

1. Sa se afiseze numele, salariul, titlul jobului, codul si numele departamentului, id-ul locatiei, orasul si tara in care lucreaza angajatii condusi direct de

hunoldalexander.

Pe ultima coloana se va afisa numele managerului (Hunold), concatenat cu spatiu, concatenat cu prenumele sau (Alexander).

Coloana o sa se numeasca Nume si Prenume Manager.

```
SELECT * FROM employees;
```

```
SELECT ang.last_name, ang.salary, j.job_title, d.department_id, d.department_name, d.location_id, l.city, c.country_name,
```

```
concat(concat(man.last_name, ' '),man.first_name) "Nume si Prenume Manager"
```

```
FROM employees ang JOIN departments d ON (ang.department_id=d.department_id)
```

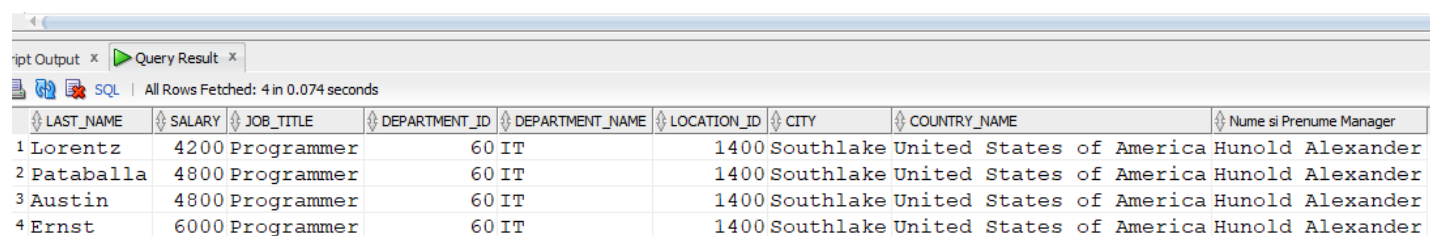
```
JOIN jobs j ON (ang.job_id=j.job_id)
```

```
JOIN locations l ON (d.location_id=l.location_id)
```

```
JOIN countries c ON (l.country_id=c.country_id)
```

```
JOIN employees man ON (ang.manager_id=man.employee_id)
```

```
WHERE lower(concat(man.last_name,man.first_name))='hunoldalexander';
```



Query Output x Query Result x

SQL | All Rows Fetched: 4 in 0.074 seconds

	LAST_NAME	SALARY	JOB_TITLE	DEPARTMENT_ID	DEPARTMENT_NAME	LOCATION_ID	CITY	COUNTRY_NAME	Nume si Prenume Manager
1	Lorentz	4200	Programmer	60	IT	1400	Southlake	United States of America	Hunold Alexander
2	Pataballa	4800	Programmer	60	IT	1400	Southlake	United States of America	Hunold Alexander
3	Austin	4800	Programmer	60	IT	1400	Southlake	United States of America	Hunold Alexander
4	Ernst	6000	Programmer	60	IT	1400	Southlake	United States of America	Hunold Alexander

2. Sa se listeze codul departamentului, numele departamentului si codul managerului de departament. In cazul in care un departament nu are manager se va afisa mesajul:

❖Departamentul <department_id> nu are manager❖ (ex: Departamentul 120 nu are manager). Coloana se va numi ❖Manageri departamente❖.

De asemenea, in cadrul aceleiasi cereri, sa se afiseze atat departamentele care au angajati, cat si cele fara angajati.

In cazul in care un departament are angajati, se va afisa si codul acestor angajati (o coloana unde se vor afisa codurile de angajati pentru fiecare departament in parte).

Daca un departament nu are angajati, se va afisa mesajul: ❖Departamentul nu are angajati❖. Coloana se va numi ❖Angajati departamente❖.

In final se vor afisa 4 coloane: department_id, department_name, Angajati departamente, Manageri departamente.

In acest caz, fiind mai multe randuri returnate, atasati un singur print screen care sa cuprinda toate cele 4 coloane, dar doar ultimele 25 de inregistrari.

SELECT d.department_id, d.department_name, NVL(to_char(e.employee_id), 'Departamentul nu are angajati') "Angajati Departamente",

NVL(to_char(d.manager_id),concat(concat('Departamentul ',to_char(d.department_id)), ' nu are manager')) "Manageri departamente"

FROM departments d LEFT JOIN employees e ON (d.department_id=e.department_id);

```
2. Sa se listeze codul departamentului, numele departamentului si codul managerului de departament. In cazul in care un departament nu are manager se va afisa mesajul:
❖Departamentul <department_id> nu are manager❖ (ex: Departamentul 120 nu are manager). Coloana se va numi ❖Manageri departamente❖.
De asemenea, in cadrul aceleiasi cereri, sa se afiseze atat departamentele care au angajati, cat si cele fara angajati.
In cazul in care un departament are angajati, se va afisa si codul acestor angajati (o coloana unde se vor afisa codurile de angajati pentru fiecare departament in parte)
Daca un departament nu are angajati, se va afisa mesajul: ❖Departamentul nu are angajati❖. Coloana se va numi ❖Angajati departamente❖.
In final se vor afisa 4 coloane: department_id, department_name, Angajati departamente, Manageri departamente.
In acest caz, fiind mai multe randuri returnate, atasati un singur print screen care sa cuprinda toate cele 4 coloane, dar doar ultimele 25 de inregistrari.

SELECT d.department_id, d.department_name, NVL(to_char(e.employee_id), 'Departamentul nu are angajati') "Angajati Departamente",
       NVL(to_char(d.manager_id),concat(concat('Departamentul ',to_char(d.department_id)), ' nu are manager')) "Manageri departamente"
FROM departments d LEFT JOIN employees e ON (d.department_id=e.department_id);
```

DEPARTMENT_ID	DEPARTMENT_NAME	Angajati Departamente	Manageri departamente
50	Shipping	188	121
89	50 Shipping	189	121
90	50 Shipping	190	121
91	50 Shipping	191	121
92	50 Shipping	192	121
93	50 Shipping	193	121
94	50 Shipping	194	121
95	50 Shipping	195	121
96	50 Shipping	196	121
97	50 Shipping	197	121
98	50 Shipping	198	121
99	50 Shipping	199	121
100	10 Administration	200	200
101	20 Marketing	201	201
102	20 Marketing	202	201
103	40 Human Resources	203	203
104	70 Public Relations	204	204
105	110 Accounting	205	205
106	110 Accounting	206	205
107	220 NOC	Departamentul nu are angajati	Departamentul 220 nu are manager
108	170 Manufacturing	Departamentul nu are angajati	Departamentul 170 nu are manager
109	240 Government Sales	Departamentul nu are angajati	Departamentul 240 nu are manager
110	210 IT Support	Departamentul nu are angajati	Departamentul 210 nu are manager
111	160 Benefits	Departamentul nu are angajati	Departamentul 160 nu are manager
112	150 Shareholder Services	Departamentul nu are angajati	Departamentul 150 nu are manager
113	250 Retail Sales	Departamentul nu are angajati	Departamentul 250 nu are manager
114	140 Control And Credit	Departamentul nu are angajati	Departamentul 140 nu are manager
115	260 Recruiting	Departamentul nu are angajati	Departamentul 260 nu are manager
116	200 Operations	Departamentul nu are angajati	Departamentul 200 nu are manager
117	120 Treasury	Departamentul nu are angajati	Departamentul 120 nu are manager
118	270 Payroll	Departamentul nu are angajati	Departamentul 270 nu are manager
119	130 Corporate Tax	Departamentul nu are angajati	Departamentul 130 nu are manager
120	180 Construction	Departamentul nu are angajati	Departamentul 180 nu are manager
121	190 Contracting	Departamentul nu are angajati	Departamentul 190 nu are manager
122	230 IT Helpdesk	Departamentul nu are angajati	Departamentul 230 nu are manager

3. Sa se creeze tabelele urmatoare CAMPANIE_PNU si SPONSOR_PNU

Unde PNU se formeaza astfel:

? P -> prima litera din prenume

? NU -> primele doua litere din nume

CAMPANIE_PNU

(cod_campanie ? cheie primara,

titlu -> titlul campaniei ? nu poate fi null,

data_start -> data la care incepe campania ? are implicit

valoarea sysdate,

data_end -> data la care se incheie campania ? este o data

inserata in momentul inserarii inregistrarii in baza de date; aceasta data trebuie sa fie mai mare decat data_start,
valoare -> pretul campaniei ? poate fi null,
cod_sponsor ? cheie externa
)

SPONSOR_PNU

(cod_sponsor ? cheie primara,
nume -> denumirea sponsorului ? nu poate fi null si trebuie sa aiba o valoare unica,
email -> poate fi null si are o valoare unica
)

OBS:

Relatia dintre cele doua tabele este one-to-many -> un sponsor poate sponsoriza mai multe campanii, iar o campanie este sponsorizata doar de un singur sponsor.

Instructiunea commit se va rula doar unde este necesara stocarea datelor/modificarilor in baza de date sau doar acolo unde cerinta specifica acest lucru.

CREATE TABLE SPONSOR_COCO

```
(  
  cod_sponsor NUMBER(3),  
  constraint pk_sponsor primary key(cod_sponsor),  
  nume VARCHAR(20) constraint nume_unic_sponsor unique  
    constraint sponsor_not_null not null,  
  email VARCHAR(20) constraint email_unic_sponsor unique  
);
```

CREATE TABLE CAMPANIE_COCO

```
(
    cod_campanie NUMBER(3),
    constraint pk_campanie primary key(cod_campanie),
    titlu_campanie VARCHAR(26) constraint titlu_campanie_not_null not null,
    data_start date default sysdate,
    data_end DATE,
    constraint data_mai_mare_ca_data_start check(data_end>data_start),
    valoare NUMBER(7),
    cod_sponsor NUMBER(3) constraint fk_cod_sponsor references SPONSOR_COCO(cod_sponsor)
);
```

Table CAMPANIE_COCO created.

Name	Null?	Type
COD_CAMPANIE	NOT NULL	NUMBER(3)
TITLU_CAMPANIE	NOT NULL	VARCHAR2(26)
DATA_START		DATE
DATA_END		DATE
VALOARE		NUMBER(7)
COD_SPONSOR		NUMBER(3)

Name	Null?	Type
COD_SPONSOR	NOT NULL	NUMBER(3)
NUME	NOT NULL	VARCHAR2(20)
EMAIL		VARCHAR2(20)

4. Sa se insereze in baza de date urmatoarele inregistrari, folosind la alegere metoda implicita sau explicita, precizand varianta aleasa.

SPONSOR

--folosesc metoda explicita

INSERT INTO SPONSOR_COCO(cod_sponsor, nume, email)

VALUES

```
--(10, 'CISCO', 'cisco@gmail.com');  
--(20, 'KFC', NULL);  
--(30, 'ADOBE', 'adobe@adobe.com');  
--(40, 'BRD', NULL);  
--(50, 'VODAFONE', 'vdf@gmail.com');  
--(60, 'BCR', NULL);  
--(70, 'SAMSUNG', NULL);  
--(80, 'IBM', 'ibm@ibm.com');  
--(90, 'OMV', NULL);  
(100, 'ENEL', NULL);
```

--metoda implicita

INSERT INTO CAMPANIE_COCO

VALUES

```
(1, 'CAMP1', sysdate, to_date('20-06-2023', 'dd-mm-yyyy'), 1200, 10);  
(2, 'CAMP2', sysdate, to_date('25-07-2023', 'dd-mm-yyyy'), 3400, 20);  
(3, 'CAMP3', sysdate, to_date('10-06-2023', 'dd-mm-yyyy'), NULL, 30);  
(4, 'CAMP4', sysdate, to_date('20-06-2023', 'dd-mm-yyyy'), NULL, 40);  
(5, 'CAMP5', sysdate, to_date('05-06-2023', 'dd-mm-yyyy'), 2200, 50);  
(6, 'CAMP6', sysdate, to_date('15-08-2023', 'dd-mm-yyyy'), NULL, 60);  
(7, 'CAMP7', sysdate, to_date('02-09-2023', 'dd-mm-yyyy'), 5500, 70);  
(8, 'CAMP8', sysdate, to_date('10-10-2023', 'dd-mm-yyyy'), NULL, 20);  
(9, 'CAMP9', sysdate, to_date('10-06-2023', 'dd-mm-yyyy'), 4000, 30);  
(10, 'CAMP10', sysdate, to_date('25-09-2023', 'dd-mm-yyyy'), 3500, NULL);
```

select * from campanie_coco;

```
--folosesc metoda explicita
```

```
INSERT INTO SPONSOR_COCO(cod_sponsor, nume, email)  
VALUES
```

```
--(10, 'CISCO', 'cisco@gmail.com');  
--(20, 'KFC', NULL);  
--(30, 'ADOBE', 'adobe@adobe.com');  
--(40, 'BRD', NULL);  
--(50, 'VODAFONE', 'vdf@gmail.com');  
--(60, 'BCR', NULL);  
--(70, 'SAMSUNG', NULL);  
--(80, 'IBM', 'ibm@ibm.com');  
--(90, 'OMV', NULL);  
(100, 'ENEL', NULL);
```

ript Output x Query Result x

SQL | All Rows Fetched: 10 in 0.122 seconds

	COD_SPONSOR	NUME	EMAIL
1	10	CISCO	cisco@gmail.com
2	20	KFC	(null)
3	30	ADOBE	adobe@adobe.com
4	40	BRD	(null)
5	50	VODAFONE	vdf@gmail.com
6	60	BCR	(null)
7	70	SAMSUNG	(null)
8	80	IBM	ibm@ibm.com
9	90	OMV	(null)
10	100	ENEL	(null)

```

1  --metoda implicita
2  INSERT INTO CAMPANIE_COCO
3  VALUES
4      (1, 'CAMP1', sysdate, to_date('20-06-2023', 'dd-mm-yyyy'), 1200, 10);
5      (2, 'CAMP2', sysdate, to_date('25-07-2023', 'dd-mm-yyyy'), 3400, 20);
6      (3, 'CAMP3', sysdate, to_date('10-06-2023', 'dd-mm-yyyy'), NULL, 30);
7      (4, 'CAMP4', sysdate, to_date('20-06-2023', 'dd-mm-yyyy'), NULL, 40);
8      (5, 'CAMP5', sysdate, to_date('05-06-2023', 'dd-mm-yyyy'), 2200, 50);
9      (6, 'CAMP6', sysdate, to_date('15-08-2023', 'dd-mm-yyyy'), NULL, 60);
10     (7, 'CAMP7', sysdate, to_date('02-09-2023', 'dd-mm-yyyy'), 5500, 70);
11     (8, 'CAMP8', sysdate, to_date('10-10-2023', 'dd-mm-yyyy'), NULL, 20);
12     (9, 'CAMP9', sysdate, to_date('10-06-2023', 'dd-mm-yyyy'), 4000, 30);
13     (10, 'CAMP10', sysdate, to_date('25-09-2023', 'dd-mm-yyyy'), 3500, NULL);
14
15 select * from campanie_coco;

```

Script Output x Query Result x

SQL | All Rows Fetched: 10 in 0.073 seconds

	COD_CAMPANIE	TITLU_CAMPANIE	DATA_START	DATA_END	VALOARE	COD_SPONSOR
1	10	CAMP10	24-APR-23	25-SEP-23	3500	(null)
2	9	CAMP9	24-APR-23	10-JUN-23	4000	30
3	8	CAMP8	24-APR-23	10-OCT-23	(null)	20
4	7	CAMP7	24-APR-23	02-SEP-23	5500	70
5	6	CAMP6	24-APR-23	15-AUG-23	(null)	60
6	5	CAMP5	24-APR-23	05-JUN-23	2200	50
7	4	CAMP4	24-APR-23	20-JUN-23	(null)	40
8	3	CAMP3	24-APR-23	10-JUN-23	(null)	30
9	2	CAMP2	24-APR-23	25-JUL-23	3400	20
10	1	CAMP1	24-APR-23	20-JUN-23	1200	10

5. Sa se stearga campaniile care vor expira inainte de data 01-07-2023.

Se va adauga un print screen cu rezultatele ramase in urma stergerii, dupa care se vor anula modificarile.

DELETE FROM CAMPANIE_CCO

WHERE data_end < to_date('01-07-2023', 'dd-mm-yyyy');

select* from campanie_cco;

rollback;

rkshet Query Builder

6 si se vor adauga in document print screen-uri cu output-ul din SQL Developer.

7

8

9

0 5. Sa se stearga campaniile care vor expira inainte de data 01-07-2023.

1 Se va adauga un print screen cu rezultatele ramase in urma stergerii, dupa care se vor anula modificarile.

2

3 DELETE FROM CAMPANIE_CCO

4 WHERE data_end<to_date('01-07-2023','dd-mm-yyyy');

5 select* from campanie_cco;

6

7 6. Sa se modifice valoarea tuturor campaniilor, aplicandu-se o majorare cu 25%. Daca o campanie nu are valoare, atun

8 ?Campanie Caritabila?. Se va atasa in document un print cu valorile modificate (output-ul dupa rulare) dupa care se

9

10

11

12 7. Sa se stearga sponsorul 20 din baza de date. Explicati in cuvinte pasii necesari rezolvarii cu succes a cerintei

Script Output x Query Result x

SQL All Rows Fetched: 5 in 0.03 seconds

	COD_CAMPANIE	TITLU_CAMPANIE	DATA_START	DATA_END	VALOARE	COD_SPONSOR
1	2	CAMP2	24-APR-23	25-JUL-23	3400	20
2	6	CAMP6	24-APR-23	15-AUG-23	(null)	60
3	7	CAMP7	24-APR-23	02-SEP-23	5500	70
4	8	CAMP8	24-APR-23	10-OCT-23	(null)	20
5	10	CAMP10	24-APR-23	25-SEP-23	3500	(null)

6. Sa se modifice valoarea tuturor campaniilor, aplicandu-se o majorare cu 25%. Daca o campanie nu are valoare, atunci ea este o campanie caritabila si se va completa cu textul

?Campanie Caritabila?. Se va atasa in document un print cu valorile modificate (output-ul dupa rulare) dupa care se vor anula modificarile.

#AM MODIFICAT TIPUL DE DATE DIN TABEL PT A PUTEA ACCEPTA SI TEXTUL „CAMPANIE CARITABILA”

ALTER TABLE CAMPANIE_CCO

DROP COLUMN valoare;

ALTER TABLE CAMPANIE_CCO

ADD valoare VARCHAR(10);

UPDATE CAMPANIE_CCO

SET valoare= 1200

WHERE cod_campanie=1;

UPDATE CAMPANIE_CCO

SET valoare= 3400

WHERE cod_campanie=2;

UPDATE CAMPANIE_CCO

SET valoare= NULL

WHERE cod_campanie=3;

UPDATE CAMPANIE_CCO

SET valoare= NULL

WHERE cod_campanie=4;

UPDATE CAMPANIE_CCO

SET valoare= 2200

WHERE cod_campanie=5;

UPDATE CAMPANIE_CCO

SET valoare= NULL

WHERE cod_campanie=6;

UPDATE CAMPANIE_CCO

SET valoare= 5500

WHERE cod_campanie=7;

UPDATE CAMPANIE_CCO

SET valoare= NULL

WHERE cod_campanie=8;

UPDATE CAMPANIE_CCO

SET valoare= 4000

WHERE cod_campanie=9;

UPDATE CAMPANIE_CCO

SET valoare= 3500

WHERE cod_campanie=10;

select* from campanie_cco;

ALTER TABLE CAMPANIE_CCO

MODIFY valoare VARCHAR(25);

UPDATE CAMPANIE_CCO

SET valoare= (NVL(to_char(1.25*to_number(valoare)), 'Campanie Caritabila'));

select* from CAMPANIE_CCO;

rollback;

Worksheet | Query Builder

```

203 SET valoare= 3500
204 WHERE cod_campanie=10;
205
206 select* from campanie_cco;
207
208 ALTER TABLE CAMPANIE_CCO
209 MODIFY valoare VARCHAR(25);
210
211 UPDATE CAMPANIE_CCO
212 SET valoare= (NVL(to_char(1.25*to_number(valoare)), 'Campanie Caritabila'));
213
214 select* from CAMPANIE_CCO;
215
216 7. Sa se stearga sponsorul 20 din baza de date. Explicati in cuvinte pasii necesari rezolvarii cu succes
217
218

```

Script Output x | Query Result x

All Rows Fetched: 10 in 0.036 seconds

	COD_CAMPANIE	TITLU_CAMPANIE	DATA_START	DATA_END	COD_SPONSOR	VALOARE
1	1	CAMP1	24-APR-23	20-JUN-23	10	1500
2	2	CAMP2	24-APR-23	25-JUL-23	20	4250
3	3	CAMP3	24-APR-23	10-JUN-23	30	Campanie Caritabila
4	4	CAMP4	24-APR-23	20-JUN-23	40	Campanie Caritabila
5	5	CAMP5	24-APR-23	05-JUN-23	50	2750
6	6	CAMP6	24-APR-23	15-AUG-23	60	Campanie Caritabila
7	7	CAMP7	24-APR-23	02-SEP-23	70	6875
8	8	CAMP8	24-APR-23	10-OCT-23	20	Campanie Caritabila
9	9	CAMP9	24-APR-23	10-JUN-23	30	5000
10	10	CAMP10	24-APR-23	25-SEP-23	(null)	4375

7. Sa se stearga sponsorul 20 din baza de date. Explicati in cuvinte pasii necesari rezolvarii cu succes a cerintei. Dupa stergere anulati modificarile.

```
select * from campanie_cco;
```

#Prima oara modificam campaniile sponsorizate de sponsorul care urmeaza sa dispara, altfel nu putem sterge sponsorul respectiv, deoarece are „copii”

```
UPDATE CAMPANIE_CCO
```

```
SET cod_sponsor=NULL
```

```
WHERE cod_sponsor=20;
```

```
DELETE FROM SPONSOR_CCO WHERE cod_sponsor=20;
```

```
rollback;
```

Script Output x Query Result x			
SQL All Rows Fetched: 9 in 0.033 seconds			
	COD_SPONSOR	NUME	EMAIL
1	100	ENEL	(null)
2	90	OMV	(null)
3	80	IBM	ibm@ibm.com
4	70	SAMSUNG	(null)
5	60	BCR	(null)
6	50	VODAFONE	vdf@gmail.com
7	40	BRD	(null)
8	30	ADOBE	adobe@adobe.com
9	10	CISCO	cisco@gmail.com

Script Output x Query Result x					
SQL All Rows Fetched: 10 in 0.055 seconds					
	COD_CAMPANIE	TITLU_CAMPANIE	DATA_START	DATA_END	COD_SPONSOR VALOARE
1	1	CAMP1	24-APR-23	20-JUN-23	10 1200
2	2	CAMP2	24-APR-23	25-JUL-23	(null) 3400
3	3	CAMP3	24-APR-23	10-JUN-23	30 (null)
4	4	CAMP4	24-APR-23	20-JUN-23	40 (null)
5	5	CAMP5	24-APR-23	05-JUN-23	50 2200
6	6	CAMP6	24-APR-23	15-AUG-23	60 (null)
7	7	CAMP7	24-APR-23	02-SEP-23	70 5500
8	8	CAMP8	24-APR-23	10-OCT-23	(null) (null)
9	9	CAMP9	24-APR-23	10-JUN-23	30 4000
10	10	CAMP10	24-APR-23	25-SEP-23	(null) 3500

8. Stergeti sponsorii care nu sponsorizeaza nicio campanie. Dupa stergere realizati un print screen output-ului (SELECT * FROM sponsor), dupa care salvati modificarile.

```
SELECT * FROM SPONSOR_CCO;
```

```
ALTER TABLE SPONSOR_CCO
```

```
ADD are_campanii NUMBER(2) default 0;
```

```
ALTER TABLE SPONSOR_CCO
```

```
DROP COLUMN are_campanii;
```

```
UPDATE SPONSOR_CCO
```

```
SET are_campanii=1
```

```
WHERE cod_sponsor in (select cod_sponsor from CAMPANIE_CCO);
```

```
delete FROM SPONSOR_CCO WHERE are_campanii=0;
```

	⚡ COD_SPONSOR	⚡ NUME	⚡ EMAIL	
1	70	SAMSUNG	(null)	
2	60	BCR	(null)	
3	50	VODAFONE	vdf@gmail.com	
4	40	BRD	(null)	
5	30	ADOBE	adobe@adobe.com	
6	20	KFC	(null)	
7	10	CISCO	cisco@gmail.com	