Tema 2

Laborator BD

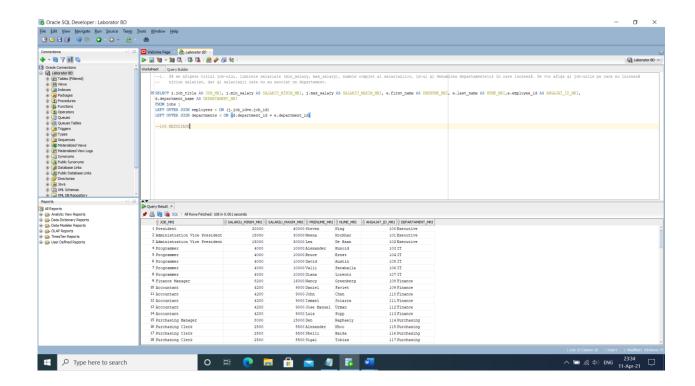
EXERCITIUL 1

Cod:

SELECT j.job_title AS JOB_MRI, j.min_salary AS
SALARIU_MINIM_MRI, j.max_salary AS SALARIU_MAXIM_MRI,
e.first_name AS PRENUME_MRI, e.last_name AS NUME_MRI,
e.employee_id AS ANGAJAT_ID_MRI,
d.department_name AS DEPARTAMENT_MRI
FROM jobs j
LEFT OUTER JOIN employees e ON (j.job_id=e.job_id)
LEFT OUTER JOIN departments d ON (d.department_id = e.department_id)

Comentarii:

Afisam datele cerute prin parcurgerea tabelelor employees si departments cu ajutorul operatorului LEFT OUTER JOIN cu care verificam egalitatile pentru job_id si department_id.



EXERCITIUL 2

Cod:

SELECT UNIQUE 'Departamentul ' || d.department_name || ' este condus de ' || NVL(TO_CHAR(d.manager_id), 'nimeni') || ' si nu are salariati.' AS INFORMATII_DEPARTAMENTE_MRI

FROM departments d

WHERE d.department_id IN (SELECT department_id FROM departments MINUS SELECT department_id FROM employees)

UNION

SELECT UNIQUE 'Departamentul ' || d.department_name || ' este condus de ' || NVL(TO_CHAR(d.manager_id), 'nimeni') || ' si are salariati.' AS INFORMATII_DEPARTAMENTE_MRI

FROM departments d

WHERE d.department_id NOT IN (SELECT department_id FROM departments MINUS SELECT department_id FROM employees)

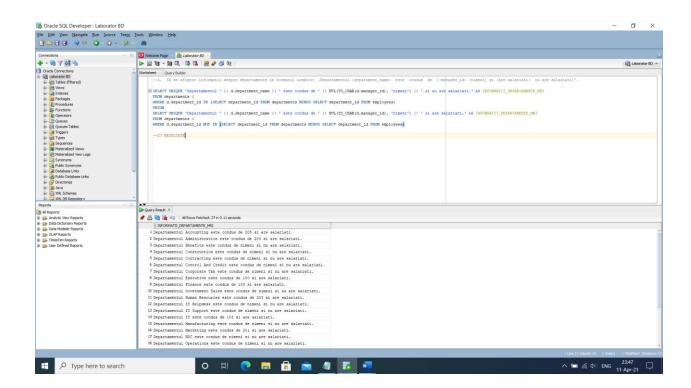
Comentarii:

Formam doua multimi: departamentele care au angajati si departamentele care nu au angajati.

La fiecare, managerul departamentului este verificat in tabelul departments, iar rezultatul (id-ul sau 'nimeni') il afisam cu ajutorul functiei NVL.

Pentru prima multime verificam daca id-ul departamentului se afla in diferenta multimilor departamentelor din departamente si cea a departamentelor in care lucreaza angajati, in timp ce pentru a doua verificam daca nu se afla in aceasta diferenta.

Nota: Se poate folosi si NVL2 pentru a verifica daca exista salariati in acel department.



EXERCITIUL 3

Cod:

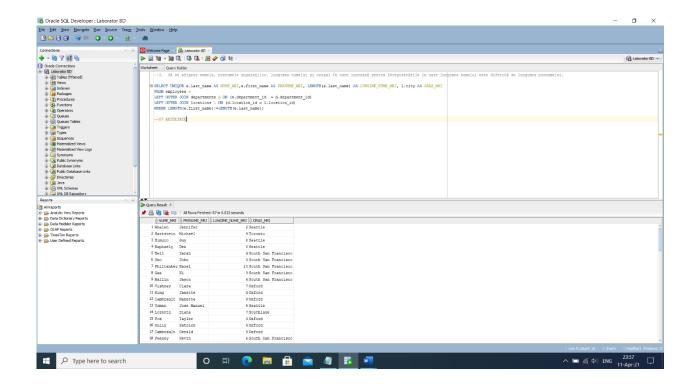
SELECT UNIQUE e.last_name AS NUME_MRI, e.first_name AS PRENUME_MRI, LENGTH(e.last_name) AS LUNGIME_NUME_MRI ,l.city AS ORAS_MRI

FROM employees e

LEFT OUTER JOIN departments d ON (e.department_id=d.department_id)
LEFT OUTER JOIN locations 1 ON (d.location_id = l.location_id)
WHERE LENGTH(e.first_name)!=LENGTH(e.last_name);

Comentarii:

Afisam datele cerute prin parcurgerea tabelelor locations si departments cu ajutorul operatorului LEFT OUTER JOIN cu care verificam egalitatile pentru location_id si department_id, la care adaugam conditia din enunt (lungimea numelui salariatului sa fie diferita de lungimea prenumelui salariatului).



EXERCITIUL 4

Cod:

Cu DECODE:

SELECT UNIQUE e.last_name AS NUME_MRI, e.hire_date AS DATA_ANGAJARE_MRI, j.job_title AS NUME_JOB_MRI, e.salary AS SALARIU_MRI,

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) AS SALARIU_FINAL_MRI

FROM employees e

LEFT OUTER JOIN jobs j ON (e.job_id=j.job_id);

Cu CASE:

SELECT UNIQUE e.last_name AS NUME_MRI, e.hire_date AS DATA_ANGAJARE_MRI, j.job_title AS NUME_JOB_MRI, e.salary AS SALARIU_MRI,

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

FROM employees e

LEFT OUTER JOIN jobs j ON (e.job_id = j.job_id);

Comentarii:

Vom afisa datele cerute din tabelele employees si jobs, pe cel din urma apelandu-l cu ajutorul functiei LEFT OUTER JOIN.

Cautam in employees salariatii care au anul angajarii 1989, 1990, 1991 cu DECODE/CASE. In cazul in care salariatii au fost angajati in acei ani, ultima coloana va reprezenta salariul dupa marire, in caz contrat afisam salariul actual.

