Tematy zajęć:

* Język definiowania zapytań DQL.
* Instrukcje SELECT, FROM. Wybieranie: kolumn, unikatowych wierszy, ograniczenia liczby zwracanych wyników.
* Formatowanie wyników. Aliasy. Sortowanie wyników.
* Filtrowanie danych (IN, NOT, LIKE). Klauzula WHERE.
* Tworzenie pól obliczeniowych.

Spis treści

[Omówienie materiału na podstawie bazy Northwind 1](#_Toc58493414)

[Structured Query Language – Podstawowe polecenia 2](#_Toc58493415)

[Operatory w MS SQL 4](#_Toc58493416)

[Zbiory wynikowe 6](#_Toc58493417)

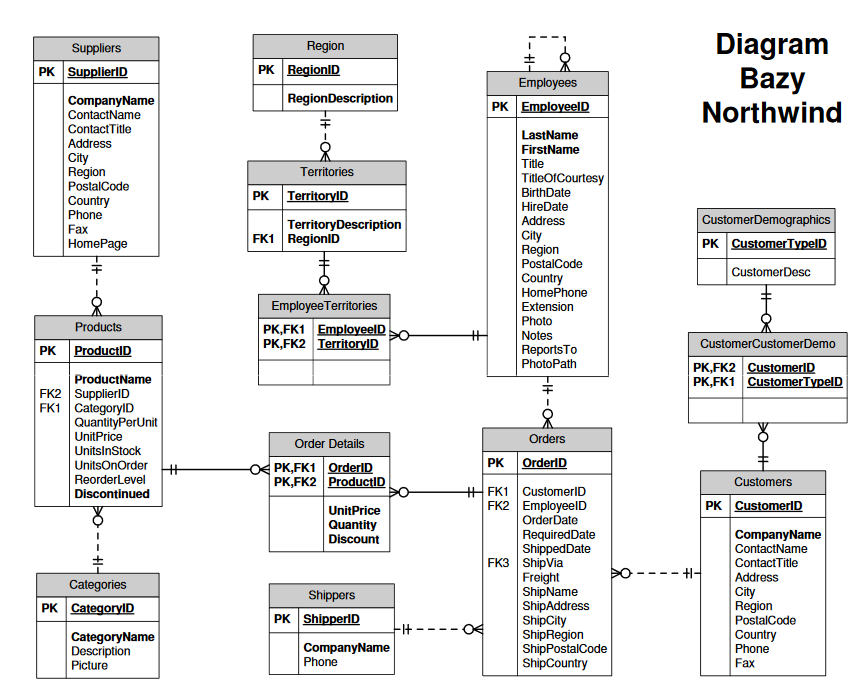
[Sortowanie danych 6](#_Toc58493418)

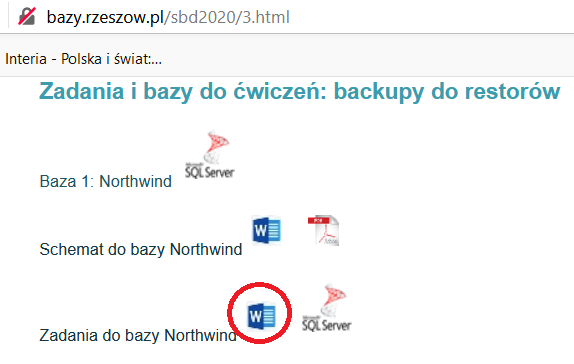
[Eliminowanie duplikatów 7](#_Toc58493419)

[SQL SELECT TOP 7](#_Toc58493420)

# ****Omówienie materiału na podstawie bazy Northwind****

* Należy wykonać restore bazy
* Zapoznać się ze schematem tej bazy
* Baza danych firmy sprzedającej artykuły spożywcze
* Informacje o towarach, dostawcach, klientach, zamówieniach klientów itp.





Rysunek 1. Ze strony przedmiotowej należy pobrać dokument w Wordzie i zapoznać się z nim

# Structured Query Language – Podstawowe polecenia

SQL a wielkość liter

SQL jest językiem, w którym wielkość liter w słowach kluczowych i identyfikatorach nie ma znaczenia. Natomiast dobrą praktyką jest pisanie słów kluczowych wielkimi literami.

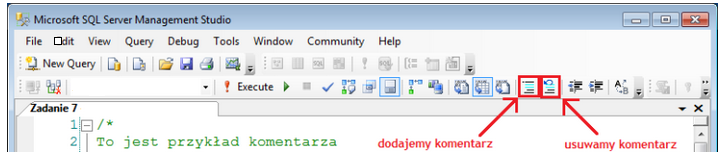
**Komentarze w sql**

Możemy dodać dwa rodzaje komentarzy:

* jednowierszowy, znak „– – treść komentarza”. W tym wariancie wszystko w danym wierszu po dwóch myślnikach zostanie potraktowane jako komentarz aż do końca wiersza i nie będzie brane pod uwagę w momencie uruchomienia zapytania,
* wielowierszowy, znak „ /\* treść komentarza \*/”. W tym wariancie wszystko co znajdzie się pomiędzy dwoma ukośnikami i gwiazdkami zostanie potraktowane jako komentarz.

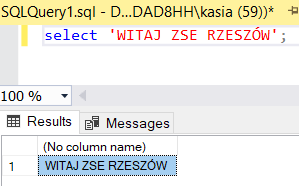
/\*komentarz wielowierszowy\*/

--komentarz liniowy



**DQL** – Data Query Language. DQL to instrukcje, za pomocą których możesz otrzymać z bazy określone dane. Najważniejszym poleceniem jest tutaj **SELECT**.

Instrukcja **select** służy do pobierania danych z bazy. Instrukcja ta może być bardzo prosta, ograniczona jedynie do nazw pobieranych kolumn i nazwy tabeli. Jak również i bardziej złożona, gdzie w połączeniu z różnymi argumentami i funkcjami może wykonywać operacje na danych, m.in. obliczenia, formatować wyniki i tworzyć podsumowania.



Rysunek 2.Najprostsza instrukcja SELECT

Instrukcja select, która pobiera wszystkie dane z wybranej tabeli.

*select \* from [nazwa tabeli];*

Instrukcja:

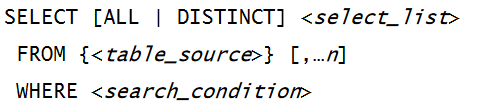
„select” - pobierz

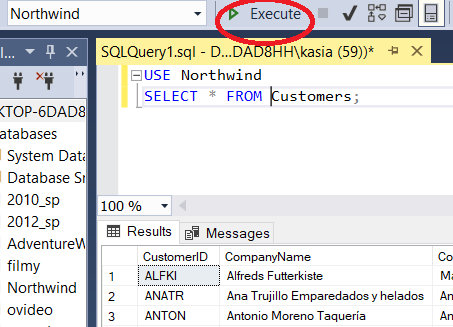
„\*” - oznacza wybranie zawartości wszystkich kolumn do wyświetlenia.

„from” - dosłownie „skąd”, po tym słowie wpisujemy nazwę tabeli, z której mają zostać pobrane dane.

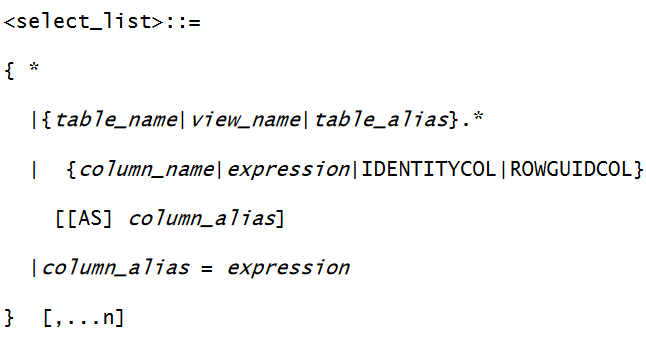
POLECENIE SELECT:

* Select\_list określa kolumny
* Klauzula WHERE specyfikuje warunki ograniczające zapytania
* Klauzula FROM określa tabele





Pierwszą rzeczą, którą wykonujemy po połączeniu z naszym serwerem jest wskazanie bazy danych, na której uruchamiane będą zapytania Query. Służy do tego polecenie **USE**



Rysunek 3. Wybór określonych kolumn tabeli

**Wybór kolumn do tabeli, przykład:**

SELECT CompanyName,ContactName, Address,City FROM Customers;

Zadanie 1 –wybór kolumn

1. Wybierz nazwy i adresy wszystkich klientów

2. Wybierz nazwiska i numery telefonów pracowników

3. Wybierz nazwy i ceny produktów

4. Pokaż wszystkie kategorie produktów (nazwy i opisy)

5. Pokaż nazwy i adresy stron www dostawców

# Operatory w MS SQL

* Operatory porównań =,>,<,>=,<=,<>=,>,<,>=,<=,<>
* Porównywanie stringów LIKE, NOT LIKE
* Operatory logiczne: AND, OR NOT
* Zakres BETWEEN, NOT BETWEEN
* Listy wartości IN, NOT IN
* Nieznane wartości IS NULL, IS NOT NULL

**Przykład na wykorzystanie klauzuli WHERE do wyboru określonych wierszy**

SELECT CompanyName,ContactName, Address,City FROM Customers

WHERE Country='Germany';

Zadanie 2 – wybór wierszy.

6. Wybierz nazwy i adresy wszystkich klientów mających siedziby w Londynie

7. Wybierz nazwy i adresy wszystkich klientów mających siedziby we Francji lub w Hiszpanii

8. Wybierz nazwy i ceny produktów o cenie jednostkowej pomiędzy 20.00 a 30.00

9. Wybierz nazwy i ceny produktów z kategorii ‘seafood’

10. Wybierz nazwy produktów oraz inf. o stanie magazynu dla produktów dostarczanych przez firmę .Tokyo Traders.

11. Wybierz nazwy produktów których nie ma w magazynie.

* Operator LIKE – porównywanie ze wzorcem
* LIKE tylko dla danych typu:
  + char, nchar, varchar, nvarchar, binary, varbinary, smalldatetime, datetime, oraz pod pewnymi założeniami dla text, ntext, image
* Znak specjalny

% 0 lub więcej znaków

\_ pojedynczy znak

[ ] pojedynczy znak z zakresu

[^] pojedynczy znak z poza zakresu

* + - LIKE 'BR%'
    - LIKE 'Br%'
    - LIKE '%een'
    - LIKE '%en%'
    - LIKE '\_en'
    - LIKE '[CK]%'
    - LIKE '[S-V]%'
    - LIKE 'M[^c]%'

**Przykład zastosowania operatora LIKE:**

SELECT CompanyName FROM Customers

WHERE CompanyName LIKE '%Restaurant%';

Zadanie 3 –porównanie napisów (stringów)

12. Szukamy informacji o produktach sprzedawanych w butelkach (bottle)

13. Wyszukaj informacje o stanowisku pracowników, których nazwiska zaczynają się na literę z zakresu od B do L

14. Wyszukaj informacje o stanowisku pracowników, których nazwiska zaczynają się na literę B lub L

15. Znajdź nazwy kategorii, które w opisie zawierają przecinek

16. Znajdź klientów, którzy w swojej nazwie mają w którymś miejscu słowo .Store.

**Przykład na użycie operatorów logicznych**

1. SELECT productid, productname, supplierid, unitprice

FROM products

WHERE (productname LIKE 'T%' OR productid = 46) AND (unitprice > 16);

1. SELECT productname, unitprice FROM products WHERE unitprice BETWEEN 10 AND 20;

Zadanie 4 –zakres wartości i warunki logiczne

17. Szukamy informacji o produktach o cenach mniejszych niż 10 lub większych niż 20

18. Wybierz nazwy i ceny produktów o cenie jednostkowej pomiędzy 20.00 a 30.00

Wybierz nazwy i kraje wszystkich klientów mających siedziby w Japonii (Japan) lub we Włoszech (Italy), wyniki posortuj według kraju, w ramach danego kraju nazwy firm posortuj alfabetycznie

**Przykład na wyszukiwanie wartości NULL**

SELECT companyname, fax FROM suppliers WHERE fax IS NULL;

### Zbiory wynikowe

* Sortowanie danych
* Eliminowanie duplikatów
* Zmian nazw kolumn
* Użycie literałów
* Kolumny wyliczane

# Sortowanie danych

Do sortowania wyników służy klauzula **order by**, po której wskazujemy kolumnę, która ma ulec procedurze sortowania. Dane możemy porządkować w kolejności rosnącej (**asc** - skrót od słowa ascending) lub malejącej (**desc** - skrót od słowa descending). W przypadku nie określenia kierunku sortowania, dane zostaną poukładane rosnąco.

Klauzula ta ma zastosowanie do kolumn zawierających dane zarówno typu tekstowego, liczbowego jak i daty.

Sortowania możemy używać po więcej niz jednej kolumnie, nalezy wtedy po **order by** wymienić kolejne kolumny, po których wynik ma zostać posortowany, oddzielając je przecinkami.

Zamiast nazw kolumn, można używać również aliasów oraz cyfr określających kolejność wymienionych kolumn po instrukcji **select**.

**Przykłady na sortowanie**

1. SELECT productid, productname, unitprice

FROM products

ORDER BY unitprice DESC;

1. SELECT productid, productname, categoryid, unitprice

FROM products

ORDER BY categoryid, unitprice DESC;

1. SELECT productid, productname, categoryid, unitprice

FROM products

ORDER BY 3,4 DESC

Zadanie 5 –sortowanie

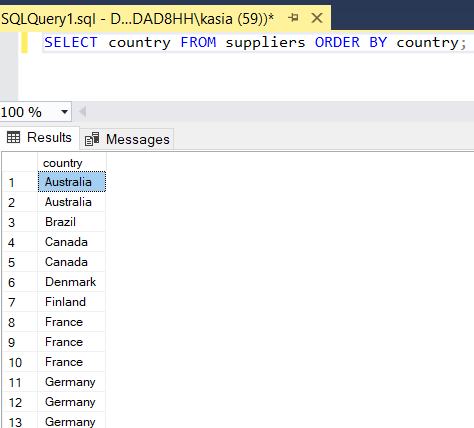
18. Wybierz nazwy i kraje wszystkich klientów, wyniki posortuj według kraju, w ramach danego kraju nazwy firm posortuj alfabetycznie

19. Wybierz informację o produktach (grupa, nazwa, cena), produkty posortuj wg grup a w grupach malejąco wg ceny

20. Wybierz nazwy i kraje wszystkich klientów mających siedziby w Japonii (Japan) lub we Włoszech (Italy), wyniki posortuj według kraju, w ramach danego kraju nazwy firm posortuj alfabetycznie

# Eliminowanie duplikatów

SELECT country FROM suppliers ORDER BY country;



W zbiorze wynikowym pojawiły się zduplikowane wiersze

**Polecenie na eliminowanie duplikatów -DISTINCT**

SELECT DISTINCT country FROM suppliers ORDER BY country;

**Aliasy –zmiana nazw kolumn - AS**

SELECT DISTINCT country AS kraj FROM suppliers ORDER BY country;

**Użycie literałów**

SELECT firstname, lastname,'Numer identyfikacyjny:', employeeid FROM employees;

**Kolumny wyliczeniowe**

1. SELECT orderid, unitprice \* 1.05 as nowa\_cena FROM [order details];

/\*cena powiększona o 5%\*/

1. SELECT firstname +' '+ lastname as imienazwisko FROM employees

# SQL SELECT TOP

Polecenie SELECT TOP służy do filtrowania wyników zapytania. Po słówku TOP wpisujemy wartość liczbową określającą ile pierwszych rekordów zwróconych przez zapytanie chcemy wyświetlić.

### SQL SELECT TOP składnia

**SELECT TOP** wartość\_liczbowa nazwy\_kolumy(kolumn)  
FROM  
nazwa\_tabeli  
ORDER BY  
nazwa\_kolumny1 ASC , nazwa\_kolumny2 DESC

**Przykład - TOP**

Wyświetl wszystkie produkty (tylko kolumny: ID, Nazwa Produktu, Kategoria Produktu) wraz z ich kategoriami. Wyniki posortuj wg nazwy produktu (rosnąco) a następnie wg nazwy kategorii (malejąco). Wyświetl pierwszych 10 rekordów zwróconych przez zapytanie.

SELECT TOP 10 P.ProductID ,P.ProductName ,C.CategoryName  
FROM  
Products P INNER JOIN Categories C

ON (P.CategoryID = C.CategoryID)  
ORDER BY  
ProductName ASC, CategoryName DESC

Zadanie.

1. Wyświetl wg wzoru: Nazwa firmy: CompanyName pochodzi z kraju: country. Posortuj rosnąco po kraju.
2. Zmniejsz cenę zamówień o 25%, nazwij to pole pomniejszona\_cena z tabeli szczegóły zamówień. Posortuj malejąco po pomniejszonej cenie.
3. Wyświetl 15 najdroższych produktów.
4. Wyświetl 10 najtańszych produktów.