SQL for Developer

Giảng viên: Bùi Quang Đăng

Contents

Stanford


MySQL®

TRUY VẤN, HÀM TRONG MYSQL

Stanford
nâng tầm tri thức

Truy vấn dữ liệu

SELECT

- Mệnh đề SELECT cho phép chỉ ra các thuộc tính mà ta muốn tìm. Thứ tự các thuộc tính trong kết quả là thứ tự mà nó xuất hiện trong lệnh SELECT. Bằng cách đó cho phép ta thực hiện được phép chiếu của quan hệ.
- Như vậy, kết quả của câu lệnh SELECT là một bảng, bảng đó là kết quả của phép chiếu qua bảng xuất phát.

Stanford
nâng tầm tri thức

Truy vấn dữ liệu

- **SELECT**
  - SELECT có thể thực hiện trên 1 bảng hoặc trên nhiều bảng.
  - SELECT có nhiều mệnh đề, mỗi mệnh đề đảm bảo một chức năng.

```
SELECT [DISTINCT]|Columns_list|Expression_list|*
FROM <Tables_list>
WHERE <Conditions>
GROUP BY <Columns>
HAVING <Conditions_for_group>
ORDER BY [ACS| DESC]
```

Stanford
nâng tầm tri thức

Truy vấn dữ liệu

- **SELECT**

Trong đó:

- Sau SELECT: Các thông tin cần đưa ra, đó chính là danh sách các thuộc tính
- Sau FROM: Danh sách các tên bảng, từ đó thông tin được lấy ra.
- **WHERE**: Các biểu thức logic, chỉ ra thông tin được lấy ra từ hàm nào hoặc điều kiện nối giữa các bảng.
- **GROUP BY**: Các cột mà trong đó được tính theo từng nhóm.

Stanford
nâng tầm tri thức

Truy vấn dữ liệu

- **SELECT**

Trong đó:

- **HAVING**: Biểu thức logic chỉ ra thông tin được lấy ra từ nhóm nào.
- **ORDER BY**: Chỉ ra các cột mà trong đó thông tin được sắp xếp theo thứ tự.
  - **ASC**: thông tin được sắp xếp theo chiều tăng dần (ASCendent)
  - **DESC**: thông tin được sắp xếp theo chiều giảm dần(DESCendent)

Stanford
nâng tầm tri thức

❖ Truy vấn dữ liệu

Các toán tử của SQL

- **[NOT] BETWEEN x AND y:** [Không] nằm giữa giá trị X và Y
- **IN (danh sách):** thuộc bất kỳ giá trị nào trong danh sách
- **x [NOT] LIKE y**: Đúng nếu x [không] giống khung mẫu y.
  Các ký tự dùng trong khuôn mẫu:
  Dấu gạch dưới (_) : Chỉ một ký tự bất kỳ
  Dấu phần trăm (%) : Chỉ một nhóm ký tự bất kỳ
- **IS [NOT] NULL:** kiểm tra giá trị rỗng
- **EXISTS**: Trả về TRUE nếu có tồn tại.

Truy vấn dữ liệu

Sử dụng các hàm

Hàm số học

- **ROUND(n[,m])**: Cho giá trị làm tròn của n (đến cấp m)
- **CEILING(n)**: Cho số nguyen nhỏ nhất lớn hơn hoặc bằng n
- **FLOOR(n)**: Cho số nguyen lớn nhất bằng hoặc nhỏ hơn n
- **TRUNCATE(number,n)**: Làm cắt n số sau dấu phẩy
- **POWER(m,n)**: Cho lũy thừa bậc n của m
- **SQRT(n)**: Cho căn bậc 2 của n, n>=0

Stanford
nâng tầm tri thức

Truy vấn dữ liệu

Sử dụng các hàm

Các hàm ký tự

- **LOWER(char)**: Chuyển chuỗi dữ liệu về dạng chữ thường
- **UPPER(char)**: Chuyển chuỗi dữ liệu về dạng chữ hoa
- **LTRIM(char)**: Bỏ các ký tự trống bên trái
- **RTRIM(char)**: Loại bỏ ký tự trống bên phải
- **TRIM(char)**: Loại bỏ ký tự trống đầu và cuối chuỗi
- **CONCAT(str1, str2,…strn)**: Lối các chuỗi với nhau

Stanford
nâng tầm tri thức

Truy vấn dữ liệu

Sử dụng các hàm

Các hàm ký tự

- **REPLACE(char, search_string, replace_string):** Thay thế tất cả các chuỗi **search_string** có trong chuỗi **char** bằng chuỗi **replace_string**.
- **SUBSTRING(char, m [, n]):** Cho chuỗi con của chuỗi char lấy từ vị trị m về phải n ký tự, nếu không chỉ n thì lấy cho đến cuối chuỗi. Nếu m là số âm thì lấy từ bên phải chuỗi với n kí tự.
- **LENGTH(char):** Cho chiều dài của chuỗi char.

Stanford
nâng tầm tri thức

Truy vấn dữ liệu

Sử dụng các hàm

Các hàm trong MySQL

- **INSTR(string, char):** Trả về vị trí của chuỗi char trong string
- **LPAD(string, length, lpad_string) :** Chèn bao nhiêu kí tự lpad vào bên trái chuỗi cho đủ độ dài chuỗi mong muốn.
- **RPAD(string, length, rpad_string) :** Chèn bao nhiêu kí tự rpad vào bên phải chuỗi cho đủ độ dài chuỗi mong muốn.
- **REVERSE(string) :** Hàm đảo ngược chuỗi.

Stanford
nâng tầm tri thức

Truy vấn dữ liệu

Sử dụng các hàm

Các hàm ngày

- **NOW(), CURDATE():** Lấy ngày giờ; ngày tháng hiện thời
- **DATE(exp):** Đưa exp về dạng ngày tháng
- **DATE_ADD(date, INTERVAL value addunit)**: Là hàm sử dụng thêm thông tin vào date với adduinit là: SECOND, MINUTE, HOUR, DAY, WEEK, MONTH, YEAR,…
  - » Ví dụ: DATE_ADD('2020/01/10', INTERVAL 10 DAY) => Thông tin thay đổi 2020/01/20
- **DATEDIFF(startdate, enddate):** So sánh 2 ngày startdate và enddate.
  - » Ví dụ: DATEDIFF('2020/01/10','2020/01/25'); // Trả về số ngày

Stanford
Nâng tầm tri thức

Truy vấn dữ liệu

Sử dụng các hàm

Các hàm ngày

- **SYSDATE()**: Hàm trả về ngày giờ hiện tại
- **EXTRACT(part FROM date)**: Hàm trả về các thành phần (part) của ngày như ngày, tháng, năm,…
- **DAY(date), MONTH(date), YEAR(date)**: Lấy thông tin ngày, tháng, năm của date

Stanford

Truy vấn dữ liệu

Sử dụng các hàm

Các hàm chuyển đổi kiểu

- **CONVERT(type, value):** Chuyển đổi kiểu dữ liệu về dạng tương ứng từ giá trị value.
- **CAST(value as type):** Chuyển đổi kiểu dữ liệu về dạng tương ứng.

Stanford

GROUP, HAVING

Stanford
nâng tầm tri thức

Truy vấn dữ liệu

Sử dụng các hàm

Hàm nhóm

- COUNT(): Đếm số lần xuất hiện của thuộc tính.
- SUM(colume): Tính tổng các giá trị của thuộc tính (thuộc loại số học)
- AVG(colume): Tính giá trị trung bình các giá trị của thuộc tính (thuộc loại số học)
- MAX(colume): Tìm giá trị cực đại của thuộc tính
- MIN(colume): Tìm giá trị cực tiểu của thuộc tính.

Truy vấn dữ liệu

Mệnh đề GROUP BY

Mệnh đề GROUP BY <các cột> cho phép đưa ra thông tin theo từng nhóm.

Ví dụ:

```
SELECT CongViec, AVG(Luong) AS LuongTB
FROM NHANVIEN
GROUP BY CongViec
```

```
SELECT CongViec, AVG(Luong) AS LuongTB
FROM NHANVIEN
WHERE Luong>200
GROUP BY CongViec
```

Stanford
nâng tầm tri thức

Truy vấn dữ liệu

Mệnh đề HAVING

Muốn đưa ra các nhóm trên cơ sở thông tin nhóm thì điều kiện phải được viết trong mệnh đề HAVING

Ví dụ:

```
SELECT CongViec, Avg(Luong) AS TBLuong
FROM NHANVIEN
GROUP BY  CongViec
HAVING (Avg(Luong)>300)
```

LẤY THÔNG TIN NHIỀU BẢNG, SUBQUERY

Stanford
nâng tầm tri thức

Truy vấn dữ liệu

- **Lấy thông tin từ nhiều bảng**
  - **Nối bằng (Equi-Join)**

  Điều kiện nối là một đẳng thức

Ví dụ:

```
SELECT HoTen, CongViec, TenDV
FROM NHANVIEN, DONVI
WHERE NHANVIEN.MaDV= DONVI.MaDV
```

Truy vấn dữ liệu

Lấy thông tin từ nhiều bảng

- **Thực hiện kết nối thông qua từ khóa Join**

Có thể thực hiện lấy dữ liệu từ hai bảng thông qua từ khóa **JOIN**

- **Cú pháp**

```
SELECT  field1, field2, field3
FROM  table1
INNER JOIN table2
ON  table1.keyfield=table2.foreign_keyfield;
```

❖ Truy vấn dữ liệu

▪ Lấy thông tin từ nhiều bảng

- **INNER JOIN (nối trong)**

Ví dụ:

```
SELECT KHACHHANG.TenKH, DONHANG.TenSP
FROM KHACHHANG
INNER JOIN DONHANG
ON KHACHHANG.MaKH=DONHANG.MaKH
```

Trong đó:

- **INNER JOIN** trả về tất cả các dòng từ hai bảng thỏa mãn điều kiện. Nếu những dòng dữ liệu có bên table1 mà không có trong table2 thì sẽ không được hiển thị.

Truy vấn dữ liệu

Lấy thông tin từ nhiều bảng

- **LEFT JOIN**

Ví dụ:

```
SELECT KHACHHANG.TenKH, DONHANG.TenSP
FROM KHACHHANG
LEFT JOIN DONHANG
ON KHACHHANG.MaKH=DONHANG.MaKH
```

Trong đó:

- **LEFT JOIN** trả về tất cả các dòng có ở bảng thứ nhất, mặc dù ở bảng thứ hai không thỏa mãn phép toán. Nếu dữ liệu có ở bảng thứ nhất mà không có ở bảng thứ hai thì dữ liệu vẫn hiển thị.

Stanford nâng tầm tri thức

Truy vấn dữ liệu

Lấy thông tin từ nhiều bảng

- **RIGHT JOIN**

Ví dụ:

```
SELECT KHACHHANG.TenKH, DONHANG.TenSP
FROM KHACHHANG
RIGHT JOIN DONHANG
ON KHACHHANG.MaKH=DONHANG.MaKH
```

Trong đó:

- **RIGHT JOIN** trả về tất cả các dòng có ở bảng 2, mặc dù bảng 1 không thỏa mãn phép toán. Nếu dữ liệu có ở bảng 2 mà không có ở bảng 1 thì vẫn được hiển thị.

Truy vấn dữ liệu

Thực hiện các phép toán trên tập hợp

Phép UNION

Sử dụng để ghép 2 kết quả của câu lệnh truy vấn lại với nhau

Ví dụ:

```
SELECT CongViec
FROM NHANVIEN
WHERE MaDV='0001'
UNION
        SELECT CongViec
        FROM NHANVIEN
        WHERE MaDV='0002'
```

Truy vấn dữ liệu

Các câu hỏi lồng nhau

- Là các lệnh SELECT trong đó có chứa các lệnh SELECT khác.
- Các câu lệnh SELECT bên trong nằm sau mệnh đề WHERE hoặc HAVING của SELECT bên ngoài.

Ví dụ:

```
SELECT Hoten, TenDV, Congviec, Luong
FROM NHANVIEN AS NV, DONVI AS DV
WHERE (NV.MaDV= DV.MaDV)
                AND (Luong> ( SELECT AVG(Luong)
                              FROM NHANVIEN ))
```

Stanford
nâng tầm tri thức

Truy vấn dữ liệu

Sử dụng Phép toán IN

- Biểu thức: <Giá trị> **IN** {Tập hợp} trả lại kết quả = TRUE nếu tập hợp các giá trị nằm trong tập hợp đứng sau IN.

Ví dụ:

```
SELECT NHANVIEN.MaNV, NHANVIEN.Hoten, NHANVIEN.Luong
FROM NHANVIEN
WHERE NHANVIEN.Luong IN (
      SELECT  Min(NHANVIEN.Luong) AS MinOfLuong
      FROM NHANVIEN
      GROUP BY NHANVIEN.MaDV)
```

Truy vấn dữ liệu

Sử dụng Limit trong MySQL

```
SELECT select_list
FROM table_name
LIMIT [offset,] row_count;
```

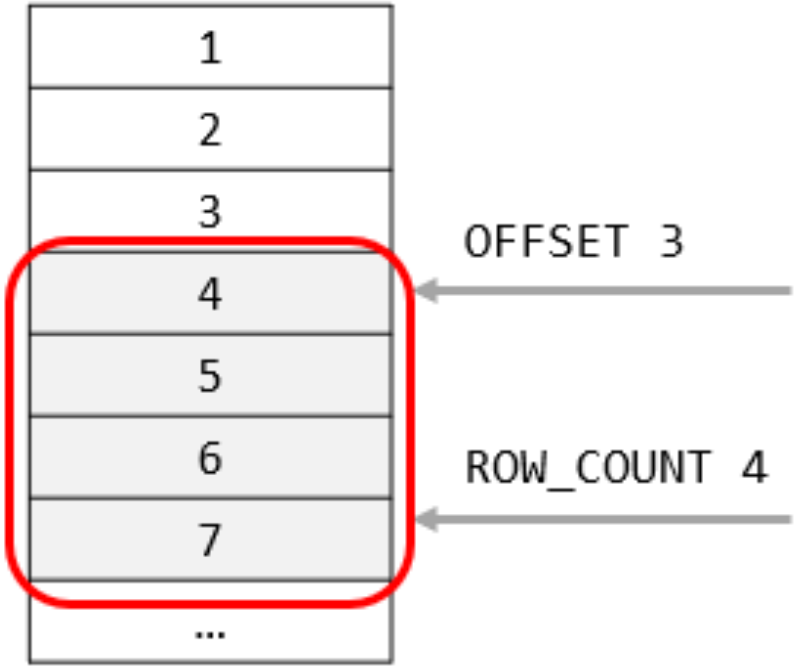
- **offset**: là vị trí bắt đầu lấy thông tin
- **row_count**: Số thông tin cần lấy
- Nếu không có offset thì sẽ bắt đầu lấy từ vị trí thứ 0

Stanford
nâng tầm tri thức

Truy vấn dữ liệu

- **Sử dụng Limit trong MySQL**

```
SELECT n FROM t
ORDER BY n
LIMIT 3, 4;
```



Truy vấn dữ liệu

Sử dụng Limit trong MySQL

- Trình tự thực hiện câu lệnh:


```
SELECT select_list
FROM table_name
ORDER BY order_expression
LIMIT offset, row_count;
```

Truy vấn dữ liệu

- **Sử dụng Limit trong MySQL**
  - **Ví dụ:**

```
1 Select * from NhanVien limit 2,3
```

MaNV	HoTen	DienThoai	Email	DiaChi
SF003	Trần Tuấn Anh	0988233128	tuananh@gmail.com	Hà Nội
SF004	Trần Minh Hoàng	0988235689	minhhoang@gmail.com	Hà Nội
SF005	Vũ Thị Hoa	0987335689	hoavt@gmail.com	Hà Nam

Truy vấn dữ liệu

- **Update Join:** Sử dụng để nối thông tin giữa các bảng để lấy thông tin phục vụ cập nhật giá trị:

```
UPDATE T1,T2
INNER JOIN
T2 ON T1.C1 = T2.C1
SET T1.C2 = T2.C2, T2.C3 = expr
WHERE condition
```

Truy vấn dữ liệu

- **Update Join:** Sử dụng để nối thông tin giữa các bảng để lấy thông tin phục vụ cập nhật giá trị:
  - Ví dụ:

```
1 Update NhanVien nv JOIN HeSoLuong hs
2 ON nv.HeSo = hs.Id set nv.Luong =
  3500000*hs.HeSo + hs.PhuCap;
```

MaNV	HoTen	DienThoai	Email	Luong
SF001	Lê Minh Hà	0988123568	minhha@gmail.com	7000000
SF002	Lê Mạnh Hùng	0988233568	hunglm@gmail.com	10500000
SF003	Trần Tuấn Anh	0988233128	tuananh@gmail.com	14200000
SF004	Trần Minh Hoàng	0988235689	minhhoang@gmail.com	17800000
SF005	Vũ Thị Hoa	0987335689	hoavt@gmail.com	21500000
SF006	Vũ Thị Lan	0987335125	lanvt@gmail.com	7000000

Truy vấn dữ liệu

- **Temporary table:** Sử dụng bảng tạm để chứa thông tin trong một phiên làm việc với MySQL:
  - Cú pháp:

```
CREATE TEMPORARY TABLE table_name
(
  column_1_definition,
  column_2_definition, ...,
  table_constraints
);
```

Exercises

Stanford
nâng tầm tri thức


Stanford
nâng tầm tri thức
Thank You !
www.stanford.com.vn