*Anhelina Belavezha*

# Bazy danych, ćwiczenia 6

```sql
--1
SELECT * FROM Customers WHERE Country = 'Germany' OR Country = 'Poland'

--2
SELECT * FROM Customers WHERE CompanyName LIKE 'N%' OR CompanyName LIKE 'C%'

--3
SELECT * FROM Products WHERE ProductName LIKE 'C%'

--4
SELECT
Orders.OrderID, Orders.OrderDate, [Order Details].ProductID, Products.ProductName,
[Order Details].UnitPrice, Quantity, CategoryName, Orders.CustomerID, CompanyName
FROM
Orders LEFT JOIN [Order Details] ON Orders.OrderID = [Order Details].OrderID
LEFT JOIN Products ON [Order Details].ProductID = Products.ProductID
LEFT JOIN Categories ON Categories.CategoryID = Products.CategoryID
LEFT JOIN Customers ON Customers.CustomerID = Orders.CustomerID

--5
SELECT LastName, FirstName FROM Employees WHERE Region IS NULL

--6
SELECT DISTINCT Country FROM Customers WHERE  Region is NOT NULL

--7
SELECT * FROM Products
WHERE CategoryID IN (
SELECT CategoryID FROM Categories WHERE CategoryName = 'Beverages')
AND UnitPrice BETWEEN 20 AND 30

--8
SELECT * FROM Customers
LEFT JOIN Orders ON Customers.CustomerID = Orders.CustomerID WHERE Orders.OrderID
IS NULL

--9
SELECT * FROM Customers WHERE CustomerID NOT IN (SELECT Orders.CustomerID FROM
Orders)

--10
SELECT * FROM Customers WHERE NOT EXISTS
    (SELECT * FROM Orders WHERE Orders.CustomerID = Customers.CustomerID)

--11
SELECT Country, COUNT(Country) AS liczba FROM Customers
WHERE Country = 'Poland' OR Country = 'Germany' GROUP BY Country

--12
SELECT COUNT(DISTINCT Country) FROM Customers

--13
SELECT YEAR(OrderDate) AS YEAR, MONTH(OrderDate) AS MONTH, DAY(OrderDate) AS DAY,
COUNT(OrderID) FROM Orders GROUP BY YEAR(OrderDate), MONTH(Orderdate),
DAY(OrderDate)
```

```sql
--14
SELECT Categories.CategoryID, COUNT([Order Details].OrderID) FROM Categories
LEFT JOIN Products ON Categories.CategoryID = Products.CategoryID
LEFT JOIN [Order Details] ON [Order Details].ProductID = Products.ProductID
GROUP BY Categories.CategoryID

--15
SELECT Products.ProductID, COUNT(DISTINCT [Order Details].OrderID)
FROM Products
LEFT JOIN [Order Details] ON [Order Details].ProductID = Products.ProductID
GROUP BY Products.ProductID

--16
SELECT Categories.CategoryID, MIN(Products.UnitPrice) AS MI,
MAX(Products.UnitPrice) AS MA
FROM Categories INNER JOIN Products ON Categories.CategoryID = Products.CategoryID
WHERE Products.UnitPrice > 30 GROUP BY Categories.CategoryID

--17
SELECT YEAR(OrderDate) AS YEAR, MONTH(OrderDate) AS MONTH, DAY(OrderDate) AS DAY,
SUM([Order Details].UnitPrice * [Order Details].Quantity) AS SUM
FROM Orders INNER JOIN [Order Details] ON [Order Details].OrderID = Orders.OrderID
GROUP BY YEAR(OrderDate), MONTH(OrderDate), DAY(OrderDate)

--18
SELECT YEAR(OrderDate) AS YEAR, MONTH(OrderDate) AS MONTH, DAY(OrderDate) AS DAY,
[Order Details].ProductID,
SUM([Order Details].UnitPrice * [Order Details].Quantity) AS SUM
FROM Orders INNER JOIN [Order Details] ON [Order Details].OrderID = Orders.OrderID
GROUP BY YEAR(OrderDate), MONTH(OrderDate), DAY(OrderDate), [Order
Details].ProductID

--19

SELECT Customers.CustomerID FROM Customers
JOIN Orders ON Customers.CustomerID = Orders.CustomerID
JOIN [Order Details] ON Orders.OrderID = [Order Details].OrderID
GROUP BY Customers.CustomerID
HAVING SUM([Order Details].UnitPrice * [Order Details].Quantity) =
(
SELECT TOP 1 SUM([Order Details].UnitPrice * [Order Details].Quantity) AS S
FROM Customers
JOIN Orders ON Customers.CustomerID = Orders.CustomerID
JOIN [Order Details] ON Orders.OrderID = [Order Details].OrderID
GROUP BY Customers.CustomerID
ORDER BY SUM([Order Details].UnitPrice * [Order Details].Quantity) DESC
)

--20

WITH TOPMAN AS
(SELECT Customers.CustomerID
FROM Customers
JOIN Orders ON Customers.CustomerID = Orders.CustomerID
JOIN [Order Details] ON Orders.OrderID = [Order Details].OrderID
GROUP BY Customers.CustomerID
HAVING SUM([Order Details].UnitPrice * [Order Details].Quantity) =
(
       SELECT TOP 1 SUM([Order Details].UnitPrice * [Order Details].Quantity) AS S
       FROM Customers
       JOIN Orders ON Customers.CustomerID = Orders.CustomerID
       JOIN [Order Details] ON Orders.OrderID = [Order Details].OrderID
       GROUP BY Customers.CustomerID
```

```
      ORDER BY SUM([Order Details].UnitPrice * [Order Details].Quantity) DESC
))
SELECT Customers.* FROM Customers
JOIN TOPMAN ON Customers.CustomerID = TOPMAN.CustomerID

--21

WITH RankedProducts AS (
    SELECT
        Products.CategoryID,
        Products.ProductID,
        Products.ProductName,
        Products.UnitPrice,
        RANK() OVER (PARTITION BY Products.CategoryID ORDER BY Products.UnitPrice)
AS PriceRankA,
        RANK() OVER (PARTITION BY Products.CategoryID ORDER BY Products.UnitPrice
DESC) AS PriceRankD
    FROM
        Products
)
SELECT
    CategoryID,
    ProductID,
    ProductName,
    UnitPrice AS VAL
FROM RankedProducts
WHERE PriceRankA = 1 OR PriceRankD = 1

--22

SELECT TOP 3 * FROM Products ORDER BY UnitPrice DESC

--23

SELECT MAX(UnitPrice) AS MaxUnitPrice FROM Products
UNION
SELECT MAX(UnitPrice) FROM Products WHERE UnitPrice < (SELECT MAX(UnitPrice) FROM
Products)
UNION
SELECT MAX(UnitPrice) FROM Products WHERE UnitPrice < (SELECT MAX(UnitPrice) FROM
Products WHERE UnitPrice < (SELECT MAX(UnitPrice) FROM Products));
WITH RankedProducts AS (
    SELECT
        UnitPrice,
        ROW_NUMBER() OVER (ORDER BY UnitPrice DESC) AS RowNumber
    FROM Products
)
SELECT UnitPrice FROM RankedProducts
WHERE RowNumber <= 3

--24

(500, 400, 300)

--25

SELECT MAX(A.UnitPrice) AS M1, MAX(B.UnitPrice) AS M2, MAX(C.UnitPrice) AS M3
FROM Products AS A, Products AS B, Products AS C
WHERE (A.UnitPrice > B.UnitPrice) AND (B.UnitPrice > C.UnitPrice)

--26

Wygeneruje 3 wartości maksymalne z kolumny „Stawka”

--27

SELECT DISTINCT UnitPrice FROM Products
```

```
WHERE UnitPrice >=
 (SELECT MAX(UnitPrice) FROM Products
 WHERE UnitPrice < (SELECT MAX(UnitPrice) FROM Products
 WHERE UnitPrice < (SELECT MAX(UnitPrice) FROM Products)))

--28
```

Maksymalnie 3 wartości z tej kolumny, ale jeśli mamy powtórzenia, to nie możemy określić wyniku

```
--29

SELECT DISTINCT COUNT(*), A.UnitPrice FROM Products A, Products B
WHERE (A.UnitPrice <= B.UnitPrice)
GROUP BY A.UnitPrice
HAVING COUNT(*) <= 3

--30

SELECT
    E1.LastName AS BossSurname,
    E1.FirstName AS BossName,
    'jest szefem pracownika ',
    E2.LastName AS SubordinateSurname,
    E2.FirstName AS SubordinateName
FROM Employees AS E1
JOIN Employees AS E2 ON E2.ReportsTo = E1.EmployeeID
```

Pierwszy wyświetli kilka wierszy z 2 kolumnami Szef i Podwladny, gdzie szef jest bezpośrednim szefem podwładnego, Drugi pominie wiersze, w których szef jest równy NULL

```
--31

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