Akilesh K

k.akilesh123@gmail.com

Data engineering - Batch 1

Date: 23-01-24

SQL-DATA CLEANING, RANK FUNCTIONS, STORED PROCEDURE

```
93 •    select *from studentdata1;
```

id	name	age	grade
NULL	stella	20	A+
1	appu	20	A+
5	bob	21	C
6	sunny	21	NULL
7	NULL	21	C

Data transformation

```
95 •    select name,count(name) as Actual_count from studentdata1
96      group by name
97      having count(name)>1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

name	Actual_count

```
106
107    select * from studentdata1
108    where name is null;
109
```

Result Grid | Filter Rows:

	id	name	age	grade
▶	7	NULL	21	C

●
```
delete from studentdata1
where name is null;
```

Data cleaning

```
112
113 ●   select *from studentdata1 where id is null;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Co

	id	name	age	grade
▶	NULL	stella	20	A+

```
update studentdata1 set id=7
where id is null;
```

```
21 •    select *from studentdata1;

22

23

24
```

esult Grid | Filter Rows: _____ | Export:

id	name	age	grade
7	stella	20	A+
1	appu	20	A+
5	bob	21	C
6	sunny	21	NULL

```
121 •    select *from studentdata1;
```

Result Grid | Filter Rows: _____ | Exp

id	name	age	grade
7	STELLA	20	A+
1	APPU	20	A+
5	BOB	21	C
6	SUNNY	21	NULL

Rank functions

```
152
153
154 •    select * from rank_demo;
```

	first_name	last_name	city
▶	Luisa	Evans	Texas
	Paul	Ward	Alaska
	Peter	Bennett	California
	Carlos	Patterson	New York
	Rose	Huges	Florida
	Marielia	Simmons	Texas
	Antonio	Butler	New York
	Diego	Cox	California

RANK() Functions

```
157 •    SELECT first_name, last_name, city,
158      RANK () OVER (ORDER BY city) AS Rank_No
159      FROM rank_demo;
160
161
```

	first_name	last_name	city	Rank_No
▶	Paul	Ward	Alaska	1
	Peter	Bennett	California	2
	Diego	Cox	California	2
	Rose	Huges	Florida	4
	Carlos	Patterson	New York	5
	Antonio	Butler	New York	5
	Luisa	Evans	Texas	7
	Marielia	Simmons	Texas	7

RANK FUNCTION USING PARTITION

```sql
162  SELECT first_name, last_name, city,
163  RANK () OVER (PARTITION BY city ORDER BY first_name) AS Rank_No
164  FROM rank_demo;
165
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: IA

first_name	last_name	city	Rank_No
Paul	Ward	Alaska	1
Diego	Cox	California	1
Peter	Bennett	California	2
Rose	Huges	Florida	1
Antonio	Butler	New York	1
Carlos	Patterson	New York	2
Luisa	Evans	Texas	1
Marielia	Simmons	Texas	2

ROW_NUMBER() Function

```sql
161  SELECT first_name, last_name, city,
162  ROW_NUMBER() OVER(ORDER BY city) AS my_rank
163  FROM rank_demo;
164
```

Result Grid | Filter Rows: | Export: | Wrap C

first_name	last_name	city	my_rank
Paul	Ward	Alaska	1
Peter	Bennett	California	2
Diego	Cox	California	3
Rose	Huges	Florida	4
Carlos	Patterson	New York	5
Antonio	Butler	New York	6
Luisa	Evans	Texas	7
Marielia	Simmons	Texas	8

ORDER IT BY DESC

```
162 •    SELECT first_name, last_name, city,
163      ROW_NUMBER() OVER(ORDER BY city DESC) AS my_rank
164      FROM rank_demo;
165
```

Result Grid | Filter Rows: | Export: | Wrap Cell Conter

first_name	last_name	city	my_rank
Luisa	Evans	Texas	1
Marielia	Simmons	Texas	2
Carlos	Patterson	New York	3
Antonio	Butler	New York	4
Rose	Huges	Florida	5
Peter	Bennett	California	6
Diego	Cox	California	7
Paul	Ward	Alaska	8

DENSE_RANK() Function

```
165
166 •    SELECT first_name, last_name, city,
167      DENSE_RANK() OVER(ORDER BY city) AS my_rank
168      FROM rank_demo;
169
```

Result Grid | Filter Rows: | Export: | Wrap Cell Cont

first_name	last_name	city	my_rank
Paul	Ward	Alaska	1
Peter	Bennett	California	2
Diego	Cox	California	2
Rose	Huges	Florida	3
Carlos	Patterson	New York	4
Antonio	Butler	New York	4
Luisa	Evans	Texas	5
Marielia	Simmons	Texas	5

NTILE(N) Function

```
171 •    SELECT first_name, last_name, city,
172      NTILE(3) OVER(ORDER BY city) AS my_rank
173      FROM rank_demo;
174
```

Result Grid | Filter Rows: | Export: | Wrap Cell

first_name	last_name	city	my_rank
Paul	Ward	Alaska	1
Peter	Bennett	California	1
Diego	Cox	California	1
Rose	Huges	Florida	2
Carlos	Patterson	New York	2
Antonio	Butler	New York	2
Luisa	Evans	Texas	3
Marielia	Simmons	Texas	3

STORED PROCEDURE

```
mysql> CREATE TABLE student (
    ->      id INT PRIMARY KEY,
    ->      name VARCHAR(50),
    ->      age INT,
    ->      salary INT
    -> );
Query OK, 0 rows affected (0.25 sec)
```

```
mysql> INSERT INTO student (id, name, age, salary) VALUES
    -> (1, 'John', 27, 20000),
    -> (2, 'Haris', 29, 28000),
    -> (3, 'Peter', 17, 25000),
    -> (4, 'Marco', 29, 28000),
    -> (5, 'Diego', 17, 35000),
    -> (6, 'Antonio', 22, 48000),
    -> (7, 'Steffan', 16, 35000);
Query OK, 7 rows affected (0.04 sec)
Records: 7  Duplicates: 0  Warnings: 0
```

CREATING THE PROCEDURE

```
mysql> DELIMITER //
mysql>
mysql> CREATE PROCEDURE studentList()
    -> BEGIN
    ->     SELECT name, age, salary
    ->     FROM STUDENT
    ->     ORDER BY salary;
    -> END //
Query OK, 0 rows affected (0.02 sec)

mysql>
mysql> DELIMITER ;
mysql>
```

CALLING THE PROCEDURE

```
mysql> CALL studentList;
+---------+------+--------+
| name    | age  | salary |
+---------+------+--------+
| John    |   27 |  20000 |
| Peter   |   17 |  25000 |
| Haris   |   29 |  28000 |
| Marco   |   29 |  28000 |
| Diego   |   17 |  35000 |
| Steffan |   16 |  35000 |
| Antonio |   22 |  48000 |
+---------+------+--------+
7 rows in set (0.03 sec)

Query OK, 0 rows affected (0.04 sec)
```

NEW TABLE

```
mysql> select * from studentdata1;
+------+--------+------+-------+
| id   | name   | age  | grade |
+------+--------+------+-------+
|    7 | STELLA |   20 | A+    |
|    1 | APPU   |   20 | A+    |
|    5 | BOB    |   21 | C     |
|    6 | SUNNY  |   21 | NULL  |
+------+--------+------+-------+
4 rows in set (0.01 sec)
```

CREATING PROCEDURE

```
mysql> DELIMITER //
mysql>
mysql> CREATE PROCEDURE dataofstudent()
    -> BEGIN
    ->     SELECT * FROM studentdata1;
    -> END //
Query OK, 0 rows affected (0.02 sec)

mysql>
mysql> DELIMITER ;
```

CALL PROCEDURE

```
mysql> CALL dataofstudent;
+------+--------+------+-------+
| id   | name   | age  | grade |
+------+--------+------+-------+
|    7 | STELLA |   20 | A+    |
|    1 | APPU   |   20 | A+    |
|    5 | BOB    |   21 | C     |
|    6 | SUNNY  |   21 | NULL  |
+------+--------+------+-------+
4 rows in set (0.00 sec)

Query OK, 0 rows affected (0.03 sec)
```