

NAME: SHARATH CHANDRA

BATCH : DATA ENGINEERING

TOPIC :RANKING FUNCTIONS.

RANKING FUNCTIONS:

The functions are used to generate sequence numbers for each row for providing a rank based on the specified condition. The functions are provided to rank the records in the table. The following functions are provided for the rank records.

i)row_number()

ii) rank ()

iii) dense_rank ()

iv) ntile ()

→ performing RANK FUNCTIONS:

Firstly I have created a table named employees1 and inserted some data into it.

```
mysql> CREATE TABLE employees1 (  
-> id INT PRIMARY KEY,  
-> name VARCHAR(255),  
-> salary DECIMAL(10, 2)  
-> );  
Query OK, 0 rows affected (0.02 sec)  
  
mysql>  
mysql> -- Insert data into Employees table  
mysql> INSERT INTO employees1 VALUES  
-> (1, 'John Doe', 60000.00),  
-> (2, 'Jane Smith', 75000.00),  
-> (3, 'Bob Johnson', 60000.00),  
-> (4, 'Alice Williams', 80000.00),  
-> (5, 'Eva Brown', 75000.00);  
Query OK, 5 rows affected (0.01 sec)  
Records: 5 Duplicates: 0 Warnings: 0  
  
mysql> select *from employees1;  
+----+-----+-----+  
| id | name       | salary |  
+----+-----+-----+  
| 1  | John Doe   | 60000.00 |  
| 2  | Jane Smith | 75000.00 |  
| 3  | Bob Johnson | 60000.00 |  
| 4  | Alice Williams | 80000.00 |  
| 5  | Eva Brown  | 75000.00 |  
+----+-----+-----+  
5 rows in set (0.00 sec)  
  
mysql> |
```

—>then i have performed the ranking functions ,

i)row_number()

```
mysql> /*Ranking Functions*/
mysql> /* row()_number-giving consecutive numbers to rank*/
mysql> select id,name,salary,ROW_NUMBER() over(order by salary desc) as rownumber
-> from employees1;
+-----+-----+-----+-----+
| id | name       | salary | rownumber |
+-----+-----+-----+-----+
| 4 | Alice Williams | 80000.00 | 1 |
| 2 | Jane Smith   | 75000.00 | 2 |
| 5 | Eva Brown    | 75000.00 | 3 |
| 1 | John Doe     | 60000.00 | 4 |
| 3 | Bob Johnson  | 60000.00 | 5 |
+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

ii) **rank ()**: its showing error in my sql mam ,so i have done it in online compiler

```
1 CREATE TABLE employees1(
2     id INT PRIMARY KEY,
3     name VARCHAR(255),
4     salary DECIMAL(10, 2)
5 );
6
7 -- Insert data into Employees table
8 INSERT INTO employees1 VALUES
9 (1, 'John Doe', 60000.00),
10 (2, 'Jane Smith', 75000.00),
11 (3, 'Bob Johnson', 60000.00),
12 (4, 'Alice Williams', 80000.00),
13 (5, 'Eva Brown', 75000.00);
14 /*rank()-used to give rank if duplicates allowed ranking will be changed
15 SELECT id,name,salary,RANK()OVER(ORDER BY salary DESC) AS RANK
16 FROM employees1;
17
```

id	name	salary	rank
4	Alice Williams	80000	1
2	Jane Smith	75000	2
5	Eva Brown	75000	2
1	John Doe	60000	4
3	Bob Johnson	60000	4

iii)dense_rank ():

```
1 CREATE TABLE employees1(  
2     id INT PRIMARY KEY,  
3     name VARCHAR(255),  
4     salary DECIMAL(10, 2)  
5 );  
6  
7 -- Insert data into Employees table  
8 INSERT INTO employees1 VALUES  
9 (1, 'John Doe', 60000.00),  
10 (2, 'Jane Smith', 75000.00),  
11 (3, 'Bob Johnson', 60000.00),  
12 (4, 'Alice Williams', 80000.00),  
13 (5, 'Eva Brown', 75000.00);  
14 /*dense_rank()-used to give ranks consecutively even if duplicates are al  
15 SELECT id,name,salary,DENSE_RANK() OVER(ORDER BY salary DESC) AS RANK  
16 FROM employees1;  
17
```

id	name	salary	rank
4	Alice Williams	80000	1
2	Jane Smith	75000	2
5	Eva Brown	75000	2
1	John Doe	60000	3
3	Bob Johnson	60000	3

iv)ntile ():

Without condition:

```
1 CREATE TABLE employees1(  
2     id INT PRIMARY KEY,  
3     name VARCHAR(255),  
4     salary DECIMAL(10, 2)  
5 );  
6  
7 -- Insert data into Employees table  
8 INSERT INTO employees1 VALUES  
9 (1, 'John Doe', 60000.00),  
10 (2, 'Jane Smith', 75000.00),  
11 (3, 'Bob Johnson', 60000.00),  
12 (4, 'Alice Williams', 80000.00),  
13 (5, 'Eva Brown', 75000.00);  
14 /*ntile() function- it will divide give the rank in groups*/  
15 SELECT id,name,salary,NTILE(2) OVER(ORDER BY salary) AS RANK  
16 FROM employees1;/*without condition*/  
17  
18
```

id	name	salary	rank
1	John Doe	60000	1
3	Bob Johnson	60000	1
2	Jane Smith	75000	1
5	Eva Brown	75000	2
4	Alice Williams	80000	2

With condition:

```
1 CREATE TABLE employees1(  
2     id INT PRIMARY KEY,  
3     name VARCHAR(255),  
4     salary DECIMAL(10, 2)  
5 );  
6  
7 -- Insert data into Employees table  
8 INSERT INTO employees1 VALUES  
9 (1, 'John Doe', 60000.00),  
10 (2, 'Jane Smith', 75000.00),  
11 (3, 'Bob Johnson', 60000.00),  
12 (4, 'Alice Williams', 80000.00),  
13 (5, 'Eva Brown', 75000.00);  
14 SELECT name,salary,NTILE(3) OVER(ORDER BY salary) AS RANK  
15 FROM employees1 WHERE salary>40000;/*with condition*/  
16
```

name	salary	rank
John Doe	60000	1
Bob Johnson	60000	1
Jane Smith	75000	2
Eva Brown	75000	2
Alice Williams	80000	3