

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.

The screenshot shows the Oracle SQL Developer interface. The 'Query Builder' window displays a query that selects job titles, salary ranges, and employee details, excluding departments without employees. The 'Query Result' window shows the output of this query, listing 18 departments with their respective managers, salaries, and employee counts.

JOB_MRI	SALARIU_MIN_MRI	SALARIU_MAX_MRI	PRENUM_MRI	NUME_MRI	ANGAJAT_ID_MRI	DEPARTAMENT_MRI
1 President	20000	40000	Steven	King	100	Executive
2 Administration Vice President	15000	30000	Henna	Hochbar	101	Executive
3 Administration Vice President	15000	30000	Lex	De Haan	102	Executive
4 Programmer	4000	10000	Alexander	Burnold	103	IT
5 Programmer	4000	10000	Bruce	Ernat	104	IT
6 Programmer	4000	10000	David	Austin	105	IT
7 Programmer	4000	10000	Valli	Pataballa	106	IT
8 Programmer	4000	10000	Diana	Lavere	107	IT
9 Finance Manager	8200	14000	Hancy	Greenberg	108	Finance
10 Accountant	4200	9000	Daniel	Faviet	109	Finance
11 Accountant	4200	9000	John	Chen	110	Finance
12 Accountant	4200	9000	Ismael	Sclarsa	111	Finance
13 Accountant	4200	9000	Jose Manuel	Orman	112	Finance
14 Accountant	4200	9000	Sula	Popp	113	Finance
15 Purchasing Manager	8000	15000	Den	Raphaely	114	Purchasing
16 Purchasing Clerk	2500	5500	Alexander	Khoo	115	Purchasing
17 Purchasing Clerk	2500	5500	Shelli	Beida	116	Purchasing
18 Purchasing Clerk	2500	5500	Sigal	Tobias	117	Purchasing

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.

The screenshot shows the Oracle SQL Developer interface. The main window displays a query in the Query Builder:

```
--2. Să se afişeze informaţii despre departamente în formatul următor: <Departamentul <department_name> este condus de <manager_id> (nimeni) şi (are salariati| nu are salariati)>.
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 departamente 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 departamente d
WHERE d.department_id NOT IN (SELECT department_id FROM departments MINUS SELECT department_id FROM employees)
```

The Results window shows the output of the query:

INFORMATII_DEPARTAMENTE_MRI
1 Departamentul Accounting este condus de 205 si are salariati.
2 Departamentul Administration este condus de 200 si are salariati.
3 Departamentul Benefits este condus de nimeni si nu are salariati.
4 Departamentul Construction este condus de nimeni si nu are salariati.
5 Departamentul Contracting este condus de nimeni si nu are salariati.
6 Departamentul Control And Credit este condus de nimeni si nu are salariati.
7 Departamentul Corporate Tax este condus de nimeni si nu are salariati.
8 Departamentul Executive este condus de 100 si are salariati.
9 Departamentul Finance este condus de 100 si are salariati.
10 Departamentul Government Sales este condus de nimeni si nu are salariati.
11 Departamentul Human Resources este condus de 203 si are salariati.
12 Departamentul IT Helpdesk este condus de nimeni si nu are salariati.
13 Departamentul IT Support este condus de nimeni si nu are salariati.
14 Departamentul IT este condus de 109 si are salariati.
15 Departamentul Manufacturing este condus de nimeni si nu are salariati.
16 Departamentul Marketing este condus de 201 si are salariati.
17 Departamentul NOC este condus de nimeni si nu are salariati.
18 Departamentul Operations este condus de nimeni si nu are salariati.

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 l 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).

The screenshot shows the Oracle SQL Developer interface with a query executed in the 'Query Builder' window. The query is as follows:

```
--3. Sa se afiseze numele, prenumele angajatilor, lungimea numelui si orasul in care lucreaza pentru inregistrările in care lungimea numelui este diferita de lungimea prenumelui.
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 l ON (d.location_id = l.location_id)
WHERE LENGTH(e.first_name) != LENGTH(e.last_name);
```

The results are displayed in the 'Query Result' window, showing 18 rows of data. The columns are NUME_MRI, PRENUME_MRI, LUNGIME_NUME_MRI, and ORAS_MRI.

NUME_MRI	PRENUME_MRI	LUNGIME_NUME_MRI	ORAS_MRI
Whalen	Jennifer	6	Seattle
Barretstein	Michael	9	Toronto
Rimmo	Roy	6	Seattle
Raphaely	Den	5	Seattle
Ball	Sarah	4	South San Francisco
Seo	John	3	South San Francisco
Philtanker	Raisel	10	South San Francisco
Lee	EL	3	South San Francisco
Wallin	Jason	6	South San Francisco
Vishney	Clara	7	Oxford
King	Janette	4	Oxford
Cambault	Patette	9	Oxford
Conat	Joie Hanuel	9	Seattle
Loorents	Diana	7	Southlake
Fox	Taylor	3	Oxford
Sully	Patrick	8	Oxford
Cambault	Gerald	9	Oxford
Feeney	Kevin	6	South San Francisco

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 contrar afisam salariul actual.

The screenshot shows the Oracle SQL Developer interface. The main window displays a SQL query in the Query Builder. The query is as follows:

```

-- 4. Se afiseaza numele, data angajarii, titlul job-ului, salariul si o coloana reprezentand salariul dupa ce se aplica o marire, astfel:
--pentru salariatii angajati in 1989 creterea este de 20%, pentru cei angajati in 1990 creterea este de 15%, iar salariul celor angajati in anul 1991 creste cu 10%.
--Pentru salariatii angajati in alti ani valoarea nu se modifica. (2 rezolvatii DECODE si 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,
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);

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);

--177 rezultate

```

The query results are displayed in a table with 5 columns: NUME_MRI, DATA_ANGAJARE_MRI, NUME_JOB_MRI, SALARIU_MRI, and SALARIU_FINAL_MRI. The results show 17 rows of employee data, including names, hire dates, job titles, and salaries before and after adjustments.

NUME_MRI	DATA_ANGAJARE_MRI	NUME_JOB_MRI	SALARIU_MRI	SALARIU_FINAL_MRI
1 Urman	07-NOV-95	Accountant	7800	7800
2 Johnson	04-JAN-00	Sales Representative	6200	6200
3 Grant	24-MAY-99	Sales Representative	7000	7000
4 Abel	11-MAY-96	Sales Representative	11000	11000
5 Fox	24-JAN-98	Sales Representative	9600	9600
6 Bloom	23-MAR-98	Sales Representative	10000	10000
7 Hall	20-JUN-97	Sales Representative	9000	9000
8 Phillips	24-OCT-97	Stock Clerk	3200	3200
9 Mills	14-JUN-96	Stock Clerk	3300	3300
10 Mikellinneni	29-SEP-98	Stock Clerk	2700	2700
11 Walsh	24-AUG-98	Shipping Clerk	3100	3100
12 Jones	17-MAR-99	Shipping Clerk	2800	2800
13 Gates	11-JUL-98	Shipping Clerk	2900	2900
14 Scudler	03-JUN-90	Programmer	9000	10350
15 Kochhar	21-SEP-99	Administration Vice President	17000	20400
16 Taylor	24-MAR-90	Sales Representative	6000	6000
17 Anderson	24-MAR-00	Sales Representative	6400	6400
18 Mavris	04-JUN-97	Sales Representative	3500	3500