Instructions

Relational Model

Matthew Pike & Yuan Yao

Univeristy of Nottingham Ningbo China (UNNC)

Overview

Content

- Relational Model - definition, structure, terminology.
- Candidate, Primary and Foreign Keys.
- Entity and Referential Integrity.

Learning Outcomes

- Understand what is the relational model.
- Familiar with each component in the model.
- Point out keys in a given relaitonal model.
- Design primary and foreign keys.
- Understand the basic concept of referential integrity.

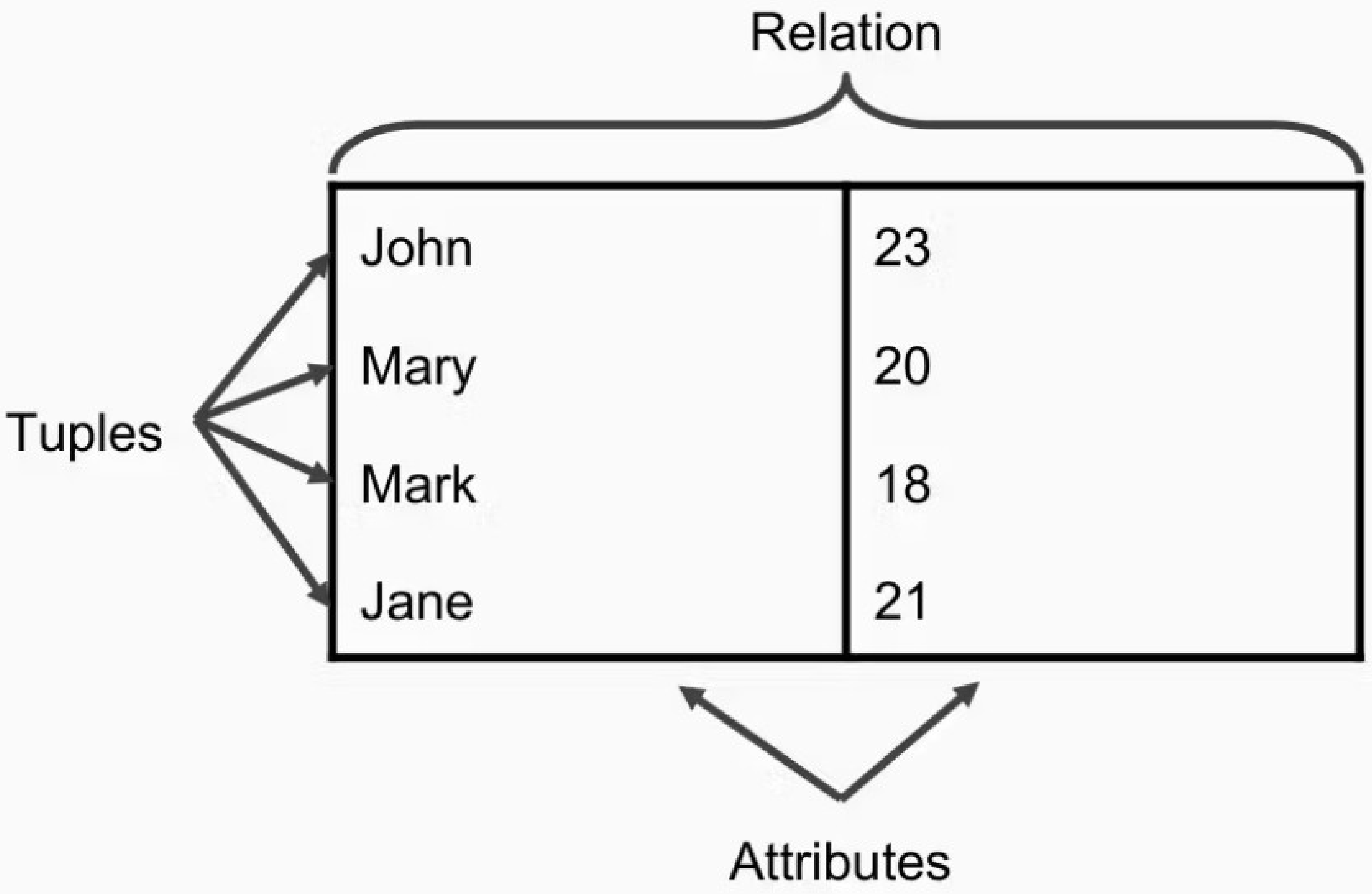
Relational Model

What is the Relational Model?

- An approach to managing data using a structure and language consistent with:
    - *First-order predicate logic (FOL)*
- Originally introduced by E.F. Codd in his paper in 1970:
    - “A Relational Model of Data for Large Shared Databanks”
- The foundation for most (but not all) modern DBMS.
- Provide a declarative method for specifying data and queries.
    - What info DB contains and what info users want
- Chapter 4 of the DB book.

Relational Data Structure

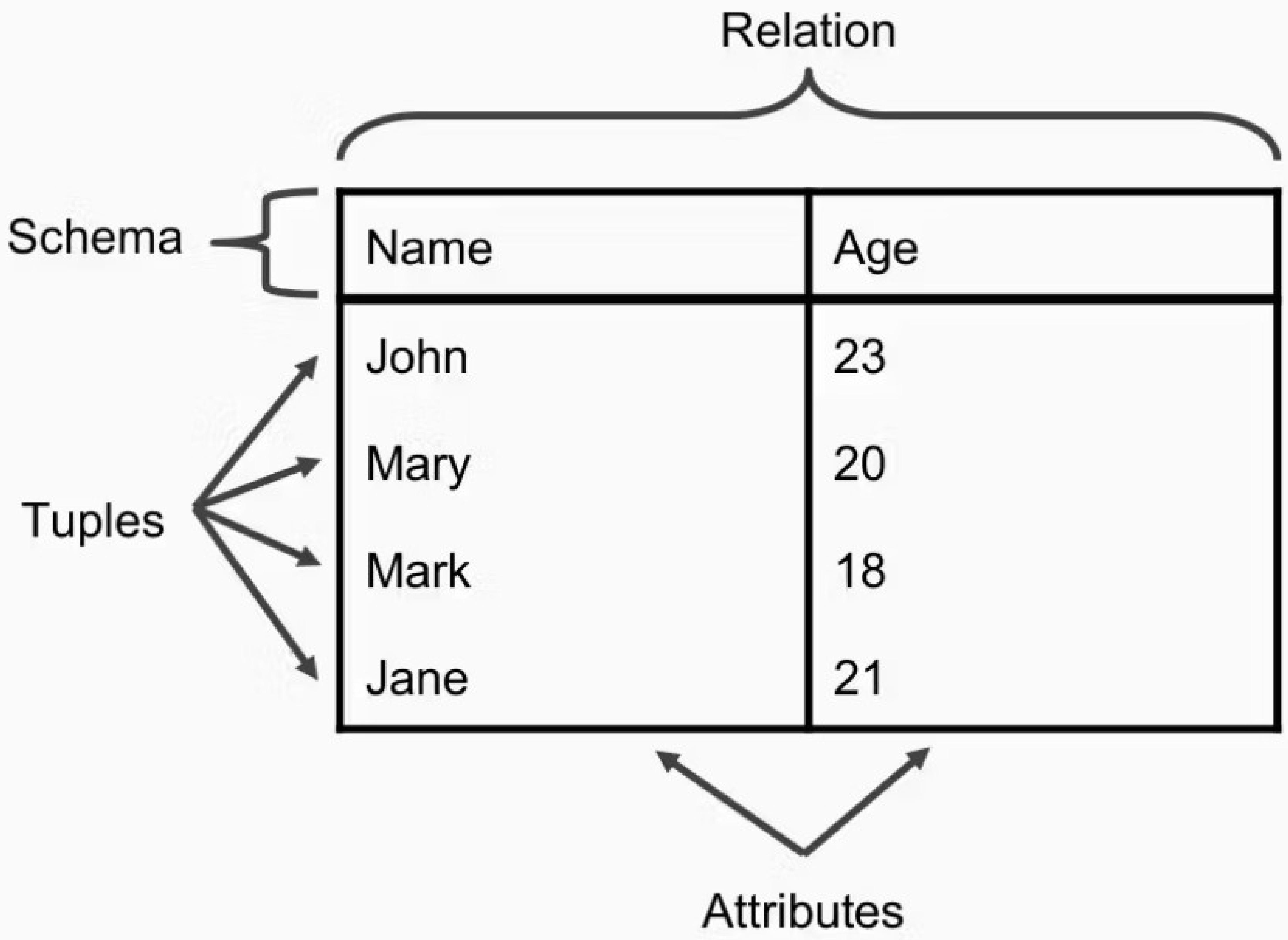
- Data is stored in **relations** which are tables with columns and rows.
- An **attribute** of a relation is a column in the table.
- A relation is a set of **tuples**.
  - e.g., a tuple *(John, 23)*.



John	23
Mary	20
Mark	18
Jane	21

Relational Data Structure

- Each relation has a **schema**.
  - Schemas define the relation's attributes.
  - Sometimes called **headings**.
- Each attribute has a corresponding **domain**.
  - i.e., A set of possible values.



Name	Age
John	23
Mary	20
Mark	18
Jane	21

How to represent relations

Student

Name	Age
John	23

- Attribute: the name of each attribute, e.g., Age
- Schema: $(A_1, \ldots, A_n)$
  - A_i: attributes in the relation R.
  - E.g., $(Name, Age)$.
- Relation: $R(A_1, \ldots, A_n)$
  - R: the name of a relation.
  - E.g., $Student(Name, Age)$
- Tuples:
  - Named: $\{(A_1 : V_1^1, \ldots, A_n : V_n^1), \ldots, (A_1 : V_1^m, \ldots, A_n : V_n^m)\}$
  - E.g., $\{(Name : John, Age : 23)\}$
  - Unnamed: $\{(V_1^1, \ldots, V_n^1), \ldots, (V_1^m, \ldots, V_n^m)\}$

Relational Data Structure

More formally:

- A relational schema is a set of attributes.
- A tuple assigns a value to each attribute in the schema.
- A relation is a set of tuples with the same schema.

Student

Name	Age
John	23
Mary	20
Mark	18

- Schema: ?
- Tuples: ?

Relational Data Structure

More formally:

- A relational schema is a set of attributes.
- A tuple assigns a value to each attribute in the schema.
- A relation is a set of tuples with the same schema.

Student

Name	Age
John	23
Mary	20
Mark	18

- Schema: $(Name, Age)$
- Tuples: $\{(Name : John, Age : 23), (Name : Mary, Age : 20), (Name : Mark, Age : 18)\}$

Relational Data Structure

More formally:

- A relational schema is a set of attributes.
- A tuple assigns a value to each attribute in the schema.
- A relation is a set of tuples with the same schema.

Student

Name	Age
John	23
Mary	20
Mark	18

- Schema: $(Name, Age)$
- Tuples: $\{(Name : John, Age : 23),$ $(Age : 20, Name : Mary),$ $(Name : Mark, Age : 18)\}$

Terminology

- **Degree of a relation:** the number of attributes in the relational schema, i.e., how many columns.
- **Cardinality of a relation:** the number of tuples in a relation, i.e., how many rows.

Example

Employee

ID	Name	Salary	Department
M139	John Smith	18000	Marketing
M140	Mary Jones	22000	Marketing
A368	Jane Brown	22000	Accounts
P222	Mark Brown	24000	Personnel
A367	David Jones	20000	Accounts

- What are the attributes?
- What is the schema?
- How to represent this relation?
- What is the degree of this relation?
- What is the cardinality of this relation?

Example

- Attributes: ID, Name, Salary, Department.
- Schema: $(ID, Name, Salary, Department)$
- Relation: $Employee(ID, Name, Salary, Department)$
- Relation:$\{(M139, JohnSmith, 18000, Marketing),$
  $(M140, MaryJones, 22000, Marketing),$
  $(A368, JaneBrown, 22000, Accounts),$
  $(P222, MarkBrown, 24000, Personnel),$
  $(A367, DavidJones, 20000, Accounts)\}$
- Degree: 4
- Cardinality: 5

Properties

- Each relation has a distinct name.
- Each cell contains exactly one single value.
- Each attribute has a distinct name.
- The values of an attribute are all from the same domain.
- The order of attributes has no significance.
- The order of tuples has no significance.

Problems without DBMS

- No standards.
- Incompatible file format.
- Data duplication.
- Data dependence.
- Fixed queries.
- Concurrency.
- Security.
- No theoretical foundations.

Data duplication

Employee

ID	Name	Salary	Department
M139	John Smith	18000	Marketing
M140	Mary Jones	22000	Marketing
A368	Jane Brown	22000	Accounts
P222	Mark Brown	24000	Personnel
A367	David Jones	20000	Accounts
M140	Mary Jones	22000	Marketing

Data duplication

Employee

ID	Name	Salary	Department
M139	John Smith	18000	Marketing
M140	Mary Jones	22000	Marketing
A368	Jane Brown	22000	Accounts
P222	Mark Brown	24000	Personnel
A367	David Jones	20000	Accounts
A369	Mary Jones	21000	Accounts

Candidate Keys

- SuperKey: an attribute, or a set of attributes, that uniquely identifies a tuple within a relation.
- Candidate Key: a superkey such that no proper subset is a superkey within the relation.

Definition (Candidate Key)

A set of attributes $\mathcal{K}$ is a candidate key for a relation $\mathcal{R}$ iff

- In each tuple of $\mathcal{R}$, the values of $\mathcal{K}$ uniquely identify that tuple. (*Uniqueness*)
- There is no subset of $\mathcal{K}$ can uniquely identify the tuples in $\mathcal{R}$. (*Irreducibility*)

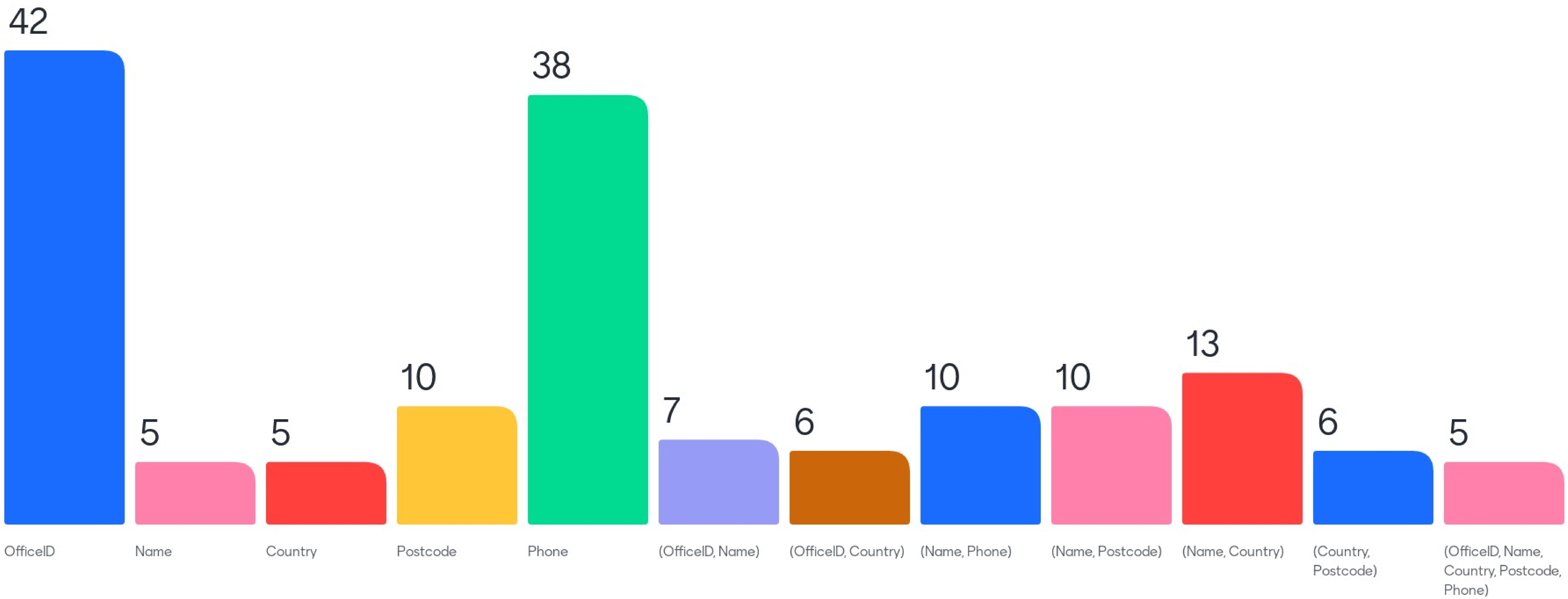
Example: Candidate Keys

Office

OfficeID	Name	Country	Postcode	Phone
O1001	Headquarters	UK	W1 1AA	0044 20 1545 3241
O1002	R&D Labs	UK	W1 1AA	0044 20 1545 4984
O1003	US West	USA	94130	001 415 665981
O1004	US East	USA	10201	001 212 448731
O1005	Telemarketing	UK	NE5 2GE	0044 1909 559862
O1006	Telemarketing	USA	84754	001 385 994763

Mentimeter

What are the potential candidate keys?



7

48

Example: Candidate Keys

Candidate Keys: OfficeID, Phone, (Name, Postcode), (Name, Country)

OfficeID	Name	Country	Postcode	Phone
O1001	Headquarters	UK	W1 1AA	0044 20 1545 3241
O1002	R&D Labs	UK	W1 1AA	0044 20 1545 4984
O1003	US West	USA	94130	001 415 665981
O1004	US East	USA	10201	001 212 448731
O1005	Telemarketing	UK	NE5 2GE	0044 1909 559862
O1006	Telemarketing	USA	84754	001 385 994763

- You cannot necessarily infer the candidate keys based solely on the data in your table.
- You must use knowledge of the real-world to help

Primary Keys and NULLs

- To identify each row in a table.
- Each relation must have a primary key to avoid data duplication.

Definition (Primary Key)
A primary key is selected from the set of candidate key to identify tuples in a relation.

- A NULL indicates a missing or unknown value in a relation.
  - Is a NULL the same as 0?
  - Is a NULL the same as a blank space character?
  - Can primary keys contains a NULL value?

Foreign Keys

Department

DID	DName
13	Marketing
14	Accounts
15	Personnel

Employee

EID	EName	DID
15	John Smith	13
16	Mary Brown	14
17	Mark Jones	13
18	Jane Smith	NULL

- What are the primary keys?
- What are the connections between these two relations?

Foreign Keys

Definition (Foreign Key)

An attribute, or set of attributes $\mathcal{F}$ within one relation $\mathcal{R}_1$ is a foreign key, if it matches the candidate key in another relation.

Definition (Referential Integrity)

If $\mathcal{F}$ is a foreign key which connects the data in $\mathcal{R}_1$ to those in another relation $\mathcal{R}_2$, then each value of $\mathcal{F}$ must:

- matche a primary/candidate key values in $\mathcal{R}_2$

or:

- be Null

Foreign Keys

Department

DID	DName
13	Marketing
14	Accounts
15	Personnel

Employee

EID	EName	DID
15	John Smith	13
16	Mary Brown	14
17	Mark Jones	13
18	Jane Smith	NULL

- What is the foreign key?

Foreign Keys

Department

DID	DName
13	Marketing
14	Accounts
15	Personnel

Employee

EID	EName	DID
15	John Smith	13
16	Mary Brown	14
17	Mark Jones	13
18	Jane Smith	NULL

- DID is the foreign key of Employee.

Referential Integrity

Referential Integerity might be violated when a referenced tuple is

- updated, e.g., the DID 13 is changed to 16 in the Department relation.

or:

- delete, e.g., the tuple starts with DID 14 is deleted.

Three options:

- RESTRICT
- CASCADE
- SET NULL

Example: Referential Integrity

- What will happen if:
  - We change the **DID** of Marketing to *16* in **Department**?
  - We delete the tuple for *Accounts* in **Department**?

Department

DID	DName
13	Marketing
14	Accounts
15	Personnel

Employee

EID	EName	DID
15	John Smith	13
16	Mary Brown	14
17	Mark Jones	13
18	Jane Smith	NULL

RESTRICT

- What will happen if:
  - We change the **DID** of Marketing to *16* in **Department**?
  - We delete the tuple for *Accounts* in **Department**?
- RESTRICT stops any actions that violates referential integrity:
  - You cannot change the **DID** of *Marketing*.
  - You are not allowed to delete the tuple for *Accounts*.

Department

DID	DName
13	Marketing
14	Accounts
15	Personnel

Employee

EID	EName	DID
15	John Smith	13
16	Mary Brown	14
17	Mark Jones	13
18	Jane Smith	NULL

CASCADE

- What will happen if:
  - We change the **DID** of Marketing to *16* in **Department**?
  - We delete the tuple for *Accounts* in **Department**?
- CASCADE allows the changes made to flow through:
  - The **DID** of *John Smith* and *Mark Jones* will be changed to *16*.
  - The tuple for *Mary Brown* will be deleted from **Employee**.

Department

DID	DName
~~13~~ 16	Marketing
~~14~~	~~Accounts~~
15	Personnel

Employee

EID	EName	DID
15	John Smith	~~13~~ 16
~~16~~	~~Mary Brown~~	~~14~~
17	Mark Jones	~~13~~ 16
18	Jane Smith	NULL

SET NULL

- What will happen if:
  - We change the **DID** of Marketing to *16* in **Department**?
  - We delete the tuple for **Accounts** in **Department**?
- SET NULL allows the changes to happen, but…
  - The **DID** of *John Smith* and *Mark Jones* will be set to Null.
  - The **DID** of Mary Brown in **Employee** will be set to Null.

Department

DID	DName
~~13~~ 16	Marketing
~~14~~	~~Accounts~~
15	Personnel

Employee

EID	EName	DID
15	John Smith	~~13~~ NULL
16	Mary Brown	~~14~~ NULL
17	Mark Jones	~~13~~ NULL
18	Jane Smith	NULL