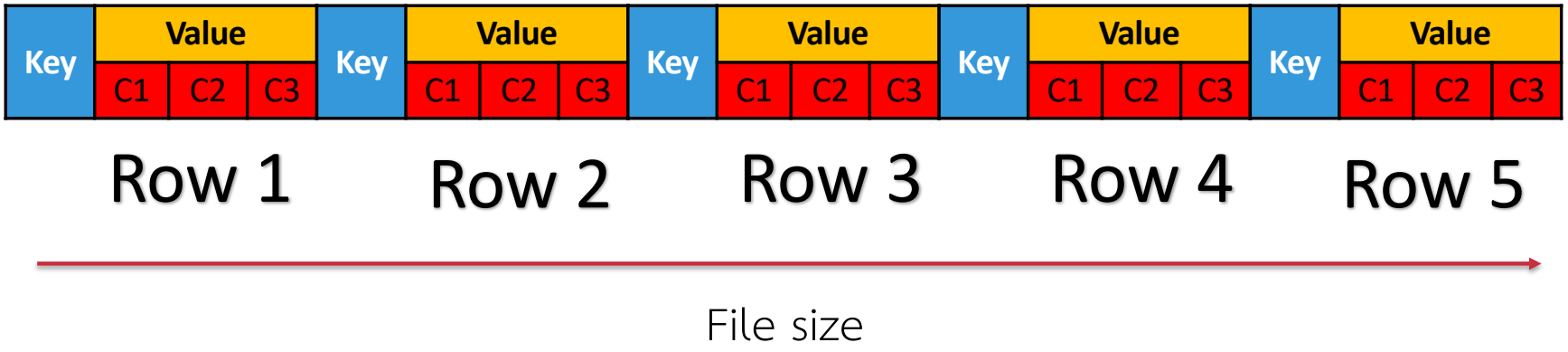


Column Families

row key	personal data		professional data	
employee ID	Name	City	Title	Salary
1342	Joseph	New York City	Product Manager	$150,000
1543	Frank	Boston	Software Engineer	$160,000
7643	David	San Francisco	Data Scientist	$130,000

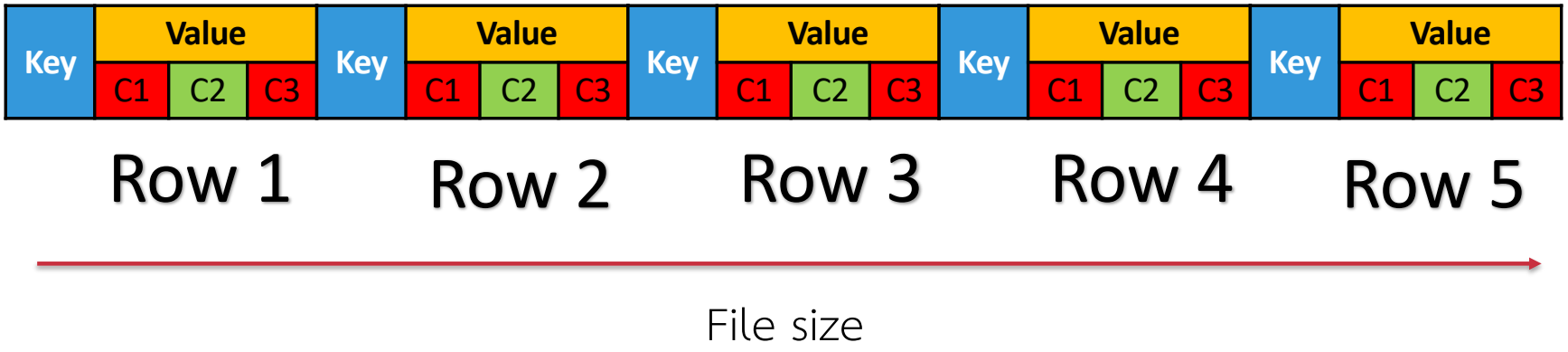
COLUMN-STORE DATABASES (CONT.)

On a disk, the **row store** database will store data like this



COLUMN-STORE DATABASES (CONT.)

So, if you want only data in column 2



Your disk must be read all data!!!

COLUMN-STORE DATABASES (CONT.)

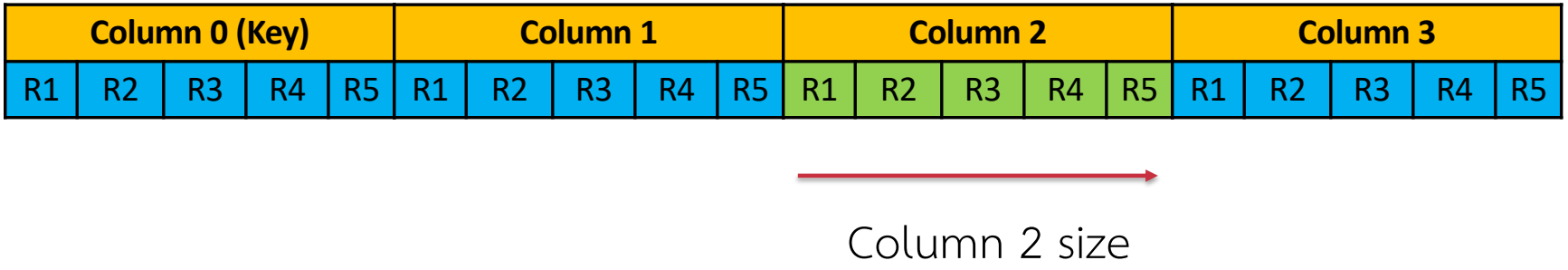
But in **Column-Store** database, it will store data like this

Column 0 (Key)					Column 1					Column 2					Column 3				
R1	R2	R3	R4	R5	R1	R2	R3	R4	R5	R1	R2	R3	R4	R5	R1	R2	R3	R4	R5

File size

COLUMN-STORE DATABASES (CONT.)

So, if you want only data in column 2



Just read the data of column 2

COLUMN-STORE DATABASES (CONT.)

id	name	address	gender	age
1	Bob	USA	Male	16
2	Lucy	Brazil	Female	50
3	Dum	Thailand	Male	38

Table A has a 10 GB data with

- column id : 1 GB
- column name : 2 GB
- column address : 5GB
- column gender : 0.5 GB
- column age :1.5 GB

- If a query **only uses column age** (e.g., compute an average), at most 1.5 GB of data will be processed by a **Column-Store**
- In a **row store**, the full 10 GB will be processed

Less Data = Faster !!!

COLUMN-STORE DATABASES (CONT.)

Criteria

- Large volumes of data, read performance, and high availability
- Applications that require the ability to always aggregate or query on column

Use cases

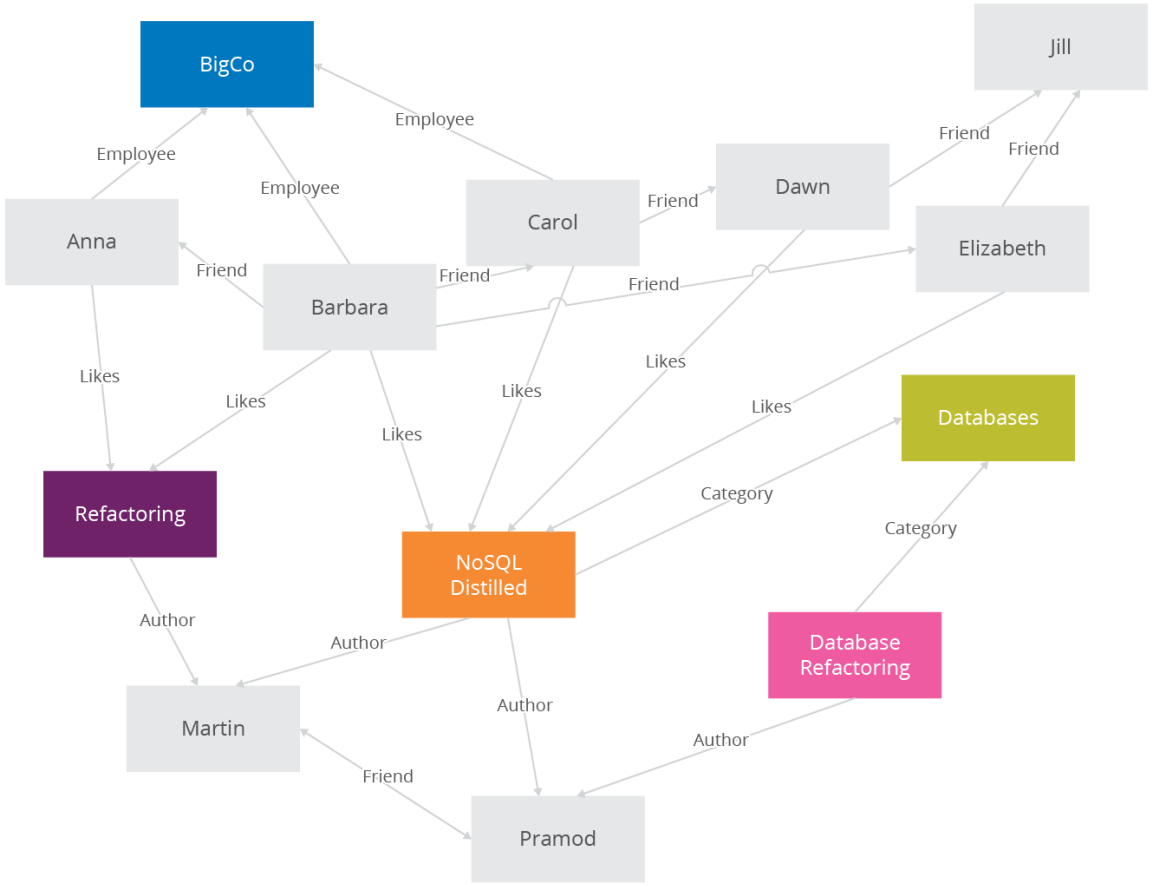
- Security analytics using network traffic and log data model
- Big Science data, e.g., bioinformatics using genetic and proteomic data
- Stock market analysis using trade data

COLUMN-STORE DATABASES (CONT.)

Column-Store based software

- Cassandra
- Hbase (Hadoop)
- Google Cloud Bigtable
- Accumulo
- ScyllaDB

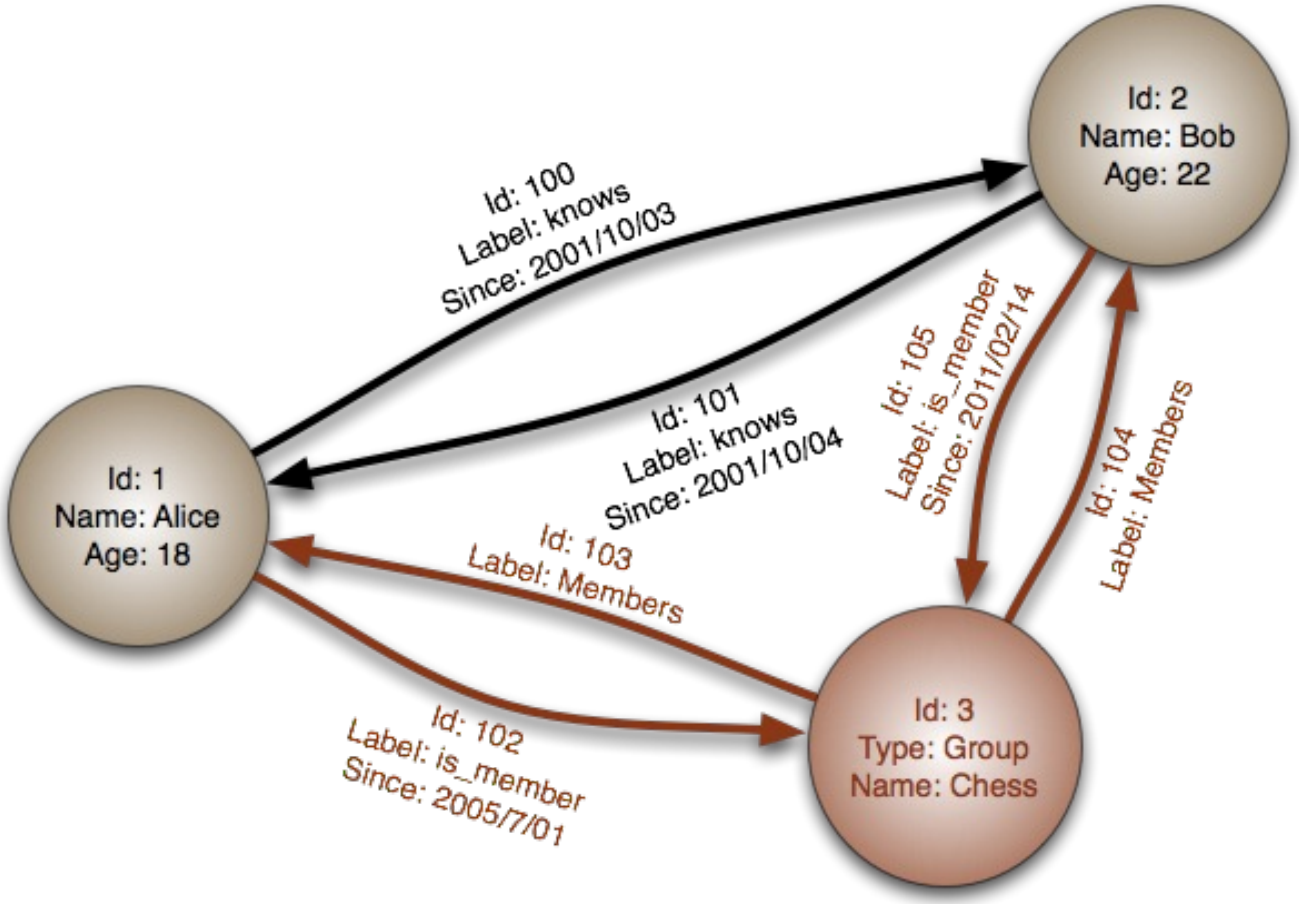
GRAPH DATABASES



GRAPH DATABASES (CONT.)

- Graph databases use edges and nodes to represent and store the data.
  - Nodes are represented as objects
  - Edges act as relations among these nodes
- Every node and edge can store additional properties.
- The relationship between nodes is not calculated at query time, but is actually persisted as a relationship.
- Nodes can have different types of relationships between them.

GRAPH DATABASES (CONT.)



GRAPH DATABASES (CONT.)

An example of mobile users that can be stored in graph databases

number	age	gender	rnCode	promotion	noOfCall	noOfReceive
089-1000000	23	male	AIS	703 B	19	19
089-1000001	20	female	AIS	422 B	13	6
089-1000002	25	male	DTAC	574 B	1	14
089-1000003	26	female	AIS	514 B	20	1
089-1000004	20	female	DTAC	449 B	9	20

a_number	b_number	startDate	startTime	callDay	duration
089-1000038	089-1000060	20160108	20.52	Sunday	327
089-1000170	089-1000141	20160114	6.16	Friday	560
089-1000000	089-1000134	20160121	21.1	Sunday	197
089-1000153	089-1000188	20160110	1.36	Wednesday	202
089-1000190	089-1000065	20160114	17.3	Saturday	300

Node Data

Edge Data

GRAPH DATABASES (CONT.)

For example, in Neo4J graph database, there are many functions to manage the data such as

```
MATCH (n:Node {property})
Return n
```

Return all nodes that match with the property

Ex: MATCH (n:Node:{name: "Tomas"})
return n

```
MATCH (n:Node) -[r:type]-> (m:Node)
WHERE conditon
Return r
```

Return all relations that match with the type and we can add the conditon in the function

Ex: (n:Node) -[r:Call]-> (m:Node)
WHERE r.startTime > 12.00 AND r.duration < 60
return r

GRAPH DATABASES (CONT.)

Criteria

- Problems that lend themselves to representations as networks of connected entities are well suited for graph databases.
- Application that need to focus on relations of entities.

Use cases

- Network and IT infrastructure management
- Recommending products and services
- Social networking

GRAPH DATABASES (CONT.)

Graph-based software

- Neo4j
- OrientDB 
- Titan
- Virtuoso
- ArangoDB