ZAD 1

Wykorzystując wyrażenie CTE zbuduj zapytanie, które znajdzie informacje na temat stawki

pracownika oraz jego danych, a następnie zapisze je do tabeli tymczasowej

TempEmployeeInfo. Rozwiąż w oparciu o AdventureWorks

**CTE – do tworzenia tymczasowych wyników zapytań. Definicja CTE (nazwa+kolumny na wynik) + Zapytanie CTE**

USE AdventureWorks2019

***Dane pracownika i jego płaca:***

SELECT \* FROM AdventureWorks2019.Person.Person

SELECT \* FROM AdventureWorks2019.HumanResources.EmployeePayHistory

***Tworzenie CTE. Łączy Person+Employee kolumną ID. Wybiera 4 kolumny i daje nowe nazwy***

WITH zapytanie (id, imie, nazwisko, stawka)

AS

(

SELECT p.BusinessEntityID, FirstName, LastName, Rate

FROM AdventureWorks2019.Person.Person p

INNER JOIN AdventureWorks2019.HumanResources.EmployeePayHistory e

ON p.BusinessEntityID=e.BusinessEntityID

)

***Wpisuje wyniki CTE do tabeli tymczasowej***

SELECT \* INTO TempEmployeeInfo FROM zapytanie;

***Wynik operacji***

SELECT \* FROM TempEmployeeInfo;

SELECT \* FROM AdventureWorks2019.HumanResources.EmployeePayHistory

ZAD 2

Uzyskaj informacje na temat przychodów ze sprzedaży według firmy i kontaktu (za pomocą

CTE i bazy AdventureWorksL). Wynik powinien wyglądać następująco:

SELECT \*FROM SalesLT.SalesOrderHeader;

SELECT \*FROM SalesLT.Customer;

***Tworzę CTE2. Łączę Customer I SalesOrderHeader na kolumnie ID. Concat łączy kilka kolumn w jedną. Te nawiasy to żeby się zgadzało ze wzorem polecenia***

WITH zapytanie2 (CompanyContact, Revenue)

AS

(

SELECT CONCAT( CompanyName, ' (', FirstName, ' ', LastName, ')' ), TotalDue

FROM SalesLT.Customer c

INNER JOIN SalesLT.SalesOrderHeader s

ON c.CustomerID=s.CustomerID

)

***Wyświetlam wynik operacji***

SELECT \* FROM zapytanie2

ZAD 3

Napisz zapytanie, które zwróci wartość sprzedaży dla poszczególnych kategorii produktów.

Wykorzystaj CTE i bazę AdventureWorksLT.

***Zawartość kolumn z których korzystam***

SELECT \* FROM SalesLT.ProductCategory

SELECT \* FROM SalesLT.Product

SELECT \* FROM SalesLT.SalesOrderDetail

***Tworzę CTE3. Łączę 3 tabele na kolumnach ID***

WITH zapytanie3 (Category, SalesValue)

AS

(

SELECT c.Name, UnitPrice

FROM SalesLT.Product p

INNER JOIN SalesLT.ProductCategory c

ON p.ProductCategoryID=c.ProductCategoryID

INNER JOIN SalesLT.SalesOrderDetail o

ON p.ProductID=o.ProductID

)

***Wyświetlam wynik CTE3***

SELECT \* FROM zapytanie3

***Wyświetlam kolumny, zmieniam nazwę tej drugiej i grupuję.***

SELECT Category, SUM(SalesValue) AS SalesValue

FROM zapytanie3

***Używam group by żeby posortować alfabetycznie. Mogłabym też order by (?)***

GROUP BY Category