

2. sprawozdanie z laboratorium Hurtownie Danych

Mikołaj Kubś, 272662

11 marca 2025

1 Zadanie 1. Ekstrakcja danych

1.1

Utworzyć zestawienie, które dla poszczególnych miesięcy i lat przedstawi informację o liczbie różnych klientów. Przygotuj zapytanie z i bez użycia polecenia pivot.

1.1.1 Wersja bez pivot

```
1 SELECT
2 YEAR(OrderDate),
3 MONTH(OrderDate),
4 COUNT(DISTINCT CustomerID)
5 FROM Sales.SalesOrderHeader
6 GROUP BY YEAR(OrderDate), MONTH(OrderDate)
7 ORDER BY YEAR(OrderDate), MONTH(OrderDate)
```

1.1.2 Wersja z użyciem pivot

```
1 WITH UniqueCustomers AS (
2     SELECT
3         YEAR(OrderDate) AS OrderYear,
4         MONTH(OrderDate) AS OrderMonth,
5         CustomerID
6     FROM Sales.SalesOrderHeader
7     GROUP BY YEAR(OrderDate), MONTH(OrderDate), CustomerID
8 )
```

```

9  SELECT * FROM UniqueCustomers
10 PIVOT (
11     COUNT(CustomerID)
12     FOR OrderMonth IN ([1], [2], [3], [4],
13                        [5], [6], [7], [8],
14                        [9], [10], [11], [12])
15 ) AS PivotTable
16 ORDER BY OrderYear;

```

1.2

Utworzyć zestawienie zawierające w wierszach imiona i nazwiska sprzedawców, a w kolumnach kolejne lata. Wartością będzie liczba obsłużonych transakcji. Wyświetlić tylko tych sprzedawców, którzy pracowali przez wszystkie 4 lata.

```

1  SELECT * FROM
2  (
3      SELECT
4          FirstName, LastName, SalesOrderID,
5          YEAR(OrderDate) AS OrderYear FROM Sales.SalesPerson
6      JOIN HumanResources.Employee ON
7          Employee.BusinessEntityID = SalesPerson.BusinessEntityID
8      JOIN Person.Person ON
9          Person.BusinessEntityID = Employee.BusinessEntityID
10     JOIN Sales.SalesOrderHeader ON
11         SalesOrderHeader.SalesPersonID = SalesPerson.BusinessEntityID
12     WHERE YEAR(HireDate) = 2011
13 ) AS SourceTable
14 PIVOT (
15     COUNT(SalesOrderID)
16     FOR OrderYear IN ([2011], [2012], [2013], [2014])
17 ) AS PivotedTable
18 ORDER BY FirstName

```

1.3

Zdefiniować zapytanie wyznaczające sumę kwot sprzedaży towarów oraz liczbę różnych produktów w zamówieniach w poszczególnych latach, miesiącach, dniach.

```

1 SELECT
2     YEAR(OrderDate) AS "Rok",
3     MONTH(OrderDate) AS "Miesiąc",
4     DAY(OrderDate) AS "Dzień",
5     SUM(LineTotal) AS "Suma",
6     COUNT(DISTINCT ProductID) AS "Liczba_różnych_produktyw"
7 FROM Sales.SalesOrderHeader
8 JOIN Sales.SalesOrderDetail ON
9     SalesOrderDetail.SalesOrderID = SalesOrderHeader.SalesOrderID
10 GROUP BY YEAR(OrderDate), MONTH(OrderDate), DAY(OrderDate)
11 ORDER BY YEAR(OrderDate), MONTH(OrderDate), DAY(OrderDate)

```

1.4

Wykorzystując polecenie CASE przygotować podsumowania do zestawienia z poprzedniego zadania tak, aby sumowane były kwoty zamówień oraz obliczana liczba różnych produktów dla poszczególnych miesięcy i dni tygodnia. Uwaga: Pamiętaj o wybraniu właściwego atrybutu funkcji datepart tak, aby zgadzała się nazwa dnia tygodnia

```

1 SET DATEFIRST 1;
2 SET LANGUAGE Polish;
3
4 SELECT
5     YEAR(OrderDate) AS "Rok",
6     DATENAME(month, OrderDate) AS "Miesiąc",
7     CASE DATEPART(dw, OrderDate)
8         WHEN 1 THEN 'Poniedziałek'
9         WHEN 2 THEN 'Wtorek'
10        WHEN 3 THEN 'Środa'
11        WHEN 4 THEN 'Czwartek'
12        WHEN 5 THEN 'Piątek'
13        WHEN 6 THEN 'Sobota'
14        WHEN 7 THEN 'Niedziela'
15    END AS "Dzień_tygodnia",
16     SUM(LineTotal) AS "Suma",
17     COUNT(DISTINCT ProductID) AS "Liczba_różnych_produktyw"
18 FROM Sales.SalesOrderHeader
19 JOIN Sales.SalesOrderDetail ON
20     SalesOrderDetail.SalesOrderID = SalesOrderHeader.SalesOrderID
21 GROUP BY

```

```

22     YEAR(OrderDate),
23     DATENAME(month, OrderDate),
24     MONTH(OrderDate),
25     DATEPART(dw, OrderDate)
26 ORDER BY
27     YEAR(OrderDate),
28     MONTH(OrderDate),
29     DATEPART(dw, OrderDate)

```

1.5

Przygotować zestawienie, w którym dla wybranych klientów przygotujemy kartę lojalnościową:

- srebrną, jeśli klient wykonał co najmniej 2 transakcje w sklepie;
- złotą, jeśli wykonał co najmniej 4 transakcje w sklepie, w tym co najmniej 2 transakcje, których łączna kwota przekraczała 250% średniej wartości zamówień w bazie;
- platynową, jeśli klient spełniał warunki otrzymania karty złotej oraz w co najmniej jednej transakcji kupił jednocześnie produkty ze wszystkich kategorii

```

1  WITH AvgOrderValue AS (
2      SELECT AVG(TotalOrderValue) AS AvgValue
3      FROM (
4          SELECT SalesOrderID, SUM(LineTotal) AS TotalOrderValue
5          FROM Sales.SalesOrderDetail
6          GROUP BY SalesOrderID
7      ) AS OrderValues
8  ),
9  CustomerOrders AS (
10     SELECT
11         SOH.CustomerID,
12         COUNT(DISTINCT SOH.SalesOrderID) AS TransactionCount,
13         SUM(SOD.LineTotal) AS TotalTransactionValue,
14         SUM(CASE WHEN OrderValues.TotalOrderValue > 2.5 * A.AvgValue
15             THEN 1 ELSE 0 END) AS HighValueOrderCount
16     FROM Sales.SalesOrderHeader SOH
17     JOIN Sales.SalesOrderDetail SOD ON SOD.SalesOrderID = SOH.SalesOrderID
18     JOIN (
19         SELECT SalesOrderID, SUM(LineTotal) AS TotalOrderValue
20         FROM Sales.SalesOrderDetail

```

```

21         GROUP BY SalesOrderID
22     ) AS OrderValues ON SOH.SalesOrderID = OrderValues.SalesOrderID
23     CROSS JOIN AvgOrderValue A
24     GROUP BY SOH.CustomerID
25 ),
26 UniqueCategories AS (
27     SELECT
28         C.CustomerID,
29         COUNT(DISTINCT PC.ProductCategoryID) AS CategoryCount
30     FROM Sales.Customer C
31     JOIN Sales.SalesOrderHeader SOH ON SOH.CustomerID = C.CustomerID
32     JOIN Sales.SalesOrderDetail SOD ON SOD.SalesOrderID = SOH.SalesOrderID
33     JOIN Production.Product PR ON PR.ProductID = SOD.ProductID
34     JOIN Production.ProductSubcategory PSC ON
35         PSC.ProductSubcategoryID = PR.ProductSubcategoryID
36     JOIN Production.ProductCategory PC
37         ON PC.ProductCategoryID = PSC.ProductCategoryID
38     GROUP BY C.CustomerID
39 )
40 SELECT
41     P.FirstName AS "Imię",
42     P.LastName AS "Nazwisko",
43     COALESCE(CO.TransactionCount, 0) AS "Liczba_transakcji",
44     COALESCE(CO.TotalTransactionValue, 0) AS "Łączna_kwota_transakcji",
45     CASE
46         WHEN CO.TransactionCount >= 4
47             AND CO.HighValueOrderCount >= 2
48             AND COALESCE(UC.CategoryCount, 0) =
49                 (SELECT COUNT(*) FROM Production.ProductCategory)
50             THEN 'Platynowa'
51         WHEN CO.TransactionCount >= 4
52             AND CO.HighValueOrderCount >= 2
53             THEN 'Złota'
54         WHEN CO.TransactionCount >= 2
55             THEN 'Srebrna'
56         ELSE 'Brak_karty'
57     END AS "Kolor_karty"
58 FROM Sales.Customer C
59 LEFT JOIN CustomerOrders CO ON CO.CustomerID = C.CustomerID
60 LEFT JOIN UniqueCategories UC ON UC.CustomerID = C.CustomerID
61 JOIN Person.Person P ON P.BusinessEntityID = C.CustomerID

```

```
62 ORDER BY CO.TransactionCount DESC;
```