Mô hình dữ liệu quan hệ:

* Cấu trúc:
  + Dữ liệu được thể hiện trong các bảng
  + Mỗi bảng gồm các hàng, các cột thể hiện thông tin về một chủ thể.
  + Các cột biểu thị các thuộc tính của chủ thể và tên cột thường là tên thuộc tính
  + Mỗi hàng biểu thị cho một cá thể, gồm một bộ các giá trị tương ứng với các cột
* Thao tác dữ liệu:
  + Có thể cập nhật dữ liệu như thêm, sửa, xóa bản ghi trong một bảng
  + Các kết quả tìm kiếm thông tin qua truy vấn dữ liệu có được nhờ thực hiện các thao tác trên dữ liệu
* Ràng buộc dữ liệu: Dữ liệu trong các bảng phải thỏa mãn một số rang buộc. Chẳng hạn không được có 2 bộ nào trong 1 bảng giống nhau.

Cơ sở dữ liệu được xây dựng dựa trên mô hình dữ liệu quan hệ được gọi là cơ sở dữ liệu quan hệ

Hệ quản trị cơ sở dữ liệu dùng để tạo, cập nhật, khai thác CSDL quan hệ được gọi là hệ quan trị cơ sở dữ liệu quan hệ.

Các tính chất của một CSDL quan hệ:

* Mỗi quan hệ có tên để phân biệt với các quan hệ khác
* Các bộ là duy nhất và không phân biệt thứ tự
* Mỗi thuộc tính có tên để phân biệt và không phân biệt thứ tự
* Quan hệ không có thuộc tính đa trị hay phức hợp(
  + Thuộc tính đa trị: 1 thuộc tính tương ứng trong nhiều bộ giá trị
  + Phức hợp một thuộc tính có 2 giá trị

)

MS SQL Server là một hệ quản trị cơ sở dữ liệu quan hệ(RDBMS) phát triển bởi Microsoft. Sản phẩm này được xây dựng với các chức năng cơ bản là lưu trữ, truy vấn dữ liệu theo yêu cầu của các ứng dụng khác. Nó có thể chạy hoặc trên cùng một máy hoặc trên máy khác thông qua mạng.

Schema là một namespace dùng để gom nhóm các table có chung đặc điểm nào đó để dễ dàng quản lý. Mặc định của schema là dbo .Trong 1 database thì tên của schema là duy nhất.

**CÁC TỪ KHÓA CƠ BẢN TRONG SQL**

**-SELECT:** Khi xử lý câu lệnh SELECT, SQL Server xử lý từ lệnh FROM trước và sau đó đến SELECT mặc dù SELECT ở đầu câu query. Để lấy tất cả các cột của 1 bảng, có thể gọi tất cả các cột trong câu SELECT, cũng có thể gõ SELECT \* là cách viết tắt để lấy tất cả các cột của bảng. SELECT \* hữu ích khi kiểm trả các cột và dữ liệu của một bảng mới. Nó hữu ích cho các truy vấn đặc biệt

Tuy nhiên không nên sử dụng SELECT \* khi code project thật vì các lý do sau:

+ Đầu tiên SELECT \* thường lấy nhiều dữ liệu hơn là ứng dụng mình viết cần. Nó khiến dữ liệu không cần thiết chuyển từ SQL Server về ứng dụng client sẽ tốn rất nhiều thời gian khi dữ liệu truyền qua mạng và làm ứng dụng chậm đi

+Thứ 2 nếu bảng được them một hoặc nhiều cột mới. SELECT \* sẽ lấy tất cả các cột bao gồm cả các cột mới – những cột mà không có ý định dùng trong ứng dụng hiện tại. Điều này có thể làm ứng dụng bị hỏng.

Để lọc các hàng dựa vào một hay nhiều điều kiện ta có thể sử dụng từ khóa WHERE.

Thứ tự thực hiện các mệnh đề trong câu query như sau: FROM>WHERE>SELECT.

Để sắp xếp kết quả dựa vào một hay nhiều cột thì dùng từ khóa ORDER BY.

Thứ tự thực hiện các mệnh đề trong câu query như sau: FROM>WHERE>SELECT>ORDER BY.

Để gom nhóm các cột ta sử dụng từ khóa GROUP BY.

Thứ tự thực hiện các mệnh đề trong câu query như sau: FROM>WHERE>GROUP BY>SELECT>ORDER BY

-**OFFSER FETCH:** Để giới hạn số lượng dòng trả về từ một query.

Cú pháp: ORDER BY column\_list [ASC |DESC]

OFFSET offset\_row\_count {ROW | ROWS}

FETCH {FIRST | NEXT} fetch\_row\_count {ROW | ROWS} ONLY

Từ khóa OFFSET quy định số dòng bỏ qua trước khi bắt đầu trả về các hang từ câu query. “offset\_row\_count” có thể là hằng số, biến số hoặc tham số sao cho nó lớn hơn hoặc bằng 0

Từ khóa FETCH chỉ định số hàng trả về sau khi mệnh đề OFFSET được thực thi. “offset\_row\_count” có thể là hằng số, biến số hoặc “scalar” sao cho nó lơn hơn hoặc bằng 1

Mệnh đề OFFSET bắt buộc trong khi FETCH thì có thể có hoặc không. Ngoài ra FIRST và NEXT và các từ tương ứng có nghĩa gần giống nhau vì thế có sử dụng chúng thay thế cho nhau.

OFFSET và FETCH phải dùng với mênh đề ORDER BY nếu không thì sẽ xảy ra lỗi.

**-NULL:** thường được sử dụng khi có bất kỳ một giá trị dữ liệu nào đó khuyết thiếu. Ví dụ tại 1 thời điểm ghi thông tin khách hàng, dữ liệu email có thể unknown, vì thế bản ghi sẽ được ghi là NULL trong db.

Thường thường kết quả của biểu thức logic là TRUE, FALSE. Tuy nhiên khi NULL tham gia vào đánh giá logic, kết quả là UNKNOWN. Đó là 3 giá trị logic: TRUE, FALSE và UNKNOWN.

NULL băng với nothing, ngay cả NULL cũng không bằng NULL bởi vì, mỗi NULL có thể khác nhau về IS Null

SELECT

    customer\_id,

    first\_name,

    last\_name,

    phone

FROM

    sales.customers

WHERE

    phone = NULL

ORDER BY

    first\_name,

    last\_name;

Câu lệnh trên tìm kiếm khách hàng, những bản ghi mà không có số điện thoại trong bảng customers

Câu truy vấn trả về một bộ kết quả rỗng. Biểu thức phone=NULL trả về giá trị là UNKNOWN. Vì vậy kết quả nhận được là 1 bộ rỗng.

Để kiểm tra giá trị của 1 trường có là NULL hay không , ta thường sử dụng “IS NULL” để kiểm tra. Ngược lại dùng “IS NOT NULL để kiểm tra”

**JOIN** Trong cơ sở dữ liệu quan hệ, data được phân bố ở nhiều bảng. Để lấy được một bộ dữ liệu có ý nghĩa hoàn chỉnh, ta cần truy vấn dữ liệu từ các bảng trên sử dụng “join”. SQL server hỗ trợ rất nhiều loại join bao gồm: inner join,left join, right join, full outer joim, cross join. Mỗi loại join chỉ ra cách SQL Server sử dụng dữ liệu từ một bảng như thế nào để chọn các row từ bảng khác.

