**Tổng hợp câu lệnh SQL**

1. **Ngôn ngữ truy vấn dữ liệu**

**SELECT:**

* Dùng để lấy dữ liệu từ cơ sở dữ liệu dựa trên các điều kiện xác định.

**CÚ PHÁP:**

SELECT cột 1, cột 2,.v.v..

FROM tên bảng

**WHERE**

* Dùng để lọc, trích xuất bản ghi

**CÚ PHÁP:**

SELECT cột 1, cột 2,.v.v..

FROM tên bảng

WHERE điều kiện

* Chú ý:

+ Các trường văn bản cần có đấu ngoặc kép

+ Với key word có đấu nên thêm n’… ’ (contry = n’Việt Nam’)

1. **Ngôn ngữ thao tác dữ liệu**
   1. INSERT:

* Dùng để thêm 1 or nhiều hàng dữ liệu mới vào bảng.

**CÚ PHÁP:**

INSERT INTO tên bảng(cột 1, cột 2,.v.v..)

VALUES giá trị 1, giá trị 2, v.v..

Thêm tất cả giá tri mà không cần gọi tên cột

INSERT INTO tên bảng

VALUES giá trị 1, giá trị 2, v.v..

* 1. **UPDATE:**
* Dùng để cập nhật dữ liệu hiện có trong bẳng dựa trên các điều kiện lọc.

**CÚ PHÁP:**

UPDATE tên bảng

SET cột 1 = giá trị 1, cột 2 = giá trị 2, v.v..

WHERE điều kiện

WHERE ở đây chỉ định cột được sửa đổi còn không có WHERE thì sẽ tất cả sẽ được sửa

* 1. **DELETE**
* Dùng để xóa 1 or nhiều dữ liệu đựa trên điều kiện lọc.

**CÚ PHÁP:**

DELETE cột 1, cột 2,.v.v..

FROM tên bảng

WHERE điều kiện

WHERE chỉ định cột xóa còn không có WHERE thì sẽ xóa hết

1. **Các loại toán tử**

|  |  |
| --- | --- |
| Toán tử | Phép toán |
| + | Phép cộng |
| – | Phép trừ |
| \* | Phép nhân |
| / | Phép chia |
| % | Phép chia lấy dư |

|  |  |
| --- | --- |
| Toán tử | **Phép toán** |
| = | Bằng |
| > | Lớn hơn |
| < | Nhỏ hơn |
| >= | Lớn hơn hoặc bằng |
| <= | Nhỏ hơn hoặc bằng |
| <> hoặc != | Khác nhau |
| !> | Không lớn hơn |
| !< | Không nhỏ hơn |

1. **Toán tử logic**

AND = Và

OR = Hoặc

* 1. **SELECT TOP**
* Dùng để chỉ số lượng bản ghi
* Nó không hỗ trợ trên tất cả CSDL

**CÚ PHÁP:**

SELECT TOP(Số) tên cột

FROM tên bảng

WHERE điều kiện

* 1. **LIKE và NOT LIKE**
* Dùng để tìm kiếm 1 giá trị khớp với 1 mẫu ký tự cụ thể
* Thường dùng với ký tự đại diện như % or \_

**CÚ PHÁP:**

SELECT tên cột

FROM tên bảng

WHERE tên cột LIKE…

* Chú ý:

+ NOT LIKE là tìm giá trị ngoài giá trị đã nhập

+ LIKE: ‘a%’ bắt đầu = a

‘%a’ kết thúc = a

‘%a%’ bắt đầu và kết thúc bất cứ đâu miễn là = a

‘\_a%’ a sẽ là ký tự thứ 2

‘[abc]%’ không băt đầu bằng abc

‘%[abc]’ không kết thúc bằng abc

* 1. **IN**
* Dùng để kiểm tra 1 giá trị có nằm trong danh sách được chỉ định hay không.

**CÚ PHÁP:**

SELECT tên cột

FROM tên bảng

WHERE tên cột IN (giá trị 1, giá trị 2, .v.v..)

* 1. **BETWEEN**
* Dùng để kiểm tra 1 giá trị có nằm trong khoảng giá trị (giá trị bắt đầu và giá trị kết thúc) hay không.

**CÚ PHÁP:**

SELECT tên cột

FROM tên bảng

WHERE tên cột BETWEEN giá trị 1 AND giá trị 2

* 1. **NOT**
* Đảo ngược kết quả của điều kiện.

**CÚ PHÁP:**

SELECT tên cột

FROM tên bảng

WHERE NOT tên cột

* 1. **NULL**
* Là Trường không có giá trị
* Kiểm tra giá trị NULL

**CÚ PHÁP:**

SELECT tên cột

FROM tên bảng

WHERE tên cột IS NULL

* Chú ý: Luôn sử dụng IS NULL để tìm giá trị NULL
  1. **IS NOT NULL**
* Kiểm tra giá trị không không trống
* Tìm kiếm giá trị không NULL

**CÚ PHÁP:**

SELECT tên cột

FROM tên bảng

WHERE tên cột IS NOT NULL

* 1. **CASE WHEN**
* Nó giống kiểu IF ELSE.

**CÚ PHÁP:**

CASE

WHEN điều kiện 1 THEN kết quả 1

WHEN điều kiện 2 THEN kết quả 2

v.v..

ELESE kết quả cuối

END

* Chú ý: WHEN còn kèm được cả IN, LIKE, BETWEEN

WHEN số IN (‘1’, ‘2’, ‘3’) THEN Kết quả

* 1. **LOWER**
* Chuyển chữ thường thành chữ hoa

**CÚ PHÁP:**

SELECT LOWER (tên cột)

FROM tên bảng

WHERE tên cột

* 1. **UPPER**
* Chuyển chữ hoa thành chữ thường

**CÚ PHÁP:**

SELECT UPPER (tên cột)

FROM tên bảng

WHERE tên cột

* 1. **DISTINCT**
* Dùng để lọc các giá trị duy nhất từ 1 cột trong bảng

**CÚ PHÁP:**

SELECT UPPER (tên cột)

FROM tên bảng

WHERE tên cột

* 1. **CONCAT**
* Dùng để nối chuỗi

**CÚ PHÁP:**

SELECT CONCAT(FirstName, ' ', LastName) AS FullName

FROM Employees

* Kết quả trả ra là “Tiến Nguyễn”

1. **Nhóm hàm tổng hợp**
   1. **SUM**

* Dùng để tính tổng của 1 tập giá trị.

CÚ PHÁP:

SELECT SUM (tên cột)

FROM tên bảng

* 1. **MIN**
* Trả về giá trị nhỏ nhất trong 1 tập giá trị.

CÚ PHÁP:

SELECT MIN(tên cột)

FROM tên bảng

* 1. **MAX**
* Trả về giá trị lớn nhất trong 1 tập giá trị.

CÚ PHÁP:

SELECT MAX(tên cột)

FROM tên bảng

* 1. **COUNT**
* Đếm số lượng bản ghi or giá trị không NULL

CÚ PHÁP:

SELECT COUNT(tên cột)

FROM tên bảng

* 1. **AVG**
* Tính giá trị trung bình của 1 tập giá trị.

CÚ PHÁP:

SELECT AVG(tên cột)

FROM tên bảng

1. **Các nhóm hàm khác**
   1. **ORDER BY**

* Sắp xếp theo thứ tụ tăng dần or giản dần
* Tăng dần: ASC or để trống 🡺 Hệ thống sẽ tự đông hiểu
* Giảm dần: DESC

CÚ PHÁP:

SELECT tên cột 1, tên cột 2, .v.v..

FROM tên bảng

WHERE or không cần cũng được tùy theo tính chất đề bài

ORDER BY cột 1, cột 2 ASC/DESC

* Chú ý:

+ Ngoài ra ORDER BY có thể kết hợp với các toán tử tổng hợp

+ VD: ORDER BY (COUNT, SUM, MAX, MIN, AVD) ASC/DESC

* 1. **UNION**
* Sử dụng để kết hợp kết quả của 2 hay nhiều câu lệch SELECT
* Mọi câu lệnh SELECT trong UNION phải có cùng số cột
* Các cột phải có cùng kiểu dữ liệu và trả ra cùng 1 thứ tự
* Các giá trị bị trùng lặp sẽ được hợp nhất thành 1

CÚ PHÁP:

SELECT tên cột 1, tên cột 2, .v.v..

FROM tên bảng

WHERE

UNION

SELECT tên cột 1, tên cột 2, .v.v..

FROM tên bảng

WHERE

,v.v…

* 1. **UNION ALL**
* Lấy tất cả các giá trị kể cả bị trùng lặp

CÚ PHÁP:

SELECT tên cột 1, tên cột 2, .v.v..

FROM tên bảng

WHERE

UNION ALL

SELECT tên cột 1, tên cột 2, .v.v..

FROM tên bảng

WHERE

,v.v…

* 1. **GROUP BY**
* Dùng để nhóm các hàng có cùng giá trị.
* Thường dùng cho các toán tử tổng hợp như: COUNT, SUM, MAX, MIN, AVG
* Đã có toán tử tổng hợp thì phải có GROUP BY
* Trừ trường hợp sau SELECT COUNT(\*) FROM 🡺 thì không cần GROUP BY

CÚ PHÁP:

SELECT tên cột 1, tên cột 2, .v.v..

FROM tên bảng

WHERE

GROUP BY toán tử + tên tên cột 1, tên cột 2, .v.v..

* 1. HAVING
* Đây là điều kiện của GROUP BY
* HAVING còn kẹp được cả điều kiện tính toán như các toán tử tổng hợp >< còn WHERE thì không

1. **Câu hỏi**
   1. **GROUP BY và HAVING**

* GROUP BY là câu lệnh để nhóm các bản ghi, giá trị, cột và tính toán các trị đó
* HAVING để áp dụng điều kiện lọc đã được nhóm = GROUP BY
  1. **DISTINCT và GROUP BY**

|  |  |  |
| --- | --- | --- |
| Tiêu chí | DISTINCT | GROUP BY |
| Chức năng | Loại bỏ các bản ghi trùng lặp trong kết quả truy vấn | Dùng để nhóm các bản ghi theo 1 or nhiều cột và thường đi kèm mới các hàm tổng hợp (COUNT, SUM,.v.v..) |
| Cách dùng | Thường đi kèm với các cột trong câu lệnh SELECT | Thường dùng khi mình thực hiện các phép toán tổng hợp trên từng nhóm |

* 1. **WHERE và HAVING**

|  |  |  |
| --- | --- | --- |
| Tiêu chí | WHERE | HAVING |
| Vị trí | Đứng trước GROUP BY | Đứng sau GROUP BY |
| Dùng được | Dùng được với các câu lệnh INSERT, UPDATE, DELETE,… | Chỉ dùng được với câu lệnh SECLECT |
| Kết quả trả ra | Trả ra kết quả là từng dòng | Trả ra kêt quả theo từng nhóm |
| Áp dụng | Áp dụng cho các dòng của bản ghi | Áp dung cho các nhóm của barg sau khi đã được tổng hợp bằng GROUP BY |

* 1. **UNION và UNION ALL**

|  |  |  |
| --- | --- | --- |
| Tiêu chí | UNION | UNION ALL |
| Chức năng | Kết hợp 2 or nhiều câu lệch truy vấn SECLECT và loại bỏ các bản ghi trùng lặp | Kết hợp 2 or nhiều câu lệch truy vấn SECLECT và không loại bỏ các bản ghi trùng lặp |
| Hiệu suất | Thường sẽ chậm hơn vì nó cần kiểm tra và loại bỏ các bản ghi trùng lặp | Nhanh hơn vì không cần phải loại bỏ các bản ghi trùng lặp |

* 1. **Thứ tự của SQL**
* SElECT > FROM > WHERE > GROUP BY > HAVING > ORDER BY
  1. **Khi join thì điều điều quan trọng nhất**
* Dữ liệu mà mình muốn lấy là gì từ đó xác định được kiểu JOIN
  1. **JOIN là gì và có bao nhiêu loại JOIN**
* Là 1 phép toán được sử dụng để kết hợp các bảng dựa trên 1 điều kiện liên kết chung giữa chúng.
* Có 4 loại JOIN phổ biến:

+ INNER JOIN

+ LEFT JOIN

+ RIGHT JOIN

+ FULL JOIN

* 1. **Tính tuổi**

GET DATE() là lấy ngày giờ hiện tại