

Operacje agregujące oraz grupowanie

Oprócz uzyskiwania informacji, wprowadzonych do bazy, często zachodzi konieczność przeprowadzenia analizy danych, zawartych w bazie. Na przykład, chcemy uzyskać liczbę zatrudnionych pracowników, a nie listę ich imion i nazwisk. Do uzyskania tej i podobnych informacji służą funkcje agregujące.

W przykładzie ze znajdowaniem ilości zatrudnionych pracowników rozwiązanie polega zazwyczaj na znalezieniu ilości wierszy w pewnej tabeli. Funkcją zliczającą wiersze jest *COUNT()*. Można jej użyć na trzy sposoby:

- **SELECT COUNT(*) FROM tabela** - zapytanie zwróci ilość wierszy w całej tabeli.
- **SELECT COUNT(kolumna) FROM tabela** - zapytanie zwróci ilość wierszy w tabeli, z pominięciem wierszy, które nie mają nadanej wartości w podanej kolumnie (NULL).
- **SELECT COUNT(DISTINCT kolumna) FROM tabela** - powyższe zapytanie zwróci ilość wierszy w tabeli, które mają unikalne wartości w podanej kolumnie, ponadto zostaną pominięte wiersze, które nie mają nadanej wartości w podanej kolumnie (NULL).

Możliwe jest ponadto zawężenie zakresu zliczanych wierszy, poprzez zastosowanie predykatu w zapytaniu np.

SELECT COUNT(*) AS Nieletni FROM personalia WHERE wiek<18;

Powyższe zapytanie wyświetli liczbę wierszy w tabeli, które mają wpis w kolumnie wiek, będący liczbą mniejszą niż 18.

Dwie kolejne funkcje agregujące to *AVG()* oraz *SUM()*. Służą one do wyznaczenia, odpowiednio: wartości średniej arytmetycznej oraz sumy, z wyrażenia, będącego ich argumentem, opartego na wartości z kolumn tabeli. Kolumny te powinny być kolumnami liczbowymi. Przykładowe zapytanie, wyznaczające koszty zatrudnienia pracowników:

SELECT SUM(pensja) FROM place

Zapytanie powyższe obliczy sumę kwot pensji wszystkich pracowników, zaś zapytanie

SELECT AVG(pensja) FROM place

obliczy średnią pensję pracownika.

Ostatnie dwie kanoniczne funkcje agregujące w języku *SQL* to *MIN()* oraz *MAX()*, dostarczające informacji o odpowiednio: najmniejszej oraz największej wartości ich argumentu. Charakterystyczne jest to, że funkcje te mogą pracować również na tekstach, gdyż język *SQL* określa reguły porównywania ze sobą tekstów. Przykładowe zapytania używające tych funkcji to:

SELECT MIN(pensja) FROM place

SELECT MAX(pensja) FROM place

Zapytania te dostarczą informacji o najmniejszej oraz największej płacy pracowników firmy.

Ważnym narzędziem w języku *SQL*, pomocnym w tworzeniu użytecznych zapytań, związanych z funkcjami agregującymi, jest grupowanie. Polega ono na wydzieleniu z tabeli grup wierszy, posiadających tę samą wartość w pewnych kolumnach. Grupy takie są traktowane przez funkcje agregujące jako rozłączne obiekty.

Jako przykład wyobraźmy sobie następującą sytuację: chcemy dowiedzieć się, jaka jest ilość osób pracujących na poszczególnych stanowiskach na uczelni wyższej. Jakkolwiek wiadomo, że jest tylko jeden rektor wyższej uczelni, to liczba asystentów, wykładowców, pracowników technicznych i

innych nie jest limitowana. Funkcja *COUNT()*, w zakresie poznanym do tej pory, umożliwia jedynie znalezienie ilości wszystkich pracowników, a nie ilości pracowników na poszczególnych stanowiskach. Do rozwiązania tego typu problemu należy wydać zapytanie, korzystające z klauzuli *GROUP BY*, włączającej mechanizm grupowania:

SELECT kolumna1,kolumna2 FROM tabela GROUP BY kolumna

W przypadku naszego przykładu zapytanie mogłoby wyglądać następująco:

SELECT stanowisko, COUNT(*) AS ilosc FROM pracownicy GROUP BY stanowisko

Zapytanie to zwróci tabelę, składającą się z dwóch kolumn: w pierwszej wyszczególnione będą wszystkie obsadzone stanowiska (np. asystent, adiunkt, profesor, rektor itd.), a w drugiej ilości pracowników na poszczególnych stanowiskach. Standard języka mówi, że grupować można tylko względem kolumny umieszczonej na liście kolumn w zapytaniu, tutaj względem kolumny *stanowisko*.

UWAGA! Jedynie w zapytaniach, używających klauzuli grupującej, można mieszać na liście kolumn do wyświetlenia kolumny i funkcje agregujące!

Przed grupowaniem możliwe jest dokonanie selekcji wierszy, poprzez umieszczenie klauzuli *WHERE* (zawężenie zestawu wierszy następuje *przed* operacją grupowania):

SELECT stanowisko, COUNT(*) AS ilosc FROM place WHERE wiek>=60 GROUP BY stanowisko

Zapytanie takie pominie pracowników, którzy nie mają jeszcze 60 lat. Klauzula *WHERE* obarczona jest ograniczeniem mówiącym, że nie można w umieszczonym po niej predykatie używać aliasów kolumn, zdefiniowanych na liście *SELECT*. Ograniczeniem tym nie jest obarczona klauzula *HAVING*, która również służy do ograniczenia wyniku zapytania, a którą należy zastosować w zapytaniu w następujący sposób:

SELECT stanowisko, COUNT(*) AS ilosc FROM place GROUP BY stanowisko HAVING stanowisko<>'rektor'

Zapytanie to pominie wyświetlenie “grupy” pracowników na stanowisku rektora.

Co różni klauzule: *HAVING* i *WHERE*?

- Klauzula *WHERE* wybiera wiersze *przed* ich grupowaniem, klauzula *HAVING* wybiera grupy *po* procesie grupowania. Z tego powodu w klauzuli *HAVING* należy umieścić kryterium, oparte o wyrażenie, umieszczone w klauzuli *GROUP BY* (z kolei klauzula *GROUP BY* powinna zawierać listę kolumn, będącą podzbiorem listy *SELECT*), klauzula *WHERE* nie ma tego ograniczenia.
- w klauzulach *HAVING* oraz *ORDER BY* można stosować aliasy kolumn z listy *SELECT*, w klauzuli *WHERE* nie ma tej możliwości.

Sortowanie wyników zapytania można uzyskać, stosując klauzulę *ORDER BY*, która musi być umieszczona po klauzuli *HAVING*. Sortowania wyników odbywa się również po wykonaniu ograniczenia zgodnego z klauzulą *HAVING*. Proces tworzenia tabeli wynikowej można przedstawić następująco:

1. Najpierw wykonywana jest operacja ograniczenia w oparciu o predykat, umieszczony po klauzuli *WHERE* (klauzula ta, o ile istnieje, musi występować w zapytaniu jako pierwsza, spośród wymienionych na niniejszej liście).
2. Następnie dokonywane jest grupowanie wierszy, wybranych w poprzednim kroku, względem listy kolumn, stojącej po klauzuli *GROUP BY* (musi ona występować w zapytaniu po klauzuli

WHERE, jeśli obie zostały użyte), przy czym kolumny te muszą być zawarte na liście *SELECT*.

1. W dalszej kolejności dokonywane jest wybranie grup, spełniających predykat, podany po klauzuli *HAVING*, powinien on być oparty na części lub całości listy kolumn, względem której nastąpiło grupowanie wierszy. Klauzula *HAVING* musi występować w zapytaniu po klauzulach *WHERE* oraz *GROUP BY*.
2. Ostatnią czynnością jest sortowanie wyniku, względem listy kolumn, umieszczonej po klauzuli *ORDER BY*, która musi występować jako ostatnia w zapytaniu.

Jeżeli na liście kolumn, względem których realizowane jest grupowanie, znajduje się więcej niż jedna kolumna, tworzone są podgrupy: najpierw następuje grupowanie względem pierwszej kolumny na liście, następnie w zakresach uzyskanych w ten sposób grup wydzielane są podgrupy względem kolejnej kolumny na liście itd. aż do wyczerpania listy kolumn. Schemat postępowania jest analogiczny, jak przy sortowaniu względem wielu kolumn.