

Subinterogări. Subinterogări “multi-row”. Formatarea rezultatelor unei interogări

Obiective

Prin parcurgerea acestui referat studentul va dobândi cunoștințele necesare:

- identificării funcțiilor grup disponibile;
- descrierii modului de utilizare a funcțiilor grup;
- grupării datelor folosind clauza `GROUP BY`;
- includerii sau excluderii câmpurilor grupate folosind clauza `HAVING`;
- descrierii tipurilor de subinterogări;
- definiții unei subinterogări;
- scrierii subinterogărilor `single-row` și `multiple-row`.
- scrierea subinterogărilor de mai multe coloane;
- descrierea și explicarea subinterogărilor atunci când sunt returnate valori `null`;
- scrierea unor subinterogări în clauza `FROM`;
- scrierea unei interogări care necesită o variabilă de intrare;
- formatarea datelor afișate;
- crearea și execuția unui fișier script.

Subinterogări

Să presupunem că dorim să interogăm o bază de date pentru a afla numele angajatului cu un salariu mai mare decât salariul lui Jones.

Pentru rezolvarea acestei probleme sunt necesare două interogări: o interogare pentru a afla ce salariu are Jones și o doua pentru a determina cine câștigă mai mult decât această sumă. Problema poate fi rezolvată combinând aceste două interogări, integrând una din cereri în cealaltă.

O cerere inclusă sau *subinterogare* returnează o valoare folosită de interogarea exterioară. Folosirea unei subinterogări este echivalentă executării a două cereri secvențiale și folosirii rezultatului primei cereri ca valoare de căutare pentru cea de a doua cerere.

```
SELECT lista_selectie1
FROM tabel1
WHERE expr operator
      (SELECT lista_selectie2
       FROM tabel2);
```

O *subinterogare* reprezintă o instrucțiune `SELECT` care este inclusă într-o clauză aparținând altei instrucțiuni `SELECT`. Se pot astfel construi instrucțiuni puternice, pornind de la instrucțiuni simple, prin utilizarea subinterogărilor. Acestea pot fi utile în cazurile în care se dorește selectarea unor rânduri dintr-un tabel în raport cu o condiție care depinde de datele din același tabel.

Subinterogările pot fi plasate în următoarele clauze SQL:

- **WHERE**
- **HAVING**
- **FROM**

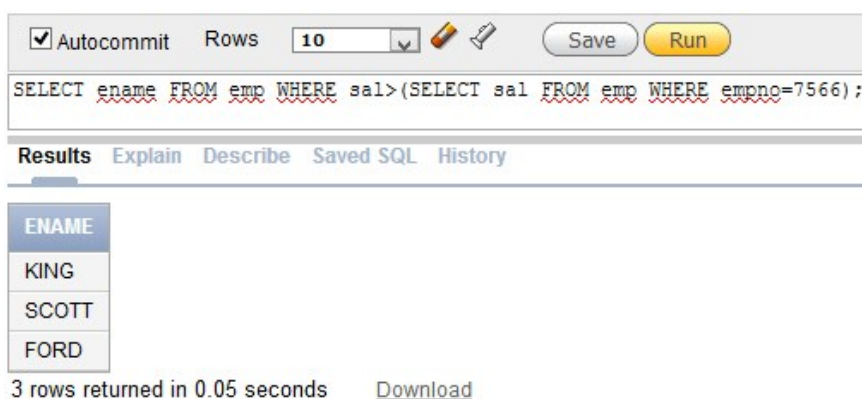
În sintaxa instrucțiunii `SELECT` prezentată mai sus *operator* indică un operator de comparație (>, =, IN, ...).

Notă: Operatorii de comparație se împart în două clase:

- operatori ce acționează asupra unui singur rând (*single-row*): >, =, >=, <, <>, <=
- operatori ce acționează asupra mai multor rânduri (*multiple-row*): IN, ANY, ALL.

Subinterogarea este deseori referită ca fiind o instrucțiune `SELECT` inclusă, un sub-`SELECT` sau o instrucțiune `SELECT` internă. În general, subinterogarea se execută prima, iar rezultatul este folosit pentru a finaliza condiția de cerere din interogarea principală.

```
SELECT ename FROM emp WHERE sal>(SELECT sal FROM emp WHERE empno=7566) ;
```



The screenshot shows a SQL query execution window. At the top, there are controls for 'Autocommit' (checked), 'Rows' (set to 10), and buttons for 'Save' and 'Run'. The query text is: `SELECT ename FROM emp WHERE sal>(SELECT sal FROM emp WHERE empno=7566);`. Below the query, there are tabs for 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History'. The 'Results' tab is active, displaying a table with one column 'ENAME' and three rows: 'KING', 'SCOTT', and 'FORD'. At the bottom, it says '3 rows returned in 0.05 seconds' and has a 'Download' link.

În exemplul anterior cererea internă determină salariul angajatului cu numărul 7566. Cererea externă preia rezultatul cererii interne și îl folosește pentru a afișa toți angajații care câștigă un salariu mai mare decât această sumă.

Indicații privind scrierea subinterogărilor:

- o subinterogare trebuie inclusă între paranteze.
- o subinterogare trebuie să apară în partea dreaptă a unui operator de comparație.
- subinterogările nu pot conține clauza `ORDER BY`. Pentru o instrucțiune `SELECT` poate exista doar o singură clauză `ORDER BY`, iar dacă această clauză este specificată, ea trebuie să fie ultima clauză din instrucțiunea `SELECT` principală.
- subinterogările folosesc două clase de operatori de comparare: operatori *single-row* și operatori *multiple-row*.

Tipuri de subinterogări:

1. subinterogări pe coloană (*single-row*): cereri care returnează doar un rând din instrucțiunea `SELECT` internă;
2. subinterogări multi-rând (*multiple-row*): cereri care returnează mai mult de un rând din instrucțiunea `SELECT` internă;
3. subinterogări multi-coloană (*multiple-column*): cereri care returnează mai multe coloane din instrucțiunea `SELECT` internă..

1. Subinterogări *single-row*

O subinterogare single-row este acea subinterogare care returnează un singur rând din instrucțiunea SELECT internă. Acest tip de subinterogare folosește un operator single-row.

Exemplu: Afișați numele și postul angajaților al căror post este același cu cel al angajatului cu numărul 7369 și al căror salariu este mai mare decât cel al angajatului 7876.

```
SQL> SELECT ename, job FROM emp WHERE job =(SELECT job FROM emp
WHERE empno = 7369)AND sal>(SELECT sal FROM emp WHERE empno=7876) ;
```

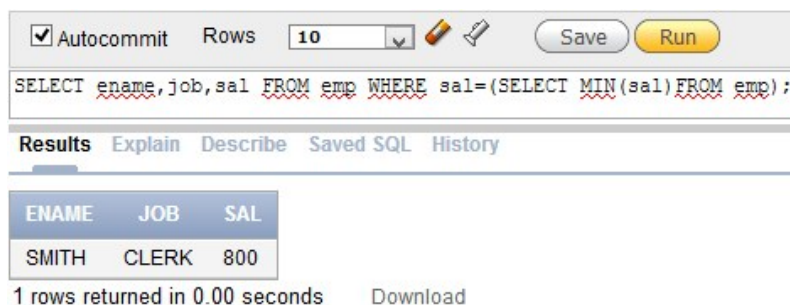
Interogarea de mai sus este formată din 3 blocuri de cereri: o cerere exterioară și două cereri interne. Blocurile de cereri interne se vor executa primele, producând rezultatele cererii: CLERK, respectiv 1100. Blocul exterior de cereri este apoi procesat și folosește valorile returnate de cererile interne pentru a-și finaliza propriile condiții de căutare. Ambele cereri interne returnează câte o singură valoare, astfel că această instrucțiune SQL este denumită *subinterogare single-row*.

Notă: Interogările exterioare și cele incluse pot prelua date din tabele diferite.

O interogare principală poate afișa date utilizând valoarea returnată de o funcție grup din subinterogare. Subinterogarea se va plasa între paranteze, după operatorul de comparație.

Exemplul următor afișează numele, postul și salariul tuturor angajaților al căror salariu este egal cu salariul minim. Funcția MIN returnează o singură valoare (și anume 800), care este folosită de interogarea principală.

```
SELECT ename, job, sal FROM emp WHERE sal=(SELECT MIN(sal)FROM
emp) ;
```



The screenshot shows a SQL query execution window. At the top, there are controls for 'Autocommit' (checked), 'Rows' (set to 10), and buttons for 'Save' and 'Run'. The query text is: `SELECT ename, job, sal FROM emp WHERE sal=(SELECT MIN(sal)FROM emp) ;`. Below the query, there are tabs for 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History'. The 'Results' tab is active, displaying a table with the following data:

ENAME	JOB	SAL
SMITH	CLERK	800

Below the table, it indicates '1 rows returned in 0.00 seconds' and a 'Download' link.

Subinterogări în clauza HAVING

Subinterogările pot fi folosite nu numai în clauza WHERE, dar și în clauza HAVING. Server-ul Oracle execută subinterogarea, returnând apoi rezultatul către clauza HAVING a interogării principale.

Următoarea instrucțiune SQL are ca scop final afișarea tuturor departamentelor la nivelul cărora salariul minim are o valoare mai mare decât valoarea salariului minim din cadrul departamentului 20.

```
SELECT deptno, MIN(sal)FROM emp GROUP BY deptno HAVING
MIN(sal)>(SELECT MIN(sal)FROM emp WHERE deptno = 20) ;
```

Autocommit Rows: 10 Save Run

```

SELECT deptno, MIN(sal) FROM emp GROUP BY deptno
HAVING MIN(sal) > (SELECT MIN(sal) FROM emp WHERE deptno=20);
  
```

Results Explain Describe Saved SQL History

DEPTNO	MIN(SAL)
30	950
10	1300

2 rows returned in 0.03 seconds [Download](#)

Exemplu: Să se găsească funcția având cel mai scăzut salariu mediu:

```

SELECT job, AVG(sal) FROM emp GROUP BY job HAVING
AVG(sal) = (SELECT MIN(AVG(sal)) FROM emp GROUP BY job);
  
```

Erori ce pot apare la folosirea subinterogărilor:

- returnarea mai multor rânduri de către o subinterogare dorită a fi de tip single-row;
- posibilitatea neselectării nici unui rând de către interogarea inclusă.

2. Subinterogări multiple-row

Subinterogările care returnează mai mult de un rând se numesc *subinterogări multiple-row*. În cazul acestui tip de subinterogări se folosesc operatori multiple-row în locul celor single-row.

Operator	Semnificatie
IN	Egal cu oricare din elementele listei
ANY	Compară valoarea cu fiecare valoare returnată de subinterogare
ALL	Compară valoarea cu toate valorile returnate de subinterogare

Următorul exemplu selectează toți angajații care câștigă un salariu egal cu salariul minim la nivel de departament.

```

SELECT ename, sal, deptno FROM emp WHERE sal IN (SELECT MIN(sal)
FROM emp GROUP BY deptno);
  
```

Autocommit Rows: 10 Save Run

```

SELECT ename, sal, deptno FROM emp WHERE sal IN
(SELECT MIN(sal) FROM emp GROUP BY deptno);
  
```

Results Explain Describe Saved SQL History

ENAME	SAL	DEPTNO
SMITH	800	20
JAMES	950	30
MILLER	1300	10

3 rows returned in 0.03 seconds [Download](#)

Interogarea internă va fi prima executată, producând ca răspuns la cerere trei rînduri: 800, 950, 1300. Blocul cererii externe este apoi procesat și folosește valorile returnate de către interogarea internă pentru a-și finaliza propria condiție de căutare. De fapt, interogarea principală este privită din perspectiva server-ului Oracle astfel:

```
SELECT ename, sal, deptno FROM emp WHERE sal IN (800, 950, 1300);
```

Operatorul ANY (și operatorul SOME, sinonim acestuia) compară o valoare cu fiecare valoare returnată de subinterogare. Exemplul de mai jos afișează angajații ale căror salarii sunt mai mici decât al oricărui angajat cu funcția CLERK și care nu au funcția CLERK. Salariul maxim pe care îl câștigă un angajat cu funcția CLERK este \$1300. Instrucțiunea SQL afișează toți angajații care nu au funcția CLERK, dar câștigă mai puțin de \$1300.

```
SELECT empno, ename, job FROM emp WHERE sal < ANY (SELECT sal  
FROM emp WHERE job = 'CLERK') AND job <> 'CLERK';
```

Autocommit Rows 10 Save Run

```
SELECT empno, ename, job FROM emp WHERE  
sal < ANY (SELECT sal FROM emp WHERE job = 'CLERK') AND job <> 'CLERK';
```

Results Explain Describe Saved SQL History

EMPNO	ENAME	JOB
7521	WARD	SALESMAN
7654	MARTIN	SALESMAN

2 rows returned in 0.01 seconds [Download](#)

Operatorul ALL compară o valoare cu toate valorile returnate de o subinterogare. Exemplul următor afișează toți angajații ale căror salarii sunt mai mari decât salariile medii la nivel de departamente. Cum cel mai mare salariu mediu al vreunui departament este \$2916.6667, rezultă că interogarea va selecta acei angajați ale căror salarii sunt mai mari decât \$2916.6667.

```
SELECT empno, ename, job FROM emp WHERE sal > ALL (SELECT  
avg(sal) FROM emp GROUP BY deptno);
```

Autocommit Rows 10 Save Run

```
SELECT empno, ename, job FROM emp WHERE  
sal > ALL (SELECT avg(sal) FROM emp GROUP BY deptno);
```

Results Explain Describe Saved SQL History

EMPNO	ENAME	JOB
7839	KING	PRESIDENT
7566	JONES	MANAGER
7788	SCOTT	ANALYST
7902	FORD	ANALYST

4 rows returned in 0.02 seconds [Download](#)

Subinterogări de coloane multiple

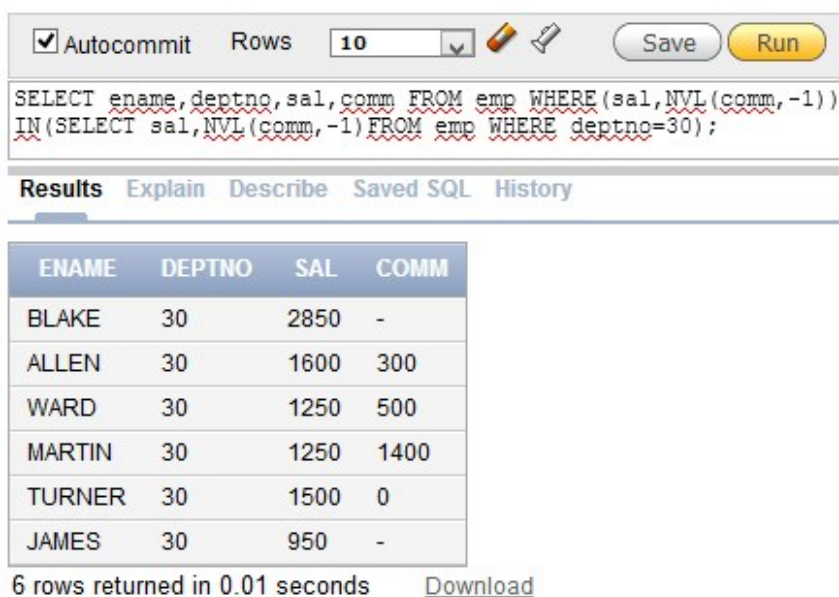
Sintaxa unei subinterogări pe mai multe coloane este:

```
SELECT coloana1, coloana2, ...
FROM tabel
WHERE (coloana1, coloana2, ...) IN
      (SELECT coloana3, coloana4, ...
       FROM tabel
       WHERE conditie);
```

În cazul în care în clauza WHERE se dorește compararea a două sau mai multe coloane, condiția corespunzătoare va trebui formulată o condiție compusă cu ajutorul operatorilor logici. Subinterogările ”multiple-column” oferă posibilitatea de a combina două condiții duplicat de selecție în una singură.

Să presupunem că se cere afișarea numelui, a numărului departamentului, a salariului și a comisionului pentru angajații ale căror salarii și comisioane coincid cu salariul și comisionul oricărui angajat din departamentul 30. Această cerere va fi formulată cu ajutorul următoarei instrucțiuni SELECT:

```
SELECT ename, deptno, sal, comm FROM emp WHERE (sal, NVL(comm, -1))
IN (SELECT sal, NVL(comm, -1) FROM emp WHERE deptno=30);
```



The screenshot shows a SQL query execution interface. At the top, there are controls for 'Autocommit' (checked), 'Rows' (set to 10), and buttons for 'Save' and 'Run'. The query text is: `SELECT ename, deptno, sal, comm FROM emp WHERE (sal, NVL(comm, -1)) IN (SELECT sal, NVL(comm, -1) FROM emp WHERE deptno=30);`. Below the query, there are tabs for 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History'. The 'Results' tab is active, displaying a table with 6 rows and 4 columns: ENAME, DEPTNO, SAL, and COMM. The data rows are: BLAKE (30, 2850, -), ALLEN (30, 1600, 300), WARD (30, 1250, 500), MARTIN (30, 1250, 1400), TURNER (30, 1500, 0), and JAMES (30, 950, -). At the bottom, it says '6 rows returned in 0.01 seconds' and has a 'Download' link.

ENAME	DEPTNO	SAL	COMM
BLAKE	30	2850	-
ALLEN	30	1600	300
WARD	30	1250	500
MARTIN	30	1250	1400
TURNER	30	1500	0
JAMES	30	950	-

În exemplul de mai sus a fost utilizată o subinterogare pe coloane multiple, deoarece subinterogarea returnează două coloane. Condiția din clauza WHERE compară perechea formată din coloanele sal și comm cu perechile returnate de subinterogare.

Comparațiile pe coloane în subinterogările de tip “multiple-row” pot fi clasificate în comparații pereche sau individuale. Exemplul anterior utilizează o comparație pereche în clauza WHERE, deoarece fiecare linie returnată de instrucțiunea SELECT exterioară trebuie să aibă același salariu și comision ca un angajat din departamentul 30. Dacă se dorește utilizarea unei comparații individuale va trebui să folosim o clauză WHERE cu mai multe condiții.

Exemplul următor folosește o comparație individuală pentru a afișa numele, numărul departamentului, salariul și comisionul oricărui angajat al cărui salariu și comision coincid cu salariul și comisionul oricărui angajat din departamentul 30:

```
SELECT ename, deptno, sal, comm FROM emp WHERE sal IN (SELECT sal
FROM emp WHERE deptno=30) AND NVL(comm, -1) IN (SELECT NVL(comm, -1)
FROM emp WHERE deptno=30);
```

The screenshot shows a SQL query execution window with the following SQL statement:

```
SELECT ename,deptno,sal,comm FROM emp WHERE sal
IN(SELECT sal FROM emp WHERE deptno=30) AND NVL(comm,-1)
IN(SELECT NVL(comm,-1) FROM emp WHERE deptno=30);
```

Below the query, there are tabs for Results, Explain, Describe, Saved SQL, and History. The Results tab is active, displaying a table with the following data:

ENAME	DEPTNO	SAL	COMM
BLAKE	30	2850	-
ALLEN	30	1600	300
MARTIN	30	1250	1400
WARD	30	1250	500
TURNER	30	1500	0
JAMES	30	950	-

At the bottom of the results, it says "6 rows returned in 0.01 seconds" and there is a "Download" link.

Deși condițiile din cele două interogări sunt diferite, rezultatele coincid. Această coincidență se datorează datelor conținute de tabelul emp.

Exercițiu: Să presupunem că salariul lui CLARK este de \$1500, iar comisionul său este de \$300. Explicați de ce rezultatele celor două interogări prezentate mai sus diferă (a doua interogare va returna înregistrarea suplimentară `CLARK 10 1500 300`).

Returnarea valorilor null în rezultatul unei subinterogări

```
SELECT employee.ename FROM emp employee WHERE employee.empno NOT
IN(SELECT manager.mgr FROM emp manager);
```

The screenshot shows a SQL query execution window with the following SQL statement:

```
SELECT employee.ename FROM emp employee WHERE employee.empno
NOT IN(SELECT manager.mgr FROM emp manager);
```

Below the query, there are tabs for Results, Explain, Describe, Saved SQL, and History. The Results tab is active, displaying the text "no data found".

Instrucțiunea SQL de mai sus afișează toți angajații care nu au nici un subordonat. Logic, această interogare SQL ar trebui să returneze 8 linii, dar instrucțiunea SQL nu returnează nici una. Una din

valoarea returnata de interogarea interioara este o valoare null si astfel, conditia de selectie din interogare principala nu este adevarata pentru nici o linie din tabelul emp. Acest lucru se explica prin aceea ca toate conditiile care compara o valoare null cu o alta valoare vor returna valoarea null.

Notă: Când există valori null în rezultatul unei subinterogări nu este indicată folosirea operatorului NOT IN, deoarece acesta este echivalent cu != ALL. În astfel de cazuri se poate utiliza operatorul IN.

De exemplu, pentru afişarea angajaţilor care au subordonaţi se poate utiliza următoarea instrucţiune SQL :

```
SELECT employee.ename FROM emp employee WHERE employee.empno IN (SELECT manager.mgr FROM emp manager) ;
```

Autocommit Rows 10 Save Run

```
SELECT employee.ename FROM emp employee WHERE employee.empno IN (SELECT manager.mgr FROM emp manager);
```

Results Explain Describe Saved SQL History

ENAME
KING
JONES
FORD
BLAKE
SCOTT
CLARK

6 rows returned in 0.01 seconds Download

Utilizarea unei subinterogări în clauza FROM

Exemplul urmator afiseaza numele angajatilor, salariile, numarul departamentului si media salariilor din fiecare departament pentru toti angajatii care castiga mai mult decat media salariilor din departamentul in care lucreaza.

```
SELECT a.ename, a.sal, a.deptno, b.salavg FROM emp a, (SELECT deptno, AVG(sal) salavg FROM emp GROUP BY deptno) b WHERE a.deptno = b.deptno AND a.sal > b.salavg;
```

Autocommit Rows 10 Save Run

```
SELECT a.ename, a.sal, a.deptno, b.salavg FROM emp a, (SELECT deptno, AVG(sal) salavg FROM emp GROUP BY deptno) b WHERE a.deptno=b.deptno AND a.sal>b.salavg;
```

Results Explain Describe Saved SQL History

ENAME	SAL	DEPTNO	SALAVG
ALLEN	1600	30	1566.666666666666666666666666666667
BLAKE	2850	30	1566.666666666666666666666666666667
FORD	3000	20	2175
SCOTT	3000	20	2175
JONES	2975	20	2175
KING	5000	10	2916.666666666666666666666666666667

6 rows returned in 0.01 seconds Download

În exemplul anterior instrucțiunea SELECT va opera pe tabelul emp (cu aliasul a) și pe tabelul returnat de interogarea din clauza FROM (cu aliasul b). Tabelul b va avea două coloane (deptno și salavg) și va conține pentru fiecare departament, numărul său și media salariilor.

Probleme:

1. Scrieți o interogare care să afișeze numele, numărul departamentului și salariul oricărui angajat al cărui număr de departament și salariu să se potrivească cu numărul departamentului și salariul oricărui angajat care poate percepe comision.
2. Afișați numele, numele departamentului și salariul oricărui angajat al cărui salariu și comision coincide cu salariul și comisionul oricărui angajat ce lucrează în Dallas.
3. Scrieți o interogare care să afișeze numele, data angajării și salariul pentru toți angajații care au același salariu și comision ca angajatul Scott.
4. Scrieți o interogare care să afișeze angajații care câștigă un salariu mai mare decât al oricărui angajat cu funcția CLERK. Sortați rezultatele descrescător după salariu.
5. Scrieți un fișier script pentru a afișa numele fiecărui angajat, funcția și numele departamentului pentru acei angajați care lucrează în localitatea furnizată la prompt de utilizator. Condiția de căutare a localității trebuie să fie *case-insensitive* (să nu depindă de tipul caracterelor: majuscule sau litere mici). Salvați fișierul script cu numele *p1.sql*.

Indicație: pentru a efectua o căutare case-insensitive se va utiliza funcția LOWER (transformă șirul argument în caractere mici) sau UPPER (transformă șirul argument în majuscule).

6. Modificați *p1.sql* pentru a crea un raport ce conține numele departamentului, numele angajatului, data angajării, salariul și salariul anual pentru toți angajații dintr-o anumită localitate, introdusă de utilizator de la tastatură. Se etichetează coloanele DEPARTMENT NAME, EMPLOYEE NAME, START DATE, SALARY și ANNUAL SALARY, plasând etichetele formate din două șiruri pe mai multe linii. Se salvează fișierul script cu numele *p2.sql*
7. Afișați numărul managerilor (pe baza valorilor din coloana mgr) din tabelul emp. Etichetați coloana "Numar Manageri".
8. Scrieți o interogare care să afișeze diferența dintre salariul cel mai mare și cel mai mic din tabelul emp. Etichetați coloana "Diferența".
9. Afișați numărul fiecărui manager și salariul celui mai prost plătit subaltern al său. Exclueți angajații ce nu au manager. Exclueți grupurile care au salariul minim mai mic decât 1000\$. Sortați rezultatele în ordine descrescătoare după salar.
10. Scrieți o interogare care să afișeze numele fiecărui departament, localitatea, numărul de angajați și salariul mediu al angajaților din acel departament. Etichetați coloanele Dname, Loc, Nr Persoane și Salariu.
11. Să se scrie o interogare care să afișeze numele și data angajării pentru toți angajații din același departament ca Blake (a se exclude Blake).
12. Să se scrie o cerere de interogare pentru a afișa numărul și numele pentru toți angajații care câștigă mai mult decât salariul mediu. Să se sorteze rezultatele în ordinea descrescătoare a salariului.
13. Să se scrie o cerere care va afișa numerele de înregistrare și numele angajaților din departamentele în care lucrează cel puțin un angajat al cărui nume conține litera 'T'.
14. Afișați numele, numărul departamentului și funcția deținută pentru toți angajații al căror departament este situat în Dallas.

15. Afișați numele și salariul tuturor angajaților subordonați lui King.
16. Afișați numărul departamentului, numele și funcția tuturor angajaților din departamentul de vânzări (Sales).
17. Afișați numărul, numele și salariul tuturor angajaților care câștigă mai mult decât salariul mediu și, în plus, lucrează într-un departament care deține cel puțin un angajat al cărui nume conține litera 'T'.