

## Relational model

## Relational model

- A formal (mathematical) model to represent
  - objects (data/information),
  - relationships between objects
  - Constraints on objects and relationships
  - Queries about information
- Well-founded on mathematical principles :
  - Precise semantics of constraints and queries
  - Can prove equivalence of different ways to express queries

## Relational model - practice

- Foundation of most Database Management Systems
- SQL language is a programming language to express constructs of formal model

## Relational Database Definitions

1. A relation is a set of tuples over specified domains
  - $R$  subset of  $D_1 \times D_2 \times D_3 \times \dots \times D_k$  (k-ary)
  - Each  $D_i$  is a declared domain
  - Domains atomic
    - types of programming languages
2. A relational database is a set of relations and possibly constraints among the relations

## Relational Database: Terminology

Schema for a relation:

1. Relation name
2. Domain (type) of each component  
i.e. declare  $D_i$  s

Equivalent:

- Instance of a scheme
- Table

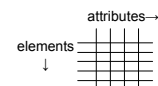
Term "relation" is used to refer to a schema and a particular instance – disambiguate by context

## Relational Database: More Terminology

Each  $D_i$  of a schema is referred to as a component or attribute or field or column of the schema

Each  $d_i$  of a tuple =  $(d_1, d_2, d_3, \dots, d_k)$  is referred to as component or attribute or field of the tuple

Each tuple of a relation is also referred to as an element or row of the relation



## Translating ER model to relational

- Domains → domains
- Entity → relation
- Relationship → one\* or more relations
  - \* come back to
- Constraints → constraints BUT
  - Not all ER constraints expressible in basic relational model

Relational model is FLAT – no hierarchy!

## Our ER Example → Relational schema

For entities, get relations:

*books*: (title, ISBN#, edition, date)

*authors*:

(name, gender, birth date, place of birth, date of death)

*publishers*: (name, country, address)

Need declare domains:

e.g. title: string

Same defs *candidate keys*, *primary key*, *superkeys*

## Our ER Example → Relational schema

For relationships:

ER *published by*: (***books***, ***publishers***, in print)

becomes

*published by*: ?

ER *written by*: (***books***, ***authors***)

becomes

*written by*: ?

## Our ER Example → Relational schema

For relationships:

ER *published by*: (***books***, ***publishers***, in print)

becomes

*published by*: (*isbn#*, *publisher\_name*, in print)

ER *written by*: (***books***, ***authors***)

becomes

*written by*: ?

## Our ER Example → Relational schema

For relationships:

ER *published by*: (***books***, ***publishers***, in print)

becomes

*published by*: (*isbn#*, *publisher\_name*, in print)

ER *written by*: (***books***, ***authors***)

becomes

*written by*:

(*isbn#*, *author\_name*, *birth date*, *place of birth*)

Keys for these?

## Our ER Example → Relational schema

For relationships:

ER *published by*: (***books***, ***publishers***, in print)

becomes

*published by*: (*isbn#*, *publisher\_name*, in print)

key constraint on entity *books* in relationship *published by* →  
A book has at most one publisher

ER *written by*: (***books***, ***authors***)

becomes

*written by*:

(*isbn#*, *author\_name*, *birth date*, *place of birth*)

### Our ER Example → Relational schema

Because ER key constraint on entity *books* in relationship *published by*  
Can fold relation *published by* into relation *books*:

*books*:  
(title, ISBN#, edition, date, pub\_name, in print)

What if some books not published?  
i.e. entity *books* not totally participate in relationship *published by*

### Our ER Example → Relational schema

*books*:  
(title, ISBN#, edition, date, pub\_name, in print)

What if some books not published?  
i.e. entity *books* not totally participate in relationship *published by*

Must allow values of attributes  
*pub\_name* and *in print* to be null

### Translating ER model to relational

General conclusion:  
Relationship → one or more relations

### Translating ER model to relational

- Get flat set of relations
- But relations are interrelated
  - Bring together primary keys of different relations to build new relation
  - Captures ER relationship
- How capture this in relational model?  
Foreign key constraints

### Foreign key constraint

- Specify that a set of attributes in schema for one relation form a primary key for a specific other relation
  - “other relation” is referred to or referenced by first relation

R1: (attrib1, attrib2, attrib3, attrib4, attrib5)  
R1 refers to/references R2  
R2: (attrib1, attrib2, attrib3, attrib4)

### Foreign Keys for Our Example

*published by*: (isbn#, publisher\_name, in print)  
*isbn#* is a foreign key referencing *books*  
Primary key of *books* understood  
*Publisher\_name* is a foreign key referencing *publishers*

*written by*:  
(isbn#, author\_name, birth date, place of birth)  
*isbn#* is a foreign key referencing *books*;  
(*author\_name*, *birth date*, *place of birth*) is a foreign key referencing *authors*

## Board Examples