

Introduction to relational databases

Tables

Relational databases organize data into tables

Tables can be linked together

A table is a relation.

Is a relation also a table?

| ISBN | Title |
|---------------|--|
| 7654321123456 | Creating relational databases for fun and profit |
| 9876543212345 | Relational databases for really, really smart people |
| 3212345678909 | My life with relational databases: a memoir |
| 8172635412345 | Relational databases: an existential journey |
| | |

Figure 1: Some sample books

Table organization

Each row in the table describes a single book

The data is organized into columns

Each *entry* (or cell) contains **a single piece of data**.

How do we handle a book with two authors?

Table 1: books with multiple authors

| ISBN | title | Authors |
|----------|--|--|
| 62112346 | Creating relational databases for fun and profit | Lopez Baranda Christina, Jones Hannah, Turay Tandice |
| 84321235 | Relational databases for really, really smart people | Novak Stanislaw, Turay Tandice |
| 64567899 | My life with relational databases: a memoir | Roy Shanta |
| 87261235 | Relational databases: an existential journey | Khatami Paree |

Not a good idea

Linked tables

- One cell, one data
- One table, one set of similar facts/situations

Two tables can be linked to obtain more information.

Needs an identifier (ID) for the each row (**primary key**)

Table 2: books

| book_id | ISBN | title |
|---------|---------------|--|
| 1 | 7654321123456 | Creating relational databases for fun and profit |
| 2 | 9876543212345 | Relational databases for really, really smart people |
| 3 | 3212345678909 | My life with relational databases: a memoir |
| 4 | 8172635412345 | Relational databases: an existential journey |

Table 3: Authors

| author_id | last_name | first_name |
|-----------|---------------|------------|
| 1 | Lopez Baranda | Christina |
| 2 | Jin-Soon | Sin |
| 3 | Jones | Hannah |
| 4 | Novak | Stanislaw |
| 5 | Turay | Tandice |
| 6 | Roy | Shanta |
| 7 | Berger | Henry |
| 8 | Khatami | Paree |

Table 4: BooksAuthors

| book_id | author_id |
|---------|-----------|
| 3 | 6 |
| 2 | 4 |
| 2 | 5 |
| 1 | 1 |
| 1 | 3 |
| 1 | 5 |
| 4 | 8 |

Bridging table

Tables, relationships, and IDs

- intermediate table: *relation* or *join* or *bridging* table
- each row must be referenced uniquely
- so that we can reference a row *from* a different table
- join tables are not necessary to refer to another table — e.g. one-to-many relations

Table 5: Editions

| edition_id | book_id | date_of_publication | edition_number |
|------------|---------|---------------------|----------------|
| 1 | 3 | 2001 | 1 |
| 2 | 3 | 2003 | 2 |
| 3 | 4 | 2003 | 1 |
| 5 | 1 | 2000 | 1 |
| 6 | 3 | 2005 | 3 |
| 8 | 2 | 2012 | 1 |
| 9 | 3 | 2009 | 4 |

Standard table with reference to another table: **foreign key**

A goal of DBMS is to translate **data** into **information**

- Structured data are more informative
- Constraints are a form of structure
- Examples: instance consistent with schema, foreign key, $\text{day} \leq 31$

NULL values

- Whenever we don't have a value, we can put a NULL
- Can mean:
 - Value does not exists
 - Value exists but is unknown
- Introduce flexible schema
 - First Name, *Second name*, Surname
- Neither TRUE nor FALSE

Data Anomalies

| Teacher | Course ID | Course Name |
|----------------|-----------|-------------|
| Mary Smith | 3 | Calculus |
| Ann Brown | 4 | Programing |
| Michael Jordan | 4 | Programing |

- Who teaches a course
- Course name and ID
- What if we fix the typo?
- Can we have a course without teachers?

Functional dependency

Definition: A, B : set of attributes

Then $A \rightarrow B$ if, for any tuples t_1 and t_2 , $t_1[A] = t_2[A] \Rightarrow t_1[B] = t_2[B]$

- $A \rightarrow B$ is a functional dependency
- A functional dependency is information
- A **bad** functional dependency is a problem
 - Goal: remove bad functional dependencies
 - How: change the schema
 - Hurdle: need the instance to have a dependency

Definition: K : set of attributes of relation R B is **superkey** of R if for any set B such that $K \cap B = \emptyset$, then $K \rightarrow B$. Equivalent to for any tuples t_1 and t_2 ,
 $t_1[K] = t_2[K] \Rightarrow t_1 = t_2$

Definition: K superkey of relation R . K is **key** if no proper subset of K is a superkey of R

A key is a minimal superkey

Example: Editions table

1. The pair (edition_id, edition_number) of the table Editions is a superkey, since there are no two rows of Editions that have the same values of edition_id and edition_number
2. The pair (edition_id, edition_number) of the table Editions is not a key, since we can remove an attribute (edition_number) and obtain a superkey (the single attribute edition_id)
3. The attribute edition_id is a key. Since it consists of a single attribute, its minimality is trivial
4. The pair (book_id, edition_number) is a superkey. Since removing any of its attribute results in a set of attributes that have duplicates, (book_id, edition_number) is also a key.
5. book_id is a foreign key, since the values of that attribute are found in the key of the Books table.

Boyce-Codd Normal Form

Definition: Let $A \rightarrow B$ for relation R . Then $A \rightarrow B$ is good if A is a superkey of R .

Definition: A relation R is in Boyce-Codd Normal Form if all its functional dependencies are good.

- Boyce-Codd Normal Form is highly desirable
- Boyce-Codd Normal Form not always achievable
- Boyce-Codd Normal Form usually achievable

Normalization

- A bad functional dependency shows which columns must be moved to a new table

| Teacher | Course ID | Course Name |
|----------------|-----------|-------------|
| Mary Smith | 3 | Calculus |
| Ann Brown | 4 | Programming |
| Michael Jordan | 4 | Programming |

- Bad functional dependency: $\text{Course ID} \rightarrow \text{Course Name}$
 - New table with attributes Course ID, Course Name
 - Remove Course Name from current table
 - Result: two linked tables

Normalization

| Teacher | Course ID |
|----------------|-----------|
| Mary Smith | 3 |
| Ann Brown | 4 |
| Michael Jordan | 4 |

| Course ID | Course Name |
|-----------|-------------|
| 3 | Calculus |
| 4 | Programming |

Duplicate rows cannot exist

Planets.csv

```
method,number,orbital_period,mass,distance,year  
Radial Velocity,1,269.3,7.1,77.4,2006  
Radial Velocity,1,874.774,2.21,56.95,2008  
Radial Velocity,1,763.0,2.6,19.84,2011
```

- different separators: space, tab
- rigid structure
- one table per file

JSON file

```
{ "menu": {  
  "id": "file",  
  "value": "File",  
  "popup": {  
    "menuitem": [  
      { "value": "New", "onclick": "CreateNewDoc()" },  
      { "value": "Open", "onclick": "OpenDoc()" },  
      { "value": "Close", "onclick": "CloseDoc()" }  
    ]  
  }  
}}
```

- nested tables
- loose structure

Except where noted, text and images for Introduction to Relational Databases by Mark Jordan is licensed under a Creative Commons Attribution 4.0 International License. Except where noted, text and images for Introduction to Relational Databases by Gianluca Della Vedova is licensed under a Creative Commons Attribution 4.0 International License. Everything in the 'scripts' directory is in the public domain (CC0).