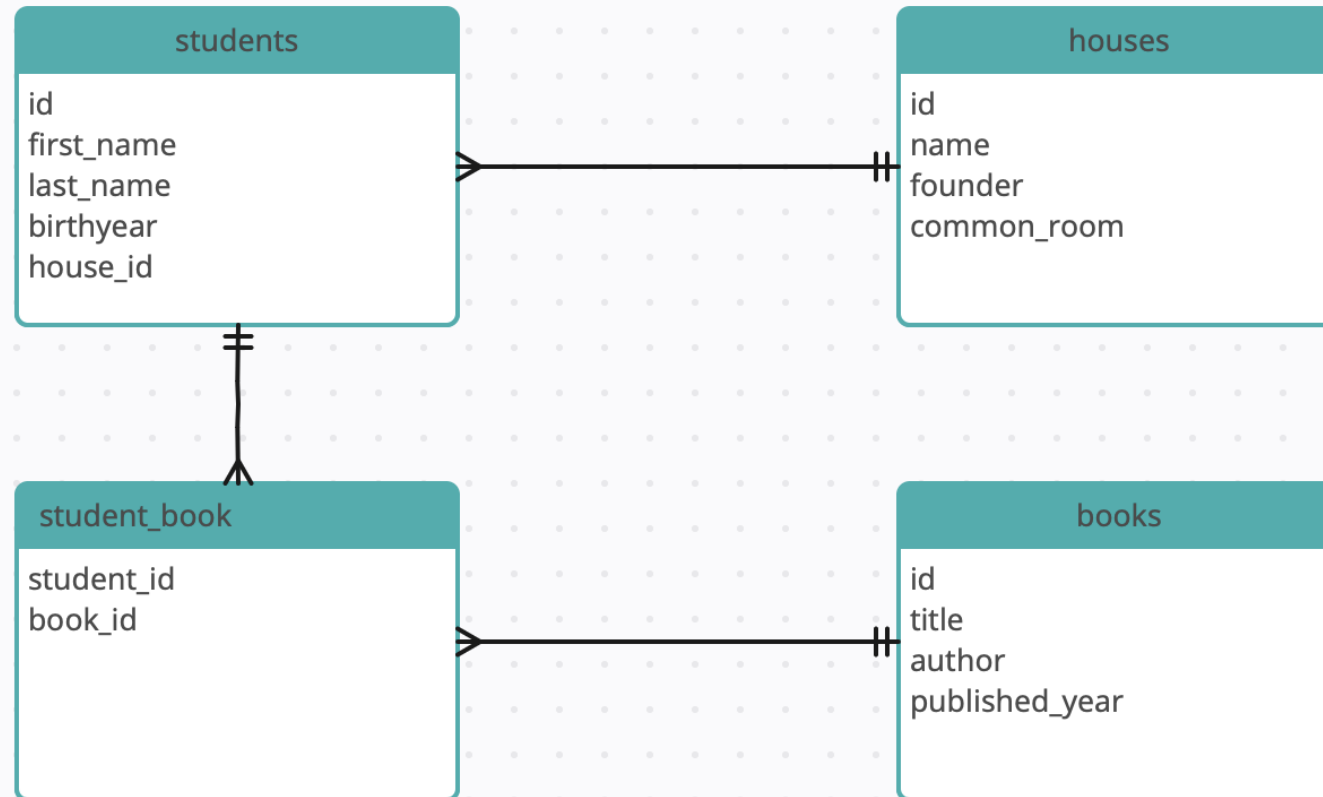# DDL & DML

# DDL: Data Definition Language

- `CREATE TABLE`
- `ALTER TABLE`
- `DROP TABLE`

2

# DML: Data Manipulation Language

- `INSERT INTO`
- `UPDATE`
- `DELETE FROM`

# Expanding the Harry Potter database

# Create a new table

```sql
CREATE TABLE books (
    id numeric,
    title text,
    author text,
    published_year numeric,
    PRIMARY KEY (id)
);
```

5

# Data types

`numeric`

- `int`
- `float`

  ...

`text`

- `varchar(n)`
- `text`

  ...

6

# Constraints to limit what values can be inserted into column

- `primary key`

- `foreign key`

- `not null`

- `unique`

- `check(exp)`

  …

# **INSERT INTO** table

```sql
INSERT INTO books
VALUES (1, 'Harry Potter and the Sorcerer''s Stone', 'J.K. Rowling', 1997);

INSERT INTO books (id, title, author, published_year)
VALUES (1, 'Harry Potter and the Sorcerer''s Stone', 'J.K. Rowling', 1997);

INSERT INTO books (id, title, author, published_year)
VALUES
    (1, 'Harry Potter and the Sorcerer''s Stone', 'J.K. Rowling', 1997),
    (2, ...),
    (3, ...);
```

# Write SQL script to create and populate tables

```sql
CREATE TABLE books (
    id int,
    title text,
    author text,
    published_year int,
    PRIMARY KEY (id)
);
CREATE TABLE student_book (
    student_id int,
    book_id int,
    PRIMARY KEY (student_id, book_id),
    FOREIGN KEY (student_id) REFERENCES students(id),
    FOREIGN KEY (book_id) REFERENCES books(id)
);
```

# Insert data into tables

```sql
INSERT INTO books (id, title, author, published_year) VALUES
(201, 'Harry Potter and the Philosopher''s Stone', 'J.K. Rowling', 1997),
(202, 'Harry Potter and the Chamber of Secrets', 'J.K. Rowling', 1998),
(203, 'Harry Potter and the Prisoner of Azkaban', 'J.K. Rowling', 1999),
(204, 'Harry Potter and the Goblet of Fire', 'J.K. Rowling', 2000),
(205, 'Harry Potter and the Order of the Phoenix', 'J.K. Rowling', 2003),
(206, 'Harry Potter and the Half-Blood Prince', 'J.K. Rowling', 2005),
(207, 'Harry Potter and the Deathly Hallows', 'J.K. Rowling', 2007);
```

```sql
INSERT INTO student_book (student_id, book_id)
VALUES
(101, 201), (101, 202), (101, 203), (102, 201), (102, 203),
(102, 204), (103, 201), (103, 204), (104, 202), (104, 205),
(105, 203), (105, 206), (106, 201), (106, 203), (106, 207),
(107, 205), (107, 206), (108, 204), (108, 206), (109, 203),
(109, 207), (110, 205), (111, 204), (112, 201), (113, 203),
(114, 202), (115, 205), (116, 204), (117, 201), (118, 202);
```

# 🖥️ Add Movies table to Harry Potter database

- table name: `movies`
- columns:
  - `id` int
  - `title` text
  - `director` text
  - `year` int
- primary key: `id`

# 🖥️ Insert data into Movies table

```
1, 'Harry Potter and the Sorcerer''s Stone', 'Chris Columbus', 2001
2, 'Harry Potter and the Chamber of Secrets', 'Chris Columbus', 2002
3, 'Harry Potter and the Prisoner of Azkaban', 'Alfonso Cuarón', 2004
4, 'Harry Potter and the Goblet of Fire', 'Mike Newell', 2005
5, 'Harry Potter and the Order of the Phoenix', 'David Yates', 2007
6, 'Harry Potter and the Half-Blood Prince', 'David Yates', 2009
7, 'Harry Potter and the Deathly Hallows – Part 1', 'David Yates', 2010
8, 'Harry Potter and the Deathly Hallows – Part 2', 'David Yates', 2011
```

# DDL: ALTER TABLE

- add columns

```
ALTER TABLE mytable
ADD column DataType OptionalTableConstraint
    DEFAULT default_value;
```

- remove columns

```
ALTER TABLE mytable
DROP column_name;
```

- rename the table

```
ALTER TABLE mytable
RENAME TO new_table_name;
```

14

# DDL: `DROP TABLE`

```sql
DROP TABLE mytable;
DROP TABLE IF EXISTS mytable;
```

# DML: **UPDATE** table

```sql
UPDATE books
SET published_year = 1998
WHERE id = 1;
```

# DML: `DELETE FROM` table

```sql
DELETE FROM books
WHERE id = 1;
```

1. remove director column from the movies table

2. add a new column `director_id` to the movies table

3. delete the movie with id 1 from the movies table

4. update the year of the movie with id 2 to 2003

5. drop the movies table