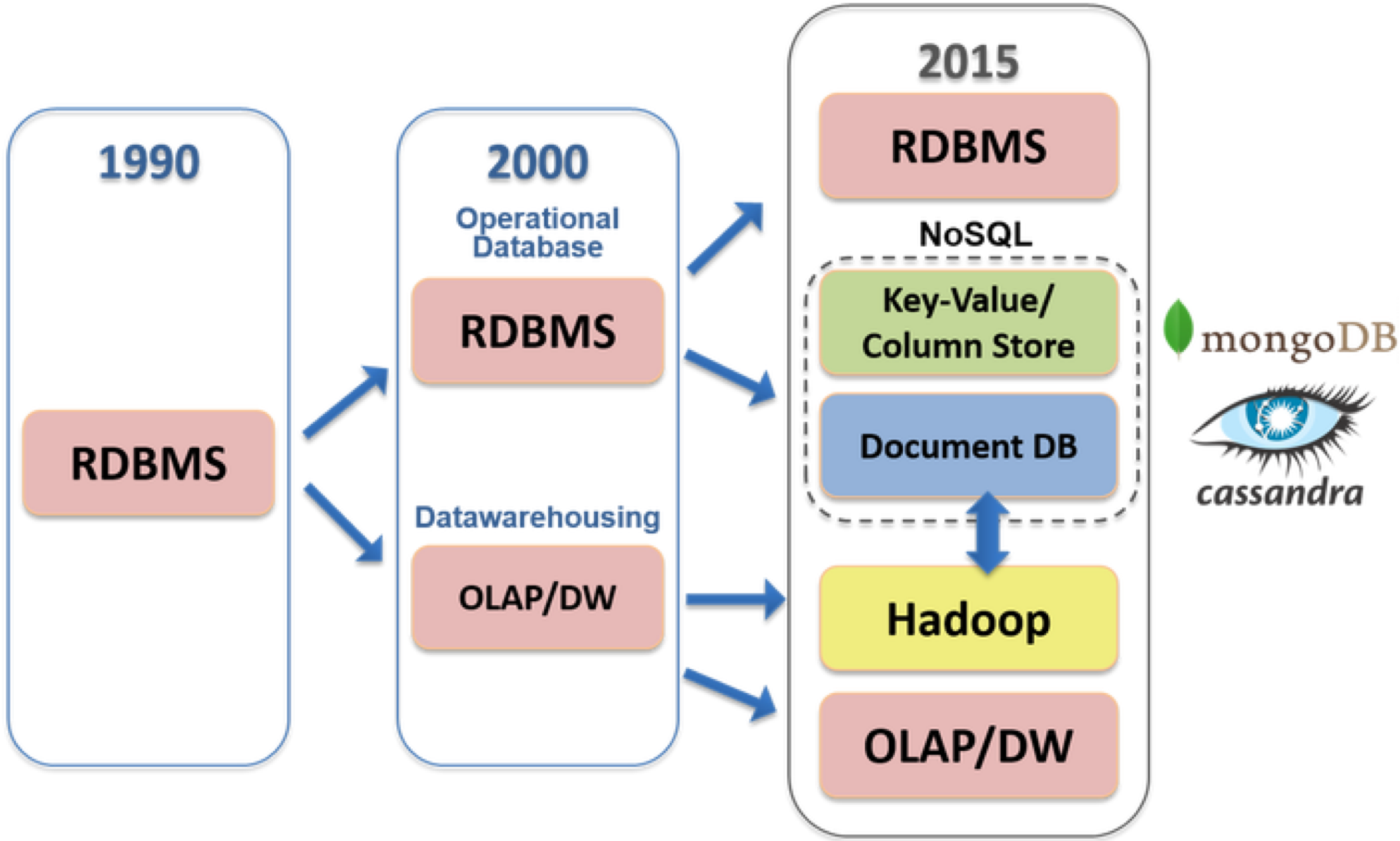
CSCI 5408

Dr. Saurabh Dey

saurabh.dey@dal.ca

Outline

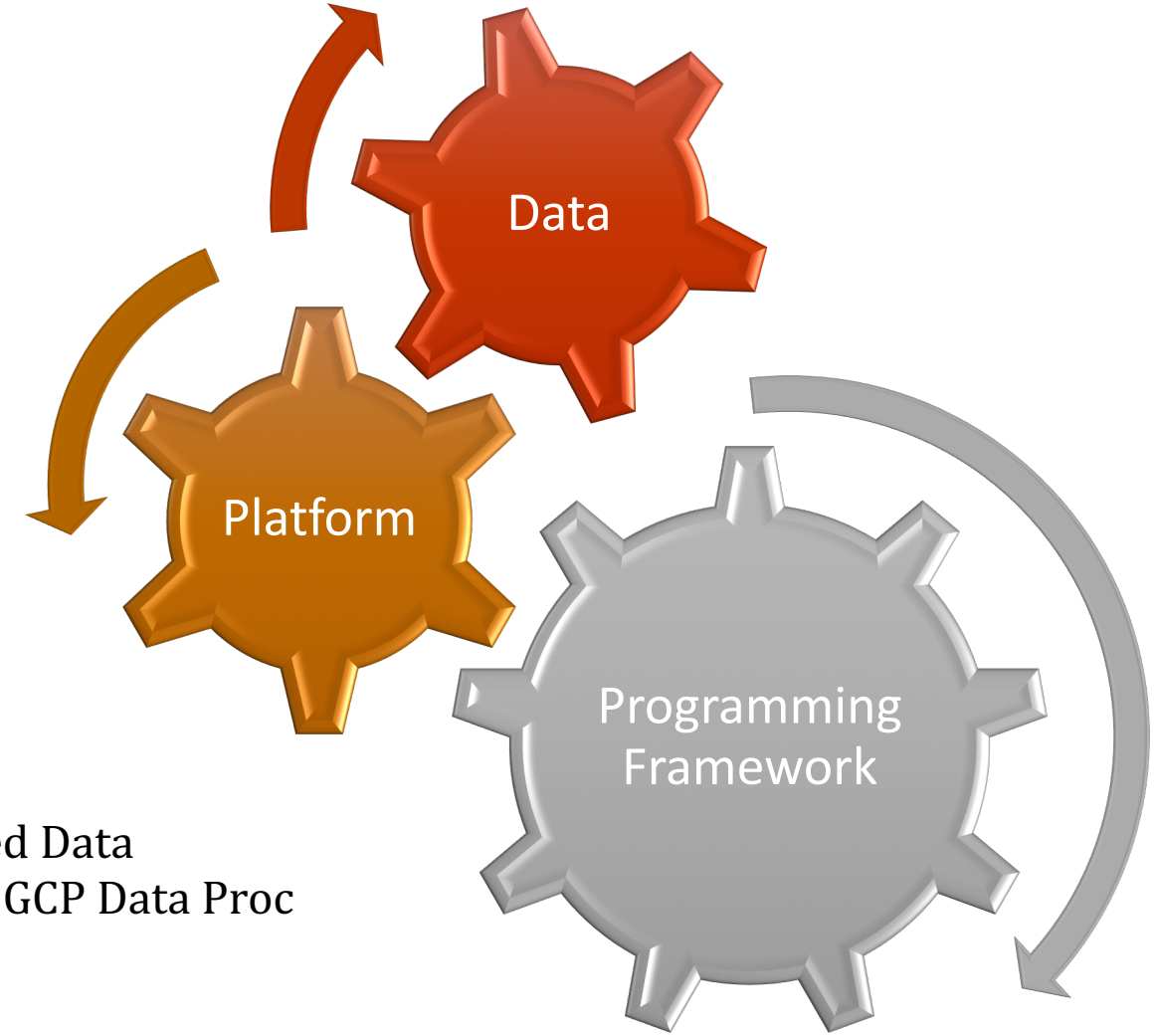
- NoSQL Database types, and structures



Source: http://www.amberoon.com/CarpeDatumRx/bid/363664/The-NoSQL-Ecoystem-A-C-Level-Guide

Can we build a framework to process data?

1. Unstructured Data in blocks
2. Hadoop & MapReduce



1. Unstructured Data
2. AWS EMR / GCP Data Proc

Where to store the data? Can I give it a structure?

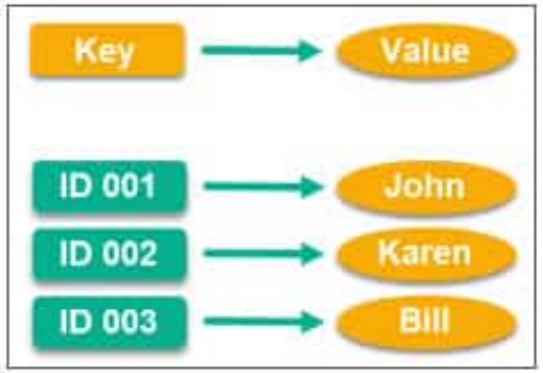
SQL Databases

Table

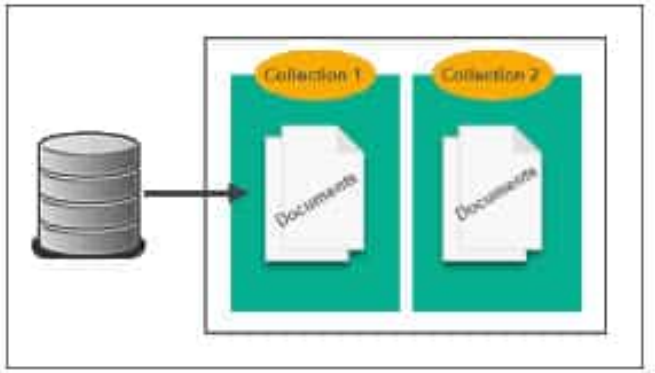
ID	Name	Grade	GPA
001	John	Senior	4.00
002	Karen	Freshman	3.67
003	Bill	Junior	3.33

NoSQL Databases

Key-value



Document



Graph

Wide-column

Row-oriented

ID	Name	Grade	GPA
001	John	Senior	4.00
002	Karen	Freshman	3.67
003	Bill	Junior	3.33

Column-oriented

Name	ID
John	001
Karen	002
Bill	003

Grade	ID
Senior	001
Freshman	002
Junior	003

GPA	ID
4.00	001
3.67	002
3.33	003

NoSQL

NoSQL CATEGORY	EXAMPLE DATABASES	DEVELOPER
Key-value database	Dynamo Riak Redis Voldemort	Amazon Basho Redis Labs LinkedIn
Document databases	MongoDB CouchDB OrientDB RavenDB	MongoDB, Inc. Apache OrientDB Ltd. Hibernating Rhinos
Column-oriented databases	HBase Cassandra Hypertable	Apache Apache (originally Facebook) Hypertable, Inc.
Graph databases	Neo4J ArangoDB GraphBase	Neo4j ArangoDB, LLC FactNexus

(1) Key-value (KV) databases: conceptually the simplest of the NoSQL data models

Store data as a collection of key-value pairs organized as buckets which are the equivalent of tables

Key-value (KV) databases

Bucket = Customer

Key	Value
10010	“LName Ramas FName Alfred Initial A Areacode 615 Phone 844-2573 Balance 0”
10011	“LName Dunne FName Leona Initial K Areacode 713 Phone 894-1238 Balance 0”
10014	“LName Orlando FName Myron Areacode 615 Phone 222-1672 Balance 0”

(2) Document databases: similar to key-value databases and can almost be considered a subtype of KV databases

Store data in key-value pairs in which the value components are encoded documents grouped into large groups called collections

Document databases (type of Key-value)

Collection = Customer

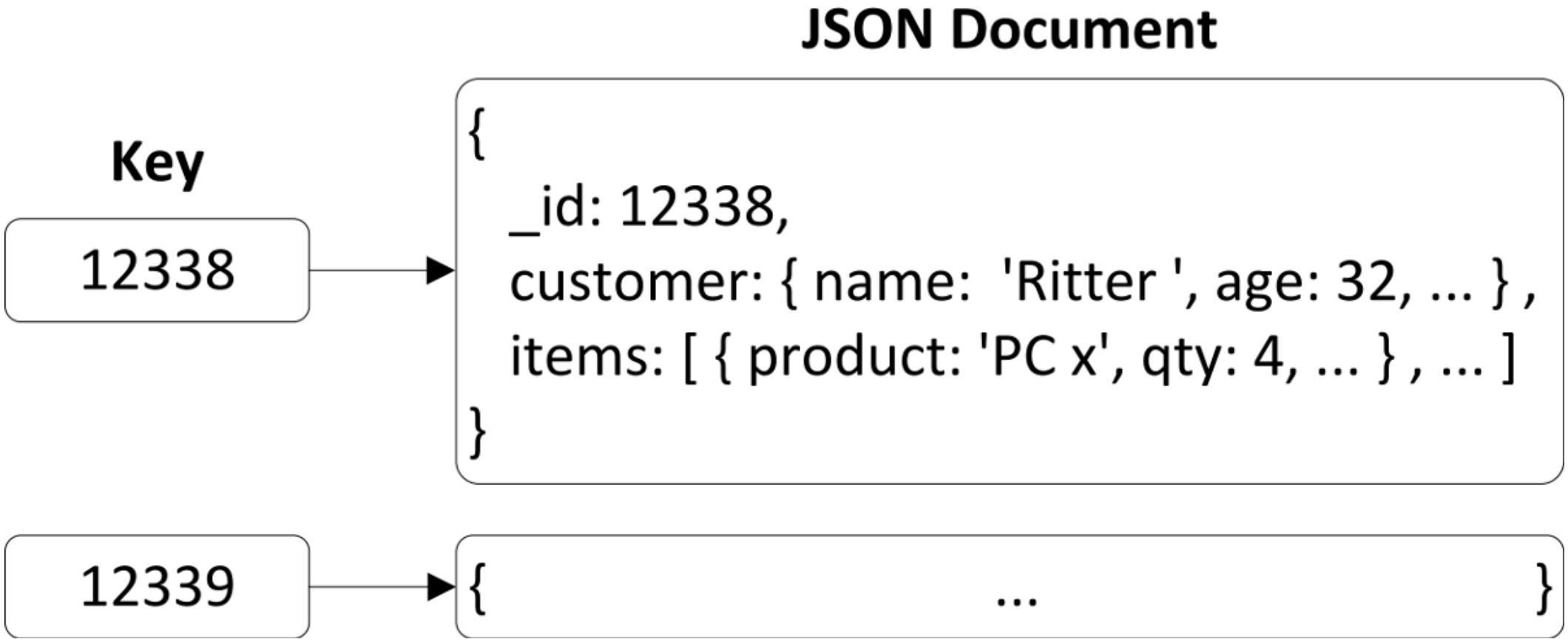
Key	Document
10010	{LName: “Ramas”, FName: “Alfred”, Initial: “A”, Areacode: “615”, Phone: “844-2573”, Balance: “0”}
10011	{LName: “Dunne”, FName: “Leona”, Initial: “K”, Areacode: “713”, Phone: “894-1238”, Balance: “0”}
10014	{LName: “Orlando”, FName: “Myron”, Areacode: “615”, Phone: “222-1672”, Balance: “0”}

Document databases

Does not store as a table. It could be in a JSON, XML or other format

```
{"firstName": "John",
 "lastName" : "Smith",
 "age"         : 25,
 "address"   :
   {"streetAdr" : "21 2nd Street",
    "city"        : "New York",
    "state"       : "NY",
    "zip"         : "10021"},
 "phoneNumber":
   [{"type"  : "home",
    "number": "212 555-1234"},
   {"type"  : "fax",
    "number" : "646 555-4567"}]
 }
```

Internally data is stored as JSON (Semi-structred)



Source: https://medium.baqend.com/nosql-databases-a-survey-and-decision-guidance-ea7823a822d

(3) Column-oriented databases

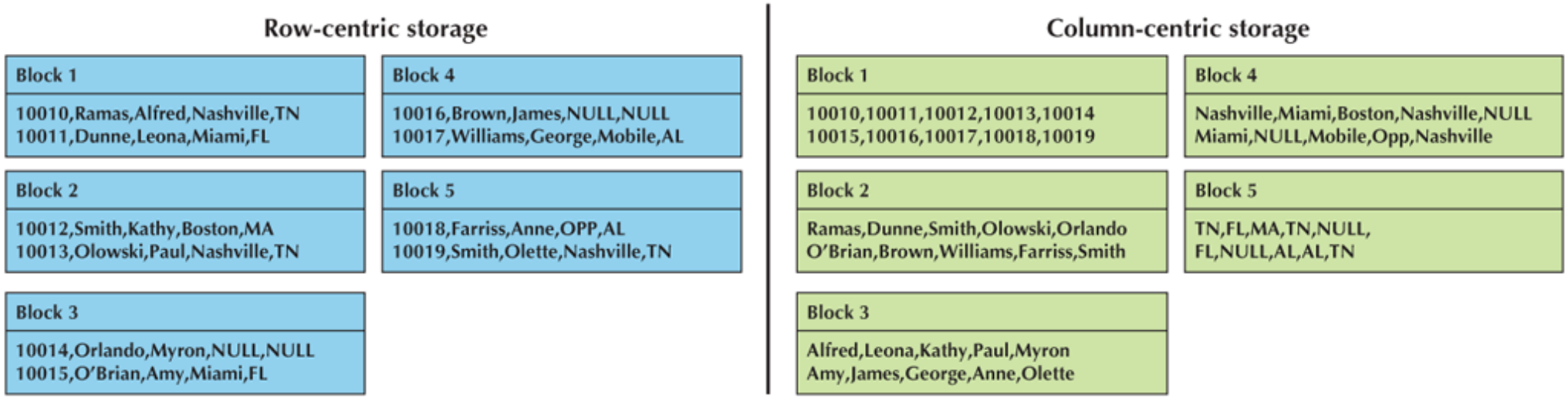
Column-oriented databases refers to two technologies

Column-centric storage: data stored in blocks which hold data from a single column across many rows

Row-centric storage: data stored in block which hold data from all columns of a given set of rows

CUSTOMER relational table

Cus_Code	Cus_LName	Cus_FName	Cus_City	Cus_State
10010	Ramas	Alfred	Nashville	TN
10011	Dunne	Leona	Miami	FL
10012	Smith	Kathy	Boston	MA
10013	Olowski	Paul	Nashville	TN
10014	Orlando	Myron		
10015	O'Brian	Amy	Miami	FL
10016	Brown	James		
10017	Williams	George	Mobile	AL
10018	Farriss	Anne	Opp	AL
10019	Smith	Olette	Nashville	TN



- The other use of the term **column-oriented database**, also called **column family** database

- This database model originated with Google's BigTable product.

Column Family Name	CUSTOMERS
Key	Rowkey 1
Columns	City: Nashville Fname: Alfred Lname: Ramas State: TN
Key	Rowkey 2
Columns	Balance: 345.86 Fname: Kathy Lname: Smith
Key	Rowkey 3
Columns	Company: Local Markets, Inc. Lname: Dunne

Column centric or Row Centric

Row centric

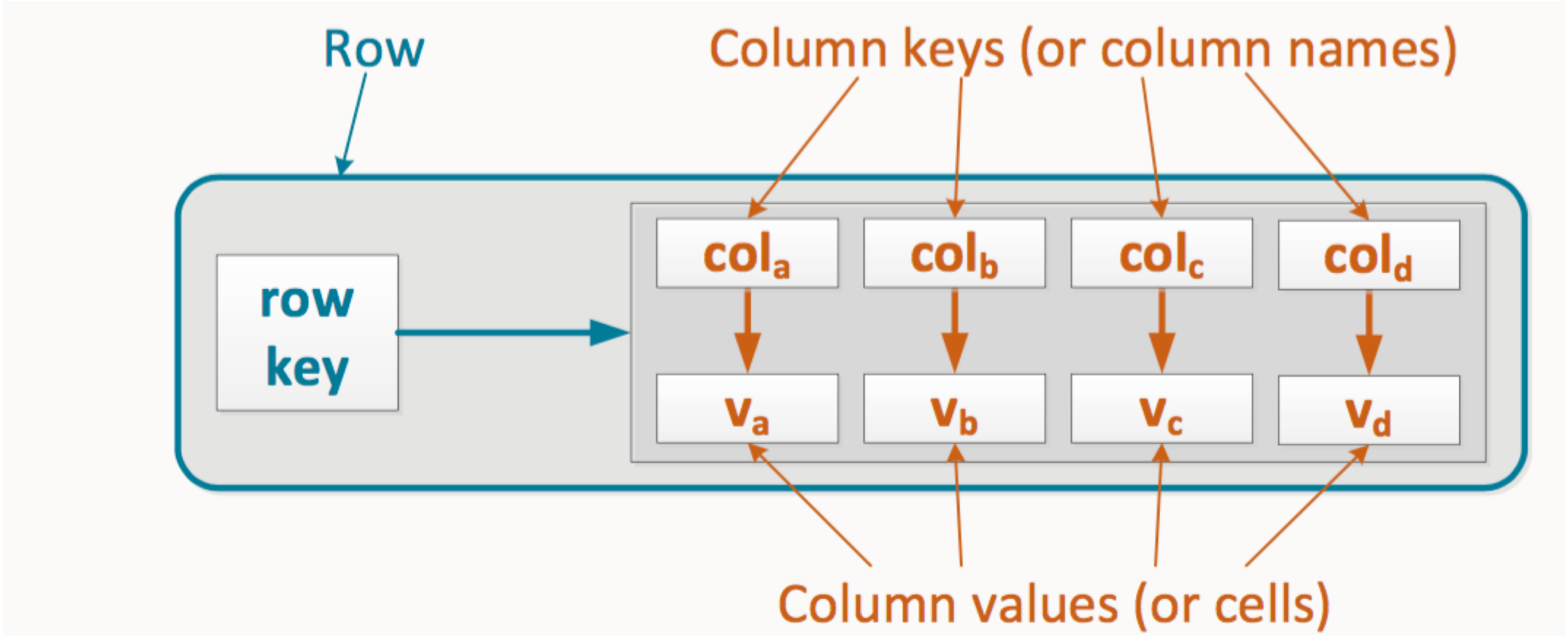
Emily - Halifax - E20 - Bob - Victoria - E23 - Alice - Moncton - E27

Name Column: Emily - Bob – Alice
City Column: Halifax – Victoria – Moncton
EmployeeID Column: E20 – E23 – E27

Column centric

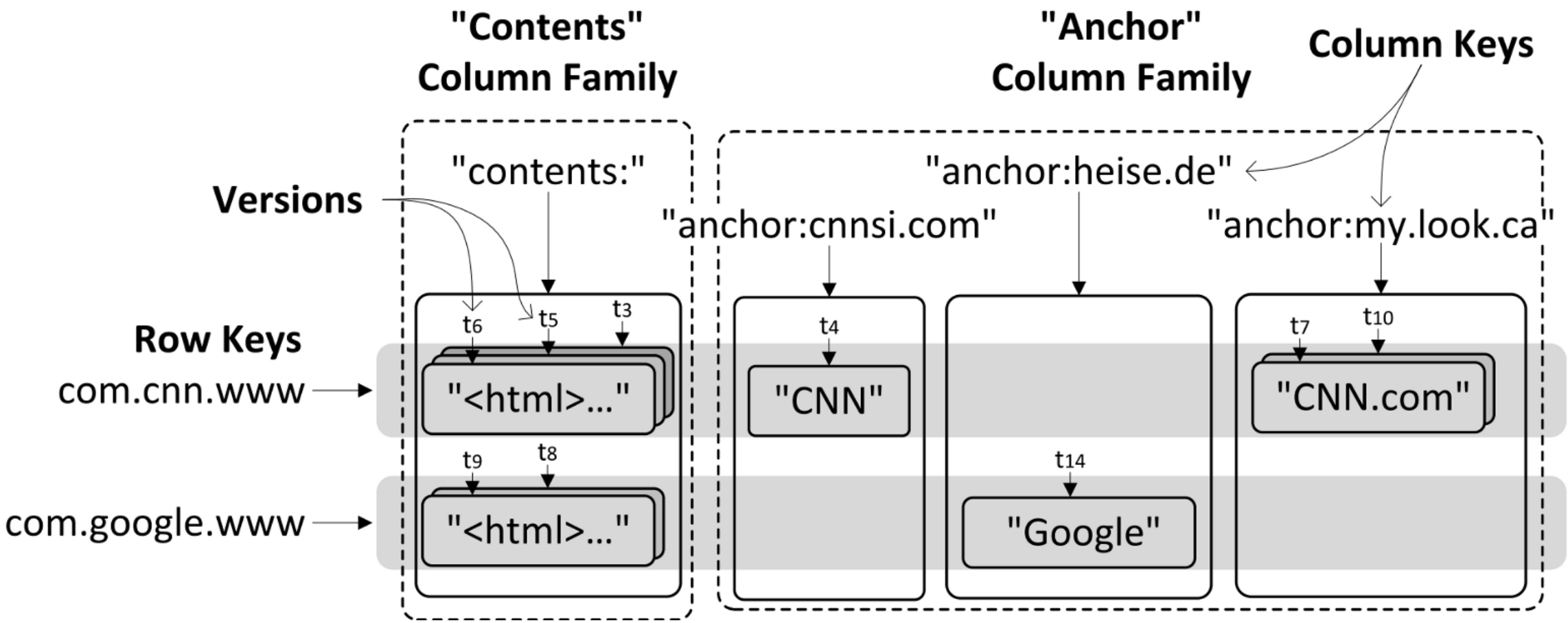
Row_key 1: **Person** [Name:Emily || City:Halifax || EmpID:E20] **Department** [Name: Sales]
Row_key 2: **Person** [Name:Bob || City:Victoria || EmpID:E23] **Department** [Name: Production]
Row_key 3: **Person** [Name:Alice || City:Moncton || EmpID:E27] **Department** [Name: Sales || Role: Manager]

Data in a wide-column store

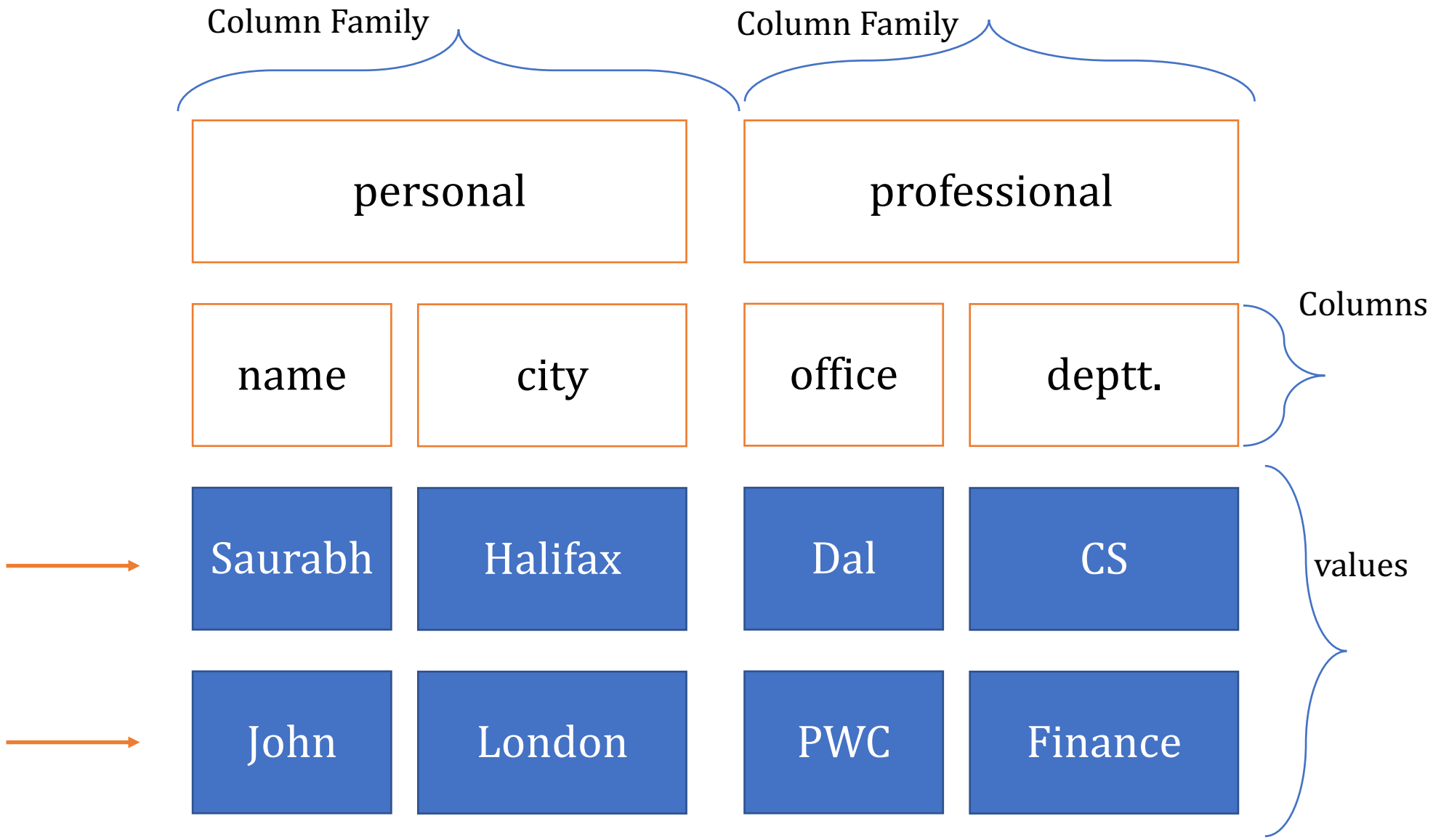


Source: https://pandaforme.gitbooks.io/introduction-to-cassandra/content/understand_the_cassandra_data_model.html

Data in a wide-column store - example



Data in a wide-column store - example

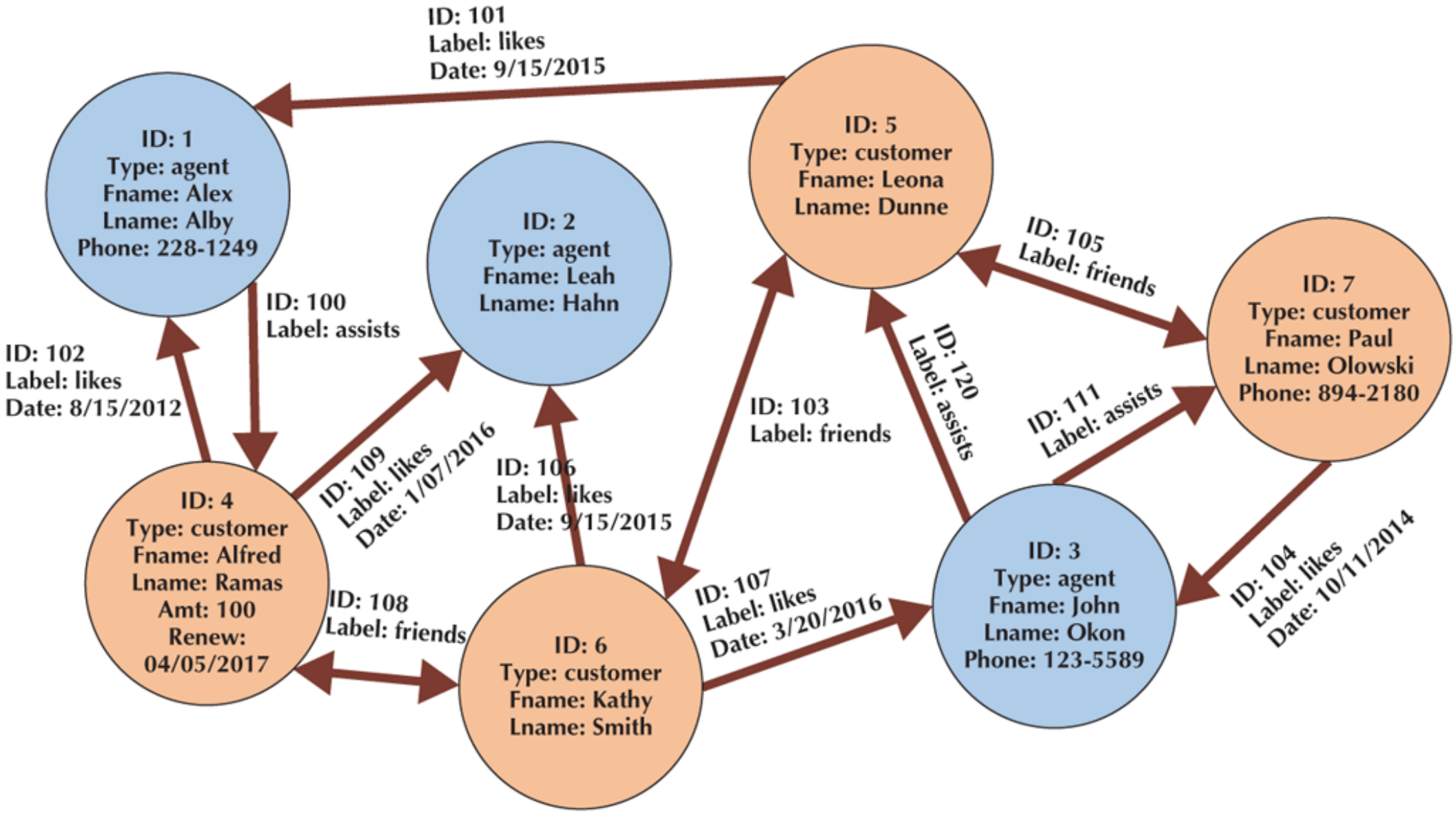


Data in a wide-column store - example

```
import java.io.IOException;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.hbase.HBaseConfiguration;
import org.apache.hadoop.hbase.client.Get;
import org.apache.hadoop.hbase.client.HTable;
import org.apache.hadoop.hbase.client.Result;
import org.apache.hadoop.hbase.util.Bytes;

public class RetriveData{
        public static void main(String[] args) throws IOException, Exception{
        // Instantiating Configuration class
        Configuration config = HBaseConfiguration.create();
        // Instantiating HTable class
        HTable table = new HTable(config, ”people");
        // Instantiating Get          class
        Get g = new Get(Bytes.toBytes("row1"));
        // Reading the data
        Result result = table.get(g);
        // Reading values from Result class object
        byte [] value = result.getValue(Bytes.toBytes("personal"),Bytes.toBytes("name"));
        byte [] value1 = result.getValue(Bytes.toBytes("personal"),Bytes.toBytes("city"));
        // Printing the values
        String name = Bytes.toString(value); String city = Bytes.toString(value1); System.out.println("name: " + name + " city: " + city); } }
```

(4) Graph databases



Questions to Consider

- Which NoSQL database will be ideal to capture a friends of friends network on Facebook?
- Are NoSQL databases safe?
- Do we lose data if kept in Hbase?