**Lý thuyết**

Hàm RANK gán thứ hạng cho mỗi hàng trong một phân vùng của tập kết quả.

Đối với mỗi phân vùng, thứ hạng của hàng đầu tiên là 1. Hàm RANK thêm số lượng hàng bị ràng buộc vào thứ hạng bị ràng buộc để tính thứ hạng của hàng tiếp theo, do đó các cấp bậc có thể không tuần tự. Ngoài ra, các hàng có cùng giá trị sẽ có cùng thứ hạng.

Sau đây minh họa cú pháp của hàm RANK:

RANK() OVER (  
    [PARTITION BY partition\_expression, ... ]  
    ORDER BY sort\_expression [ASC | DESC], ...  
)

Trong cú pháp này:

* Đầu tiên, mệnh đề PARTITION BY phân phối các hàng của tập kết quả thành các phân vùng mà hàm RANK được áp dụng.
* Sau đó, mệnh đề ORDER BY chỉ định thứ tự các hàng trong mỗi phân vùng mà hàm được áp dụng

Hàm RANK có thể hữu ích để tạo báo cáo theo dạng top-N và bottom-N.

Với dữ liệu của bảng ranks

| **c** |
| --- |
| A |
| A |
| B |
| B |
| B |
| C |
| E |

SELECT c, RANK () OVER(ORDER BY c) rank\_number   
FROM ranks;

Đầu ra như sau

| **c** | **rank\_number** |
| --- | --- |
| A | 1 |
| A | 1 |
| B | 3 |
| B | 3 |
| B | 3 |
| C | 6 |
| E | 7 |

Như bạn có thể thấy rõ từ đầu ra:

* Hàng thứ nhất và thứ hai nhận được cùng thứ hạng vì chúng có cùng giá trị A.
* Hàng thứ ba, thứ tư và thứ năm nhận được thứ hạng 3 vì hàm RANK bỏ qua thứ hạng 2 và tất cả chúng đều có cùng giá trị B.