

# Tutoriat 11

## VIEW; ROLLUP

### VIEW

**VIEW**-urile sunt tabele ce au o durata de viata mica. Acestea sunt folosite pentru a memora temporar anumite date. Sintaxa:

```
1. --creeare
2. CREATE VIEW nume_view AS (SELECT...);
3.
4. --stergere
5. DROP VIEW nume_view;
```

Asa cum se poate vedea **VIEW**-ul are sintaxa aproape identica cu ce-a de creare/stergere a unui tabel, diferenta fiind ca un **VIEW** poate fi creat numai cu date rezultate dintr-o cerere, nu putem sa ii cream o structura in care definim tipul si numele coloanelor ca la **CREATE TABLE**.

Sa exemplificam acest lucru creand un **VIEW** ce contine toate salariile managerilor:

```
1. CREATE VIEW salarii_manageri
2. AS (SELECT salary
3.     FROM employees
4.     WHERE employee_id IN ( SELECT DISTINCT manager_id
5.                           FROM employees));
```

### ROLLUP

**ROLLUP** este o functie folosita in clauza **GROUP BY**, ea primeste coloane si intoarce reuniunea cererilor cand clauza **GROUP BY** contine: primele N coloane, primele N-1 coloane,..., primele 2 coloane, prima coloana si cand **GROUP BY** nu exista. Sintaxa:

```
1. SELECT ...
2. FROM ...
3. GROUP BY ROLLUP(col1, col2, ..., colN);
```

Ce este echivalent cu:

```
1. SELECT ... FROM ...
2. GROUP BY col1, col2, ..., colN
3.
4. UNION
5.
6. SELECT ... FROM ...
7. GROUP BY col1, col2, ..., colN-1
8.
9. UNION
10. ...
11. UNION
12.
13. SELECT ... FROM ...
14. GROUP BY col1, col2
15.
16. UNION
17.
18. SELECT ... FROM ...
19. GROUP BY col1
20.
21. UNION
22.
23. SELECT ... FROM ...;
```

Ca exemplu sa afisam toate salariile atat singure cat si cu toate persoanele ce le au:

```
1. SELECT salary, last_name
2. FROM employees
3. GROUP BY ROLLUP(salary, last_name);
```