Paula Alexandrescu
Grupa 132

Laborator BD

Tema 2

Exercițiul 1

SELECT j.job_title as Job_AP, j.min_salary as Minim_Salariu_AP, j.max_salary as Maxim_Salariu_AP, e.last_name as Nume_AP, e.first_name as
Prenume_AP, d.department_id as ID_Departament_AP, d.department_name as Department_AP
from jobs j
left join employees e on (j.job_id=e.job_id)
left join departments d on (e.department_id=d.department_id);

Comentariu:
Se afiseaza datele cerute prin folosirea tabelelor jobs, employees si departments cu ajutorul a doua left join-uri, la care folosim egalitatile j.job_id=e.job_id, respectiv e.department_id=d.department_id.

<u>Exercițiul 2</u>

SELECT 'Departamentul '||d.department_name||' este condus de '|| NVL(TO_CHAR(d.manager_id),
'nimeni')||' si '||
CASE
WHEN (SELECT count(employee_id)
   FROM employees
   WHERE d.department_id=department_id)=0
THEN 'nu are salariati'
ELSE 'are salariati'
END as Informatii_Departamente_AP
FROM departments d;

Comentariu:
ID-ul managerului de departament este luat din tabelul departments, iar rezultatul este afisat cu ajutorul functiei NVL, pentru a hotara daca se afiseaza id-ul sau 'nimeni'. Pentru a vedea ce se afiseaza la salariati, folosim functia CASE pentru a vedea daca numarul de angajati din departament este 0, caz in care se afiseaza 'nu are salariati' sau diferit de 0 (se afiseaza 'are salariati').

Exercițiul 3

SELECT e.last_name Nume_AP, e.first_name Prenume_AP, length(e.last_name) Lungime_nume_AP, l.city Oras_AP
FROM employees e
LEFT JOIN departments d ON (e.department_id=d.department_id)
LEFT JOIN locations l ON (d.location_id=l.location_id)
WHERE length(e.last_name)!=length(e.first_name);

Comentariu:
Se afiseaza datele cerute prin folosirea tabelelor employees, departments si jobs, ultimele doua fiind parcurse cu ajutorul left join-uri, la care folosim egalitatile e.department_id=d.department_id, respectiv d.location_id=l.location_id. Adaugam cu ajutorul lui where conditia ca lungimea numelui sa fie diferita de lungimea prenumelui.

Exercițiul 4

CASE

```
SELECT e.last_name Nume_AP, e.first_name Prenume_AP, e.hire_date Data_angajare_AP, j.job_title
Job_AP, e.salary Salariu_AP,
CASE
WHEN EXTRACT(YEAR FROM e.hire_date)=TO_CHAR(1989)
THEN 1.2*e.salary
WHEN EXTRACT(YEAR FROM e.hire_date)=TO_CHAR(1990)
THEN 1.15*e.salary
WHEN EXTRACT(YEAR FROM e.hire_date)=TO_CHAR(1991)
THEN 1.1*e.salary
ELSE e.salary
END Salariu_dupa_marire_AP
FROM employees e
LEFT OUTER JOIN jobs j ON (e.job_id=j.job_id);
```

DECODE

```
SELECT e.last_name Nume_AP, e.first_name Prenume_AP, e.hire_date Data_angajare_AP,
j.job_title Job_AP, e.salary Salariu_AP,
DECODE (EXTRACT(YEAR FROM e.hire_date), TO_CHAR(1989), 1.2*e.salary, TO_CHAR(1990),
1.15*e.salary, TO_CHAR(1991), 1.1*e.salary,
e.salary) Salariu_dupa_marire_AP
FROM employees e
LEFT JOIN jobs j ON (e.job_id=j.job_id);
```

Comentariu:
Comparam anul angajarii fiecarui salariat cu anii 1989, 1990, 1991. Daca anul angajarii coincide cu unul din acesti ani, afisam in coloanal Salariu_dupa_marire_AP noul salariu. Daca nu, afisam salariul actual.

```sql
--4.  Să se afişeze numele, data angajării, titlul job-ului, salariul şi o coloană reprezentând salariul după ce se aplică o mărire,
--astfel: pentru salariaţii angajaţi în 1989 creşterea este de 20%, pentru cei angajaţi în 1990 creşterea este de 15%, iar salariul celor
--angajaţi în anul 1991 creşte cu 10%. Pentru salariaţii angajaţi în alţi ani valoarea nu se modifică.(2 rezolvăriş DECODE si CASE).

SELECT e.last_name Nume_AP, e.first_name Prenume_AP, e.hire_date Data_angajare_AP, j.job_title Job_AP, e.salary Salariu_AP,
CASE
WHEN EXTRACT(YEAR FROM e.hire_date)=TO_CHAR(1989)
THEN 1.2*e.salary
WHEN EXTRACT(YEAR FROM e.hire_date)=TO_CHAR(1990)
THEN 1.15*e.salary
WHEN EXTRACT(YEAR FROM e.hire_date)=TO_CHAR(1991)
THEN 1.1*e.salary
ELSE e.salary
END Salariu_dupa_marire_AP
FROM employees e
LEFT OUTER JOIN jobs j ON (e.job_id=j.job_id);
--107 REZULTATE
```

	NUME_AP	PRENUME_AP	DATA_ANGAJARE_AP	JOB_AP	SALARIU_AP	SALARIU_DUPA_MARIRE_AP
1	King	Steven	17-JUN-87	President	24000	24000
2	De Haan	Lex	13-JAN-93	Administration Vice President	17000	17000
3	Kochhar	Neena	21-SEP-89	Administration Vice President	17000	20400
4	Whalen	Jennifer	17-SEP-87	Administration Assistant	4400	4400
5	Greenberg	Nancy	17-AUG-94	Finance Manager	12000	12000
6	Popp	Luis	07-DEC-99	Accountant	6900	6900
7	Urman	Jose Manuel	07-MAR-98	Accountant	7800	7800
8	Sciarra	Ismael	30-SEP-97	Accountant	7700	7700
9	Chen	John	28-SEP-97	Accountant	8200	8200

```sql
--4.  Să se afişeze numele, data angajării, titlul job-ului, salariul şi o coloană reprezentând salariul după ce se aplică o mărire,
--astfel: pentru salariaţii angajaţi în 1989 creşterea este de 20%, pentru cei angajaţi în 1990 creşterea este de 15%, iar salariul celor
--angajaţi în anul 1991 creşte cu 10%. Pentru salariaţii angajaţi în alţi ani valoarea nu se modifică.(2 rezolvăriş DECODE si CASE).

SELECT e.last_name Nume_AP, e.first_name Prenume_AP, e.hire_date Data_angajare_AP, j.job_title Job_AP, e.salary Salariu_AP,
DECODE (EXTRACT(YEAR FROM e.hire_date), TO_CHAR(1989), 1.2*e.salary, TO_CHAR(1990), 1.15*e.salary, TO_CHAR(1991), 1.1*e.salary,
e.salary) Salariu_dupa_marire_AP
FROM employees e
LEFT JOIN jobs j ON (e.job_id=j.job_id);
--107 REZULTATE
```

|	NUME_AP	PRENUME_AP	DATA_ANGAJARE_AP	JOB_AP	SALARIU_AP	SALARIU_DUPA_MARIRE_AP
1	King	Steven	17-JUN-87	President	24000	24000
2	De Haan	Lex	13-JAN-93	Administration Vice President	17000	17000
3	Kochhar	Neena	21-SEP-89	Administration Vice President	17000	20400
4	Whalen	Jennifer	17-SEP-87	Administration Assistant	4400	4400
5	Greenberg	Nancy	17-AUG-94	Finance Manager	12000	12000
6	Popp	Luis	07-DEC-99	Accountant	6900	6900
7	Urman	Jose Manuel	07-MAR-98	Accountant	7800	7800
8	Sciarra	Ismael	30-SEP-97	Accountant	7700	7700
9	Chen	John	28-SEP-97	Accountant	8200	8200

I.Afisati informatii complete (toate coloanele din tabela jobs) despre job-urile pe care lucreaza cei mai multi angajati.

SELECT job_id ID_Job_AP, job_title Nume_Job_AP, min_salary Salariu_Minim_AP, max_salary Salariu_Maxim_AP
FROM jobs
WHERE job_id IN (SELECT job_id
            FROM (SELECT j.job_id, COUNT(e.employee_id) NR_ANGAJATI
                FROM jobs j
                INNER JOIN employees e ON j.job_id = e.job_id
                GROUP BY j.job_id, j.job_title
                ORDER BY NR_ANGAJATI DESC)
            WHERE ROWNUM=1);

Comentariu:
Prima data, a fost creat un tabel cu doua coloane: id-ul jobului si numarul de angajati ai acelui job, grupat in functie de id si numele joburilor si sortat descrescator in functie de numarul angajatilor. Folosind acest tabel am selectat id-ului jobului cu cei mai multi angajati folosind ROWNUM, iar mai apoi am afisat toate informatiile pentru acesta.
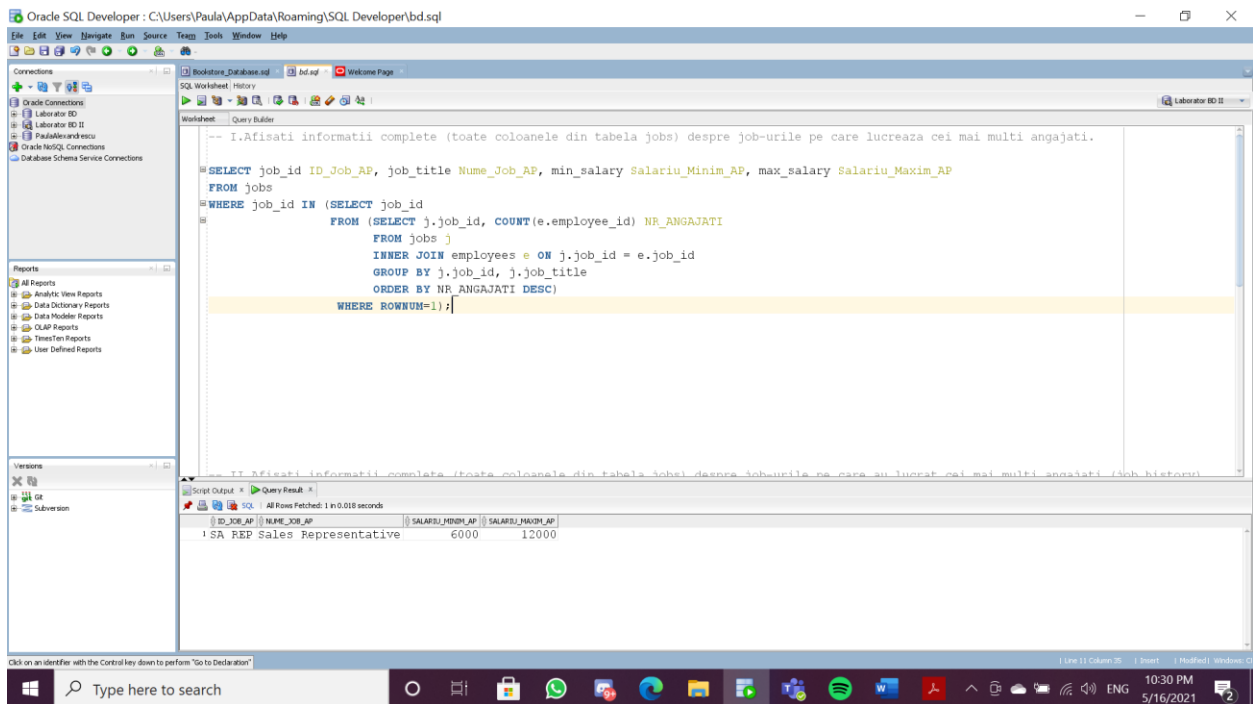
II.Afisati informatii complete (toate coloanele din tabela jobs) despre job-urile pe care au lucrat cei mai multi angajati (job_history).

SELECT job_id ID_Job_AP, job_title Nume_Job_AP, min_salary Salariu_Minim_AP, max_salary Salariu_Maxim_AP
FROM jobs

```
            WHERE job_id IN (SELECT job_id
                    FROM (SELECT j.job_id, COUNT(e.employee_id) NR_ANGAJATI
                    FROM job_history j
                    INNER JOIN employees e ON j.job_id = e.job_id
                    GROUP BY j.job_id
                    ORDER BY NR_ANGAJATI DESC)
                    WHERE ROWNUM=1);
```

Comentariu:

Prima data, a foist create un tabel cu doua coloane: id-ul jobului si numarul de fosti angajati ai acelui job, grupat in functie de id si numele joburilor si sortat descrescator in functie de numarul angajatilor. Folosind acest tabel am selectat id-ului jobului cu cei mai multi fosti angajati folosind ROWNUM, iar mai apoi am afisat toate informatiile pentru acesta.
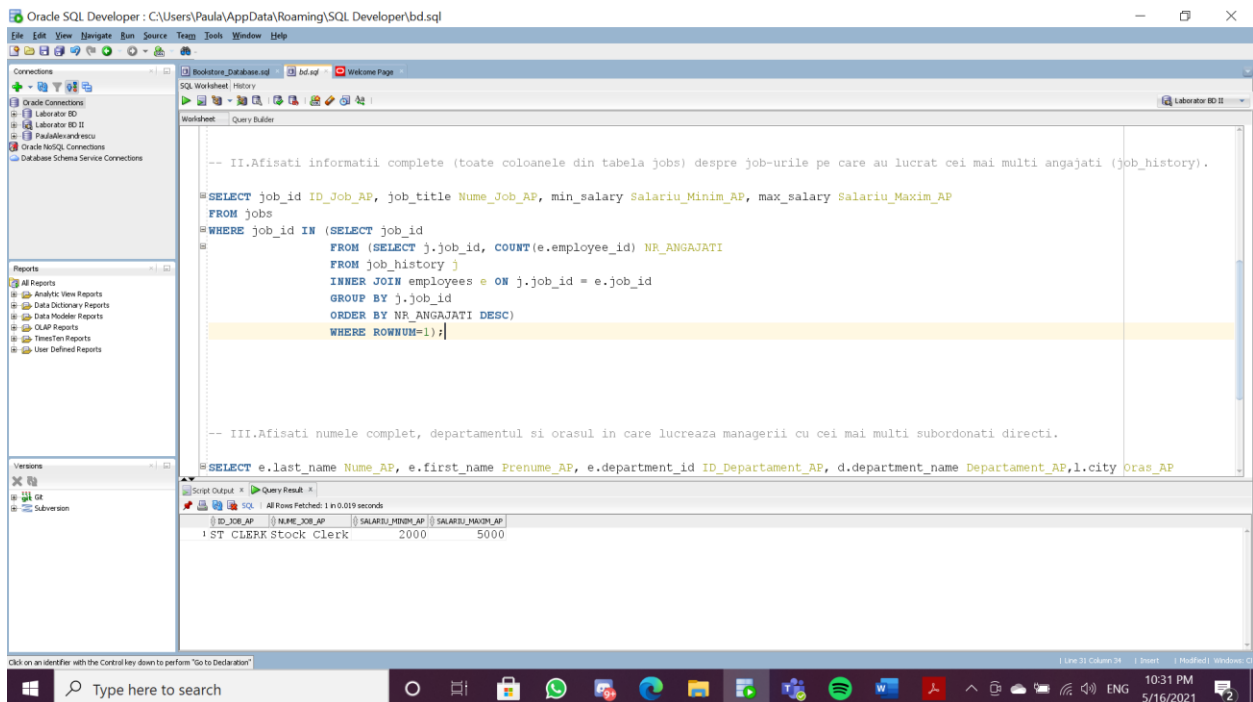
III.Afisati numele complet, departamentul si orasul in care lucreaza managerii cu cei mai multi subordonati directi.

```
        SELECT  e.last_name  Nume_AP,  e.first_name  Prenume_AP,  e.department_id  ID_Departament_AP,
        d.department_name Departament_AP,l.city Oras_AP
        from employees e, departments d, locations l
        where e.department_id=d.department_id
        and d.location_id=l.location_id
        and employee_id IN (SELECT manager_id
                FROM (SELECT manager_id, count(manager_id) NR_ANGAJATI_MANAGER
                    FROM employees
                    group by manager_id
                    ORDER BY NR_ANGAJATI_MANAGER DESC)
                WHERE ROWNUM=1);
```

Comentariu:

Am retinut manager _id si numarul de angajati care au acest id ca manager_id din tabela employees si am sortat descrescator in funtie de numarul de angajati. Folosind ROWNUM am ramas doar cu id-ul managerului cu cei mai multi subordonati directi. Am afisat toate datele angajatului care are employee-id egal cu manager id-ul gasit anterior.