

Język SQL część II

Na dzisiejszej lekcji będziemy kontynuować zagadnienia związane z wyszukiwaniem i przetwarzaniem informacji zawartej w bazie danych. Dalej będziemy wykorzystywać tylko jedną tabelę.

1. Przećwiczmy jeszcze instrukcję **SELECT** języka SQL:
 - a. Znajdź wszystkie transakcje wykonane przez Stefana
 - b. Znajdź wszystkie transakcje wykonane pomiędzy 1 sierpnia a 1 listopada 2019 roku.
 - c. Znajdź wszystkie transakcje wykonane przez Ritę o kwocie powyżej 3000 zł
 - d. Wyświetl zestawienie wszystkich transakcji raz z ceną brutto dla każdej transakcji
(cena brutto = cena netto * 1.22)

Funkcje agregujące.

Funkcje agregujące zwracają podsumowania dla całej tabeli lub grupy wierszy w tabeli. Podsumowaniem może być suma, średnia, ilość wierszy z tabeli lub określonej grupy rekordów. Do funkcji agregujących należą: **SUM** – suma, **COUNT** – liczba wystąpień, **MIN** – minimum, **AVG** – średnia,

1. Na początek wykorzystamy instrukcję **COUNT()** do policzenia ilości rekordów w tabeli:

```
SELECT COUNT(*)  
FROM zestawienie;
```

2. Do poprzedniego zapytania możemy dodać instrukcję **WHERE**, aby znaleźć ilość interesujących nas rekordów:

```
SELECT COUNT(*)  
FROM zestawienie  
WHERE reprezentant = 'Rita'
```

3. Policz listę wszystkich transakcji, która została wykonana w styczniu
4. Oblicz sumę wszystkich transakcji
5. Oblicz sumę wszystkich transakcji zrealizowanych przez Ritę w styczniu

Grupowanie danych

Przypuśćmy, że chcemy zobaczyć wszystkich reprezentantów firmy. Aby ich wyłowić użyjemy klauzuli **GROUP BY**:

```
SELECT reprezentant  
FROM zestawienie  
GROUP BY reprezentant
```

Teraz policzymy ilość transakcji wykonanych przez nich:

```
SELECT COUNT(reprezentant) AS 'Ilość transakcji', reprezentant  
FROM zestawienie  
GROUP BY reprezentant
```

1. Policz ilość transakcji wykonanych w marcu.
2. Policz ilość transakcji wykonanych w poszczególnych miesiącach 2019 roku

3. Policz sumę transakcji wykonanych przez poszczególnych odbiorców i posortuj ją malejąco względem nazwy firmy.

Klauzula **HAVING**

Język SQL dostarcza jeszcze jedną metodę filtrowania wyników zapytania w połączeniu z klauzulą **GROUP BY**.

Klauzula **WHERE** filtruje wyniki zapytania zanim są one grupowane, natomiast klauzula **HAVING** filtruje wyniki po wykonaniu grupowania. Klauzula **HAVING** pojawia się przed **ORDER BY** ale za **GROUP BY**.

HAVING i WHERE mogą być stosowane w jednym zapytaniu.

1. Aby wyświetlić wszystkich przedstawicieli, którzy mają mniej niż 30 transakcji posłużymy się poniższym zapytaniem:

```
SELECT COUNT(*) as ile, reprezentant
FROM zestawienie
GROUP BY reprezentant
HAVING COUNT(*) < 30
```

2. Wyświetl wszystkich kontrahentów, którzy mają powyżej trzech zamówień
3. Wyświetl wszystkich przedstawicieli, którzy mają obroty większe niż 70000 zł
4. Wyświetl wszystkich odbiorców, którzy kupili towary za kwotę powyżej 10000 zł. i posortuj ich malejąco względem sumy.