BIG DATA MANAGEMENT

CZ4123

DATA MODELS

Siqiang Luo

Assistant Professor

- In previous lectures, we discussed Big Data 5V's
  - Understand how to classify a big data application

- In this lecture, we will learn typical data models in big data systems
  - Geared to the mainstream big data systems

DATA MODEL AND PHYSICAL STORAGE SCHEME

- **Data model** describes how data are logically organized.

- Each data model can have different **storage schemes**.
  - Example: Relational model can be stored in row-oriented or column oriented.

DATA MODELS

- Relational Data Model
  - Corresponding to relational database
- Key-Value Data Model
  - Corresponding to key-value systems
- Graph Data Model
  - Corresponding to graph database

Relational Data Model

RELATIONAL DATA MODEL

Relational data contains a set of **Relations**

ID	name	age	gender
0001	Alex	25	M
0002	Mary	35	F

An example relation (Employee table)

RELATIONAL DATA MODEL

Schema -- specifies the **relation name**, and the **attribute** of each column.

❖Example:

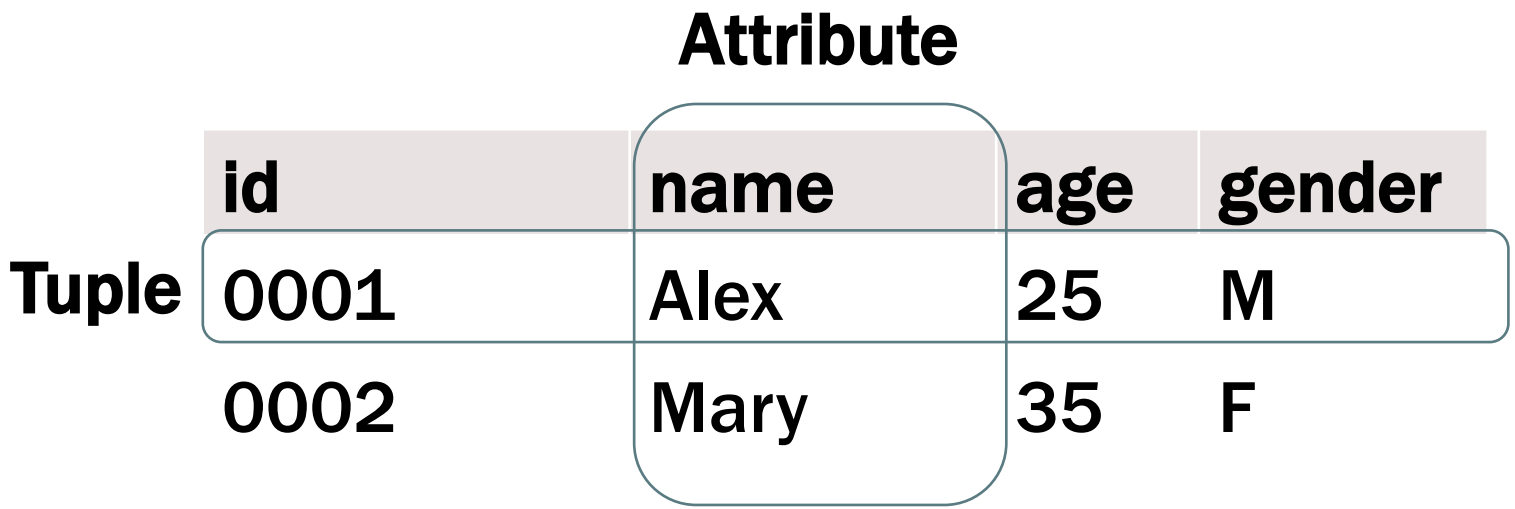
Employee (id, name, age, gender) ⟵ ***schema***

ID	name	age	gender
0001	Alex	25	M
0002	Mary	35	F

RELATIONAL DATA MODEL

Tuple: typically refers to a row of the relation

Attribute: corresponds to a column of the relation



RELATIONAL DATA MODEL

Primary Key: A set of attributes that uniquely specify a row (usually there is an ID column)

Primary Key

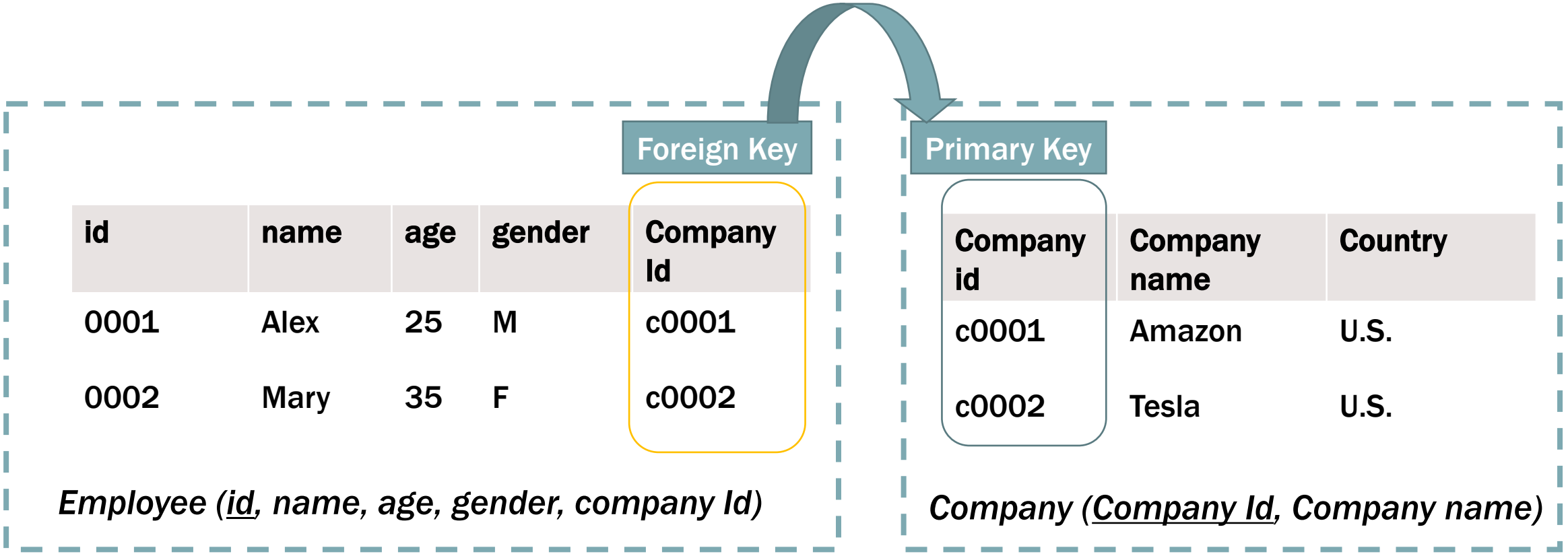
id	name	age	gender
0001	Alex	25	M
0002	Mary	35	F

Employee (<u>id</u>, name, age, gender)

Primary Key underlined

RELATIONAL DATA MODEL

Primary key – Foreign key relationship



A foreign key is a set of one or more columns in a table that refers to the primary key in another table.

TWO REPRESENTATIVE STORAGE SCHEMES

- A relation can be stored row-by-row (such a data system is often called a **row store**)
- A relation can be stored column-by-column (such a data system is often called a **column store**)
- We will discuss in more details when we in later lectures about “column store”.

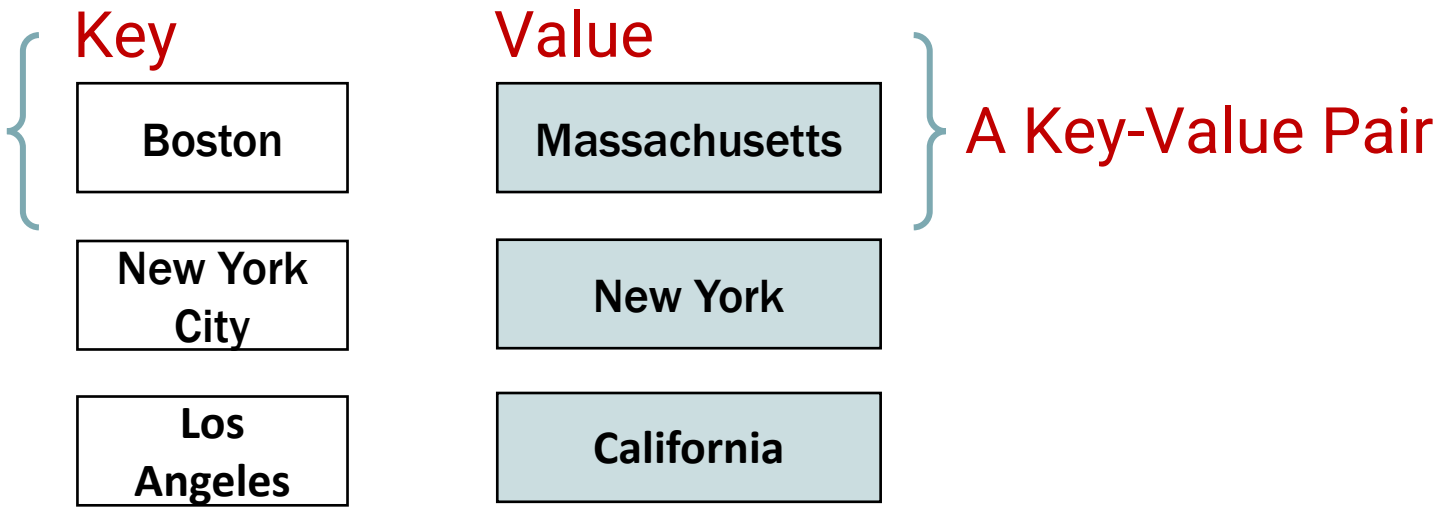
Key-Value Data Model

KEY-VALUE DATA MODEL

- The relational model has strict schemas.
- Some big data systems may require **schema-less** models.
- Key-value data model is one such kind.

KEY-VALUE DATA MODEL

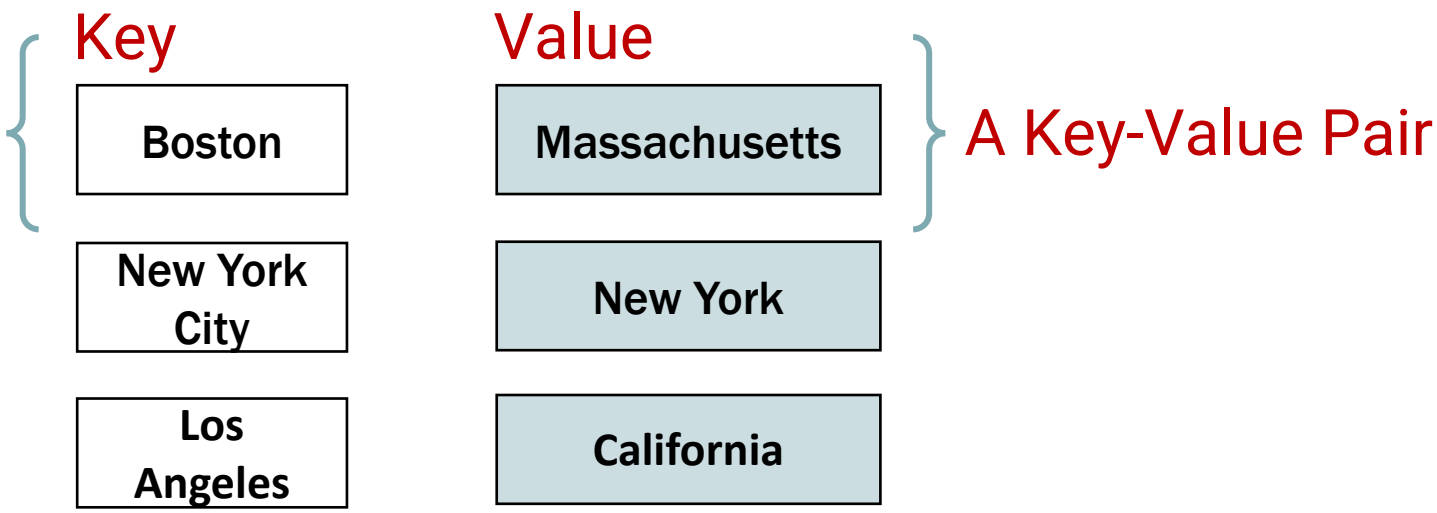
- ❑ Data is represented as a collection of key–value pairs.
- ❑ Key uniquely decides the pair



Above are (key, value) pairs describing the mapping between cities and states in the United States.

KEY-VALUE DATA MODEL

- It is usually less expressive than relational model but much simpler.
- It is preferred by a lot of real-systems including Facebook and Google in analyzing big data.
  - e.g., Google's levelDB, Facebook's RocksDB.



Key-value Data Model is ubiquitous!

For any A that can determine B

Key Value

CHOOSING THE RIGHT KEY

Key: Name

Value: ID

Alex	STU001
Bob	STU002

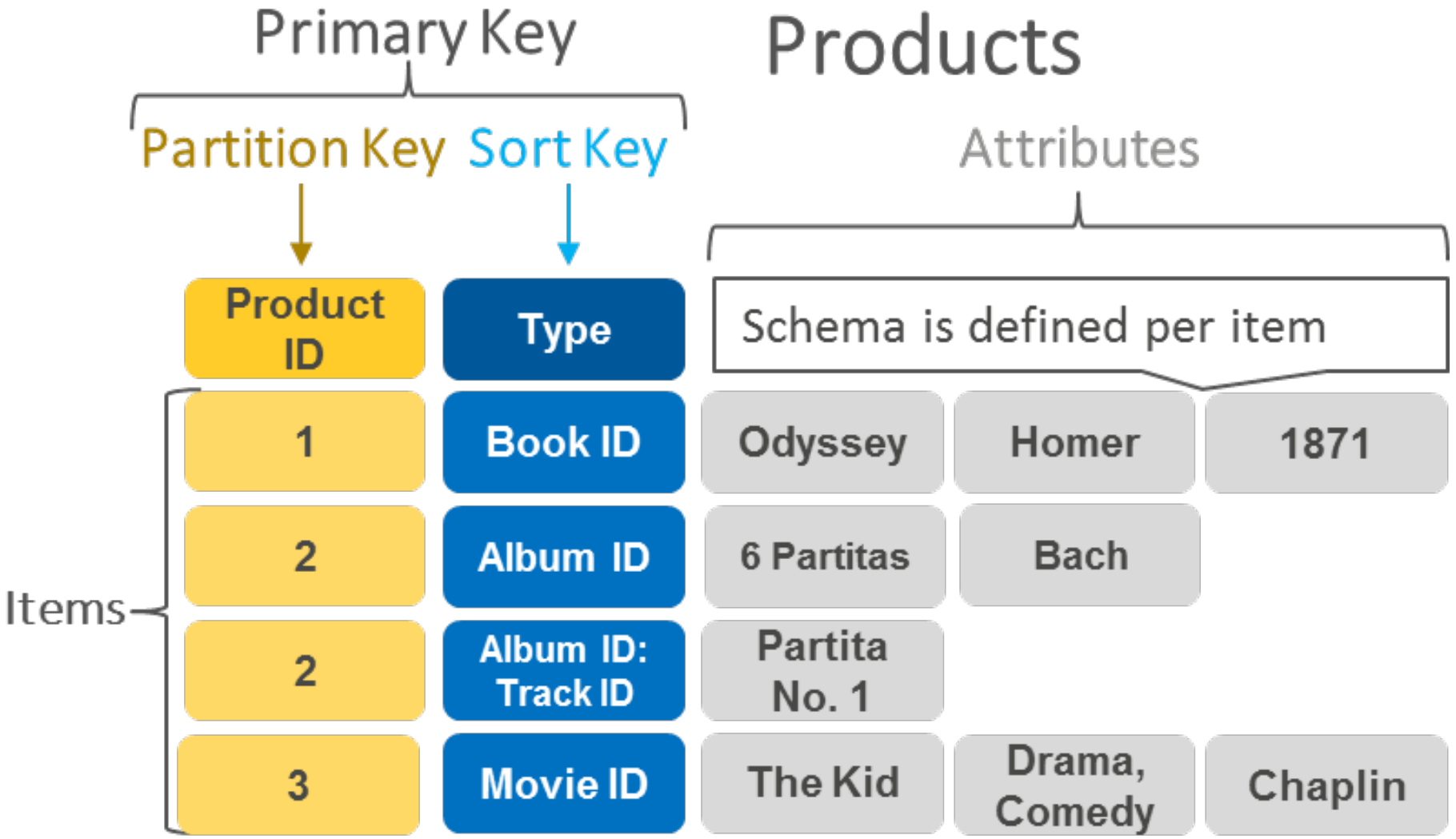
A good key-value model?

CHOOSING THE RIGHT KEY

How to put the Tweets data into key-value model?

You may not have an explicit key in the dataset

AMAZON'S CASE



REMARK 1

Key-value model can “store”
the information of a relation

EXERCISE 1

Converting the following Relation/Table to key-value model

Primary Key

id	name	age	gender
0001	Alex	25	M
0002	Mary	35	F

SOLUTION

Key

Value

Primary Key

Concatenate Other Attributes

id	name	age	gender
0001	Alex	25	M
0002	Mary	35	F

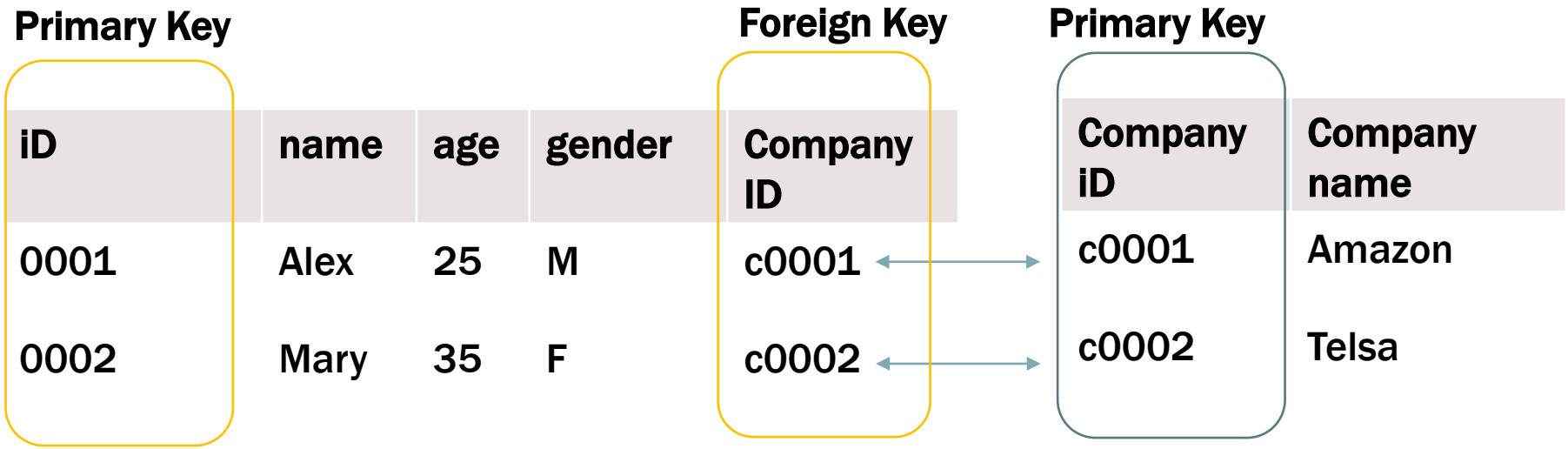
0001 Alex;25;M

0001 Mary;35;F

REMARK 2

Key-value model can be mapped to a conceptual big table in the relational model!

EXERCISE 2



Given the above two tables (Employee and Company).
If you do not worry about the storage and always want to query the information of employees, how would you convert them into a key-value model?

SOLUTION

Primary Key

Foreign Key

Primary Key

id	name	age	gender	Company Id
0001	Alex	25	M	c0001
0002	Mary	35	F	c0002

Company id	Company name
c0001	Amazon
c0002	Telsa

Step 1: Join the table (Left outer-join from Employees)

ID	name	age	gender	Company-name	Company-ID
0001	Alex	25	M	Amazon	c0001
0002	Mary	35	F	Telsa	c0002

SOLUTION

Step 2: Make primary key as “Key”, and the others concatenated as values

Key: Id
Value: name; net worth; rank; company-name; company-id; CEO-name

ID	name	Net worth	rank	Company-name	Company-ID	CEO-name
0001	Jeff Bezos	$201.4B	1	Amazon	c0001	Jeff Bezos
0002	Bernard Arnault & Family	$181.6B	2	LVMH	c0003	Bernard Arnault

Note: We let key be ID because ID still uniquely defines a row in the big table.

COMPARING RELATIONAL MODEL AND KEY-VALUE MODEL

Advantages,
disadvantages?

COMPARING RELATIONAL MODEL AND KEY-VALUE MODEL

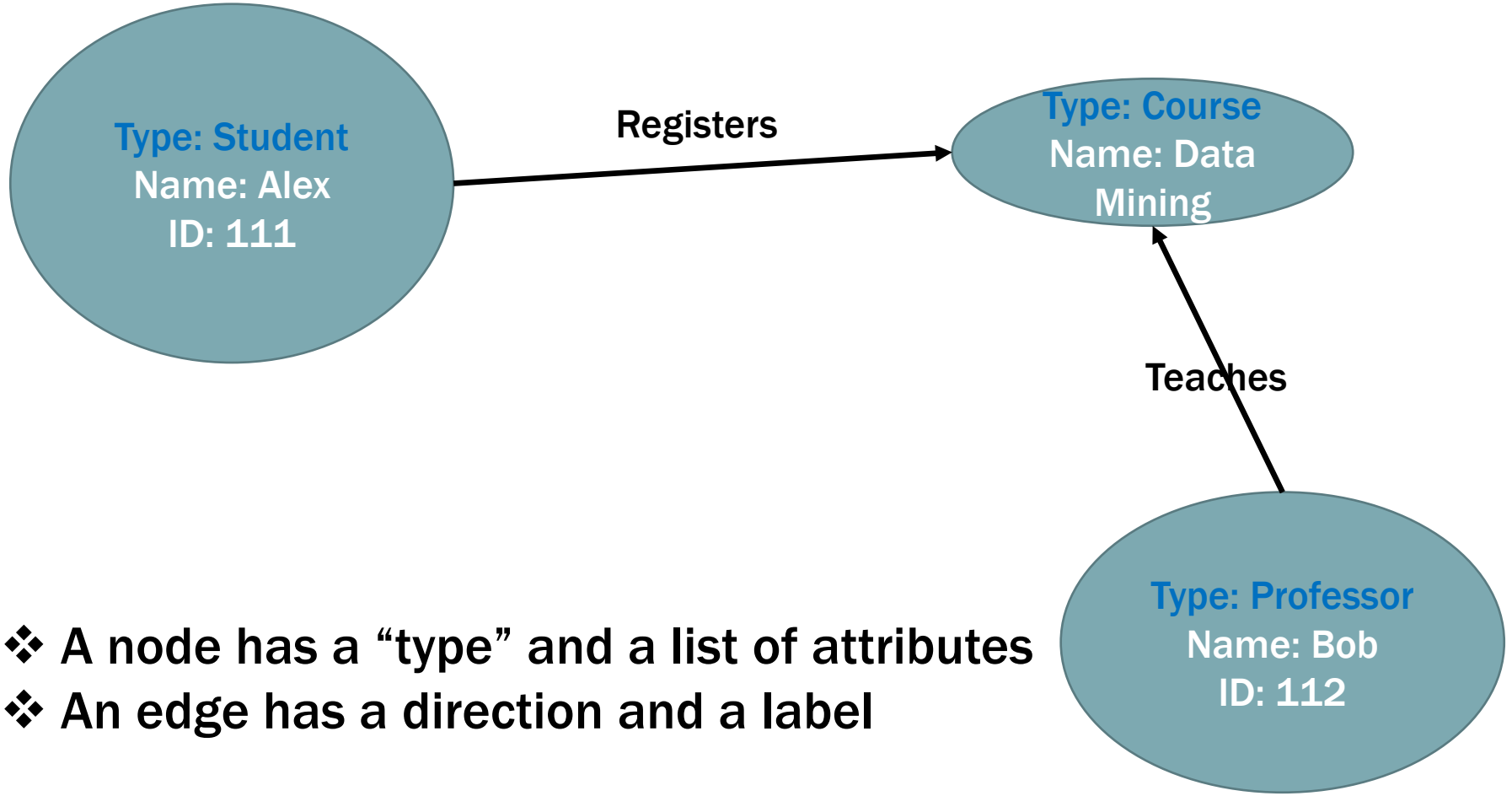
- Key-Value model is more flexible
  - Favoured by a lot of industrial-level big data systems, e.g., Facebook's RocksDB, Google's LevelDB
  - Assume most of the queries are simple (example: find a value corresponding to a key or a key range)
  - It is schema-less, making it commonly used in real-time web-based applications (highly partitionable, easy scaling)
  - Flexible to handle schema changes
- Relational model is more structured
  - Suitable to handle tabular data
  - Favoured by accuracy-sensitive systems, e.g., data systems in the bank
  - It has strict schemas, and is easy to design query languages (e.g., SQLs)

Graph Data Model

GRAPH MODEL

- There is another type of database called graph database
  - E.g., Neo4j, OrientDB
- Graphs are the underlying data model of graph databases
  - A graph is formed by nodes and edges
  - A node represents an entity
  - An edge represents the relationship between entities

GRAPHS ARE UBIQUITOUS



- A node has a "type" and a list of attributes
- An edge has a direction and a label

Primary Key

Foreign Key

id	name	age	gender	Company Id
0001	Alex	25	M	c0001
0002	Mary	35	F	c0002

Primary Key

Company id	Company name
c0001	Amazon
c0002	Telsa

How to convert the above relations/tables into a graph model?

Primary Key

Foreign Key

Primary Key

iD	name	age	gender	Company ID
0001	Alex	25	M	c0001
0002	Mary	35	F	c0002

Company iD	Company name
c0001	Amazon
c0002	Telsa

Type: Employee
id: 0001
Name: Alex
age: 25
gender: M

Type: Company
id: c0001
Name: Amazon

Type: Employee
id: 0002
Name: Mary
Net worth: 35
Gender: F

Type: Company
id: c0002
Name: Telsa

Looks like relational model can do the same thing. Why do we need graphs?

Suppose we want to model a social network like Facebook

One possible way is to build two relations:

User ID	Name
ID001	Alex
ID002	Mark
ID003	Mary
ID004	Bob

User Table

User ID1	User ID2
ID001	Id002
ID001	Id003
ID002	Id003
ID003	Id004

Friendship Table

Suppose we want to model a social network like Facebook

One possible way is to build two relations:

User ID	Name
ID001	Alex
ID002	Mark
ID003	Mary
ID004	Bob

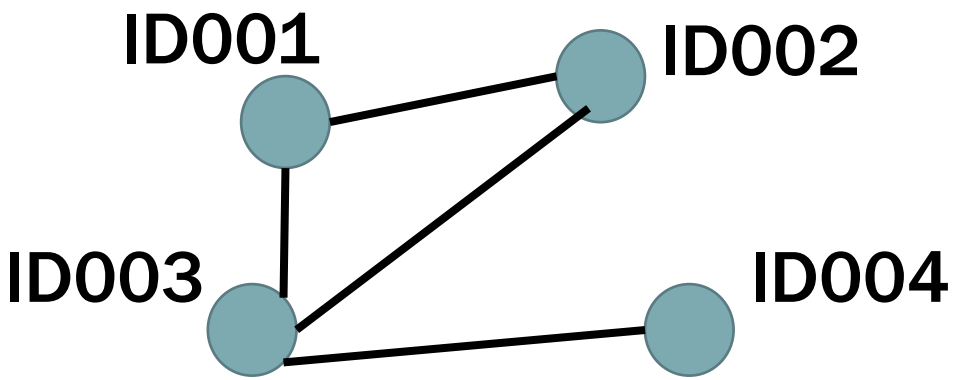
User Table

User ID1	User ID2
ID001	Id002
ID001	Id003
ID002	Id003
ID003	Id004

Friendship Table

Now, how do we answer a typical query :
"find the two most distant users"?

- There are some queries that require to explore the complex structures of the entities.
- For these queries, it is more suitable to consider the data as a graph.
- The social network is modeled as a graph as follows



We can then compute all-pair shortest path algorithms on the graph to answer the query

We finish Data Models!

Next lecture: Big data and Memory Hierarchy