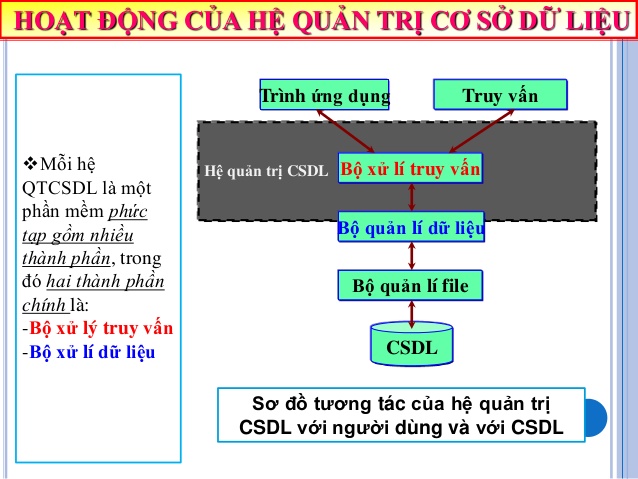
**DAY 3: TỔNG QUAN VỀ SQL**

1. **Hệ quản trị cơ sở dữ liệu:**

+ Định nghĩa:**Hệ quản trị cơ sở dữ liệu** (*Database Management System)*có thể hiểu là hệ thống được thiết kế để quản lí một khối lượng dữ liệu nhất định một cách tự động và có trật tự. Các hành động quản lý này bao gồm chỉnh sửa, xóa, lưu thông tin và tìm kiếm (truy xuất thông tin) trong một nhóm dữ liệu nhất định.

+ Hoạt động của hệ quản trị cơ sở dữ liệu:



+ Hệ quản trị cơ sở dữ liệu là hệ thống tự động giúp người dùng có thể kiểm soát các thông tin, tạo, cập nhật và duy trì các CSDL. Trong đó, hai thành phần chính trong một hệ quản trị cơ sở dữ liệu là: Bộ xử lí truy vấn (bộ xử lí yêu cầu) và bộ quản lí dữ liệu.

+ Các chức năng của hệ quản trị cơ sở dữ liệu :

***-* Cung cấp môi trường tạo lập cơ sở dữ liệu:**Hệ quản trị CSDL đóng vai trò cung cấp cho người dùng một *ngôn ngữ định nghĩa dữ liệu* để mô tả, khai báo kiểu dữ liệu, các cấu trúc dữ liệu.

***-* Cung cấp cách cập nhật và khai thác dữ liệu:** Hệ quản trị CSDL cung cấp cho người dùng *ngôn ngữ thao tác dữ liệu* để diễn tả các yêu cầu, các thao tác cập nhật và khai thác cơ sở dữ liệu. Thao tác dữ liệu bao gồm: Cập nhật (nhập, sửa, xóa dữ liệu), Khai thác (tìm kiếm, kết xuất dữ liệu).

***-* Cung cấp các công cụ kiểm soát, điều khiển các truy cập vào cơ sở dữ liệu** nhằm đảm bảo thực hiện một số yêu cầu cơ bản của hệ cơ sở dữ liệu. Bao gồm:

(1) Đảm bảo an ninh, phát hiện và ngăn chặn các truy cập bất hợp pháp.

(2) Duy trì tính nhất quán của dữ liệu.

(3) Tổ chức và điều khiển các truy cập.

(4) Khôi phục cơ sở dữ liệu khi có sự cố về phần cứng hay phần mềm.

(5) Quản lí các mô tả dữ liệu.

+ Các hệ quản trị cơ sở dữ liệu phổ biến :

* **Mysql:** MySQL là [hệ quản trị cơ sở dữ liệu](https://vi.wikipedia.org/wiki/H%E1%BB%87_qu%E1%BA%A3n_tr%E1%BB%8B_c%C6%A1_s%E1%BB%9F_d%E1%BB%AF_li%E1%BB%87u) sử dụng mã nguồn mở phổ biến nhất thế giới và được các chueyen gia lập trình web rất ưa chuộng trong quá trình phát triển web, phát triển ứng dụng. Với tốc độ và tính bảo mật cao, MySQL rất thích hợp cho các ứng dụng có truy cập CSDL trên internet.
* **Oracle:** [Oracle](https://www.oracle.com/) xuất phát từ tên của một hãng phần mềm và cũng là một hệ quản trị cơ sở dữ liệu phổ biến trên thế giới. Khởi đầu với phần mềm quản trị cơ sở dữ liệu cách đây hơn 50 năm, hiện tại ngoài sản phẩm Oracle Database Server, Oracle còn cung cấp nhiều sản phẩm khác phục vụ doanh nghiệp.
* **SQlite:** [SQLite](https://www.sqlite.org/) là hệ thống cơ sở dữ liệu quan hệ nhỏ gọn, hoàn chỉnh, có thể cài đặt bên trong các trình ứng dụng khác. SQLite được viết dưới bằng ngôn ngữ lập trình [C](https://vi.wikipedia.org/wiki/C).

1. **SQL**

**+ Định nghĩa:** SQL viết tắt của Structured Query Language, là ngôn ngữ truy vấn có cấu trúc, cho phép bạn truy cập và thao tác với các cơ sở dữ liệu để tạo, xóa, sửa đổi, trích xuất dữ liệu.

**+ Tác dụng của sql**:

- Tạo cơ sở dữ liệu mới

- Tạo bảng mới trong cơ sở dữ liệu

- Tạo veiw(khung nhìn) mới

- Thực hiện truy vấn trên cơ sở dữ liệu

- Giúp mô tả dữ liệu

- Tạo, chèn, xóa, sửa đổi bản gi trong cơ sở dữ liệu

- Trích xuất dữ liệu từ cơ sở dữ liệu

- Thiết lập quyền trên bảng, thủ tục và view

- Nhúng các ngôn ngữ khác sử dụng mô-đun SQL, thư viện, trình biên dịch sẵn.

+ **Các lệnh SQL**

**DDL - Data Definition Language (Ngôn ngữ định nghĩa dữ liệu)**

* CREATE: Tạo bảng mới, view của bảng và các đối tượng khác trong cơ sở dữ liệu.
* ALTER: Chỉnh sửa các đối tượng dữ liệu đã có, như bảng.
* DROP: Xóa toàn bộ bảng, view của bảng hoặc các đối tượng khác trong cơ sở dữ liệu.

**DML - Data Manipulation Language (Ngôn ngữ để thao tác với dữ liệu)**

* SELECT: Trích xuất bản ghi cụ thể từ một hoặc nhiều bảng
* INSERT: Chèn dữ liệu mới vào cơ sở dữ liệu.
* UPDATE: Sửa đổi, cập nhật dữ liệu trong cơ sở dữ liệu.
* DELETE: Xóa dữ liệu từ cơ sở dữ liệu.

**DCL - Data Control Language (Ngôn ngữ kiểm soát dữ liệu)**

* GRANT: Cấp đặc quyền cho user
* REVOKE: Lấy lại quyền đã cấp cho user

1. **Phân biệt SQL & NoSQL**

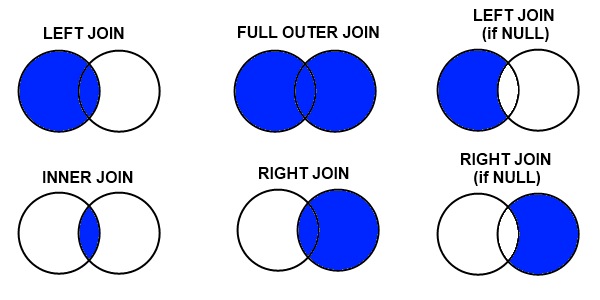
+ **Định nghĩa NoSQL:** là một DMS không quan hệ, không yêu cầu một lược đồ cố định, tránh các phép nối, và dễ dàng mở rộng. Cơ sở dữ liệu NoSQL được sử dụng cho các kho dữ liệu phân tán với nhu cầu lưu trữ dữ liệu khổng lồ. NoSQL được sử dụng cho dữ liệu lớn và các ứng dụng web thời gian thực. Ví dụ như các công ty như Twitter, Facebook, Google thu thập hàng terabyte dữ liệu người dùng mỗi ngày.

**+**BMS truyền thống sử dụng cú pháp SQL để lưu trữ và truy xuất dữ liệu để có thêm thông tin chi tiết. Thay vào đó, một hệ thống cơ sở dữ liệu NoSQL bao gồm một loạt các công nghệ cơ sở dữ liệu có thể lưu trữ dữ liệu có cấu trúc, bán cấu trúc, không có cấu trúc và đa hình.

**+ So sánh: SQL và NoSQL**

| **Tham số** | **SQL** | **NoSQL** |
| --- | --- | --- |
| Định nghĩa | Cơ sở dữ liệu SQL chủ yếu được gọi là RDBMS hoặc Cơ sở dữ liệu quan hệ | Cơ sở dữ liệu NoSQL chủ yếu được  gọi là cơ sở dữ liệu không liên quan  hoặc phân tán |
| Design for | RDBMS truyền thống sử dụng cú pháp và truy vấn SQL để phân tích và lấy dữ liệu để có thêm thông tin chi tiết. Chúng được sử dụng cho các hệ thống OLAP. | Hệ thống cơ sở dữ liệu NoSQL bao  gồm nhiều loại công nghệ cơ sở dữ  liệu khác nhau. Các cơ sở dữ liệu  này được phát triển để đáp ứng nhu cầu trình  bày cho sự phát triển của ứng dụng hiện  đại. |
| Ngôn ngữ Query | Structured query language (SQL) | Không có ngôn ngữ query |
| Type | SQL databases là cơ sở dữ liệu dựa trên bảng | NoSQL databases có thể dựa trên tài liệu,  c  ặp khóa-giá trị, cơ sở dữ liệu biểu đồ |
| Schema | SQL databases có lược đồ được xác định trước | NoSQL databases sử dụng lược đồ động cho  dữ liệu phi cấu trúc. |
| Khả năng mở rộng | SQL databases có thể mở rộng theo chiều dọc | NoSQL databases có thể mở rộng theo chiều  ngang |
| Ví dụ | Oracle, Postgres, and MS-SQL. | MongoDB, Redis, , Neo4j, Cassandra, Hbase. |
| Phù hợp cho | Đây là 1 lựa chọn lý tưởng cho môi trường truy vấn phức tạp | Không phù hợp với truy vấn phức tạp |
| Lưu trữ dữ liệu phân cấp | SQL databases không thích hợp cho việc lưu trữ dữ liệu phân cấp. | Phù hợp hơn cho kho lưu trữ dữ liệu phân cấp  vì nó hỗ trợ phương thức cặp khóa-giá trị. |
| Variations | Một loại có biến thể nhỏ | Nhiều loại khác nhau bao gồm các kho  khóa-giá trị, cơ sở dữ liệu tài liệu và cơ sở  dữ liệu đồ thị. |
| Năm phát triển | Nó được phát triển vào những năm 1970 để giải quyết các vấn đề với lưu trữ tệp phẳng | Được phát triển vào cuối những năm 2000  để khắc phục các vấn đề và hạn chế của SQL  databases. |
| Open-source | Một sự kết hợp của mã nguồn mở như Postgres & MySQL, và thương mại như Oracle Database. | Open-source |
| Tính nhất quán | Nó phải được cấu hình cho sự nhất quán chặt chẽ. | Nó phụ thuộc vào DBMS như một số cung cấp  tính nhất quán mạnh mẽ như MongoDB, trong  khi những người khác cung cấp chỉ cung cấp  sự nhất quán cuối cùng, như Cassandra. |
| Được sử dụng tốt nhất cho | RDBMS database là tùy chọn thích hợp để giải quyết các vấn đề về ACID. | NoSQL được sử dụng tốt nhất để giải quyết  các vấn đề về tính khả dụng của dữ liệu |
| Tầm quan trọng | Nó nên được sử dụng khi hiệu lực dữ liệu là siêu quan trọng | Sử dụng khi nó quan trọng hơn để có dữ liệu  nhanh hơn dữ liệu chính xác |
| Lựa chọn tốt nhất | Khi bạn cần hỗ trợ truy vấn động | Sử dụng khi bạn cần mở rộng quy mô dựa trên  yêu cầu thay đổi |
| Hardware | Specialized DB hardware (Oracle Exadata, etc.) | Commodity hardware |
| Network | Highly available network (Infiniband, Fabric Path, etc.) | Commodity network (Ethernet, etc.) |
| Loại lưu trữ | Highly Available Storage (SAN, RAID, etc.) | Commodity drives storage (standard HDDs,  JBOD) |
| Tính năng tốt nhất | Hỗ trợ đa nền tảng, Bảo mật và miễn phí | Dễ sử dụng, hiệu suất cao và công cụ linh hoạt. |
| Mô hình ACID và BASE | ACID (Atomicity, nhất quán, cách ly và độ bền) là một chuẩn cho RDBMS | Cơ bản (Về cơ bản có sẵn, trạng thái mềm,  phù hợp cuối cùng) là một mô hình của nhiều  hệ thống NoSQL |
| Performance | SQL hoạt động tốt và nhanh thì việc desgin tốt là cực kì quan trọng và ngược lại. | Nhanh hơn SQL NoSQL thì denormalized cho  phép bạn lấy được tất cả thông tin về một item  cụ thể với các codition mà không cần JOIN  liên quan hoặc truy vấn SQL phức tạp. |
| Kết luận | Dự án đã có yêu cầu dữ liệu rõ ràng xác định quan hệ logic có thể được xác định trước. | Phù hợp với những dự án yêu cầu dữ liệu  không liên quan, khó xác định, đơn giản mềm  dẻo khi đang phát triển |

1. **Phân biệt** JOIN, LEFT JOIN, RIGHT JOIN, FULL JOIN



1. **Kiểu dữ liệu phù hợp với từng đối tượng**

* **INT:** Một số nguyên có kích thước bình thường có thể được sử dụng trong phạm vi số âm và số dương. Nếu được sử dụng khoảng âm, phạm vi cho phép là từ -2147483648 đến 2147483647. Nếu không phạm vi cho phép là từ 0 đến 4294967295. Bạn có thể chỉ định chiều rộng tối đa là 11 chữ số.
* **TINYINT**: Một số nguyên rất nhỏ có thể được sử dụng trong phạm vi số âm và số dương. Nếu bao gồm số âm, phạm vi cho phép là từ -128 đến 127. Nếu không, phạm vi cho phép là từ 0 đến 255. Bạn có thể chỉ định chiều rộng tối đa 4 chữ số.
* **SMALLINT:** Một số nguyên nhỏ có thể sử dụng trong phạm vi số âm và số dương. Nếu bao gồm số âm, phạm vi cho phép là từ -32768 đến 32767. Nếu không phạm vi cho phép là từ 0 đến 65535. Bạn có thể chỉ định chiều rộng tối đa 5 chữ số.
* **MEDIUMINT:** Một số nguyên có kích thước trung bình có thể được sử dụng trong phạm vi số âm và số dương. Nếu bao gồm số âm, phạm vi cho phép là từ -8388608 đến 8388607. Nếu không, phạm vi cho phép là từ 0 đến 16777215. Bạn có thể chỉ định chiều rộng tối đa 9 chữ số.
* **BIGINT:** Một số nguyên lớn có thể sử dụng trong phạm vi số âm và số dương. Nếu bao gồm số âm, phạm vi cho phép là từ -9223372036854775808 đến 9223372036854775807. Nếu không, phạm vi cho phép là từ 0 đến 18446744073709551615. Bạn có thể chỉ định chiều rộng tối đa 20 chữ số.
* **FLOAT (M, D):** Một số dấu phẩy động được sử dụng để ngăn cách phần nguyên và phần hữu tỷ. Độ chính xác của số thập phân có thể lên tới 24 vị trí cho một FLOAT.
* **DOUBLE (M, D):**Một số dấu chấm động được sử dụng để ngăn cách phần nguyên và phần hữu tỷ. Bạn có thể xác định độ dài hiển thị (M) và số thập phân (D). Độ chính xác thập phân có thể tới 53 vị trí cho một DOUBLE. REAL là một từ đồng nghĩa với DOUBLE.
* **DECIMAL (M, D):** Một số dấu phẩy động được sử dụng để ngăn cách phần nguyên và phần hữu tỷ. Trong các số thập phân được giải nén, mỗi số thập phân tương ứng với một byte. Xác định độ dài hiển thị (M) và số thập phân (D) là bắt buộc. **NUMERIC** là một từ đồng nghĩa với **DECIMAL**.
* **DATE**: Ngày theo định dạng YYYY-MM-DD, trong khoảng từ 1000-01-01 đến 9999-12-31. Ví dụ, ngày 30 tháng 12 năm 2018 sẽ được lưu trữ như 2018-12-30.
* **DATETIME**: Kết hợp ngày và giờ theo định dạng HH: MM: SS HHYYY-MM-DD, trong khoảng từ 1000-01-01 00:00:00 và 9999-12-31 23:59:59. Ví dụ: 5:30 chiều ngày 30 tháng 12 năm 2018 sẽ được lưu trữ là 2018-12-30 17:30:00.
* **TIMESTAMP:** Mốc thời gian lúc nửa đêm. Kiểu này giống như định dạng **DATETIME** trước đó chỉ không có dấu gạch nối giữa các số; ví dụ: 3:30 chiều ngày 30 tháng 12 năm 2018 sẽ được lưu giữ là 20181230153000 (YYYYMMDDHHMMSS).
* **TIME:** Lưu trữ thời gian theo định dạng HH: MM: SS.
* **YEAR (M):** Lưu trữ một năm ở định dạng 2 chữ số hoặc 4 chữ số. Nếu độ dài được xác định là 2 (ví dụ NĂM (2)), NĂM có thể nằm trong khoảng từ 1970 đến 2069 (70 đến 69). Nếu độ dài được chỉ định là 4, thì YEAR có thể là 1901 đến 2155. Độ dài mặc định là 4.
* **CHAR (M):**Một chuỗi có độ dài cố định trong khoảng từ 1 đến 255 ký tự (ví dụ CHAR (5)), được ghi thêm bên phải bằng dấu cách với độ dài được chỉ định khi lưu trữ. Việc xác định độ dài là không bắt buộc, nhưng mặc định là 1.
* **VARCHAR (M):** Một chuỗi có độ dài từ 1 đến 255 ký tự. Ví dụ, VARCHAR (25). Bạn phải xác định độ dài khi tạo trường VARCHAR.
* **BLOB hoặc TEXT:** Trường có độ dài tối đa 65535 ký tự. BLOB là “Đối tượng lớn nhị phân” và được sử dụng để lưu trữ lượng lớn dữ liệu nhị phân, chẳng hạn như hình ảnh hoặc các loại tệp khác. Các trường được định nghĩa là TEXT cũng chứa một lượng lớn dữ liệu. Sự khác biệt giữa hai loại và trên là dữ liệu được lưu trữ phân biệt chữ hoa chữ thường trên các BLOB và không phân biệt chữ hoa chữ thường trong các trường TEXT. Bạn không chỉ định độ dài cho BLOB hoặc TEXT.
* **TINYBLOB hoặc TINYTEXT:** Cột BLOB hoặc TEXT có độ dài tối đa 255 ký tự. Bạn không chỉ định độ dài cho TINYBLOB hoặc TINYTEXT.
* **MEDIUMBLOB hoặc MEDIUMTEXT:** Cột BLOB hoặc TEXT có độ dài tối đa 16777215 ký tự. Bạn không được chỉ định độ dài cho MEDIUMBLOB hoặc MEDIUMTEXT.
* **LONGBLOB hoặc LONGTEXT:** Cột BLOB hoặc TEXT có chiều dài tối đa là 4294967295 ký tự. Bạn không được chỉ định độ dài cho LONGBLOB hoặc LONGTEXT.
* **ENUM**: dùng để chỉ định một hằng số. Khi xác định một ENUM, bạn tạo một danh sách các mục mà từ đó giá trị phải được chọn (hoặc nó có thể là NULL).