

Gruparea 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;

Funcții grup

Funcțiile grup operează pe mulțimi de înregistrări și furnizează un singur rezultat pentru întregul grup. Exemplul de mai jos afișează salariul maxim din tabelul `emp`.

<u>DEPTNO</u>	<u>SAL</u>
10	2450
10	5000
10	1300
20	800
20	1100
20	3000
20	3000
20	2975
30	1600
30	2850
30	1250
30	950
30	1500
30	1250

” salariul maxim în tabelul EMP ”

<u>MAX(SAL)</u>
5000

În tabelul următor sunt descrise funcțiile grup ce pot fi utilizate în clauza `SELECT`. Parametrii implicați apar subliniați.

Funcție	Descriere
AVG([DISTINCT ALL] n)	Returnează valoarea medie a lui <i>n</i> , ignorând valorile null;
COUNT({*[DISTINCT ALL] expr})	Returnează numărul de rânduri selectate, unde <i>expr</i> evaluează altceva decât valori null. Pentru a număra toate rândurile selectate, inclusiv rândurile duplicate și cele cu valori null, se va folosi '*';
MAX([DISTINCT ALL]expr)	Returnează valoarea maximă pentru <i>expr</i> , ignorând valorile null;
MIN([DISTINCT ALL] expr)	Returnează valoarea minimă pentru <i>expr</i> , ignorând valorile null;
STDDEV([DISTINCT ALL] n)	Calculează abaterea standard a lui <i>n</i> , ignorând valorile null;
SUM([DISTINCT ALL] n)	Calculează suma valorilor lui <i>n</i> , ignorând valorile null;
VARIANCE([DISTINCT ALL] n)	Calculează varianța lui <i>n</i> , ignorând valorile null;

Folosirea funcțiilor grup



```
SELECT coloana, functie_grup(coloana)
FROM tabel
[WHERE conditie]
[ORDER BY coloana];
```

- DISTINCT implică luarea în considerare numai a valorilor neduplicate. Argumentul ALL ia în considerare fiecare valoare, inclusiv valorile duplicate; este opțiunea implicită.
- tipurile de date admise pentru *expr* sunt: **CHAR**, **VARCHAR2**, **NUMBER** sau **DATE**.
- toate grupurile de funcții, cu excepția funcției **COUNT** (*) ignoră valorile null. Pentru a înlocui o valoare null cu o altă valoare se poate utiliza funcția **NVL** (e.g. funcția **NVL (COMM, 0)** va returna 0 dacă **COMM** are valoarea null și valoarea **COMM** dacă aceasta e diferită de null).

1. Funcțiile AVG, SUM, MIN și MAX

Funcțiile AVG și SUM pot fi aplicate doar coloanelor ce conțin date numerice, în timp ce funcțiile MIN și MAX operează asupra oricărui tip de date. Exemplul următor afișează media salariilor lunare, salariul maxim, respectiv minim și suma salariilor lunare pentru toți angajații cu funcția **MANAGER**.

```
SELECT AVG(SAL), MAX(sal), MIN(sal), SUM(sal) FROM emp WHERE job LIKE 'MANAGER';
```

Autocommit Rows  

```
SELECT AVG(sal), MAX(sal), MIN(sal), SUM(sal) FROM emp WHERE job LIKE 'MANAGER';
```



Results Explain Describe Saved SQL History

AVG(SAL)	MAX(SAL)	MIN(SAL)	SUM(SAL)
2758.333333333333333333333333333333	2975	2450	8275

1 rows returned in 0.07 seconds [Download](#)

Pentru a afișa numele primului, respectiv ultimului angajat din lista ordonată alfabetic a tuturor angajaților se va executa următoarea interogare:

```
SELECT MIN(ename), MAX(ename) FROM emp;
```

Autocommit Rows  

```
SELECT MAX(ename), MIN(ename) FROM emp;
```

Results Explain Describe Saved SQL History

MAX(ENAME)	MIN(ENAME)
WARD	ADAMS

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

Notă : Funcțiile AVG, SUM, VARIANCE și STDDEV pot fi utilizate doar cu tipuri de date numerice.



2. Funcția COUNT

Funcția COUNT are două formate:

- COUNT (*) și
- COUNT (expr)

COUNT (*) returnează numărul de rânduri din interogare, inclusiv liniile duplicat și cele conținând valori null, în timp ce COUNT (expr) returnează numărul valorilor diferite de null din coloana identificată prin expr. Exemplul de mai jos afișează numărul angajaților din departamentul 30.

```
SELECT COUNT(*) FROM emp WHERE deptno = 30;
```

Autocommit Rows  

```
SELECT COUNT(*) FROM emp WHERE deptno=30;
```

Results Explain Describe Saved SQL History

COUNT(*)
6

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

Clauza GROUP BY

EMP

DEPTNO	SAL	
10	2450	
10	5000	2916.6667
10	1300	
20	800	
20	1100	
20	3000	2175
20	3000	
20	2975	
30	1600	
30	2850	
30	1250	1566.6667
30	950	
30	1500	
30	1250	

” salariul mediu în tabelul scott.emp pentru fiecare departament “

DEPTNO	AVG (SAL)
10	2916.6667
20	2175
30	66.6667

Până acum, toate funcțiile grup au tratat tabelul ca fiind un grup larg de informații. Pentru împărțirea tabelului în grupuri mai mici se poate utiliza clauza GROUP BY.

```
SELECT coloana, functie_grup(coloana)
FROM tabel
[WHERE conditie]
[GROUP BY expresie_de_grupare]
[ORDER BY coloana];
```

unde:

expresie_de_grupare specifică acele coloane ale căror valori determină criteriul de grupare a liniilor.

Clauza **GROUP BY** determină împărțirea rândurilor din tabel în grupuri. În cazul utilizării unei funcții grup în conjuncție cu clauza **GROUP BY** se va returna rezultatul funcției grup pentru fiecare grup.

Observatii:

- dacă se include o funcție grup într-o clauză **SELECT**, lista coloanelor individuale (coloanele care nu apar în funcția grup) **trebuie** să apară în clauza **GROUP BY**. În caz contrar se va genera un mesaj de eroare.
- utilizarea unei clauze **WHERE** exclude rândurile care nu satisfac condiția impusă înainte de a se efectua divizarea în grupuri;
- în clauza **GROUP BY** trebuie incluse *coloane*; este interzisă utilizarea unui alias de coloană în clauza **GROUP BY**;

- rândurile sunt sortate implicit în ordinea ascendentă a coloanelor incluse în lista **GROUP BY**. Este posibilă încălcarea acestei reguli folosind clauza **ORDER BY**.

Următorul exemplu afișează numărul fiecărui departament împreună cu media salariilor pe acel departament. Modul de evaluare al instrucțiunii **SELECT** conținând o clauză **GROUP BY** este următorul:

- clauza **SELECT** specifică acele coloane ce urmează să fie afișate:
 - coloana *deptno* corespunzătoare numărului departamentului
 - media salariilor din grupul specificat în clauza **GROUP BY**
- clauza **FROM** indică tabelul ce trebuie accesat - *emp*.
- clauza **WHERE** specifică rândurile ce vor fi selectate. Dacă nu există clauză **WHERE** vor fi luate în considerare toate rândurile.
- clauza **GROUP BY** specifică modul de grupare al rândurilor. Rândurile sunt grupate după numărul departamentului, deci funcția AVG (aplicată coloanei *sal*) va calcula *media salariilor pentru fiecare departament*.

SELECT deptno,AVG(sal) FROM emp GROUP BY deptno ;

Autocommit Rows 10 Save Run

```
SELECT deptno, AVG(sal) FROM emp GROUP BY deptno;
```

Results Explain Describe Saved SQL History

DEPTNO	AVG(SAL)
30	1566.6666666666666666666666666667
20	2175
10	2916.6666666666666666666666666667

3 rows returned in 0.10 seconds Download

Notă: Coloana ce apare în clauza **GROUP BY** nu trebuie să apară obligatoriu în clauza **SELECT**. De exemplu, instrucțiunea **SELECT** de mai jos afișează media salariilor pentru fiecare departament, fără să afișeze numărul departamentului respectiv. Este permisă utilizarea unei funcții grup în clauza **ORDER BY**.

SELECT AVG(sal) FROM emp GROUP BY deptno ORDER BY AVG(sal) ;

Autocommit Rows 10 Save Run

```
SELECT AVG(sal) FROM emp GROUP BY deptno ORDER BY AVG(sal);
```

Results Explain Describe Saved SQL History

AVG(SAL)
1566.6666666666666666666666666667
2175
2916.6666666666666666666666666667

3 rows returned in 0.06 seconds Download

Gruparea după mai multe coloane

EMP

DEPTNO	JOB	SAL
10	MANAGER	2450
10	PRESIDENT	5000
10	CLERK	1300
20	CLERK	800
20	CLERK	1100
20	ANALYST	3000
20	ANALYST	3000
20	MANAGER	2975
30	SALESMAN	1600
30	MANAGER	2850
30	SALESMAN	1250
30	CLERK	950
30	SALESMAN	1500
30	SALESMAN	1250

” suma salariilor din tabelul scott.emp pentru fiecare funcție, grupate după departament”

DEPTNO	JOB	SUM(SAL)
10	CLERK	1300
10	MANAGER	2450
10	PRESIDENT	5000
20	ANALYST	6000
20	CLERK	1900
20	MANAGER	2975
30	CLERK	950
30	MANAGER	2850
30	SALESMAN	5600

Unele situații cer afișarea unor rezultate de sinteză pentru grupurile din interiorul altor grupuri. Figura de mai sus reprezintă totalul salariilor plătite pentru fiecare funcție, în cadrul fiecărui departament. Tabelul emp este grupat mai întâi după numărul departamentului și apoi, în fiecare grup obținut se realizează o grupare după numele funcției. De exemplu, cei doi angajați cu funcția CLERK din departamentul 20 sunt grupați împreună și este returnat un singur rezultat (salariul total) pentru toți angajații din acest grup.

Se pot returna rezultate sinteză pentru grupuri și subgrupuri prin specificarea mai multor coloane în clauza GROUP BY. Ordinea implicită a rezultatelor afișate poate fi controlată prin ordinea în care apar coloanele în clauza GROUP BY.

```
SELECT deptno, job, sum(sal) FROM emp GROUP BY deptno, job;
```

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'. Below this is the SQL query: `SELECT deptno, job, SUM(sal) FROM emp GROUP BY deptno, job;`. Underneath the query, there are tabs for 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History'. The 'Results' tab is active, displaying a table with 9 rows. The table has columns 'DEPTNO', 'JOB', and 'SUM(SAL)'. The data is as follows:

DEPTNO	JOB	SUM(SAL)
20	MANAGER	2975
20	CLERK	1900
30	SALESMAN	5600
30	CLERK	950
10	PRESIDENT	5000
30	MANAGER	2850
10	CLERK	1300
10	MANAGER	2450
20	ANALYST	6000

At the bottom of the interface, it says '9 rows returned in 0.10 seconds' and there is a 'Download' button.

EMP

DEPTNO	SAL	
10	2450	
10	5000	5000
10	1300	
20	800	
20	1100	
20	3000	3000
20	3000	
20	2975	
30	1600	
30	2850	
30	1250	2850
30	950	
30	1500	
30	1250	

” salariul maxim pe departament mai mare de 2900 \$ “

DEPTNO	AVG (SAL)
10	5000
20	3000

Pentru a păstra în rezultat doar acele grupuri ce îndeplinesc o condiție se va utiliza clauza **HAVING**.

```
SELECT coloana, functie_grup
FROM tabel
[WHERE conditie ]
[GROUP BY expresie_de_grupare]
[HAVING conditie_grup]
[ORDER BY coloana];
```

unde:

conditie_grup restricționează grupurile de rânduri la acele grupuri ce satisfac condiția specificată.

Serverul Oracle parcurge următorii pași când întâlnește o clauză **HAVING** :

- grupează rândurile
- aplică funcția grup fiecărui grup
- afișează doar grupurile care îndeplinesc criteriul din clauza **HAVING**.

Exemplul următor afișează numărul departamentului și salariul maxim pentru acele departamente în care salariul maxim este mai mare de \$2900.

```
SELECT deptno,max(sal) FROM emp GROUP BY deptno HAVING
max(sal)>2900 ;
```

The screenshot shows the Oracle SQL Developer interface. At the top, there are controls for Autocommit (checked), Rows (set to 10), and buttons for Save and Run. Below this is the SQL query: `SELECT deptno,MAX(sal) FROM emp GROUP BY deptno HAVING MAX(sal)>2900;`. Underneath the query are tabs for Results, Explain, Describe, Saved SQL, and History. The Results tab is active, displaying a table with two rows: (20, 3000) and (10, 5000). At the bottom, it indicates "2 rows returned in 0.05 seconds" and a Download button.

Probleme:

1. Afișați numărul managerilor (pe baza valorilor din coloana mgr) din tabelul emp. Etichetați coloana "Numar Manageri".
2. Scrieți o interogare care să afișeze diferența dintre salariul cel mai mare și cel mai mic din tabelul emp. Etichetați coloana "Diferența".
3. 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.
4. 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.
5. Să se scrie o interogare care să afișeze numele și data angajării pentru toți angajații din același departament ca Blake (a se exclude Blake).
6. 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.
7. 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'.
8. Afișați numele, numărul departamentului și funcția deținută pentru toți angajații al căror departament este situat în Dallas.
9. Afișați numele și salariul tuturor angajaților subordonați lui King.
10. Afișați numărul departamentului, numele și funcția tuturor angajaților din departamentul de vânzări (Sales).
11. 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'.