



Universitatea Constantin Brâncuși din Târgu Jiu

Facultatea: Inginerie

Program de conversie profesională a cadrelor didactice din învățământul preuniversitar: Informatică, Tehnologia Informației și a Comunicațiilor

Laborator 6

Limbajul SQL – Subinterogari (Subqueries)

O subinterogare este o comandă SELECT încapsulată într-o clauză a altei instrucțiuni SQL, numită instrucțiune „părinte“. Utilizând subinterogari, se pot construi interogări complexe pe baza unor instrucțiuni simple. Subinterogările mai sunt numite *instrucțiuni SELECT imbricate sau interioare*.

Subinterogarea returnează o valoare care este utilizată de către instrucțiunea „părinte“. *Utilizarea unei subinterogari este echivalentă cu efectuarea a două interogari secvențiale și utilizarea rezultatului interogarii interne ca valoare de căutare în interogarea externă (principală).*

Subinterogările sunt de 2 feluri:

1. **Subinterogări necorelate**, de forma :

```
SELECT lista_select  
FROM nume_tabel  
WHERE expresie_operator (SELECT lista_select  
FROM nume_tabel);
```

- interogarea internă este executată prima și determină o valoare (sau o mulțime de valori);
- interogarea externă se execută o singură dată, utilizând valorile returnate de interogarea internă.

2. **Subinterogări corelate**, de forma:

```
SELECT nume_coloană_1[, nume_coloană_2 ...]  
FROM nume_tabel_1 extern  
WHERE expresie_operator  
      (SELECT nume_coloană_1 [, nume_coloană_2 ...]  
      FROM nume_tabel_2  
      WHERE expresie_1 = extern.expresie_2);
```

- interogarea externă determină o linie candidat;
- interogarea internă este executată utilizând valoarea liniei candidat;
- valorile rezultate din interogarea internă sunt utilizate pentru calificarea sau descalificarea liniei candidat;



- pașii precedenți se repetă până când nu mai există linii candidat.

Observatie: **operator** poate fi:

- 1) **single-row operator** (>, =, >=, <, <>, <=), care poate fi utilizat dacă subinterogarea returnează o singură linie;
- 2) **multiple-row operator** (IN, ANY, ALL), care poate fi folosit dacă subinterogarea returnează mai mult de o linie.
- 3) Operatorul **NOT** poate fi utilizat în combinație cu IN, ANY și ALL.

Exercitii - subinterogari necorelate

- Accesati platforma **ORACLE APEX (Application Express)**, cu datele de conectare cunoscute (a se vedea laboratorul nr. 1).
- Problemele si solutiile prezentate in continuare utilizeaza cateva tabele predefinite:

1) **Tabela EMP** - gestioneaza datele despre angajatii dintr-o firma

2) **Tabela DEPT** - gestioneaza informatii despre departamentele dintr-o firma

1. Folosind subinterogari, să se afișeze numele și data angajării pentru salariații care au fost angajați după angajatul cu numele ALLEN.

```
SELECT ename, hiredate
FROM emp
WHERE hiredate > (SELECT hiredate
                  FROM emp
                  WHERE ename='ALLEN');
```

2. Folosind subinterogari, scrieți o interogare pentru a afișa numele și salariul pentru toți colegii (din același departament) lui ALLEN. Se va exclude ALLEN.

```
SELECT ename, sal
FROM emp
WHERE deptno IN
      (SELECT deptno
       FROM emp
       WHERE UPPER(ename)='ALLEN')
AND UPPER(ename) <> 'ALLEN';
```

Intrebare: Se putea pune "=" în loc de "IN"? In care caz nu se poate face această înlocuire?



3. Folosind subinterogari, să se afișeze numele și salariul angajaților conduși direct de președintele companiei (acesta este considerat angajatul care nu are manager).

```
SELECT ename, sal
FROM emp
WHERE mgr =
      (SELECT empno
       FROM emp
       WHERE mgr IS NULL);
```

4. Scrieti o interogare pentru a afișa numele, codul departamentului și salariul angajaților al caror număr de departament și salariu coincid cu numărul departamentului și salariul unui angajat care castiga comision.

```
SELECT ename, deptno, sal
FROM emp
WHERE (deptno, sal) IN
      (SELECT deptno, sal
       FROM emp
       WHERE comm IS NOT NULL);
```

5. Scrieti o interogare pentru a afișa angajații care castiga mai mult decât oricare functionar (job-ul conține șirul "CLERK"). Sortati rezultatele dupa salariu, in ordine descrescatoare.

```
SELECT ename, sal
FROM emp
WHERE sal > ALL
      ( SELECT sal
        FROM emp
        WHERE LOWER(job) LIKE '%clerk%' )
ORDER BY 2 DESC;
```

Intrebare: Ce rezultat este returnat dacă se înlocuiește "ALL" cu "ANY"?

6. Scrieți o interogare pentru a afișa numele, numele departamentului și salariul angajaților care nu câștigă comision, dar al căror șef direct coincide cu șeful unui angajat care câștigă comision.

```
SELECT e.ename, d.dname, e.sal
FROM emp e, dept d
WHERE e.deptno = d.deptno
AND comm IS NULL
AND e.mgr IN
      ( SELECT mgr
        FROM emp
        WHERE comm IS NOT NULL );
```



7. Sa se afiseze numele salariatilor al caror salariu este mai mare decat salariul lui JONES.

```
SELECT ename AS "Nume"  
FROM emp  
WHERE sal >  
      ( SELECT sal  
        FROM emp  
        WHERE ename = 'JONES');
```

8. Sa se afiseze numele si prenumele celui mai recent angajat din departamentui 30.

```
SELECT ename, hiredate  
FROM emp  
WHERE hiredate = (  
      SELECT max(hiredate)  
      FROM emp  
      WHERE deptno = 30)  
AND deptno = 30;
```

9. Sa se afiseze numarul si salariul mediu final (la care se adauga comision), cu 3 zecimale, al angajatilor din departamentui SALES, (se vor folosi tabelele EMP si DEPT, iar coloanele se vor numi **numar angajati**, respectiv **salariu mediu angajati**).

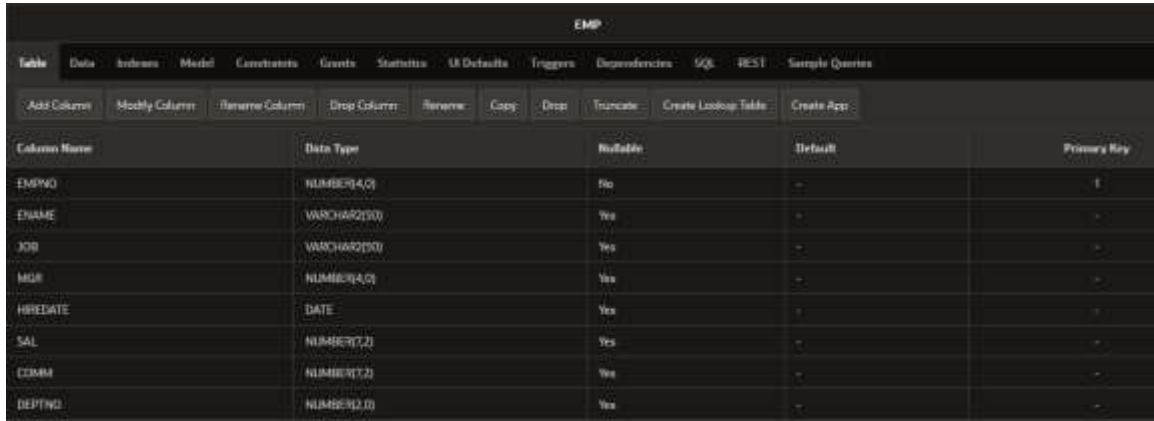
```
SELECT count(*) as "Numar de angajati", round (avg( sal*sal*nvl(c  
omm,0)), 3 ) as "Salariu mediu final"  
FROM emp  
WHERE deptno = (  
      SELECT deptno  
      FROM dept  
      WHERE dname='SALES' );
```

10. Sa se afiseze id-ul si numele departamentelor al caror nume incepe si se termina cu aceeasi litera nu se face deosebire intre litere mari si mici. (se va folosi tabela DEPT)

```
SELECT deptno, dname  
FROM DEPT  
WHERE UPPER(SUBSTR(dname,1,1)) = SUBSTR(dname,-1,1);
```

Probleme propuse spre rezolvare

Se considera tabela **EMP** avand urmatoarea structura:



Column Name	Data Type	Nullable	Default	Primary Key
EMPNO	NUMBER(4,0)	No	-	1
ENAME	VARCHAR2(50)	Yes	-	-
JOB	VARCHAR2(50)	Yes	-	-
MGR	NUMBER(4,0)	Yes	-	-
HIREDATE	DATE	Yes	-	-
SAL	NUMBER(7,2)	Yes	-	-
COMM	NUMBER(7,2)	Yes	-	-
DEPTNO	NUMBER(2,0)	Yes	-	-

Column Name	Data Type	Nullable	Default	Primary Key
EMPNO	NUMBER(4,0)	No		1
ENAME	VARCHAR2(50)	Yes		
JOB	VARCHAR2(50)	Yes		
MGR	NUMBER(4,0)	Yes		
HIREDATE	DATE	Yes		
SAL	NUMBER(7,2)	Yes		
COMM	NUMBER(7,2)	Yes		
DEPTNO	NUMBER(2,0)	Yes		

si avand urmatoarele valori introduse:

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT		11/17/1981	5000		10
7698	BLAKE	MANAGER	7839	5/1/1981	2850		30
7782	CLARK	MANAGER	7839	6/9/1981	2450		10
7566	JONES	MANAGER	7839	4/2/1981	2975		20
7788	SCOTT	ANALYST	7566	12/9/1982	3000		20
7902	FORD	ANALYST	7566	12/3/1981	3000		20
7369	SMITH	CLERK	7902	12/17/1980	800		20
7499	ALLEN	SALESMAN	7698	2/20/1981	1600	300	30
7521	WARD	SALESMAN	7698	2/22/1981	1250	500	30
7654	MARTIN	SALESMAN	7698	9/28/1981	1250	1400	30
7844	TURNER	SALESMAN	7698	9/8/1981	1500	0	30
7876	ADAMS	CLERK	7788	1/12/1983	1100		20
7900	JAMES	CLERK	7698	12/3/1981	950		30
7934	MILLER	CLERK	7782	1/23/1982	1300		10



EMP_ID	EMPNO	ENAME	JOB	MANAGER	START_DATE	SAL	COMMISSION	DEPTNO	REPORTING_MANAGER
101	7839	BLAKE	MANAGER	7838	15/08/1981	2900	14	10	7838
102	7840	CLERK	MANAGER	7839	15/08/1981	1900	14	10	7839
103	7841	ALLEN	SALESMAN	7839	17/06/1981	1600	14	30	7839
104	7842	WARD	SALESMAN	7839	17/06/1981	1200	14	30	7839
105	7843	MARTIN	SALESMAN	7839	28/09/1981	1200	14	30	7839
106	7844	MANZONI	SALESMAN	7839	21/07/1981	1100	14	30	7839
107	7845	FRANK	SALESMAN	7839	17/12/1981	800	14	30	7839
108	7846	MACKEY	SALESMAN	7839	01/09/1981	1200	14	30	7839
109	7847	BURNETT	SALESMAN	7839	24/03/1981	1100	14	30	7839
110	7848	DEWETT	SALESMAN	7839	08/07/1981	1300	14	30	7839
111	7849	TURNER	SALESMAN	7839	08/09/1981	1100	14	30	7839
112	7850	AMBROSE	SALESMAN	7839	09/09/1981	1300	14	30	7839
113	7851	PEWEE	SALESMAN	7839	07/04/1981	800	14	30	7839
114	7852	FERRELL	SALESMAN	7839	14/08/1981	1300	14	30	7839
115	7853	BLAKE	MANAGER	7838	15/08/1981	2900	14	10	7838
116	7854	CLERK	MANAGER	7839	15/08/1981	1900	14	10	7839
117	7855	ALLEN	SALESMAN	7839	17/06/1981	1600	14	30	7839
118	7856	WARD	SALESMAN	7839	17/06/1981	1200	14	30	7839
119	7857	MARTIN	SALESMAN	7839	28/09/1981	1200	14	30	7839
120	7858	MANZONI	SALESMAN	7839	21/07/1981	1100	14	30	7839
121	7859	FRANK	SALESMAN	7839	17/12/1981	800	14	30	7839
122	7860	MACKEY	SALESMAN	7839	01/09/1981	1200	14	30	7839
123	7861	BURNETT	SALESMAN	7839	24/03/1981	1100	14	30	7839
124	7862	DEWETT	SALESMAN	7839	08/07/1981	1300	14	30	7839
125	7863	TURNER	SALESMAN	7839	08/09/1981	1100	14	30	7839
126	7864	AMBROSE	SALESMAN	7839	09/09/1981	1300	14	30	7839
127	7865	PEWEE	SALESMAN	7839	07/04/1981	800	14	30	7839
128	7866	FERRELL	SALESMAN	7839	14/08/1981	1300	14	30	7839

Lab6_1) Scrieti o interogare pentru a afisa angajatii care castiga mai mult decat oricare functionar (job-ul conține șirul “CLERK”). Sortati rezultatele dupa salariu, in ordine descrescatoare. (ALL).

Lab6_2) Ce rezultat este returnat dacă se înlocuiește “ALL” cu “ANY”?

- ALL - compara cu toate liniile din subcerere
- afisarea angajatilor care castiga mai mult decat cel puțin unul dintre cei cu clerk

Lab6_3) Să se afiseze codul si numele angajatilor care câștiga mai mult decât salariul mediu din firma. Se va sorta rezultatul în ordine descrescătoare a salariilor.

Lab6_4) Să se afiseze salariatii care au fost angajati în aceeași zi a lunii în care cei mai multi dintre salariați deja angajati.

Lab6_5) Să se afiseze numele si salariul celor mai prost platiti angajati din fiecare departament (se cer 2 solutii: subinterogare sincronizata, subinterogare nesincronizata).

Lab6_6) Să se afiseze numele si salariul tuturor angajatilor care au job-ul ca manager pe cel care are valoarea NULL in campul MGR.

Lab6_7) Să se afiseze numele, departamentul su salarial angajatilor care au comisió.

Lab6_8) Să se obțină numele primilor 3 angajati avand salariul maxim. Rezultatul se va afisa în ordine crescătoare a salariilor.

Se cer doua solutii:

- a) Să se afiseze primii trei angajati, cu sau fără duplicate
- b) primele trei linii după ce a fost ordonat
- c) sau primele trei salarii



Lab6_9) Să se afișeze codul, numele, data angajării, salariul și managerul pentru subalternii direcți ai lui BLAKE:

Lab6_10) Să se afișeze numele și salariul angajaților al căror salariu este mai mare decât salariile medii din toate departamentele. Se cer 2 variante de rezolvare: cu operatorul ALL sau cu funcția MAX.

Operatorul EXISTS

- ✓ În instrucțiunile **SELECT** imbricate, este permisă utilizarea oricărui operator logic.
- ✓ Pentru a testa dacă valoarea recuperată de interogarea externă există în mulțimea valorilor regăsite de interogarea internă corelată, se poate utiliza operatorul **EXISTS**. Dacă subinterogarea returnează cel puțin o linie, operatorul returnează valoarea **TRUE**. În caz contrar, va fi returnată valoarea **FALSE**.
- ✓ Operatorul **EXISTS** asigură că nu mai este continuată căutarea în interogarea internă după ce aceasta regăsește o linie.

Lab6_11) Să se obțină numele salariaților care lucrează într-un departament în care există cel puțin un angajat cu salariul egal cu salariul maxim din departamentul 30 (operatorul **EXISTS**).

Obs: Deoarece nu este necesar ca instrucțiunea **SELECT** interioară să returneze o anumită valoare, se poate selecta o constantă ('x', ' ', 1, etc.). De altfel, din punct de vedere al performanței, selectarea unei constante asigură mai multă rapiditate decât selectarea unei coloane.

Lab6_12) Să se afișeze codul, numele și job-ul angajaților care au cel puțin doi subalterni.

Lab6_13) Să se selecteze angajații care nu s-au angajat în lunile decembrie, ianuarie și februarie.

Bibliografie:

1. <https://ocw.cs.pub.ro/courses/bd/laboratoare/08>
2. <https://ocw.cs.pub.ro/courses/bd/laboratoare/09>
3. <https://www.w3resource.com/sql-exercises/sql-subqueries-exercises.php>
4. <https://www.mysqltutorial.org/mysql-subquery/>
5. <https://www.mysqltutorial.org/mysql-exists/>