

Baze de date-Anul 1

Laborator 8 SQL

Clauza WITH. Analiza top-n.

I. [Clauza WITH]

- Cu ajutorul clauzei *WITH* se poate defini un bloc de cerere înainte ca acesta să fie utilizat într-o interogare.
- Clauza permite reutilizarea aceluiași bloc de cerere într-o instrucțiune *SELECT* complexă. Acest lucru este util atunci când o cerere face referință de mai multe ori la același bloc de cerere, care conține operații *join* și funcții de agregare.

Exerciții:

1. Utilizând clauza *WITH*, să se scrie o cerere care afișează numele departamentelor și valoarea totală a salariilor din cadrul acestora. Se vor considera departamentele a căror valoare totală a salariilor este mai mare decât media valorilor totale ale salariilor tuturor angajaților.

```
WITH val_dep AS (SELECT department_name, SUM(salary) AS total
                  FROM departments d, employees e
                  WHERE d.department_id = e.department_id
                  GROUP BY department_name),
val_medie AS (SELECT SUM(total)/COUNT(*) AS medie
               FROM val_dep)
SELECT *
FROM val_dep
WHERE total > (SELECT medie
               FROM val_medie)
ORDER BY department_name;
```

2. Utilizând clauza *WITH*, să se scrie o cerere care afișează numele managerilor și valoarea totală a salariilor subordonaților lor. Se vor afișa doar managerii a căror valoare totală a salariilor subordonaților este mai mică decât media valorilor totale ale salariilor tuturor angajaților din departamentul în care lucrează respectivul manager.

II. [Analiza top-n]

Pentru aflarea primelor *n* rezultate ale unei cereri, este utilă pseudocoloana *ROWNUM*. Aceasta returnează numărul de ordine al unei linii în rezultat.

Exerciții:

3. Să se determine primii 10 cei mai bine plătiți angajați.

```
SELECT * FROM (SELECT * FROM employees ORDER BY salary DESC)
WHERE ROWNUM < 11;
```

```
SELECT * FROM employees
WHERE ROWNUM < 11
ORDER BY salary DESC;
```

Ce returnează cea de-a doua cerere?

4. Să se determine cele mai prost plătite 3 job-uri, din punct de vedere al mediei salariilor.

5. Să se afișeze cele mai populate top 4 departamente(denumire) și nr de angajați din fiecare.

III. [Exerciții – utilizarea alternativă a funcției DECODE sau a structurii CASE; din nou NVL și NVL2; NULLIF]

Obs:

- *NVL(a, b)* – întoarce a, dacă a este NOT NULL, altfel întoarce b;
- *NVL2(a, b, c)* - întoarce b, dacă a este NOT NULL, altfel întoarce c;
- *COALESCE (expr_1, expr_2, ...expr_n)* – întoarce prima expresie NOT NULL din listă;
- *NULLIF(a, b)* – întoarce a, dacă a!=b; altfel întoarce NULL ;
- *DECODE (expresie, val_1, val_2, val_3, val_4, ..., val_2n-1, val_2n, default)* – dacă *expresie* = *val_1*, întoarce *val_2*; dacă *expresie* = *val_3*, întoarce *val_4*; ...; altfel întoarce *default*.
- *DECODE* este echivalent cu *CASE*, a cărei structură este:

CASE expresie

WHEN val_1 THEN val_2

WHEN val_3 THEN val_4

...

ELSE default

END

CASE poate avea și forma:

CASE

WHEN expr_logica_1 THEN val_2

WHEN expr_logica_3 THEN val_4

...

ELSE default

END

6. Să se afișeze informații despre departamente, în formatul următor: „Departamentul <department_name> este condus de {<manager_id> | nimeni} și {are numărul de salariați <n> | nu are salariați}“.

7. Să se afișeze numele, prenumele angajaților și lungimea numelui pentru înregistrările în care aceasta este diferită de lungimea prenumelui.

SELECT last_name, first_name,

NULLIF (LENGTH(last_name), LENGTH(first_name)) Rezultat

FROM employees;

8. Să se afișeze numele, data angajării, salariul și o coloană reprezentând salariul după ce se aplică o mărire, astfel: pentru salariații angajați în 1989 creșterea este de 20%, pentru cei angajați în 1990 creșterea este de 15%, iar salariul celor angajați în anul 1991 crește cu 10%. Pentru salariații angajați în alți ani valoarea nu se modifică. (CASE și DECODE)

9. Să se afișeze:

- suma salariilor, pentru job-urile care încep cu litera S;
- media generală a salariilor, pentru job-ul având salariul maxim;
- salariul minim, pentru fiecare din celelalte job-uri.

10. Scrieți o cerere pentru a afișa job-ul, salariul total pentru job-ul respectiv pe departamente si salariul total pentru job-ul respectiv pe departamentele 30, 50, 80. Se vor eticheta coloanele corespunzător. Rezultatul va apărea sub forma de mai jos:

Job	Dep30	Dep50	Dep80	Total

.....				
.....				

Metoda 1: (NVL, SUM, DECODE)

```
SELECT job_id, NVL(SUM(DECODE(department_id, 30, salary)),0) Dep30,
      NVL(SUM(DECODE(department_id, 50, salary)),0) Dep50,
      NVL(SUM(DECODE(department_id, 80, salary)),0) Dep80,
      NVL(SUM(salary),0) Total
FROM employees
GROUP BY job_id;
```

Metoda 2: (cu cereri corelate în clauza SELECT)

```
SELECT job_id, (SELECT NVL(SUM(salary),0)
                FROM employees
                WHERE department_id = 30
                AND job_id = e.job_id) Dep30,
      (SELECT NVL(SUM(salary),0)
       FROM employees
       WHERE department_id = 50
       AND job_id = e.job_id) Dep50,
      (SELECT NVL(SUM(salary),0)
       FROM employees
       WHERE department_id = 80
       AND job_id = e.job_id) Dep80,
      NVL(SUM(salary),0) Total
FROM employees e
GROUP BY job_id;
```

11. Să se creeze o cerere prin care să se afișeze numărul total de angajați și, din acest total, numărul celor care au fost angajați în 1997, 1998, 1999 si 2000. Denumiți capetele de tabel în mod corespunzător. (Ambele metode ca la exercițiul anterior.)