

Grupowanie danych

Grupowanie (agregowanie) danych jest operacją w wyniku której wartości kolumn, w ramach wierszy, które zostały uznane za podstawę grupowania, zostają potraktowane jako elementy niepowtarzalne zbioru i tak wybrane.

Możliwe jest wówczas określenie na nowo utworzonym zbiorze przyporządkowania, za pomocą funkcji grupowych, wartości każdemu z wierszy.

Idea grupowania

Kol A	Kol B
X	1
X	1
Y	1
Y	2

Kol A (zgrupowana)	max(Kol B)	min(Kol B)	avg(Kol B)	sum(Kol B)	count(*)
X	1	1	1	2	2
Y	2	1	1,5	3	2

Określanie kolumn grupujących możliwe jest przez zastosowanie klauzuli GROUP BY.

Przykłady:

```
SELECT d.NAME , j.FUNCTION
FROM EMPLOYEE e, JOB j, DEPARTMENT d
WHERE e.job_id = j.job_id
      AND e.department_id = d.department_id
GROUP BY j.FUNCTION, d.NAME
ORDER BY 1, 2
```

```
SELECT d.NAME , j.FUNCTION, MIN( e.HIRE_DATE ), MAX( e.SALARY ), COUNT(*)
FROM EMPLOYEE e, JOB j, DEPARTMENT d
WHERE e.job_id = j.job_id
      AND e.department_id = d.department_id
GROUP BY j.FUNCTION, d.NAME
ORDER BY 1, 2
```

Wszystkie funkcje grupowe ignorują wartość NULL oprócz COUNT(*), która zlicza wiersze niezależnie od zawartości kolumn.

W klauzuli GROUP BY muszą być umieszczone minimum te kolumny, czy wyrażenia, które nie są umieszczone w funkcjach grupowych, odstępstwem są stałe.

Argumentami funkcji grupowych mogą być liczby, a dla COUNT, MAX i MIN daty i kolumny znakowe.

Nie można używać wprost funkcji grupowych do zastosowania w klauzuli WHERE, ponieważ zawartość WHERE jest wykorzystywana do określenia zakresu danych przed realizacją grupowania.

Do filtrowania z użyciem funkcji grupowych dedykowana jest klauzula HAVING.

Przykład:

```
SELECT d.NAME , j.FUNCTION, MIN( e.HIRE_DATE ), MAX( e.SALARY ), COUNT(*)
FROM EMPLOYEE e, JOB j, DEPARTMENT d
WHERE e.job_id = j.job_id
      AND e.department_id = d.department_id
GROUP BY j.FUNCTION, d.NAME
HAVING COUNT(*) > 1 AND MAX( e.salary ) > 2000
ORDER BY 1, 2
```

Pod względem kolejności klauzule HAVING i GROUP BY są zamienne.

Podzapytania (zapytania zagnieżdżone)

Podzapytanie to realizacja warunku na zapytaniu w sposób zagnieżdżony, tj. taki, który umożliwia zastosowanie w zapytaniu zagnieżdżającym rezultatów pośrednich zwróconych przez zdanie zagnieżdżone.

Podzapytanie w takiej postaci, gdzie wykonywanie jest autonomicznie wykonane zostaje tylko raz, a potem następuje wykonanie zapytania nadrzędnego.

Przykład:

```
SELECT first_name, last_name
FROM EMPLOYEE
WHERE department_id = ( SELECT department_id
                        FROM EMPLOYEE
                        WHERE last_name = 'TURNER' )
AND last_name != 'TURNER';
```

W przypadku porównania podzapytanie musi zwrócić dokładnie jeden wiersz.

Przykład:

```
SELECT first_name, last_name, salary
FROM EMPLOYEE
WHERE salary = ( SELECT MAX(salary)
                 FROM EMPLOYEE e, JOB j
                 WHERE e.job_id = j.job_id
                 AND j.FUNCTION = 'ANALYST'
                 );
```

Podzapytania skorelowane

Podzapytanie skorelowane to szczególny przypadek zagnieżdżenia zapytań, gdzie w warunku podzapytania występuje odwołanie do tabel zewnętrznych (z zapytania nadrzędnego).

W takim przypadku dla każdego wiersza zapytania nadrzędnego wykonywane jest zapytanie podrzędne.

Przykład:

```
SELECT last_name, first_name, NAME || ' (' || d.department_id || ')'
, salary
FROM DEPARTMENT d, EMPLOYEE e
WHERE e.department_id = d.department_id
AND commission > ( SELECT AVG( commission )
                    FROM EMPLOYEE e2
                    WHERE e2.department_id = e. department_id );
```

Możliwe jest aby po lewej stronie znaku równości (nierówności, nieostrej równości) była umieszczona liczba.

Przykład:

```
SELECT last_name, first_name, employee_id
FROM EMPLOYEE EMP
WHERE 3 < ( SELECT COUNT(*)
           FROM CUSTOMER cust
           WHERE cust.salesperson_id = EMP.employee_id
         );
```

Podzapytanie wielowierszowe

W przypadku gdy zdanie zagnieżdżone zwraca więcej wierszy konieczna jest zastąpienie znaku równości przez operator IN.

Przykład:

```
SELECT cust.NAME, cust.city
FROM CUSTOMER cust
WHERE city IN ( SELECT regional_group FROM LOCATION)
```

Możliwe jest aby po lewej stronie znaku równości (nierówności, nieostrej równości) była umieszczona para (trójka, itd.) wartości (kolumn). Musi jej odpowiadać liczba kolumn lub wyrażeń w ramach podzapytania.

Przykład:

```
SELECT first_name, last_name, salary, FUNCTION
FROM EMPLOYEE e, JOB j
WHERE j.job_id = e.job_id
      AND (j.job_id, e.salary) IN ( SELECT job_id, MAX(salary)
                                   FROM EMPLOYEE e2
                                   GROUP BY job_id
                                 );
```

Do obsługi sytuacji w przypadku gdy podzapytanie może zwrócić wiele wierszy możliwe jest zastosowanie operatorów ANY, ALL lub EXISTS

gdzie:

ALL jest stosowany do porównania ze wszystkimi zwracanymi w podzapytaniu wartościami

ANY jest stosowany do porównania z dowolną wartością zwracanymi w podzapytaniu wartościami

EXISTS jest używany do zbadania warunku logicznego „czy istnieje co najmniej jeden wiersz”

Przykłady:

```
SELECT last_name, salary, NAME || ' (' || d.department_id || ')'  
FROM EMPLOYEE e, DEPARTMENT d  
WHERE salary > ALL ( SELECT salary  
                     FROM EMPLOYEE e2, DEPARTMENT d2  
                     WHERE e2.department_id = d2.department_id  
                           AND d2.NAME = 'SALES' )  
   AND e.department_id = d.department_id  
ORDER BY 3, 2, 1;
```

```
SELECT last_name, salary, department_id  
FROM EMPLOYEE e  
WHERE salary > ANY ( SELECT salary  
                    FROM EMPLOYEE e2  
                    WHERE e.department_id = e2.department_id)  
ORDER BY 3,2,1;
```

```
SELECT last_name, first_name, FUNCTION  
FROM EMPLOYEE e, JOB j  
WHERE j.job_id = e.job_id  
   AND EXISTS ( SELECT 1  
                FROM CUSTOMER  
                WHERE salesperson_id = E.employee_id )
```

Zadania:

1. Utworzyć listę nazw towarów od najlepiej się sprzedających (w ilościach) do najmniej, pokazać również ilości.
2. Utworzyć listę nazwisk i imion sprzedawców, od najbardziej efektywnych do najmniej – za efektywność przyjąć sumy rachunków klientów i również je pokazać.
3. Znaleźć pracowników zarabiających największą pensję na ich stanowiskach pracy.
4. Znaleźć departament w którym nikt nie pracuje, używając dwóch różnych sposobów.
5. Dla zadanych zarobków (1500) znaleźć płacę bezpośrednio większą i mniejszą od niej.