

Bazy danych, ćwiczenia 6.

Język SQL, zadania.

Zadania będą dotyczyły bazy danych Northwind.

Zadanie 1.

Proszę wypisać wszystkich klientów z Polski i z Niemiec.

Zadanie 2.

Proszę wypisać wszystkich klientów, których nazwa rozpoczyna się od litery N oraz tych, których nazwa rozpoczyna się od litery C.

Zadanie 3.

Proszę wypisać dane wszystkich towarów, które są z kategorii o nazwach rozpoczynających się na literę C.

Zadanie 4.

Proszę wypisać następujące dane z zamówień: OrderID, OrderDate, ProductID, ProductName, UnitPrice (z tabeli [Order Details]), Quantity, CategoryName, CustomerID, CompanyName. Ile wierszy ma taki zestaw danych?

Zadanie 5.

Proszę wypisać wszystkich klientów, którzy nie mają wpisane nic do kolumny Region.

Zadanie 6.

Proszę wypisać wszystkie kraje, z których mamy klientów.

Zadanie 7.

Proszę wypisać wszystkie produkty kategorii o nazwie 'Beverages' (napoje) o cenie jednostkowej większej bądź równej niż 20 i mniejszej bądź równej 30.

Zadanie 8.

Proszę wypisać wszystkich klientów, którzy jeszcze nic nie zamówili. Wskazówka: należy połączyć tabelę Customers z tabelą Orders przez złączenie zewnętrzne (np. LEFT JOIN) i dodać odpowiednią klauzulę WHERE.

Zadanie 9.

Proszę wypisać wszystkich klientów, którzy jeszcze nic nie zamówili. Należy wykorzystać złożenie operatorów NOT oraz IN.

Zadanie 10.

Proszę wypisać wszystkich klientów, którzy jeszcze nic nie zamówili. Należy wykorzystać złożenie operatorów NOT oraz EXISTS.

Zadanie 11.

Proszę wypisać ilu mamy klientów z Polski i z Niemiec (zestaw ma mieć dwie kolumny: kraj i liczbę klientów).

Zadanie 12.

Proszę wypisać z ilu krajów mamy klientów (zdanie ma wypisać jedną liczbę).

Zadanie 13.

Proszę wypisać dla każdego dnia ile przyjęto w ciągu tego dnia zamówień.

Zadanie 14.

Proszę wypisać, ile złożono zamówień na każdą kategorię towarów.

Zadanie 15.

Proszę wypisać dla każdego towaru ilu klientów złożyło zamówienie na ten towar.

Zadanie 16.

Proszę wypisać najwyższe i najniższe ceny towarów w każdej kategorii, przy czym chodzi o towary o cenach wyższych od 30.

Zadanie 17.

Proszę wypisać dla każdego dnia jaka jest suma kwot na wszystkich zamówieniach w danym dniu.

Zadanie 18.

Proszę zmodyfikować rozwiązanie poprzedniego zadania tak, by w każdym dniu pokazana była suma kwot uzyskanych za zamówienia poszczególnych produktów.

Zadanie 19.

Proszę wypisać identyfikatory klientów (może być jeden taki klient lub więcej), którzy zamówili złożyli zamówienia w sumie na największą kwotę.

Zadanie 20.

Proszę wypisać wszystkie dane klientów (może być jeden taki klient lub więcej), którzy zamówili złożyli zamówienia w sumie na największą kwotę.

Zadanie 21.

Proszę wypisać najwyższe i najniższe ceny towarów w każdej kategorii. Przy każdej cenie należy również podać identyfikator i nazwę towaru o tej cenie (może być kilka towarów o jednakowej cenie, należy je wszystkie uwzględnić).

Zadanie 22.

Proszę wypisać trzy najwyższe ceny w tabeli Products. Proszę wykorzystać konstrukcję TOP 3. Czyż różni się TOP 3 od TOP 3 WITH TIES?

Zadanie 23.

Proszę wypisać trzy najwyższe ceny w tabeli Products. Tym razem proszę nie wykorzystywać konstrukcji z TOP 3.

Należy wykorzystać operator UNION:

```
SELECT ... FROM Tab1  
UNION
```

```
SELECT ... FROM Tab2
```

Uwaga: jeśli pierwszy i drugi SELECT zwracają takie same wiersze (przynajmniej po jednym), to powtórzenia nie są wypisywane.

Jest też możliwość wypisania wszystkich wierszy łącznie z powtórzeniami:

```
SELECT ... FROM Tab1  
UNION ALL  
SELECT ... FROM Tab2
```

Wskazówka jak rozwiązać zadanie:

```
SELECT MAX(UnitPrice) FROM Products  
UNION  
SELECT MAX(UnitPrice) FROM Products ... WHERE  
UNION  
....
```

Zadanie 24.

Założmy, że mamy tabelę T1 (K1 INT PRIMARY KEY, K2 INT), do której wprowadzono rekordy (1,100), (2,200), (3,300), (4,400), (5,500). Co wypisze następujące zdanie SQL?

```
SELECT MAX(A.K2), MAX(B.K2), MAX(C.K2)  
FROM T1 AS A, T1 AS B, T1 AS C  
WHERE (A.K2>B.K2) AND (B.K2>C.K2)
```

Zadanie 25.

Korzystając z rozwiązania poprzedniego zadania, proszę w podobny sposób wypisać trzy najwyższe ceny z tabeli Products.

Zadanie 26.

Co wypisze następujące zdanie SQL?

```
SELECT DISTINCT Stawka FROM Pracownicy  
WHERE Stawka >=  
  (SELECT MAX(Stawka) FROM Pracownicy  
   WHERE Stawka < (SELECT MAX(Stawka) FROM Pracownicy)  
   WHERE Stawka < (SELECT MAX(Stawka) FROM Pracownicy)))
```

Zadanie 27.

Korzystając z rozwiązania poprzedniego zadania, proszę w podobny sposób wypisać trzy najwyższe ceny z tabeli Products.

Zadanie 28.

Co wypisze następujące zdanie SQL?

```
SELECT DISTINCT COUNT(*), A.Stawka
FROM Pracownicy A, Pracownicy B
WHERE (A.Stawka <= B.Stawka)
GROUP BY A.Stawka
HAVING COUNT(*) <= 3
```

Zadanie 29.

Korzystając z rozwiązania poprzedniego zadania, proszę w podobny sposób wypisać trzy najwyższe ceny z tabeli Products.

Zadanie 30.

Proszę przeglądnąć dane w tabeli Employees. Kolumna ReportsTo zawiera identyfikator zwierzchnika danego pracownika.

```
SELECT * FROM Employees
SELECT EmployeeID, LastName, FirstName, ReportsTo
FROM Employees
```

Co wyświetlą następujące zdanie SQL?

```
SELECT ReportsTo AS Szef, EmployeeID AS Podwładny
FROM Employees
```

```
SELECT ReportsTo AS Szef, EmployeeID AS Podwładny
FROM Employees
WHERE ReportsTo IS NOT NULL
```

Proszę wyświetlić zestaw rekordów podobny do powyższego, jednak tym razem ma być podane nazwisko (LastName) i imię (FirstName) zarówno szefa, jak i podwładnego.

Zestaw rekordów ma zawierać pięć kolumn:

NazwiskoSzefa, ImięSzefa, ZależnośćSłużbowa (w tej kolumnie będzie stały napis 'jest szefem pracownika: '), NazwiskoPodwładnego, ImięPodwładnego.

Wskazówka – należy złączyć tabelę Employees z samą sobą (tj. z tabelą Employees). Jest to tzw. autozłączenie:

```
Employees AS E1 JOIN Employees AS E2 ON ...
```

Zadanie 31.

Proszę przeanalizować dwie wersje instrukcji CASE:

```
/*
```

Simple CASE expression:

```
CASE input_expression
```

```
    WHEN when_expression THEN result_expression [ ...n ]
```

```
    [ ELSE else_result_expression ]
```

```
END
```

Searched CASE expression:

```
CASE
  WHEN Boolean_expression THEN result_expression [ ...n ]
  [ ELSE else_result_expression ]
END
*/
```

```
SELECT CompanyName,Region
FROM Customers
ORDER BY Region
```

Zadanie: wykorzystując CASE posortować dane tak, by NULL były na końcu, ale sortowanie generalnie ma być rosnące według identyfikatora regionu.

Rozwiązanie:

```
SELECT CompanyName,Region
FROM Customers
ORDER BY CASE
  WHEN Region IS NULL THEN 2
  ELSE 1
END, Region
```