



# Cơ sở dữ liệu

Ba Nguyễn



# Nội dung

1. Giới thiệu về Cơ sở dữ liệu (CSDL)
2. Ứng dụng
3. Các dạng CSDL
4. Một số Hệ quản trị CSDL phổ biến
5. Thực hành một số câu lệnh truy vấn



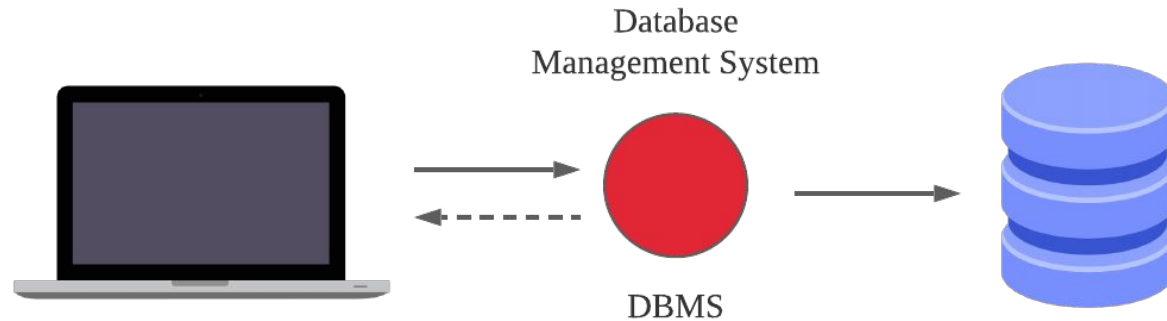
## Dữ liệu, Cơ sở dữ liệu, Hệ quản trị CSDL

**Dữ liệu** (data) là các thông tin mô tả về một sự vật, sự việc hoặc hiện tượng nào đó. Dữ liệu có thể là các chữ số, văn bản, hình ảnh hoặc âm thanh, ...

**Cơ sở dữ liệu** (database - DB) là một tập hợp dữ liệu được tổ chức theo một cách nào đó, CSDL thường được lưu trữ và truy cập trên các hệ thống máy tính

**Hệ quản trị Cơ sở dữ liệu** (Database Management System - DBMS) là một phần mềm cung cấp tương tác giữa người dùng, ứng dụng và cơ sở dữ liệu, cho phép kết nối và truy vấn dữ liệu trong CSDL

# Dữ liệu, Cơ sở dữ liệu, Hệ quản trị CSDL





## Ứng dụng

DBMS cung cấp nhiều chức năng cho phép các hệ thống, trang web và ứng dụng thao tác với DB:

- Định nghĩa dữ liệu: Tạo, chỉnh sửa các định nghĩa xác định cấu trúc dữ liệu
- Cập nhật dữ liệu: Thêm, sửa, xóa dữ liệu
- Truy xuất: Tìm kiếm và cung cấp thông tin để sử dụng trong các ứng dụng, trang web, ...
- Quản trị: Kiểm soát người truy cập, sao lưu dữ liệu để phòng sự cố, ...



# Các dạng Database



## Các dạng Database

Hiện nay có nhiều DBMS khác nhau, chúng được chia thành 2 dạng chính dựa theo cách thức lưu trữ và truy vấn dữ liệu:

- CSDL quan hệ (Relational - SQL) ([Danh sách RDBMS - Wiki](#))
- CSDL phi quan hệ (Non-Relational - NoSQL) ([Danh sách NoSQL - Wiki](#))



Relational

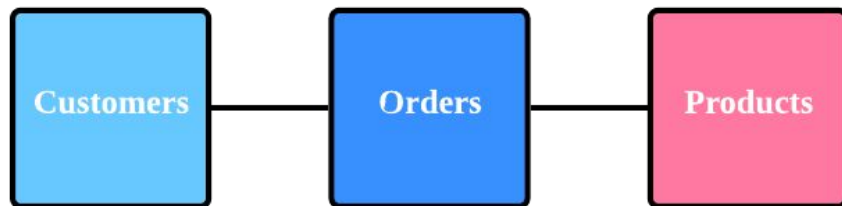


Non-Relational

# Relational DBMS

**RDBMS** sử dụng mô hình dữ liệu quan hệ, dữ liệu được cấu trúc và lưu trữ dưới dạng các bảng (tương tự excel).

Mỗi bảng dữ liệu sẽ lưu trữ về một đối tượng, bao gồm các cột và hàng. Mỗi cột thể hiện một thông tin về đối tượng, mỗi hàng đại diện cho một đối tượng cụ thể. Dữ liệu trong các bảng liên kết với nhau thông qua các mối quan hệ.







## Relational DBMS

	product_id	name	quantity_in_stock	unit_price
▶	1	Foam Dinner Plate	70	1.21
	2	Pork - Bacon,back Peameal	49	4.65
	3	Lettuce - Romaine, Heart	38	3.35
	4	Brocolinni - Gaylan, Chinese	90	4.53
	5	Sauce - Ranch Dressing	94	1.63
	6	Petit Baguette	14	2.39
	7	Sweet Pea Sprouts	98	3.29
	8	Island Oasis - Raspberry	26	0.74
	9	Longan	67	2.26
	10	Broom - Push	6	1.09
*	NULL	NULL	NULL	NULL



# Relational DBMS

Các khái niệm cơ bản trong RBDMS:

- **Bảng** (table): Bao gồm tập hợp dữ liệu có liên quan đến nhau lưu trữ trong các hàng và cột
- **Trường** (field): Là thành phần nhỏ nhất trong bảng, mô tả thông tin của một bản ghi
- **Hàng/bản ghi** (row/record): Chứa toàn bộ thông tin về một đối tượng riêng biệt



# Relational DBMS

Các khái niệm cơ bản trong RBDMS:

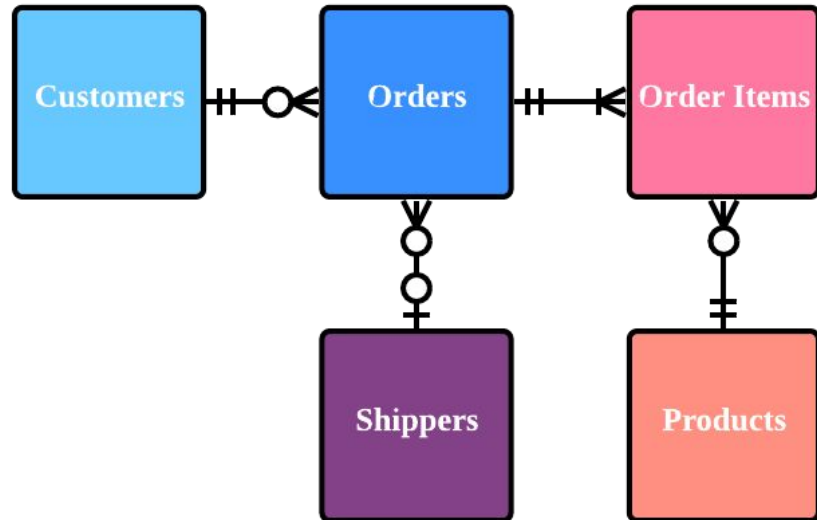
- **Cột** (column): Chứa toàn bộ thông tin liên kết đến một trường cụ thể trong bảng
- **NULL**: Là giá trị đặc biệt cho biết một trường không có giá trị (để trống) khi tạo bản ghi
- **Ràng buộc** (constraint): Là quy tắc cho các trường trong bảng, các ràng buộc được sử dụng để giới hạn kiểu dữ liệu (giá trị) có thể nhập vào khi tạo bản ghi, điều này đảm bảo tính chính xác và tin cậy của dữ liệu

# Relational DBMS

Mối quan hệ giữa các bảng:

- 1 - 1 (one to one)
- 1 - n (one to many)
- n - m (many to many)
- Mandatory (bắt buộc)
- Optional (tùy chọn)

💡 Tìm hiểu về [Crow's Foot Notation](#)





## Relational DBMS

**SQL** (Structured Query Language - ngôn ngữ truy vấn có cấu trúc) là ngôn ngữ phổ biến được sử dụng trong các RDBMS để tạo, chỉnh sửa, hoặc truy vấn dữ liệu trong DB

```
SELECT *  
FROM products  
WHERE category = "food"  
ORDER BY price
```



## Non-Relational DBMS (NoSQL)

**NoSQL** là một thuật ngữ chung để chỉ tất cả DB nào không sử dụng mô hình dữ liệu quan hệ (**SQL**), nghĩa là dữ liệu được lưu trữ theo một mô hình nào đó **không phải dạng bảng**, với các hàng và cột

**NoSQL** trở nên nổi tiếng nhờ những công ty lớn như Facebook, Google, ... và ngày càng được sử dụng phổ biến



## Non-Relational DBMS (NoSQL)

Ưu điểm của **NoSQL** so với **SQL**:

- Thiết kế đơn giản
- Linh hoạt
- Hiệu năng cao
- Khả năng mở rộng



## Non-Relational DBMS (NoSQL)

Các mô hình dữ liệu **NoSQL** phổ biến:

- **Mô hình khóa - giá trị** (key - value) hay còn được gọi là từ điển: Là mô hình đơn giản nhất, dữ liệu bao gồm một tập hợp các cặp khóa - giá trị, mỗi khóa đảm bảo chỉ xuất hiện một lần duy nhất.

Các DB sử dụng mô hình khóa - giá trị điển hình như: **Redis, Memcached, ...**





## Non-Relational DBMS (NoSQL)

Các mô hình dữ liệu **NoSQL** phổ biến:

- **Mô hình tài liệu** (document): Dữ liệu được đóng gói trong các document. Document có định dạng khác nhau (**XML**, **JSON**, **BSON**, ...) tùy thuộc vào database, tuy nhiên, mô hình này có điểm tương đồng và các khái niệm có thể so sánh được với RDBMS.

Các DB sử dụng mô hình tài liệu điển hình như: **MongoDB**, **CouchDB**, ...



## Non-Relational DBMS (NoSQL)

Ví dụ: **MongoDB** là một **NoSQL DB** sử dụng mô hình tài liệu.

Dữ liệu trong MongoDB được lưu trữ trong các **document**, các documents được lưu trữ trong một tập hợp được gọi là **collection**, và một **database** trong MongoDB là một tập hợp các collections



## Non-Relational DBMS (NoSQL)

Document có định dạng BSON, bao gồm các cặp **field: value**

- **Field** là khóa - định danh (identifier) cho một thông tin
- **Value** là dữ liệu tương ứng với khóa đó

```
{  
  "name": "Ba",    ← field: value  
  "age": 29,       ← field: value  
  "job": "dev"     ← field: value  
}
```



## Non-Relational DBMS (NoSQL)

<i>Các khái niệm SQL tương ứng trong MongoDB</i>	
SQL	MongoDB
Database	Database
Table	Collection
Row	Document
Column	Field
Primary Key	ObjectId
Join	Embedded Document



# SQL vs NoSQL

Sự khác biệt chính giữa **SQL** và **NoSQL**:

- Với SQL, cần phải định nghĩa dữ liệu (lược đồ - schema) trước khi thêm mới, còn NoSQL thì không
- SQL hỗ trợ ACID (tính toàn vẹn dữ liệu) như một tiêu chuẩn
- NoSQL cho hiệu suất cao hơn, để đạt được hiệu suất cao với SQL đòi hỏi thiết kế phải thực sự tốt



# SQL vs NoSQL

Sử dụng SQL khi:

- Dữ liệu có cấu trúc rõ ràng
- Ưu tiên tính toàn vẹn dữ liệu
- Cần sử dụng các phép nối và truy vấn phức tạp



# SQL vs NoSQL

Sử dụng NoSQL khi:

- Dữ liệu không có cấu trúc rõ ràng
- Ưu tiên hiệu suất truy vấn hơn là tính chính xác của dữ liệu
- Cần khả năng mở rộng, hoặc thay đổi cấu trúc linh hoạt



# Thực hành





## Thực hành

- Thực hành với SQLite3, MySQL: [SQLite Online](#)
- Thực hành với Redis: [Try Redis](#)
- Thực hành với MongoDB: Sử dụng Python, kết nối đến database và thao tác đơn giản



Fork code: [Database](#)