Baze de date - Anul 1

Laborator 6

Subcereri nesincronizate (necorelate). Gruparea datelor (1)

I. [Obiective]

Prezentarea conceptului de subcerere. Unde pot fi folosite subcererile? Clasificarea subcererilor.

Introducere in gruparea datelor.

II. [Subcereri]

O **subcerere** este o comandă *SELECT* încapsulată într-o clauză a altei instrucţiuni *SQL*, numită instrucţiune „părinte“. Utilizând subcereri, se pot construi interogări complexe pe baza unor instrucţiuni simple. Subcererile mai sunt numite instrucţiuni *SELECT* imbricate sau interioare.

Subcererea returnează o valoare care este utilizată de către instrucţiunea „părinte“. Utilizarea unei subcereri este echivalentă cu efectuarea a două cereri secvenţiale şi utilizarea rezultatului cererii interne ca valoare de căutare în cererea externă (principală).

În general, subcererile pot aparea în clauzele *SELECT*, *FROM*, *WHERE*, *HAVING* ale comenzii *SELECT*.

Subcererile sunt de 2 tipuri :

- **Necorelate (nesincronizate)**, de forma :

SELECT *lista_select*
FROM *nume_tabel*
WHERE *expresie operator* (***SELECT*** *lista_select*
FROM *nume_tabel*);

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

- **Corelate (sincronizate)**, 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*);

- cererea externă determină o linie candidat;
- cererea internă este executată utilizând valoarea liniei candidat;
- valorile rezultate din cererea internă sunt utilizate pentru calificarea sau descalificarea liniei candidat;
- paşii precedenţi se repetă până când nu mai există linii candidat.

Observaţiie: *operator* poate fi:

- *single-row operator* (>, =, >=, <, <>, <=), care poate fi utilizat dacă subcererea returnează o singură linie;
- *multiple-row operator* (*IN*, *ANY*, *ALL*), care poate fi folosit dacă subcererea returnează mai mult de o linie.
  Operatorul *NOT* poate fi utilizat în combinaţie cu *IN*, *ANY* şi *ALL*.

III. [Funcţii grup şi clauza GROUP BY]

Clauza *GROUP BY* este utilizată pentru **a diviza liniile unui tabel în grupuri**. Pentru a returna informaţia corespunzătoare fiecărui astfel de grup, pot fi utilizate funcţiile agregat. Aceste funcţii pot apărea în clauzele:

- *SELECT*
- *ORDER BY*
- *HAVING*.

Server-ul *Oracle* aplică aceste funcţii fiecărui grup de linii şi returnează **un singur rezultat** pentru fiecare mulţime.

Dintre funcţiile grup definite în sistemul *Oracle*, se pot enumera: *AVG*, *SUM*, *MAX*, *MIN*, *COUNT*, *STDDEV*, *VARIANCE* etc. Tipurile de date ale argumentelor funcţiilor grup pot fi *CHAR*, *VARCHAR2*, *NUMBER* sau *DATE*.

- Funcţiile *AVG*, *SUM*, *STDDEV* şi *VARIANCE* operează numai asupra valorilor numerice.
- Funcţiile *MAX* şi *MIN* pot opera asupra valorilor numerice, caracter sau dată calendaristică.
- Absenţa clauzei GROUP BY conduce la aplicarea funcţiei grup pe mulţimea tuturor liniilor tabelului.
- Toate funcţiile grup, cu excepţia lui *COUNT*(*), ignoră valorile *null*. *COUNT*(*expresie*) returnează numărul de linii pentru care expresia dată nu are valoarea *null*. Funcţia *COUNT* returnează un număr mai mare sau egal cu zero şi nu întoarce niciodată valoarea *null*.

Expresiile din clauza *SELECT* a unei cereri care conţine opţiunea *GROUP BY* trebuie să reprezinte **o proprietate unică de grup**, adică fie un atribut de grupare, fie o funcţie de agregare aplicată tuplurilor unui grup, fie o expresie formată pe baza primelor două. Toate expresiile din clauza *SELECT*, cu excepţia funcţiilor de agregare, se trec în clauza *GROUP BY* (unde pot apărea cel mult 255 expresii).

IV. [Clauza HAVING]

- Clauza *HAVING* a comenzii *SELECT* permite restricţionarea grupurilor de linii returnate, la cele care îndeplinesc o anumită condiţie.
- Dacă această clauză este folosită în absenţa unei clauze *GROUP BY*, aceasta presupune că **gruparea se aplică întregului tabel**, deci este returnată o singură linie, care este reţinută în rezultat doar dacă este îndeplinită condiţia din clauza *HAVING*.

V. [Exercitii - subcereri necorelate]

1. Folosind subcereri, să se afişeze numele şi data angajării pentru salariaţii care au fost angajaţi după Gates.

```
SELECT last_name, hire_date
FROM     employees
WHERE  hire_date > (SELECT hire_date
                    FROM     employees
                    WHERE  INITCAP(last_name)='Gates');
```

2. Folosind subcereri, scrieţi o cerere pentru a afişa numele şi salariul pentru toţi colegii (din acelaşi departament) lui Gates. Se va exclude Gates.

```
SELECT last_name, salary
FROM     employees
WHERE   department_id IN (SELECT department_id
                          FROM     employees
                          WHERE  LOWER(last_name)='gates')
AND      LOWER(last_name) <> 'gates';
```

În care caz nu se poate înlocui "=" cu "IN"??

3. Folosind subcereri, să se afişeze numele şi salariul angajaţilor conduşi direct de preşedintele companiei (acesta este considerat angajatul care nu are manager).

4. Scrieți o cerere pentru a afișa numele, codul departamentului și salariul angajaților al căror cod de departament și salariu coincid cu codul departamentului și salariul unui angajat care câștigă comision.

Soluție:

```
SELECT last_name, department_id, salary
```

```
FROM employees
WHERE (department_id, salary) IN
    (
     SELECT department_id, salary
     FROM employees
     WHERE commission_pct IS NOT NULL
    );
```

5. Rezolvaţi problema 6 din laboratorul precedent utilizând subcereri: să se afișeze codul, numele și salariul tuturor angajaților care caștigă mai mult decât salariul mediu pentru job-ul corespunzător și lucrează într-un departament cu cel puțin unul dintre angajații al căror nume conține litera “t”. Vom considera salariul mediu al unui job ca fiind egal cu media aritmetică a limitelor sale admise (specificate în coloanele *min_salary*, *max_salary* din tabelul JOBS).

Soluție:

```
SELECT e.employee_id,e.last_name,e.salary
FROM employees e
WHERE e.salary >
      (
       SELECT (j.min_salary+j.max_salary)/2
       FROM jobs j
       WHERE j.job_id=e.job_id
      )
AND e.job_id IN
      (
       SELECT job_id
       FROM employees m
       WHERE e.department_id=m.department_id
       AND LOWER(m.last_name) LIKE '%t%'
      );
```

6. Scrieti o cerere pentru a afișa angajații care câștigă mai mult decât oricare funcționar (job-ul conţine şirul “CLERK”). Sortați rezultatele dupa salariu, în ordine descrescătoare.

Ce rezultat este returnat dacă se înlocuieşte “ALL” cu “ANY”?

Soluție:

```
SELECT *
FROM employees e
WHERE salary > ALL
      (
       SELECT salary
       FROM employees
       WHERE job_id LIKE '%CLERK'
      );
```

7. Scrieţi o cerere pentru a afişa numele, numele departamentului şi salariul angajaţilor care nu câştigă comision, dar al căror şef direct câştigă comision.

8. Să se afișeze numele, departamentul, salariul şi job-ul tuturor angajaților al căror salariu și comision coincid cu salariul și comisionul unui angajat din Oxford.

9. Să se afişeze numele angajaţilor, codul departamentului şi codul job-ului salariaţilor al căror departament se află în Toronto.

VI. [Exercitii – gruparea datelor]

10. a) Functiile grup includ valorile *NULL* in calcule?
b) Care este deosebirea dintre clauzele *WHERE* şi *HAVING*?

11. Să se afişeze cel mai mare salariu, cel mai mic salariu, suma şi media salariilor tuturor angajaţilor. Etichetaţi coloanele Maxim, Minim, Suma, respectiv Media. Să se rotunjească rezultatele.

12. Să se afişeze minimul, maximul, suma şi media salariilor pentru fiecare job.

13. Să se afişeze numărul de angajaţi pentru fiecare job.

14. Să se determine numărul de angajaţi care sunt şefi. Etichetați coloana “Nr. manageri”.
Observație: Este necesar cuvântul cheie DISTINCT. Ce obţinem dacă îl omitem?

15. Să se afişeze diferenţa dintre cel mai mare și cel mai mic salariu mediu pe departamente. Etichetați coloana “Diferenta”.

16. Scrieţi o cerere pentru a se afişa numele departamentului, locaţia, numărul de angajaţi şi salariul mediu pentru angajaţii din acel departament. Coloanele vor fi etichetate corespunzător.
Observație: În clauza *GROUP BY* se trec obligatoriu toate coloanele prezente în clauza *SELECT*, care nu sunt argument al funcţiilor grup (a se vedea ultima observaţie de la punctul I).

17. Să se afişeze codul şi numele angajaţilor care câştigă mai mult decât salariul mediu din firmă. Se va sorta rezultatul în ordine descrescătoare a salariilor.

18. Pentru fiecare şef, să se afişeze codul său şi salariul celui mai puțin platit subordonat al său. Se vor exclude cei pentru care codul managerului nu este cunoscut. De asemenea, se vor exclude grupurile în care salariul minim este mai mic de 1000$. Sortaţi rezultatul în ordine descrescătoare a salariilor.

19. Pentru departamentele in care salariul maxim depăşeşte 3000$, să se obţină codul, numele acestor departamente şi salariul maxim pe departament.

20. Care este salariul mediu minim al job-urilor existente? Salariul mediu al unui job va fi considerat drept media arirmetică a salariilor celor care îl practică.

21. Să se afişeze codul, numele departamentului şi suma salariilor pe departamente.

22. Să se afişeze maximul salariilor medii pe departamente.

23. Să se obțină codul, titlul şi salariul mediu al job-ului pentru care salariul mediu este minim.

24. Să se afişeze salariul mediu din firmă doar dacă acesta este mai mare decât 2500. (clauza *HAVING* fără *GROUP BY*)