

AT82.02

DATA MODELING AND MANAGEMENT

UNIT 2-3: COLUMN FAMILY MODEL

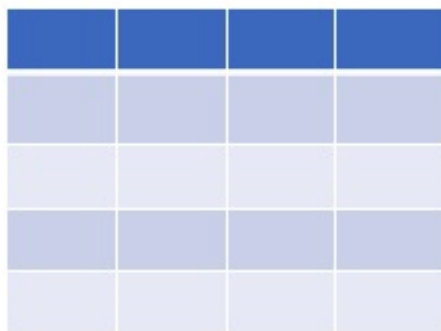
CHUTIPORN ANUTARIYA (CHUTI AT AIT DOT AC DOT TH)

Database Family

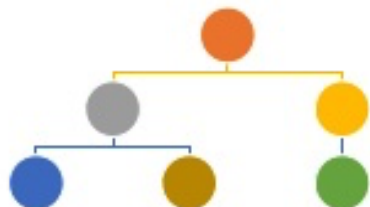
RECAP

Databases

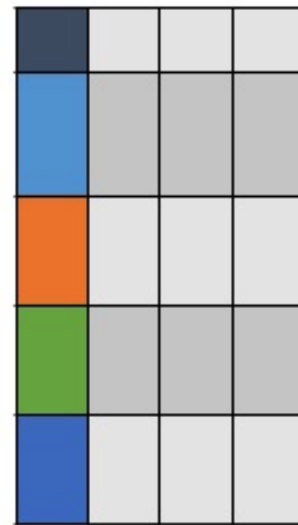
Relational



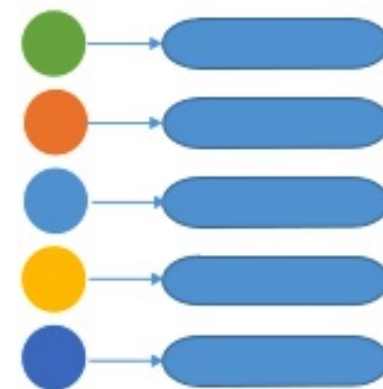
Document



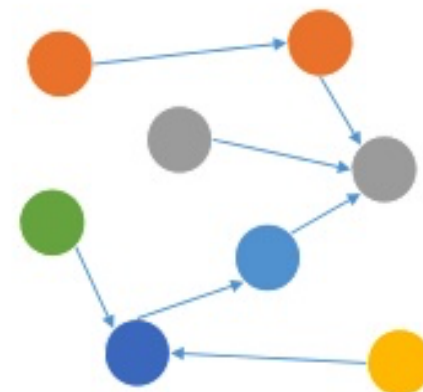
Column-Family



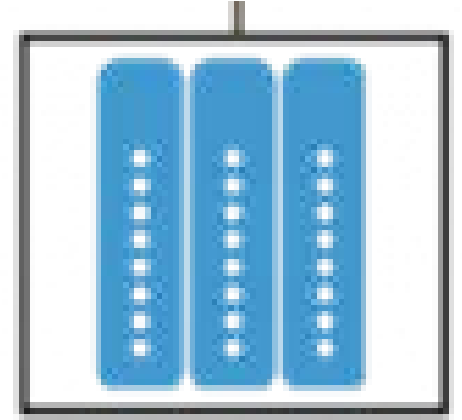
Key-Value



Graph



Column-Family Model



(AKA. COLUMNAR AND WIDE-COLUMN MODEL)

NOTE: MOST TERMINOLOGY USED HERE ARE BASED ON APACHE CASSANDRA SINCE IT IS ONE OF THE MOST POPULAR COLUMN-FAMILY STORES.

Column-Family Model

Column-family stores **enhance the key-value concept** by providing additional structure.

One of the most influential NoSQL database was Google's BigTable.

Other stores: Cassandra, HBase, Hypertable, Amazon DynamoDB.

Column-Family Model

Most RDB databases has rows as unit of storage, which helps in writing performances.

In practical use, it has shown to be more efficient for optimizing read operations to store the data in relational tables not per row, but per column.

This is because all columns in one row are rarely needed at once, but there are groups of columns that are often read together.

Therefore, in order to optimize access, it is useful to structure the data in such groups of columns—column families—as storage units.

Column-Family Model



Columns can be grouped to column families to support organization and partitioning.



When you place related columns in the same family, instances of those columns will be stored as physically close to each other as possible.

Popular Column-Family Stores



Cassandra



HBase



BigTable

- It uses a concept called keyspace, which works in a similar way to a relational model scheme.
- Each column is treated separately;
- They have high performance in aggregation queries (SUM, COUNT, AVG, MIN) since the data is available in one column;
- Widely used to manage data warehouses, business intelligence, CRM, card catalogs from the Library

Column

A column can be considered like a Key-Value pair:

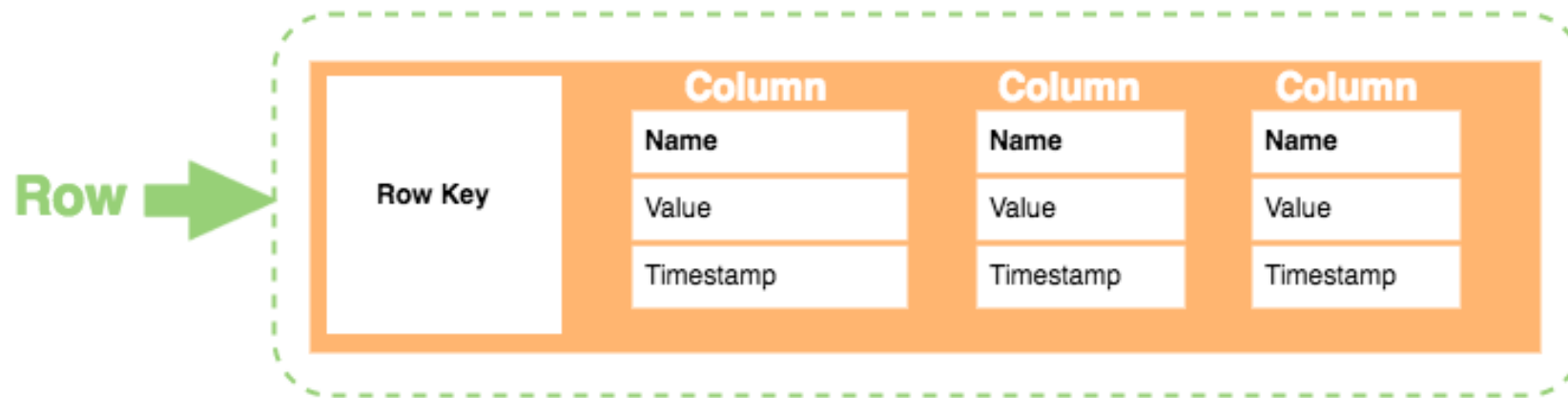
- Key is the Column Name with an associated value.
- A column in a certain implementation may have an **additional field called timestamp.**

Column Name
Column Value
Timestamp

Source: <https://www.codeproject.com/Articles/518134/CassandraplusChapterplusThreeplus-e2-80-93plusData>

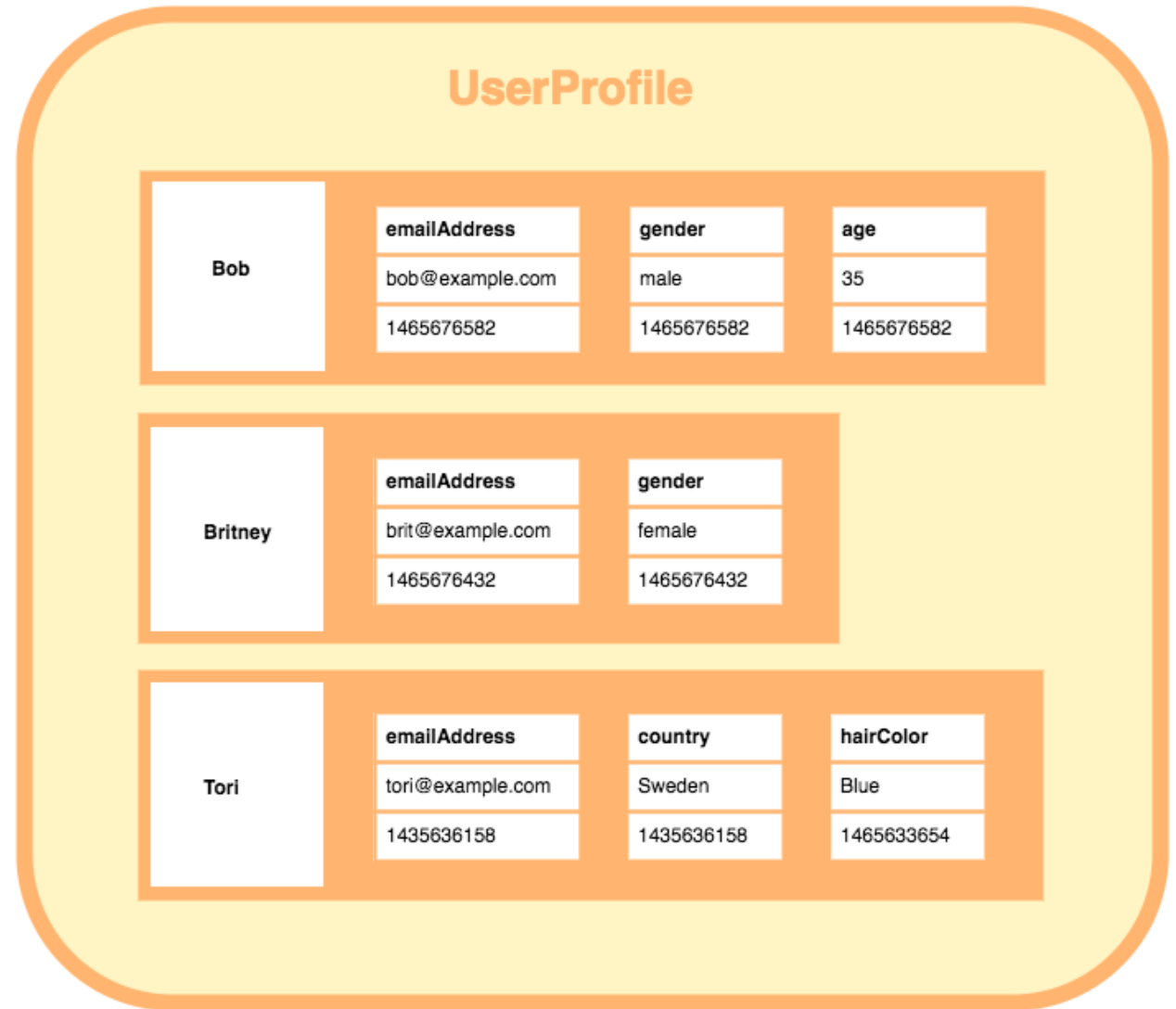
Column Family Data Model

- Row Key: Each row has a unique key, which is a unique identifier for that row.
- Column: Each column contains a name, a value, and timestamp.
 - Name: the name of the name/value pair.
 - Value: the value of the name/value pair.
 - Timestamp: provides the date and time that the data was inserted, which can be used to determine the most recent version of data.

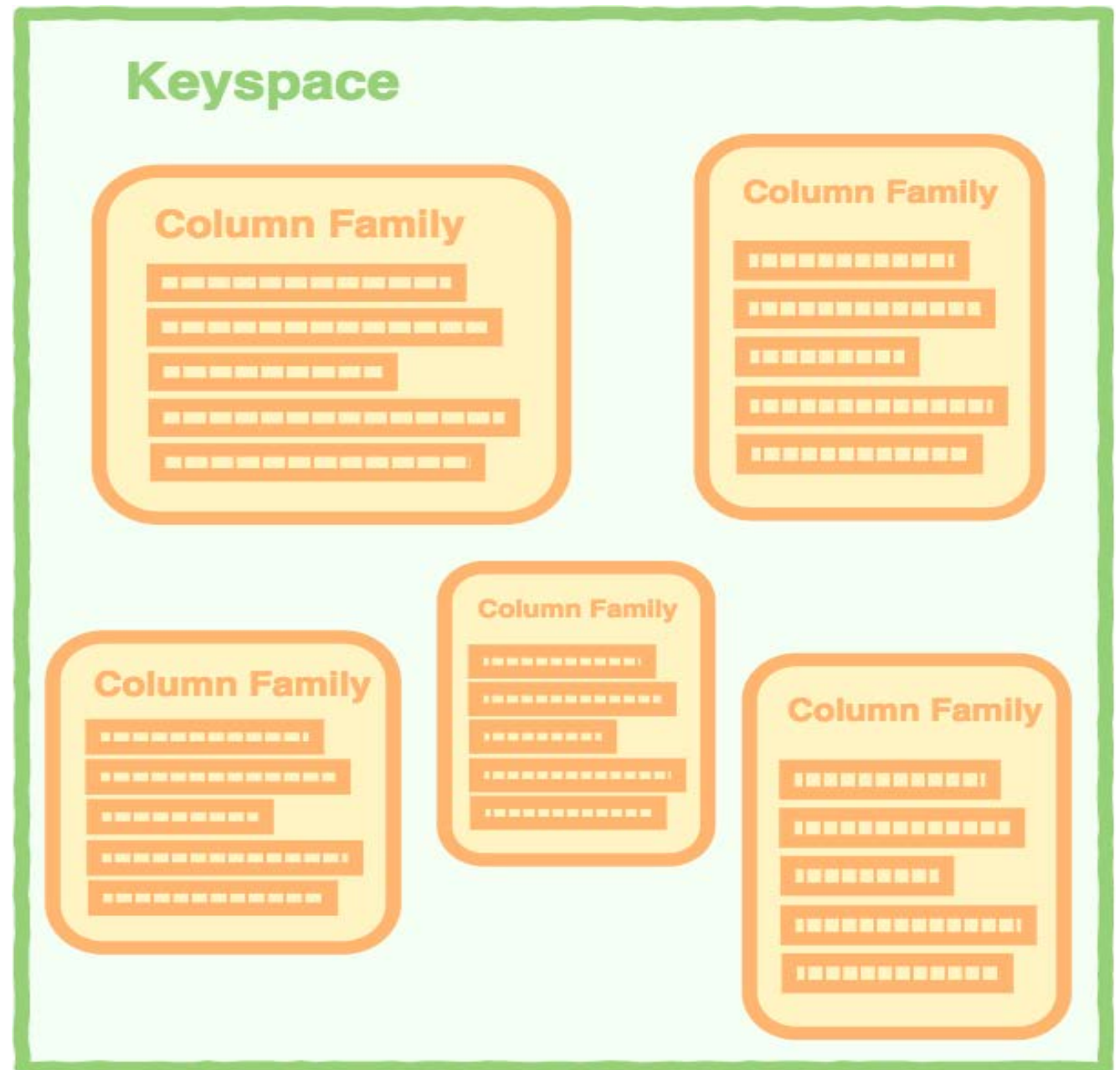


src: <https://database.guide/what-is-a-column-store-database/>

Example: A
Column Family
with 3 rows



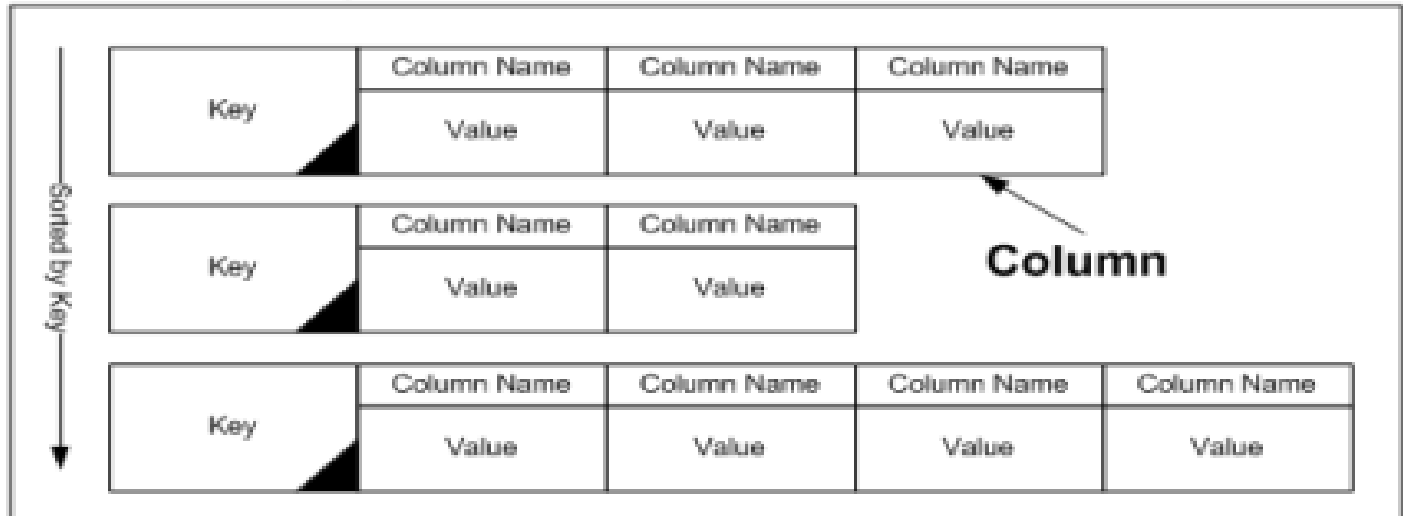
Keyspace



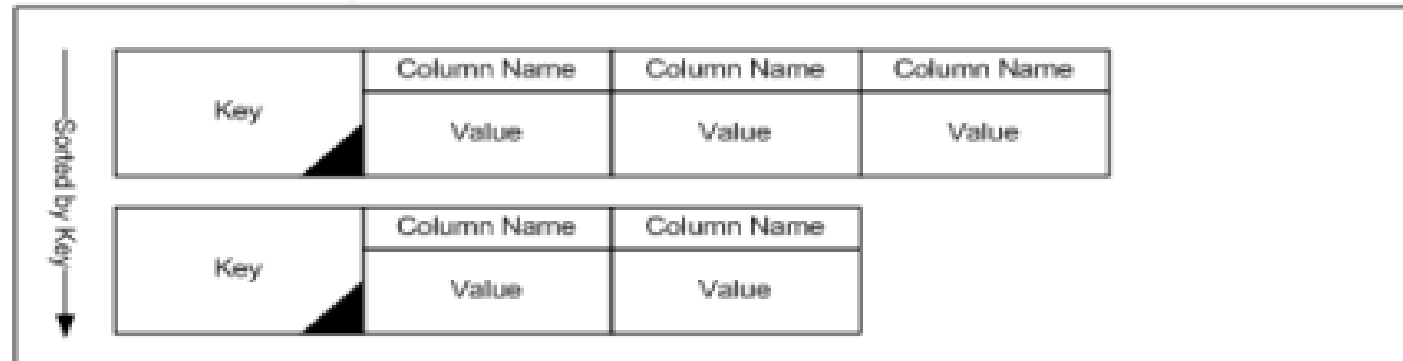
KeySpace

KeySpace

Column Family



Column Family



KeySpace

Four Building Blocks of Column Family Model

Figure source:

<https://neo4j.com/blog/aggregating-stores-tour/>

name
value

Column

super column name		
name	...	name
value		value

Super Column

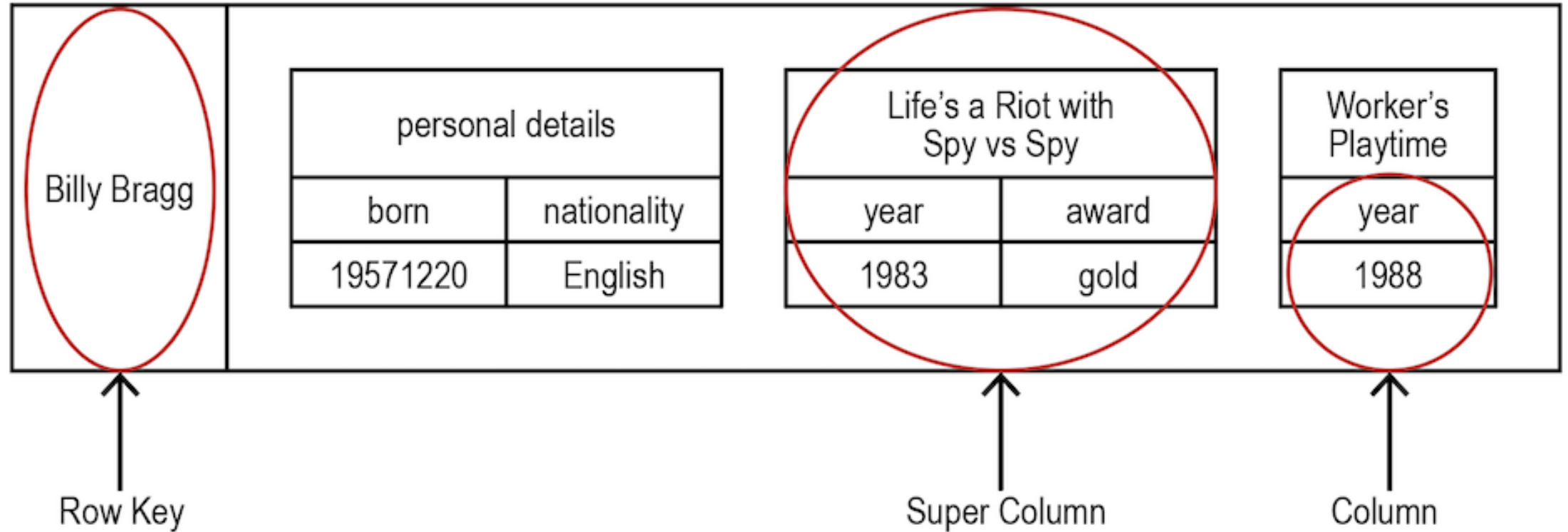
row key	name	...	name
	value		value

Column Family

row key	super column name			...	super column name		
	name	...	name		name	...	name
	value		value		value		value

Super Column Family

Super Column Family



Storing Data in a Super Column Family

Figure source: <https://neo4j.com/blog/aggregate-stores-tour/>

Static vs. Dynamic Column Family

Static Column Family

- We can specify schema such as Column Names, their Data Types and indexes
- Rows are not required to reserve storage for every column defined in the schema and can be sparse.
- Space is only used for the columns that are present in the row.

Dynamic Column Family

- There is no schema. The application is free to store whatever columns it wants and their data types at run-time.

Static vs. Dynamic Column Family

row key columns ...

jbellis	name	email	address	state
	jonathan	jb@ds.com	123 main	TX
dhutch	name	email	address	state
	daria	dh@ds.com	45 2 nd St.	CA
egilmore	name	email		
	eric	eg@ds.com		

row key columns ...

jbellis	dhutch	egilmore	datastax	mzcassie
dhutch	egilmore			
egilmore	datastax	mzcassie		

Summary



Keyspace: Top level container for Column Families



Column Family: A container for Row Keys and Column Families



Row Key: The unique identifier for data stored within a Column Family



Super Column: A Dictionary of Columns identified by Row Key



Column: Name-Value pair with an additional field: timestamp

Row-oriented vs. Column-oriented

Most RDB databases has rows as unit of storage, which helps in writing performances.

However, there are many scenarios where:

- Write are rares, but
- You need to read a few columns of many rows at once

Sales				
	saleid	prodid	date	region
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				

(a) Column Store with Virtual Ids

Sales				
	saleid	prodid	date	region
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				

(b) Column Store with Explicit Ids

Sales				
	saleid	prodid	date	region
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				

(c) Row Store

Row-oriented vs. Column-oriented

Benefits & Limitations

The biggest benefit of a column-oriented database is fast data aggregation:

- Extracting data from a single column and providing summaries of that data.
- Google uses one, for example, to aggregate Web page visitation data. When the software has compiled the Web sites visited and the number of times each was visited, the data are archived.

Benefits & Limitations

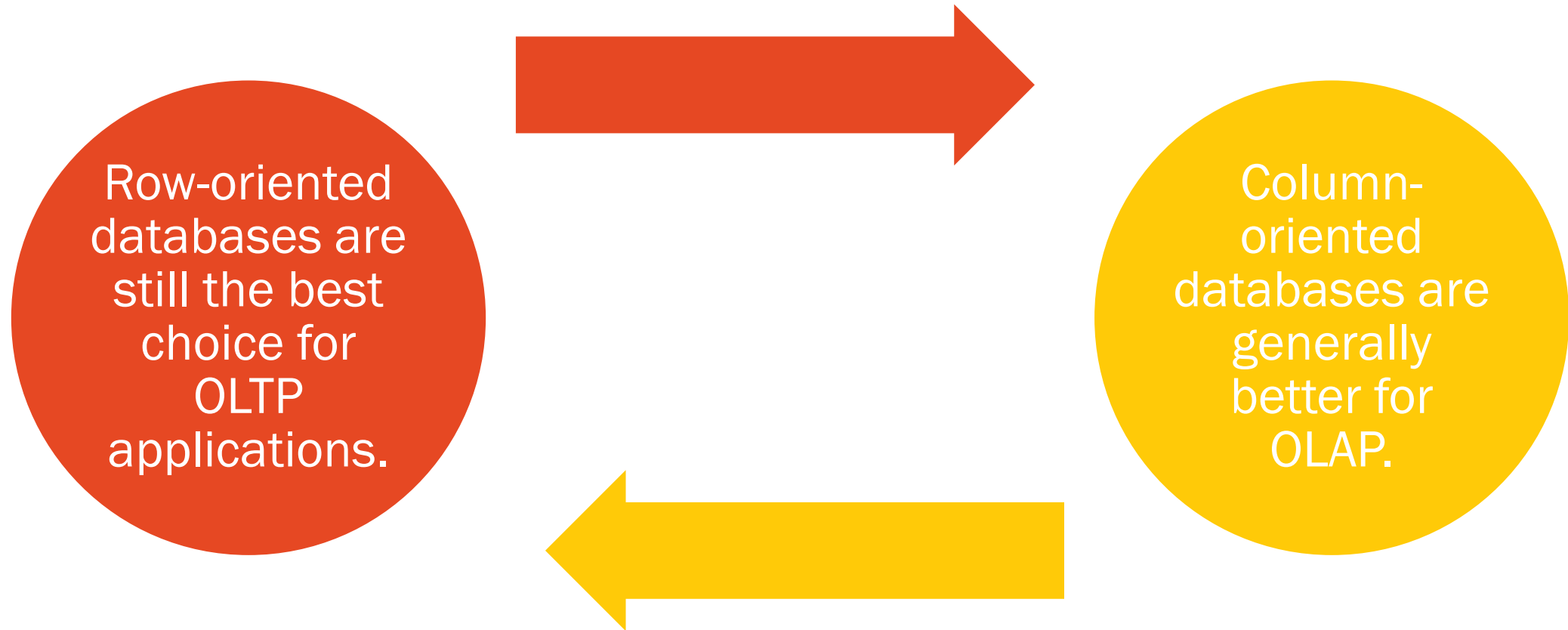
Unlike transactional data, which is written frequently, analytical data doesn't change often. It's usually created by infrequent bulk writes – data dumps.

Columnar store lets you ignore all data that do not apply to a particular query, because you can retrieve the information from just the columns you want.

To insert a new record in a row-oriented database, it takes a single operation.

However, it takes more computing resources to write a record to a columnar database, because you have to write all the fields to the proper columns one at a time.

Applications



- ◎ P. Sadalage and M. Fowler: NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence, Addison-Wesley Professional, 2013
- ◎ Jan L. Harrington: Relational Database Design and Implementation, 4th edition, Morgan Kaufmann, 2016
- ◎ A. Makris, K. Tserpesa, V. Andronikou Dimosthenis Anagnostopoulos: A Classification of NoSQL Data Stores Based on Key Design Characteristics, Procedia Computer Science, Vol. 97, 2016, pp. 94-103.