Universität Zürich[UZH]

Databases I

Language Technology and Web Applications

Johannes Graën

Department of Computational Linguistics &
Linguistic Research Infrastructure (LiRI)

October 4, 2023

Learning Goals for this Week

- You can draw an Entity Relationship diagram
- You can create a database (using command line and Python) and insert values

Topics

1. Data Modeling

2. Creating a Database

Motivation

We want to manage data (e.g. a list of books) in a structured way
⇒ **Databases**

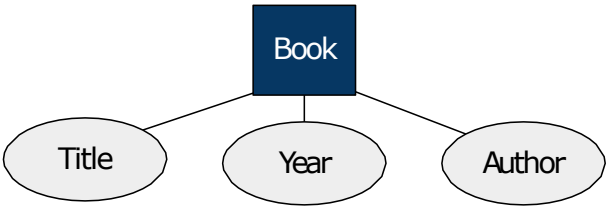
To create a database, we need to understand the logical structure of the data
⇒ **Data Modeling** using an **Entity Relationship Model**

Topics

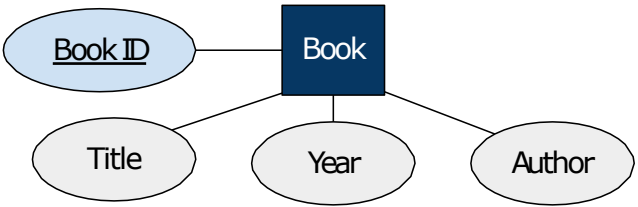
1. Data Modeling

2. Creating a Database

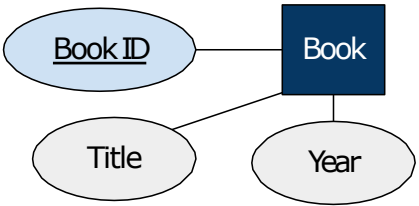
Entities and Attributes

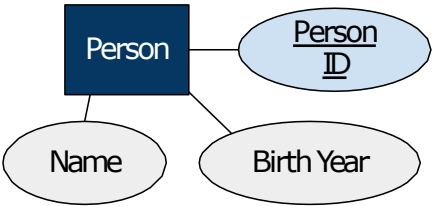


Identification Key

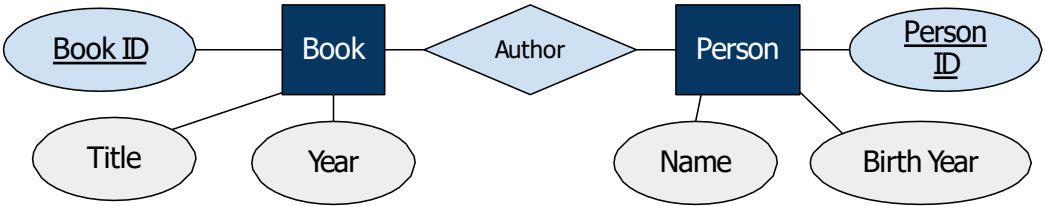


Multiple Entities

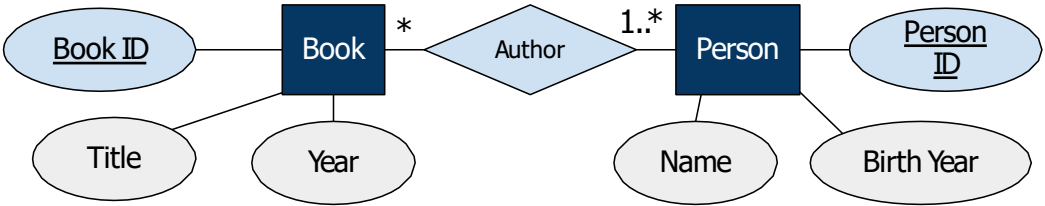




Relationships



Cardinalities



Exercise

How might YouTube's database be structured? Draw an Entity Relationship diagram for a video sharing platform.

Topics

1. Data Modeling

2. Creating a Database

Relational Databases

Definition

A **Relational Database** represents entities and relationships as tables (mathematically: as relations, i.e., sets of tuples).

Book ID	**Title**	**Year**
1	War and Peace	1869
2	Hamlet	1603
...	...	...

Features of Relational Databases

Relational database management systems usually implement the following features:

- **Schema**: Tables and attributes are formally defined.
- **SQL**: The database supports *Structured Query Language* (SQL).
- **Multi-User**: Several users can access the database without conflicts.
- **Permission Management**

ACID principle

- Atomicity
- Consistency
- Isolation
- Durability

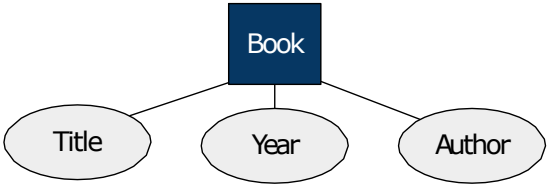
Examples of Relational Database Management Systems

- PostgreSQL
- MySQL
- Oracle (proprietary)
- SQLite
- DB2 (proprietary)
- SQL Server (proprietary)

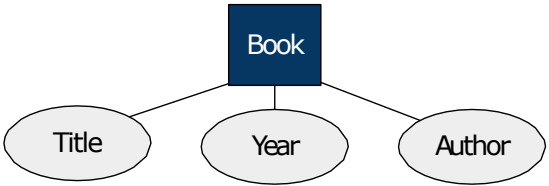
Creating an SQLite Database in Python

```
import sqlite3

# Creates file "example.db" if necessary
connection = sqlite3.connect('example.db')
cursor = connection.cursor()

# SQL statement to create a table
cursor.execute('''
CREATE TABLE book (
    title TEXT,
    year  INT
);
''')

# Save (commit) the changes
connection.commit()
connection.close()
```

Entities as Database Tables

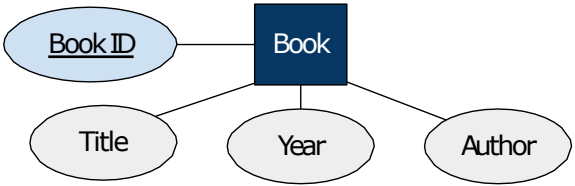


Entities as Database Tables



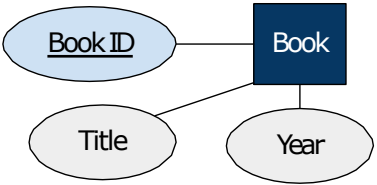

```
CREATE TABLE book (
    title  TEXT,
    year   INT,
    author TEXT
);
```

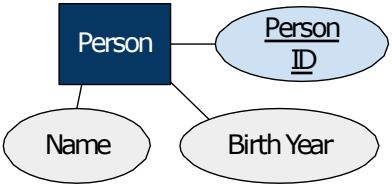
Primary Keys



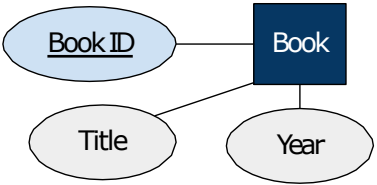

```
CREATE TABLE book (
    id      INT PRIMARY KEY,
    title   TEXT,
    year    INT,
    author TEXT
);
```

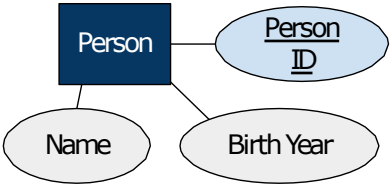
Multiple Tables





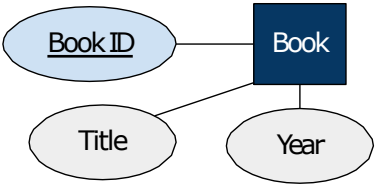
Multiple Tables

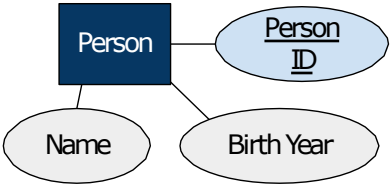





```
CREATE TABLE book (
    id      INT PRIMARY KEY,
    title   TEXT,
    year    INT
);
```

Multiple Tables

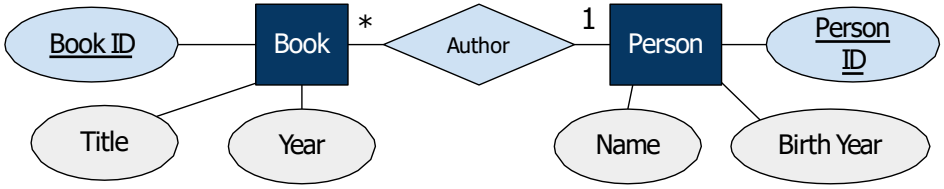





```sql
CREATE TABLE book (
    id      INT PRIMARY KEY,
    title   TEXT,
    year    INT
);
```

```sql
CREATE TABLE person (
    id          INT PRIMARY KEY,
    name        TEXT,
    birth_year  INT
);
```

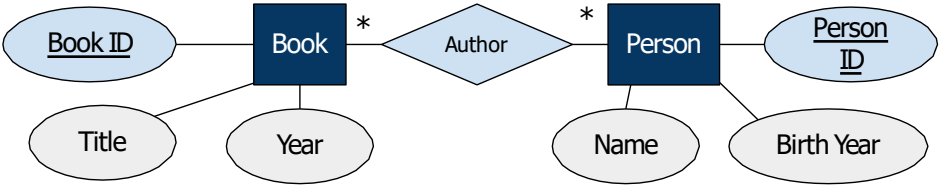
Foreign Keys



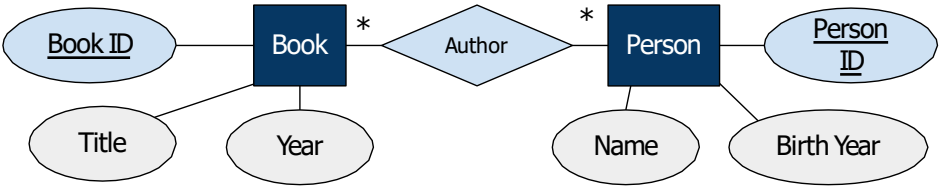

```sql
CREATE TABLE book (
    id          INT PRIMARY KEY,
    title       TEXT,
    year        INT,
    author_id INT REFERENCES person(id)
);
```

```sql
CREATE TABLE person (
    id          INT PRIMARY KEY,
    name        TEXT,
    birth_year INT
);
```

Relationships as Tables



Relationships as Tables




```
CREATE TABLE author (
     id         INT PRIMARY KEY,
     book_id    INT REFERENCES book(id),
     person_id INT REFERENCES
     person(id)
);
```

Common Data Types

- `INT`
- `REAL/FLOAT`
- `TEXT`
- `BOOLEAN`
- `DATE`
- `BLOB` (= binary large object)

https://www.postgresql.org/docs/current/datatype.html

Inserting Values

```
/*
CREATE TABLE book
    (
     id      INT PRIMARY KEY,
     title   TEXT,
     year    INT,
     author TEXT
    );
*/

INSERT INTO book VALUES
    (1, 'War and Peace', 1869, 'Leo Tolstoy'),
    (2, 'Hamlet', NULL, 'William Shakespeare');
```

Constraints

```
CREATE TABLE book
   (
      id      INT  PRIMARY KEY,
      title   TEXT NOT NULL,
      isbn    TEXT UNIQUE
   );
```

Deleting a Table

```
DROP TABLE book;

-- or:

DROP TABLE IF EXISTS book;
```