BAZY DANYCH

Lab. 5 Funkcje agregujące, klauzule GROUP BY, HAVING

Laboratorium nr 5

Temat: Funkcje agregujące, klauzule GROUP BY, HAVING

Celem ćwiczenia jest zaprezentowanie zagadnień dotyczących stosowania w zapytaniach języka SQL predefiniowanych funkcji agregujących.

PLAN LABORATORIUM:

1. Charakterystyka funkcji agregujących
2. Zapytania z jedną grupą
3. Zapytania z wieloma grupami
4. Podział grup na podgrupy
5. Filtrowanie grup
6. Najczęstsze błędy przy konstrukcji zapytań z funkcjami agregującymi
7. Konstrukcje zaawansowane

1. Charakterystyka funkcji agregujących

Funkcje agregujące działają na zbiorach rekordów, nazywanych **grupami** (w przeciwieństwie do funkcji wierszowych, które zawsze działają na jednym rekordzie). Przed zastosowaniem funkcji agregującej konieczne jest podzielenie rekordów na grupy, tzw. **grupowanie.** Do jednej grupy należą te rekordy relacji, dla których tzw. wyrażenie grupujące zwraca tą samą wartość. Wyrażeniem grupującym jest najczęściej pojedynczy atrybut relacji. Po podziale rekordów na grupy w każdej z grup zostaje zastosowana funkcja agregująca, która wylicza **pojedynczą wartość** dla grupy. Stąd w wyniku zapytania otrzyma się tyle rekordów, ile grup zostało utworzonych w wyniku operacji grupowania.

Problem: znajdź średnie wynagrodzenie pracowników dla każdej grupy stanowisk.

SQL Query Area

```
1 SELECT imie, nazwisko, wynagrodzenie, id_stanowisko
2 FROM pracownicy
3 ORDER BY id_stanowisko
```

imie	nazwisko	wynagrodzenie	id_stanowisko
Jan	Kowalczuk	1700.00	1
Tomasz	Makarski	2000.00	1
Danuta	Zielinska	2000.00	1
Lukasz	Borowik	1900.00	1
Mariusz	Potepa	1700.00	1
Emilia	Ignatowicz	2000.00	1
Radoslaw	Tomaszewki	2100.00	1
Wojciech	Potepa	1900.00	1
Krzysztof	Potepa	9000.00	1
Mateusz	Zielonka	2000.00	1
Dariusz	Malinowski	3000.00	2
Krystyna	Czuj	2850.00	2
Mateusz	Brzeski	2900.00	3
Antoni	Darecki	2700.00	4
Anna	Molek	1200.00	5

Grupy dla poszczególnych stanowisk

id_stanowisko	avg(wynagrodzenie)
1	2630.000000
2	2925.000000
3	2900.000000
4	2700.000000
5	1200.000000

Przykład zapytania z grupowaniem i funkcją agregującą: „znajdź średnie wynagrodzenie pracowników dla każdej grupy stanowisk". Wyrażeniem grupującym, które dostarcza wartości dzielące zbiór rekordów relacji EGZAMINATORZY na grupy, jest atrybut STANOWISKO. Przykładowy zbiór rekordów został podzielony na trzy grupy: pierwszą dla wartości ADIUNKT, znajdują się w niej trzy rekordy, następną dla wartość ASYSTENT, należą do niej dwa rekordy, wreszcie ostatnią dla wartości WYKŁADOWCA z pięcioma rekordami. Następnie w każdej z grup wartości atrybutu WYNAGRODZENIE zostają poddane dzianiu funkcji agregującej, wyliczającej średnią. W wyniku zapytania otrzyma się po jednym rekordzie dla każdej grupy etatowej: rekord zawiera wartość wyrażenia grupującego, a więc atrybutu STANOWISKO i wyliczonym średnim wynagrodzeniem w tej grupie.

1. Rodzaje funkcji agregujących

- Funkcje:
  - **MAX** - maksimum,
  - **SUM**- suma.
  - **COUNT** - liczba wystąpień,
  - **MIN**- minimum,
  - **AVG** - średnia,

  Składnia: **nazwa_funkcji(all** | distinct *wyrażenie)*

- Szczególny przypadek - funkcja **COUNT:**
  - **COUNT(*)** - liczba rekordów,
  - **COUNT(all** | distinct wyrażenie) - liczba niepustych wartości wyrażenia.

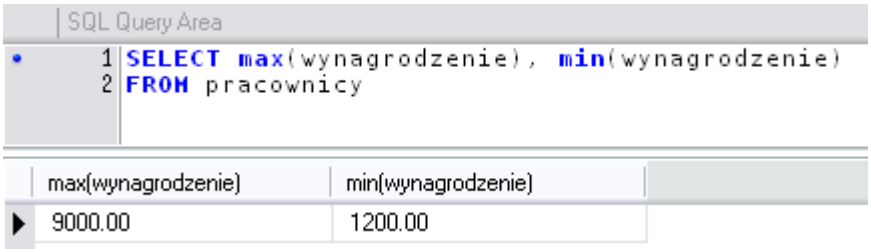
Do wyliczenia średniej służy funkcja **AVG.** Funkcja **COUNT** pozwala na znalezienie liczby wystąpień. Funkcje **MAX** i **MIN** umożliwiają znalezienie, odpowiednio, wielkości maksymalnej i minimalnej w zbiorze. Funkcja **SUM** umożliwia wyliczenie sumy elementów. Każda z funkcji posiada jeden parametr, będący wyrażeniem dostarczającym wartości do obliczeń. W przypadku funkcji AVG i SUM wyrażenie musi dostarczać wartości liczbowe, pozostałe funkcje agregujące przyjmują wartości dowolnego typu podstawowego. Przed wyrażeniem można umieścić słowo kluczowe **DISTINCT,** wówczas do obliczeń zostaną wzięte wartości wyrażenia po eliminacji powtórzeń. Umieszczenie w wywołaniu funkcji samego wyrażenia lub poprzedzenie wyrażenia słowem **ALL** powoduje, że do obliczeń będą brane wszystkie wartości wyrażenia.

Dodatkowego komentarza wymaga użycie funkcji COUNT. Funkcja zwróci liczbę niepustych wartości wyrażenia w grupie (wartości różnych od null). Dodanie słowa DISTINCT spowoduje policzenie różnych niepustych wystąpień wartości wyrażenia w grupie. Z kolei jeśli w wywołaniu funkcji wyrażenie zastąpi się gwiazdką (symbol *), wówczas zostanie policzona liczba rekordów należących do grupy. Pozostałe funkcje agregujące również pomijają przy obliczeniach wartości puste, stąd nie ma konieczności stosowania mechanizmów eliminujących wartości puste.

2. Zapytania z jedną grupą

Są to zapytania bez wyrażenia grupującego, stąd wszystkie rekordy, odczytane przez zapytanie, trafiają do tej samej jednej grupy, a wynikiem zapytania będzie zawsze **jeden rekord** z wartościami wyliczonymi przez umieszczone w klauzuli SELECT zapytania funkcje agregujące.

Przykładowe zapytanie wylicza dwie wartości: minimalne wynagrodzenie i maksymalne wynagrodzenie. Wynikiem zapytania jest jeden rekord z dwiema wartościami.



3. Zapytania z wieloma grupami

Konstruując zapytania z wieloma grupami konieczne jest zdefiniowanie wyrażenia grupującego. Wyrażenie grupujące umieszcza się w klauzuli **GROUP BY**. W przykładzie zbiór rekordów relacji PRACOWNICY zostaje podzielony na grupy ze względu na wartość atrybutu id_stanowisko, następnie w każdej z grup zostaje wyliczona wartość średniego wynagrodzenia pracowników, zatrudnionych na danym stanowisku. W wyniku zapytania otrzyma się tyle rekordów, ile jest stanowisk. Wynik zapytania zostaje posortowany ze względu na wartość wyrażenia grupującego, a więc atrybutu id_stanowisko.

```
SQL Query Area
1 SELECT id_stanowisko,avg(wynagrodzenie)
2 FROM pracownicy
3 GROUP BY id_stanowisko
```

id_stanowisko	avg(wynagrodzenie)
1	2630.000000
2	2925.000000
3	2900.000000
4	2700.000000
5	1200.000000

4. *Podział grup na podgrupy*

Istnieje możliwość podziału grup, odczytywanych przez zapytanie, na podgrupy. Realizuje się to umieszczając w klauzuli GROUP BY **kilka wyrażeń grupujących.** W przykładzie zbiór rekordów relacji PRACOWNICY zostaje podzielony na grupy ze względu na wartość atrybutu id_stanowisko (pierwsze wyrażenie grupujące), następnie rekordy w każdej z grup zostają podzielona na podgrupy ze względu na wartość atrybutu wymagrodzenie (drugie wyrażenie grupujące).

Funkcja agregująca zostaje wykonana w każdej z podgrup. Interpretacja wyniku przykładowego zapytania jest następująca: otrzymuje się dla każdego wynagrodzenia informację o liczbie pracowników zarabiających daną kwotę pracujących na poszczególnych stanowiskach.

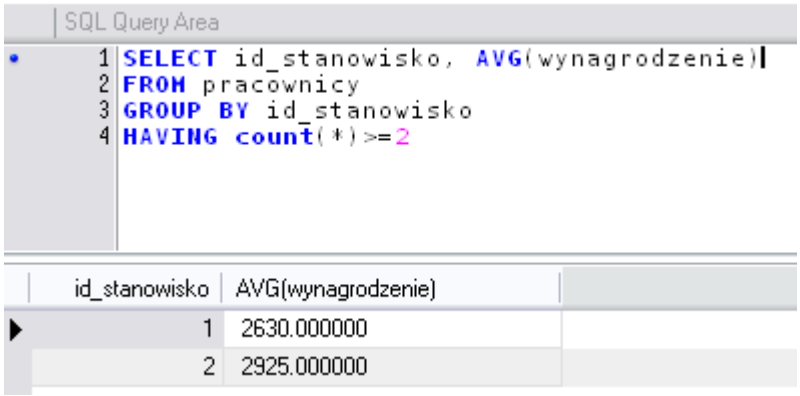
```
SQL Query Area
1 SELECT id_stanowisko, wynagrodzenie, count(*)
2 FROM pracownicy
3 GROUP BY id_stanowisko, wynagrodzenie
```

id_stanowisko	wynagrodzenie	count(*)
1	1700.00	2
1	1900.00	2
1	2000.00	4
1	2100.00	1
1	9000.00	1
2	2850.00	1
2	3000.00	1
3	2900.00	1
4	2700.00	1
5	1200.00	1

5. Filtrowanie grup

Zapytanie z grupowaniem można dodatkowo wyposażyć w mechanizm filtrowania grup. Realizuje się to umieszczając warunek logiczny w dodatkowej klauzuli HAVING. Należy pamiętać, że w warunku tym można użyć jedynie wyrażeń grupujących i/lub funkcji agregujących. Wartościowanie warunku następuje po utworzeniu grup. Grupy, dla których warunek nie jest spełniony, zostają odrzucone (nie pojawiają się w wyniku zapytania).

W przykładzie zbiór rekordów relacji PRACOWNICY zostaje pogrupowany ze względu na wartość atrybutu id_stanowisko, jednak odrzucone zostają te grupy, w których jest mniej niż dwa rekordy. W każdej z pozostałych grup zostaje wyliczone średnie wynagrodzenie pracowników, należących do stanowisk.



6. Najczęstsze błędy przy konstrukcji zapytań z funkcjami agregującymi

- Umieszczenie w warunku w klauzuli WHERE funkcji agregującej

```
SELECT id_stanowisko, wynagrodzenie
FROM pracownicy
WHERE count(*)>=2
```

- Umieszczenie w klauzuli SELECT zapytania z wieloma grupami wyrażenia nie będącego wyrażeniem grupującym lub funkcją agregującą.

  Kolejny błąd dotyczy zapytań z wieloma grupami i polega na umieszczeniu w klauzuli SELECT wyrażenia, nie będącego wyrażeniem grupującym (a więc nie występującym w klauzuli GROUP BY) lub funkcją agregującą. W przykładzie w klauzuli SELECT umieszczono atrybut NAZWISKO, tymczasem wyrażeniem

grupującym jest atrybut id_stanowisko. Pomimo, iż zapytanie zwraca wynik - jest on błędny!!! Proszę przeanalizować przykład

SQL Query Area

```
1 SELECT id_stanowisko, nazwisko, SUM(wynagrodzenie)
2 FROM pracownicy
3 GROUP BY id_stanowisko
```

id_stanowisko	nazwisko	SUM(wynagrod...
1	Kowalczuk	26300.00
2	Czuj	5850.00
3	Brzeski	2900.00
4	Darecki	2700.00
5	Molek	1200.00

SQL Query Area

```
1 SELECT id_stanowisko, nazwisko, imie
2 FROM pracownicy
3
```

id_stanowisko	nazwisko	imie
1	Kowalczuk	Jan
2	Czuj	Krystyna
3	Brzeski	Mateusz
4	Darecki	Antoni
5	Molek	Anna
1	Potepa	Krzysztof
1	Potepa	Wojciech
1	Tomaszewki	Radoslaw
1	Ignatowicz	Emilia
1	Potepa	Mariusz
1	Borowik	Lukasz
2	Malinowski	Dariusz
1	Zielinska	Danuta
1	Makarski	Tomasz
1	Zielonka	Mateusz

- Umieszczenie w warunku w klauzuli HAVING wyrażenia nie będącego funkcją agregującą lub wyrażeniem grupującym. Taki warunek powinien zostać umieszczony w klauzuli WHERE

```
SELECT miasto, count(distinct wynagrodzenie)
FROM pracownicy
GROUP BY miasto
HAVING id_stanowisko=1
```

Poprawnie byłoby

```
SQL Query Area
1 SELECT miasto, count(distinct wynagrodzenie)
2 FROM pracownicy
3 WHERE id_stanowisko=1
4 GROUP BY miasto
5
6
```

miasto	count(distinct w...
Biala Podlaska	1
Lublin	5
Warszawa	2

7. *Konstrukcje zaawansowane*

- Przykład prezentuje zapytanie z jedną grupą (a więc bez zdefiniowanego wyrażenia grupującego), w którym zastosowano klauzulę HAVING. W takim przypadku warunek filtrujący zostaje zastosowany do jedynej grupy zapytania, jeśli warunek nie jest spełniony, zapytanie zwraca wynik pusty. W przykładzie otrzymuje się wynik, maksymalną płacę podstawową pracowników z miast Lublin, Biała Podlaska, pod warunkiem, że z obu miast zatrudniono w sumie przynajmniej 2 osoby.

```
SQL Query Area
1 SELECT MAX(wynagrodzenie) FROM pracownicy
2 WHERE miasto IN ('Lublin', 'Biala Podlaska')
3 HAVING COUNT(*)>=2
```

MAX(wynagrodzenie)
9000.00

- Zapytanie z klauzulą WHERE i HAVING; przykład: dla każdego miasta, z którego pochodzą pracownicy, w którym średnie wynagrodzenie nie jest mniejsze niż 2100, podaj liczbę zatrudnionych pracowników, pomiń pracowników pracujących na stanowisku o identyfikatorze 4, wynik uporządkuj ze względu na sumę wynagrodzeń w poszczególnych miastach.

SQL Query Area

```
1 SELECT miasto, count(*)
2 FROM pracownicy
3 WHERE id_stanowisko<>4
4 GROUP BY miasto
5 HAVING AVG(wynagrodzenie)>=2100
6 ORDER BY SUM(wynagrodzenie);
```

miasto	count(*)
Biala Podlaska	3
Lublin	9

Jest to przykład zapytania, w którym użyto wszystkich zaprezentowanych dotąd klauzul. Należy pamiętać o kolejności wykonywania klauzul.

Jako pierwszy zostaje przetworzony warunek w klauzuli **WHERE**, dokonujący filtrowania rekordów relacji PRACOWNICY ze względu na wartość atrybutu ID_STANOWISKO. Do dalszego przetwarzania zostaną wzięte tylko te rekordy, gdzie ID_STANOWISKO różni się od 4. Następnie realizowane jest grupowanie, wyrażeniem grupującym, umieszczonym w klauzuli **GROUP BY**, jest atrybut MIASTO. Powstałe grupy są filtrowane ze względu na warunek logiczny w klauzuli **HAVING**. Wreszcie wyliczana jest wartość funkcji agregującej, umieszczonej w klauzuli **SELECT** a wynik zostaje posortowany ze względu na wyrażenie umieszczone w klauzuli **ORDER BY**

Ćwiczenie do samodzielnej realizacji:

1. Wyświetl najniższą i najwyższą pensję oraz różnicę dzielącą najlepiej i najgorzej zarabiających pracowników.
2. Jaka jest średnia kwota wynagrodzenia pracowników?
3. Wyświetl średnie pensje dla poszczególnych stanowisk. Wyniki uporządkuj wg malejącego średniego wynagrodzenia.
4. Wyświetl liczbę zatrudnionych pracowników.
5. Znajdź sumaryczne miesięczne wynagrodzenie dla każdego stanowiska wśród pracowników.
6. Wyświetl liczbę zatrudnionych pracowników pracujących na każdym ze stanowisk (Wyniki należy wyświetlić wg wzoru: „Na stanowisku: id_stanowisko pracuje liczba pracowników")
7. Wyświetl nawy miast, które z których pochodzi więcej niż 2 pracowników. Wyniki uporządkuj wg malejącej liczby pracowników.
8. Wyświetl średnie pensje wypłacane w ramach poszczególnych stanowisk i liczbę pracowników zatrudnionych na danym stanowisku. Pomiń pracowników zatrudnionych po 2007 roku.
9. Wyświetl pensje najgorzej zarabiających pracowników pochodzących z poszczególnych miast. Wyniki uporządkuj wg malejącej pensji.
10. Podaj ile nazwisk czytelników zaczyna się na poszczególne litery alfabetu.