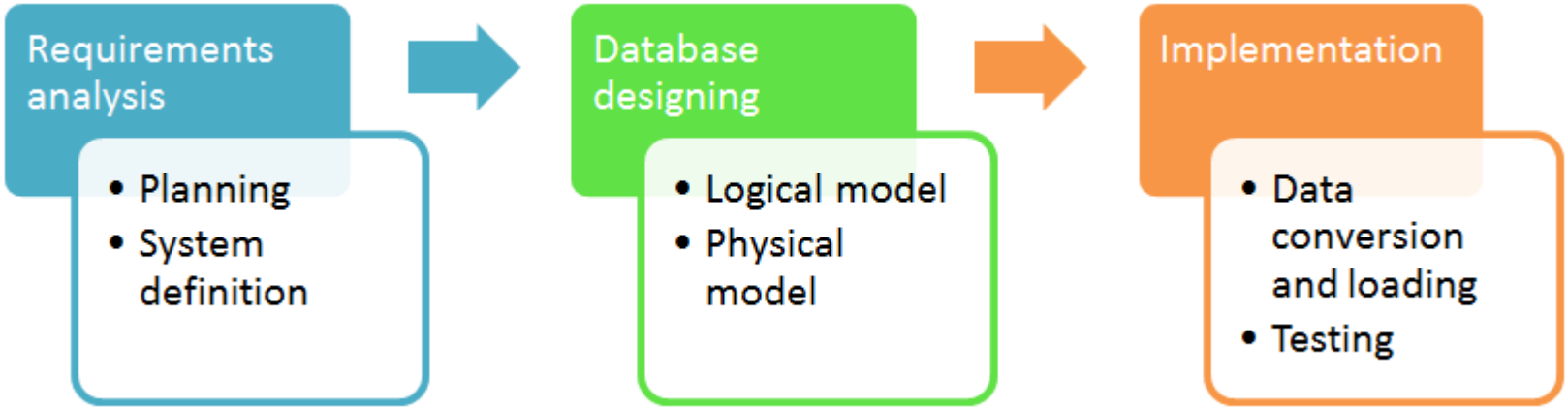
Database Design Concept

Database design (thiết kế CSDL) là quá trình xác định và tổ chức cách dữ liệu được lưu trữ và quản lý trong một hệ thống CSDL. Nó liên quan đến thiết kế cấu trúc dữ liệu, quy định mối quan hệ giữa các thành phần dữ liệu và thiết lập các ràng buộc để đảm bảo tính nhất quán, hiệu quả và bảo mật của CSDL

Các bước thực hiện database design



Database Design Concept

Requirements analysis – Phân tích yêu cầu

Trong database design chia ra 2 nhóm đối tượng là: người thực hiện (database designer) và người sử dụng (database users). Database designer phải tổng hợp và lập kế hoạch triển khai sao cho đáp ứng được tất cả mong đợi, yêu cầu và mục đích sử dụng của database users. Kế hoạch này phải thể hiện được rõ ràng các bước tổng hợp, chọn lọc, cách cấu trúc dữ liệu và ước tính kết quả trả ra.

Database designer và database users cần bàn bạc thống nhất với nhau về kế hoạch. Các nghiệp vụ, các trường dữ liệu, phạm vi, điều kiện ràng buộc…

Database designing – Thiết kế Cơ sở dữ liệu

Đây được xem là bước quan trọng nhất của thiết kế cơ sở dữ liệu. Mục tiêu chính của bước này là tạo ra thiết kế logic của dữ liệu (logical design) và thiết kế vật lý của dữ liệu (physical design) theo mô hình cơ sở dữ liệu.
Thiết kế logic là cách các dữ liệu kết nối với nhau, có thể hiểu là sơ đồ vận hành của các nhóm dữ liệu nhằm trả ra kết quả theo yêu cầu

Physical model - Còn thiết kế vật lý là cách các dữ liệu ở sơ đồ trên được lưu trữ cụ thể ở phần cứng nào và các chi tiết của dữ liệu sẽ được sắp đặt ra sao. Hay hiểu đơn giản là cài đặt và thiết lập database lên server vật lý theo mô hình nào đó để có hiệu suất cao, bảo mật…phù hợp với tính chất yêu cầu của ứng dụng

Ví dụ:
Google có rất nhiều server (máy chủ dữ liệu) ở nhiều quốc gia.
Tổ chức logical model tất cả các server đều giống nhau
Còn server tại mỗi quốc gia thì lưu trữ dữ liệu người dùng tại quốc gia đó → hiệu suất truy vấn cao.

Implementation – Triển khai

Ở bước triển khai thì dữ liệu sẽ chính thức được chuyển đổi, tải từ hệ thống cũ sang cơ sở dữ liệu mới theo định hướng của logical design và physical design. Lúc này, các database designer sẽ giám sát dữ liệu tuần tự được trả về các nhóm, đảm bảo tính kết nối theo sơ đồ đã vạch ra.

Giai đoạn này liên quan đến việc xác định lỗi trong hệ thống mới triển khai. Nó kiểm tra cơ sở dữ liệu so với các yêu cầu kỹ thuật.

Tầm quan trọng của Database Design

1 Tính nhất quán và độ tin cậy

Nó đảm bảo rằng dữ liệu được lưu trữ và quản lý một cách chính xác và không bị trùng lặp

2 Hiệu suất và tối ưu hóa

Giảm thiểu thời gian truy cập và truy vấn dữ liệu, đồng thời cải thiện hiệu suất toàn bộ hệ thống

3 Bảo mật dữ liệu

Phân quyền truy cập dữ liệu, ngăn chặn truy cập trái phép và bảo vệ dữ liệu quan trọng

4 Dễ dàng bảo trì và mở rộng

Giảm thiểu công việc bảo trì, dễ dàng nâng cấp mở rộng

5 Tìm kiếm, truy xuất nhanh

Tổ chức dữ liệu hợp lý, làm cho việc tìm kiếm, truy xuất và phân tích dữ liệu trở nên dễ dàng hơn

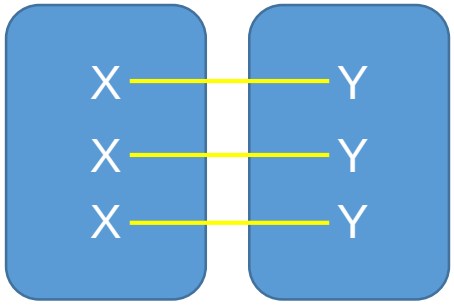
Relationships

PostgreSQL là một hệ quản trị cơ sở dữ liệu quan hệ mạnh mẽ và phổ biến

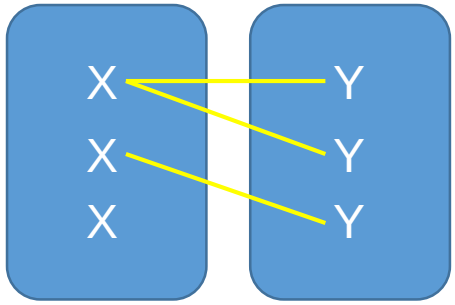
Mối quan hệ (relationship) trong PostgreSQL cho phép liên kết giữa các bảng dữ liệu trong cơ sở dữ liệu (CSDL)

Có 3 kiểu quan hệ phổ biến trong thiết kế CSDL:

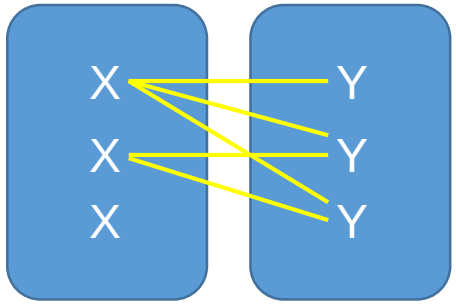
Kiểu quan hệ	Ví dụ
One-to-One	A User has ONE address
One-to-Many	**A** Book has **MANY** reviews
Many-to-Many	**A** user has **MANY** books and a book has **MANY** users



One-to-one



One-to-Many



Many-to-Many

1 Quan hệ một-một (One-to-One Relationship)

Một hàng trong bảng A chỉ liên kết với một hàng trong bảng B và ngược lại.

Ví dụ:
- Một User chỉ có 1 địa chỉ, và một địa chỉ thuộc về chỉ một User
- Một người chỉ có một số CCCD duy nhất và một số CCCD chỉ thuộc về một người duy nhất.

```sql
CREATE TABLE users (
  id serial,
  username VARCHAR(25) NOT NULL,
  enabled boolean DEFAULT TRUE,
  last_login timestamp NOT NULL DEFAULT NOW(),
  PRIMARY KEY (id)
);

CREATE TABLE addresses (
  user_id int NOT NULL,
  street VARCHAR(30) NOT NULL,
  city VARCHAR(30) NOT NULL,
  state VARCHAR(30) NOT NULL,
  PRIMARY KEY (user_id),
  CONSTRAINT fk_user_id FOREIGN KEY (user_id) REFERENCES users (id)
);
```

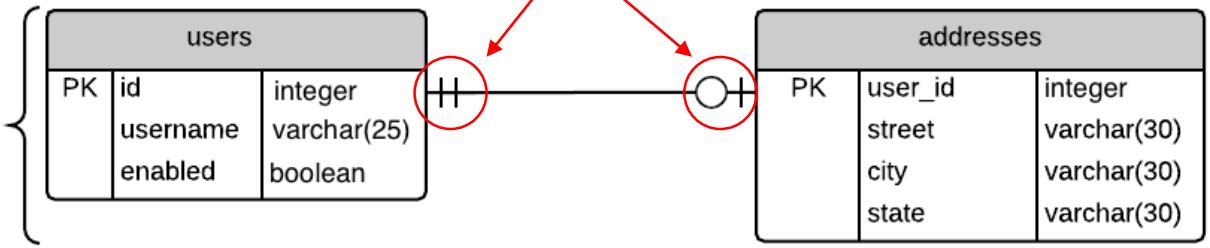
1 Quan hệ một-một (One-to-One Relationship)

Table **Users** được xem là table CHA và table **Addresses** được xem là CON có quan hệ với CHA là Users

Mối quan hệ đó được thể hiện bằng một **khóa ngoại user_id**, tham chiếu đến trường **id** của table CHA là Users

Kí hiệu biểu diễn mối quan hệ giữa các table 1-1

Schema



Data

Users

enabled	username	id
1	John Smith	1
1	Jane Smiley	2

PK

Addresses

user_id	street	city	state
1	1 Market Street	San Francisco	CA
2	2 Elm Street	San Francisco	CA

PK,FK

user_id is PK and FK referenced from users(id)

2 Quan hệ một-nhiều (One-to-Many Relationship)

Một hàng trong bảng A có thể liên kết với một hoặc nhiều hàng trong bảng B, nhưng một hàng trong bảng B chỉ liên kết với một hàng trong bảng A

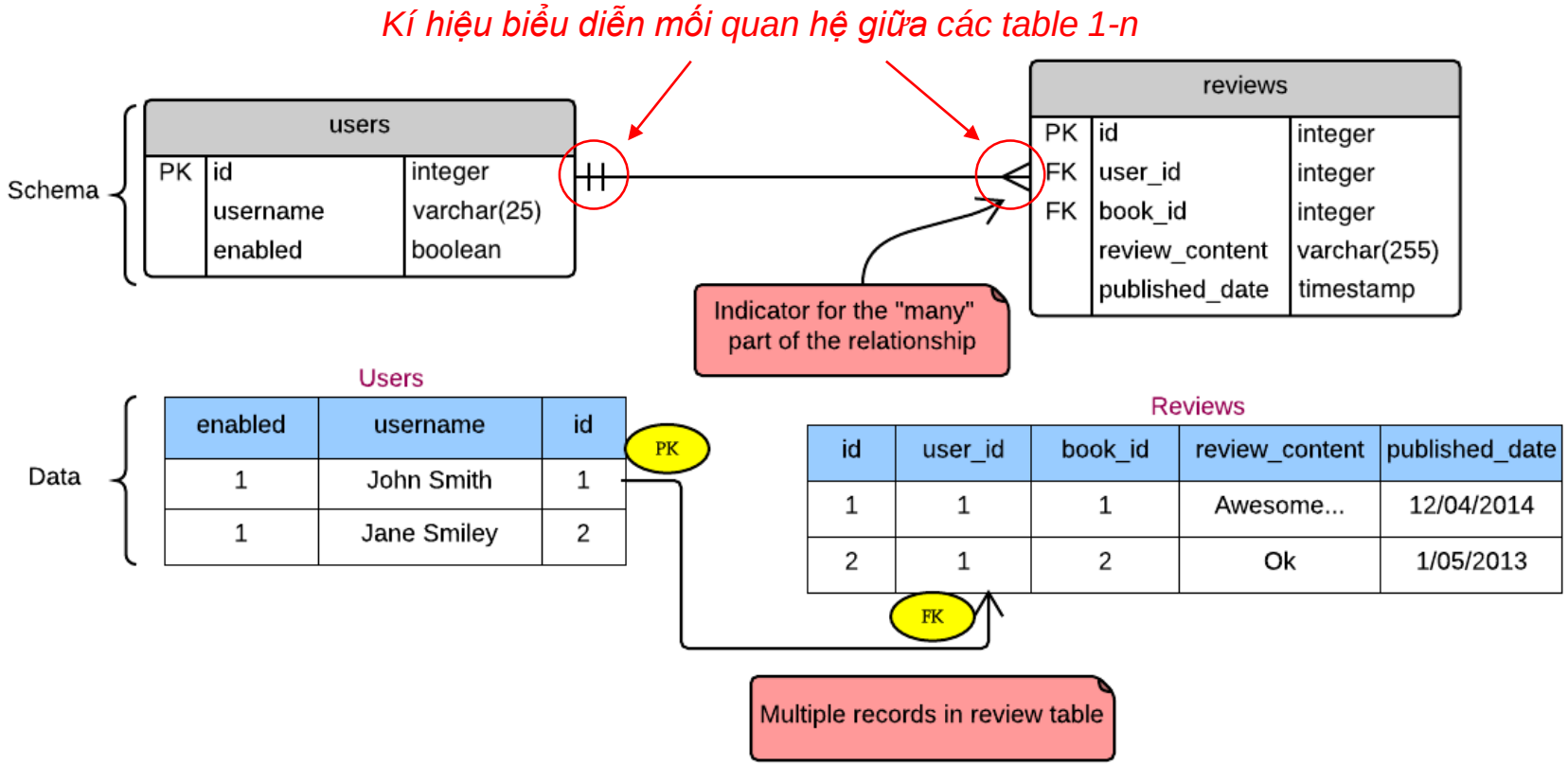
```sql
CREATE TABLE books (
  id serial,
  title VARCHAR(100) NOT NULL,
  author VARCHAR(100) NOT NULL,
  published_date timestamp NOT NULL,
  isbn int,
  PRIMARY KEY (id),
  UNIQUE (isbn)
);

CREATE TABLE reviews (
  id serial,
  book_id int NOT NULL,
  user_id int NOT NULL,
  review_content VARCHAR(255),
  rating int,
  published_date timestamp DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (id),
  FOREIGN KEY (book_id) REFERENCES books(id) ON DELETE CASCADE,
  FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE CASCADE
);
```

Ví dụ:
Một cuốn sách có nhiều người reviews
Và một review đó thì chỉ thuộc về 1 cuốn sách

2 Quan hệ một-nhiều (One-to-Many Relationship)

Table **Books** được xem là table CHA và table **Reviews** được xem là CON có quan hệ với CHA là **Books**

Mối quan hệ đó được thể hiện bằng một **khóa ngoại book_id**, tham chiếu đến trường **id** của table CHA là **Books**



3 Quan hệ nhiều-nhiều (Many-to-Many Relationship)

Một hàng trong bảng A có thể liên kết với một hoặc nhiều hàng trong bảng B và ngược lại

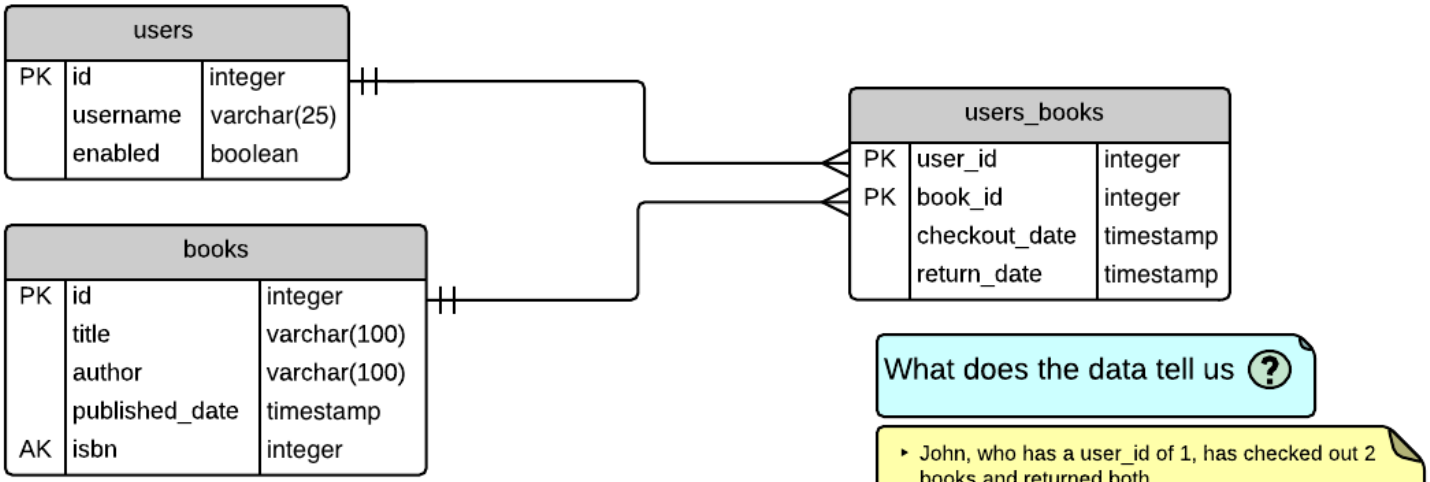
Với kiểu quan hệ này thông thường tạo ra một **bảng phụ** với khóa chính được tạo từ 2 columns (Composite Primary Keys) để ràng buộc quan hệ như dưới đây:

```sql
CREATE TABLE users_books (
  user_id int NOT NULL,
  book_id int NOT NULL,
  checkout_date timestamp,
  return_date timestamp,
  PRIMARY KEY (user_id, book_id),
  FOREIGN KEY (user_id) REFERENCES users(id) ON UPDATE CASCADE,
  FOREIGN KEY (book_id) REFERENCES books(id) ON UPDATE CASCADE
);
```

khóa ngoại **user_id** tham chiếu tới trường **id** của table **Users**

khóa ngoại **book_id** tham chiếu tới trường **id** của table **Books**

3 Quan hệ nhiều-nhiều (Many-to-Many Relationship)



What does the data tell us ?

- John, who has a user_id of 1, has checked out 2 books and returned both.
- Later, Jane, who has a user_id of 2, checked out a book that John had previously checked out
- John and Jane both checked out "My Second SQL Book", which has a book_id of 2

PK

Users

enabled	username	id
1	John Smith	1
1	Jane Smiley	2

FK FK

Users_books

user_id	book_id	checkout_date	return_date
1	2	11/04/2014	12/04/2014
1	1	11/02/2013	11/22/2013
2	2	12/01/2014	

PK

Books

author	title	published_date	isbn	id
Mary Parker	My First SQL book	2014-12-04	123..	1
John Mayer	My Second SQL book	2014-12-04	344...	2