

Capitolul 3

FUNCTII STATISTICE SI GRUPURI

STUD

MATR	NUME	AN	GRUPA	DATAN	LOC	TUTOR	PUNCTAJ	CODS
----	-----	--	-----	-----	-----	-----	-----	----
1456	GEORGE	4	1141A	12-MAR-82	BUCURESTI		2890	11
1325	VASILE	2	1122A	05-OCT-84	PITESTI	1456	390	11
1645	MARIA	3	1131B	17-JUN-83	PLOIESTI		1400	11
3145	ION	1	2112B	24-JAN-85	PLOIESTI	3251	1670	21
2146	STANCA	4	2141A	15-MAY-82	BUCURESTI		620	21
3251	ALEX	5	2153B	07-NOV-81	BRASOV		1570	21
2215	ELENA	2	2122A	29-AUG-84	BUCURESTI	2146	890	21
4311	ADRIAN	3	2431A	31-JUL-83	BUCURESTI		450	24
3514	FLOREA	5	2452B	03-FEB-81	BRASOV		3230	24
1925	OANA	2	2421A	20-DEC-84	BUCURESTI	4311	760	24
2101	MARIUS	1	2412B	02-SEP-85	PITESTI	3514	310	24
4705	VOICU	2	2421B	19-APR-84	BRASOV	4311	1290	24

SPEC si BURSA

CODS	NUME	DOMENIU			
11	MATEMATICA	STIINTE EXACTE			
21	GEOGRAFIE	UMANIST			
24	ISTORIE	UMANIST			
TIP			PMIN	PMAX	SUMA
FARA BURSA			0	399	
BURSA SOCIALA			400	899	100
BURSA DE STUDIU			900	1799	150
BURSA DE MERIT			1800	2499	200
BURSA DE EXCEPTIE			2500	9999	300

OBIECTIV

- ◆ Până acum fiecare linie a rezultatului era calculată **dintr-o linie** a unei tabele din baza de date sau a produsului cartezian al unor tabele.
- ◆ În unele cazuri este însă necesar calculul unor **valori statistice** pornind de la toate liniile parcurgerii curente sau ale unor **grupuri de linii** care au aceleași valori pentru o listă de expresii.

OBIECTIV – cont.

- ◆ Rezultatul contine valori care caracterizează ansamblul din care provine și poate conține:
- ◆ Valori ale unor funcții statistice (MIN, MAX, AVG, SUM, COUNT, STDDEV, VARIANCE).
- ◆ Constante (numerice, șir de caractere sau dată calendaristică). În categoria constantelor sunt incluse și pseudocoloanele puse la dispoziție de sistem cum este SYSDATE.
- ◆ Valori ale expresiilor după care s-a făcut gruparea, în cazul în care aceasta este prezentă în cerere.

REZULTAT

Rezultatul unei cereri SELECT care conține funcții statistice are:

- ◆ O singură linie în cazul în care cererea nu conține clauza de grupare GROUP BY
- ◆ Un număr de linii egal cu numărul de grupuri formate pe baza criteriilor de grupare din GROUP BY, dacă această clauză există și nu este însoțită de clauza HAVING.
- ◆ Un număr de linii egal cu numărul de grupuri formate pe baza criteriilor de grupare din GROUP BY care îndeplinesc condiția de grup din HAVING, dacă ambele clauze sunt prezente în cerere.

Numărul de coloane al rezultatului este ca și înainte egal cu numărul expresiilor aflate pe clauza SELECT.

FUNCTII STATISTICR

- ◆ MIN si MAX – valoare minima si maxima
- ◆ SUM si AVG – suma si medie
- ◆ COUNT – numarare
- ◆ STDDEV si VARIANCE – deviatia standard si varianta

MIN si MAX

Sintaxa funcțiilor:

- ◆ MIN([ALL | DISTINCT] expresie)
- ◆ MAX([ALL | DISTINCT] expresie)

Descriere:

- ◆ MIN(expresie) întoarce **valoarea minimă nenula** din lista de valori ale expresiei, fiecare valoare fiind calculată pe baza unei linii din parcurgerea curentă.
- ◆ MAX(expresie) întoarce **valoarea maximă nenula** din aceeași listă.

Folosirea cuvintelor cheie ALL (care este implicit) sau DISTINCT **nu are în acest caz nici un efect**, minimul și maximul fiind același dacă din listă se elimină duplicatele.

EXEMPLU

```
SELECT 'MATEMATICA' FACULTATE,  
       MIN(SUMA) MINIM, MAX(SUMA) MAXIM  
FROM STUD, SPEC, BURSA  
WHERE STUD.PUNCTAJ BETWEEN PMIN AND  
       PMAX AND  
       STUD.CODS = SPEC.CODS AND SPEC.NUME =  
       'MATEMATICA'
```

- ◆ Rezultatul obținut va avea o singură linie având conținutul:

```
MATEMATICA 150 300
```

SUM

- ◆ Sintaxa funcției:

`SUM([ALL | DISTINCT] expresie)`

- ◆ Descriere:

- ◆ `SUM(expresie)` întoarce suma **valorilor nenule ale expresiei**, fiecare valoare fiind calculată pe baza unei linii din parcurgerea curentă.

- ◆ `SUM(DISTINCT expresie)` face același lucru ignorând însă valorile duplicate.

AVG

- ◆ Sintaxa funcției:

`AVG([ALL | DISTINCT] expresie)`

- ◆ Descriere:

- ◆ `AVG(expresie)` întoarce media aritmetică a valorilor nenule ale expresiei.

- ◆ `AVG(DISTINCT expresie)` ignora valorile duplicat.

EXEMPLU

Cererea care calculeaza sume si medii ale burselor studentilor de la ISTORIE:

```
SELECT 'ISTORIE' FACULT,  
       SUM(SUMA) SUM1, SUM(DISTINCT SUMA) SUM2,  
       AVG(SUMA) AVG1, AVG(DISTINCT SUMA) AVG2  
FROM STUD, SPEC, BURSA  
WHERE STUD.PUNCTAJ BETWEEN PMIN AND PMAX AND  
STUD.CODS = SPEC.CODS AND SPEC.NUME='ISTORIE'
```

REZULTAT

FACULT	SUM1	SUM2	AVG1	AVG2
ISTORIE	650	550	162.5	183.3

◆ După cum se observă, suma și media sunt diferite pentru considerarea sau ignorarea valorilor duplicat:

- ◆ $650 = 100 + 100 + 150 + 300$
- ◆ $550 = 100 + 150 + 300$
- ◆ $162.5 = 650 / 4$
- ◆ $183.3 = 550 / 3$

COUNT

Sintaxa funcției:

- ◆ COUNT(*)
- ◆ COUNT([ALL | DISTINCT] expresie)

Descriere:

- ◆ COUNT(*) întoarce numărul de linii pe baza cărora se calculează rezultatul => nu se poate adăuga DISTINCT sau nu se poate vorbi de valori nule
- ◆ COUNT(expresie) întoarce numărul de valori nenule ale expresiei.
- ◆ COUNT(DISTINCT expresie) întoarce numărul de valori nenule și distincte ale expresiei.

EXEMPLU

- ◆ Tot pentru ISTORIE se dorește numărului studenților, numărul de bursieri și numărul de valori diferite ale sumei primite ca bursă. Cererea este:

```
SELECT 'ISTORIE' FACULT, COUNT(*) NRSTUD,  
       COUNT(SUMA) BURSIERI,  
       COUNT(DISTINCT SUMA) BURSE  
FROM STUD, SPEC, BURSA  
WHERE STUD.PUNCTAJ BETWEEN PMIN AND PMAX  
       AND STUD.CODS = SPEC.CODS - cond.join  
       AND SPEC.NUME='ISTORIE';
```

REZULTAT

FACULT	NRSTUD	BURSIERI	BURSE
-----	-----	-----	-----
ISTORIE	5	4	3

- ◆ Într-adevăr, sunt 5 studenți, doar 4 au bursa nenulă și sunt trei valori distincte pentru aceasta: 100, 150 și 300.

STDDEV si VARIANCE

Sintaxa funcțiilor:

- ◆ `STDDEV([ALL | DISTINCT] expresie)`
- ◆ `VARIANCE([ALL | DISTINCT] expresie)`

Descriere:

- ◆ Din punct de vedere matematic, deviația standard dă o măsură a abaterii față de medie iar varianța este pătratul deviației standard.
- ◆ `STDDEV(expresie)` întoarce deviația standard a valorilor nenule ale expresiei.
- ◆ `VARIANCE(expresie)` întoarce varianța valorilor respective.

Cuvântul cheie `DISTINCT` duce la ignorarea valorilor duplicat.

EXEMPLU

```
SELECT STDDEV (PUNCTAJ) ,  
       VARIANCE (PUNCTAJ)  
FROM STUD;
```

◆ Rezultat:

STDDEV (PUNCTAJ)	VARIANCE (PUNCTAJ)
949.8465	902208.3

CLAUZA GROUP BY

- ◆ În cererile anterioare toate liniile parcurgerii curente formau un grup din care se calculau valorile funcțiilor statistice.
- ◆ Dacă se dorește însă partiționarea acestora în grupuri pentru a calcula valori statistice pentru fiecare grup în parte este necesară folosirea clauzei GROUP BY.
- ◆ Sintaxa acesteia este:

```
GROUP BY expresie1 [, expresie2,  
expresie 3 ...]
```

EFFECT

Această clauză trebuie să apară în cerere după cele discutate anterior (SELECT, FROM și WHERE) și are următorul efect:

- ◆ Liniile parcurgerii curente (filtrate anterior de WHERE dacă aceasta este prezentă) sunt împărțite în grupuri.
- ◆ Fiecare grup este format din liniile care au aceleași valori (inclusiv valoarea nulă) pentru expresiile specificate în GROUP BY.
- ◆ Funcțiile statistice se calculează pentru fiecare grup în parte, rezultatul având un număr de linii egal cu numărul de grupuri formate.
- ◆ În clauza SELECT, pe lângă constante și funcții statistice, pot apare de asemenea și expresiile aflate în GROUP BY deoarece valorile acestora sunt constante la nivelul fiecărui grup în parte.

EXEMPLU

- ◆ Listă conținând codurile specializărilor și numărul de studenți înscriși la fiecare, cererea este:

```
SELECT CODS, COUNT(*) NRSTUD
FROM STUD
GROUP BY CODS;
```

- ◆ Rezultatul obținut este:

```
CODS NRSTUD
-----
```

11	3
21	4
24	5

EXEMPLU – cont.

- ◆ Pentru a avea numele specializării gruparea se face pe joinul STUD cu SPEC:

```
SELECT SPEC.NUME, COUNT(*) NRSTUD
FROM STUD, SPEC
WHERE STUD.CODS = SPEC.CODS
GROUP BY SPEC.NUME;
```

- ◆ obținându-se rezultatul:

NUME	NRSTUD
-----	-----
GEOGRAFIE	4
ISTORIE	5
MATEMATICA	3

OBSERVATII

- ◆ Dacă în cererea anterioară gruparea s-ar fi făcut după STUD.CODS sau SPEC.CODS s-ar fi semnalat eroarea: *ORA-00979: not a GROUP BY expression* din cauza prezenței lui SPEC.NUME în linia SELECT.
- ◆ Pentru a include în rezultat și numele și codul specializării clauza GROUP BY trebuie să le conțină pe amândouă. Grupurile vor fi aceleași dar cererea va fi validă și va putea fi executată:

```
SELECT SPEC.NUME, SPEC.CODS, COUNT(*)  
NRSTUD  
FROM STUD, SPEC  
WHERE STUD.CODS = SPEC.CODS  
GROUP BY SPEC.NUME, SPEC.CODS;
```

ALT EXEMPLU

- ◆ Se dorește pentru fiecare tip de bursă valoarea acesteia, valoarea mărită cu 10% și numărul de studenți beneficiari:

```
SELECT SUMA, SUMA*1.1 BMARITA, COUNT(*)  
FROM STUD, BURSA  
WHERE PUNCTAJ BETWEEN PMIN AND PMAX  
GROUP BY SUMA;
```

- ◆ Această cerere va semnala de asemenea eroare: gruparea se face doar după SUMA, în consecință prezența în SELECT a expresiei SUMA*1.1 este ilegală, deși este evident că doi studenți care au burse egale au și burse mărite cu 10% egale.
- ◆ Sistemul de gestiune nu face astfel de deducții, pentru afișarea datelor cerute fiind necesară adăugarea în GROUP BY a expresiei SUMA*1.1.

ORDONARE IMPLICITA

Ordinea expresiilor din clauza de grupare are efect asupra prezentării rezultatului: sistemul Oracle face implicit o sortare în funcție de aceste expresii, ascendent, în ordinea în care ele sunt prezente în cerere, începând însă cu valorile nule.

EXEMPLU

```
SELECT TUTOR, CODS, COUNT (*)  
FROM STUD  
WHERE CODS IN (11, 24)  
GROUP BY TUTOR, CODS;
```

REZULTAT: ordonare dupa TUTOR si CODS

TUTOR	CODS	COUNT (*)
	11	2
	24	2
1456	11	1
3514	24	1
4311	24	2

EXEMPLU

```
SELECT TUTOR, CODS, COUNT (*)  
FROM STUD  
WHERE CODS IN (11, 24)  
GROUP BY CODS, TUTOR;
```

REZULTAT: ordonare dupa CODS si TUTOR

TUTOR	CODS	COUNT (*)
	11	2
1456	11	1
	24	2
3514	24	1
4311	24	2

ORDONARE IMPLICITA – cont.

- ◆ Se poate folosi clauza ORDER BY pentru a schimba ordinea implicită.
- ◆ În ea pot fi prezente doar elementele constante la nivel de grup (funcții statistice, expresiile de grupare) sau numărul coloanei din rezultat.
- ◆ În cererea următoare ordonarea se face după coloana a treia din rezultat și, la valori egale pe aceasta, după valoarea minimă a punctajului la nivel de grup, deși acest minim nu este prezent în lista SELECT.

EXEMPLU

```
SELECT TUTOR, CODS, COUNT (*)  
FROM STUD  
WHERE CODS IN (11, 24)  
GROUP BY CODS, TUTOR  
ORDER BY 3, MIN(PUNCTAJ);
```

◆ Rezultatul este:

TUTOR	CODS	COUNT (*)
3514	24	1
1456	11	1
	24	2
4311	24	2
	11	2

IMBRICARE FCT. STATISTICE

- ◆ Funcțiile statistice se pot imbrica.
- ◆ În acest caz valoarea rezultată se calculează pe baza valorilor întoarse de funcția imbricată pentru fiecare grup în parte.
- ◆ De exemplu, dacă se dorește o medie a burselor medii pe specializări, aceasta poate fi calculată cu cererea:

```
SELECT AVG(AVG(NVL(SUMA, 0))) MEDIE1  
FROM STUD, BURSA  
WHERE PUNCTAJ BETWEEN PMIN AND PMAX  
GROUP BY CODS;
```

REZULTAT

- ◆ Rezultatul obținut este următorul:

```
MEDIE1  
-----  
135
```

- ◆ De remarcat că această valoare este diferită de media burselor studenților care este de 133.33 și este calculată de cererea:

```
SELECT AVG (NVL (SUMA, 0) ) MEDIE2  
FROM STUD, BURSA  
WHERE PUNCTAJ BETWEEN PMIN AND PMAX;
```

CLAUZA HAVING

- ◆ Fiecare grup format de GROUP BY are, în cazul cererilor precedente, o linie corespondentă în rezultat.
- ◆ Clauzele descrise până acum nu permit o filtrare la nivel de grup.
- ◆ De exemplu, dacă se dorește obținerea de date doar despre specializările care au studenți cu un punctaj mediu mai mare decât 1200, cererea următoare va semnala eroare:

```
SELECT CODS, AVG(PUNCTAJ)  
FROM STUD  
WHERE AVG(PUNCTAJ) > 1500  
GROUP BY CODS;
```


CLAUZA HAVING – cont.

- ◆ Eroarea se datorează faptului că WHERE poate conține doar condiții la nivelul unei linii din parcurgerea curentă și nu la nivel de grup.
- ◆ Pentru filtrarea grupurilor este necesară o nouă clauză, HAVING, având următoarea sintaxă:
HAVING *conditie_de_grup*
- ◆ Condiția poate fi simplă sau compusă. În ea pot fi folosite doar elemente constante la nivel de grup (literali, funcții statistice, etc.).

EXEMPLU

- ◆ Pentru cererea anterioara formularea corectă este:

```
SELECT CODS, AVG(PUNCTAJ)
FROM STUD
GROUP BY CODS
HAVING AVG(PUNCTAJ) > 1500
```

- ◆ Rezultatul va conține doar o linie, celelalte două grupuri fiind filtrare de condiția din HAVING:

```
CODS  AVG(PUNCTAJ)
-----
11    1560
```

HAVING FARA GROUP BY

- ◆ Această clauză poate fi folosită și în absența lui GROUP BY.
- ◆ În acest caz cererea întoarce fie o linie, fie nici o linie, în funcție de satisfacerea sau nu a condiției HAVING de liniile parcurgerii curente.
- ◆ De exemplu, datorită faptului că punctajul mediu pentru întreaga tabelă STUD este mai mic decât 1500, cererea:

```
SELECT AVG(PUNCTAJ)  
FROM STUD  
HAVING AVG(PUNCTAJ) > 1500;
```

nu va întoarce nici o linie de rezultat.

HAVING SI IMBRICARE

- ◆ În cazul în care avem funcții statistice imbricate, rezultatul acestora va fi calculat doar pe baza valorilor funcției imbricate aplicată grupurilor care satisfac condiția din HAVING.
- ◆ De exemplu, următoarea cerere va calcula media burselor medii pentru două specializări (cele cu punctaj mediu peste 1200) și nu pentru toate. Rezultatul va fi 140 și nu 135 ca atunci când nu există HAVING:

```
SELECT AVG (AVG (NVL (SUMA, 0) ) ) MEDIE1  
FROM STUD, BURSA  
WHERE PUNCTAJ BETWEEN PMIN AND PMAX  
GROUP BY CODS  
HAVING AVG (PUNCTAJ) > 1200;
```

Sfarsitul capitolului

FUNCTII STATISTICE SI GRUPURI