|  |  |
| --- | --- |
| Indeks | Marcin Blicharski 258996, Adam Bednarski 260338 |
| Grupa |  |

# Zadanie 1

## A

SELECT Product.ProductID AS ID, Product.Name AS Nazwa, Product.Color AS Kolor, Product.ListPrice AS Cena, Product.ListPrice - Product.StandardCost AS Zysk,

CASE

WHEN Product.Class = 'H' THEN 'High'

WHEN Product.Class = 'M' THEN 'Medium'

WHEN Product.Class = 'L' THEN 'Low'

END AS Klasa,

CASE

WHEN Product.ProductLine = 'R' THEN 'Road'

WHEN Product.ProductLine = 'M' THEN 'Mountain'

WHEN Product.ProductLine = 'T' THEN 'Touring'

WHEN Product.ProductLine = 'S' THEN 'Standard'

END AS Typ

FROM Production.Product

ORDER BY Product.Name

Obraz zawierający tekst

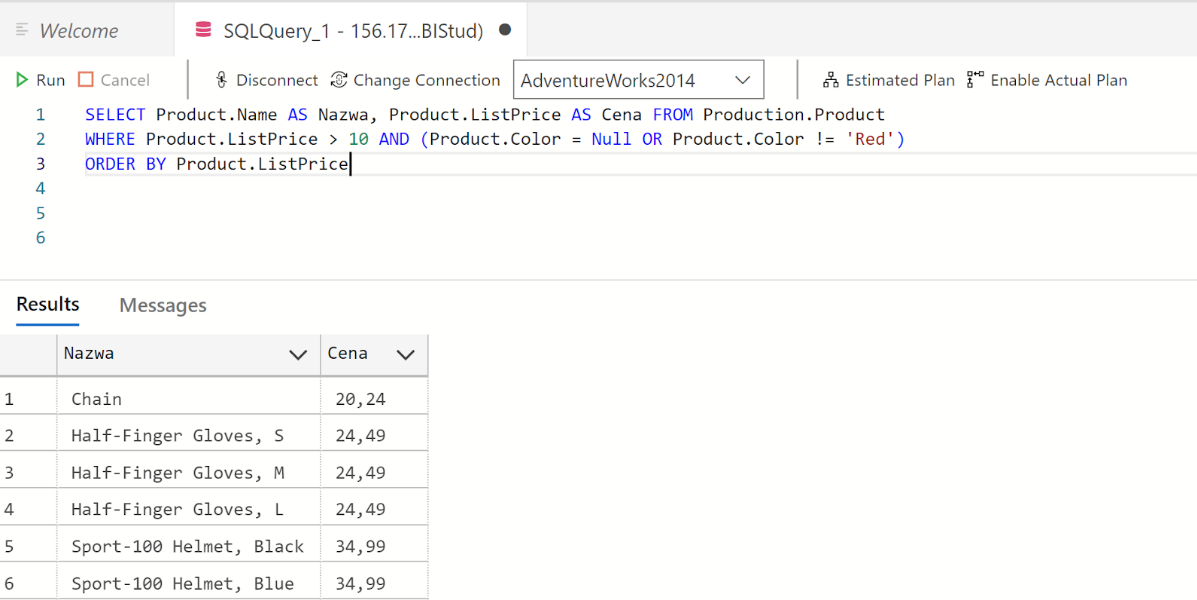
Opis wygenerowany automatycznie

## B

SELECT Product.Name AS Nazwa, Product.ListPrice AS Cena FROM Production.Product

WHERE Product.ListPrice > 10 AND (Product.Color = Null OR Product.Color != 'Red')

ORDER BY Product.ListPrice



## C

SELECT Product.Name AS Nazwa, ProductSubcategory.Name AS Podkategoria FROM Production.Product

LEFT JOIN Production.ProductSubcategory ON Production.Product.ProductSubcategoryID = Production.ProductSubcategory.ProductSubcategoryID

Obraz zawierający tekst

Opis wygenerowany automatycznie

## D

SELECT Product.Name AS Nazwa, ProductCategory.Name AS Kategoria, ProductSubcategory.Name AS Podkategoria FROM Production.Product

LEFT JOIN Production.ProductSubcategory ON Production.Product.ProductSubcategoryID = Production.ProductSubcategory.ProductSubcategoryID

LEFT JOIN Production.ProductCategory ON Production.ProductCategory.ProductCategoryID = Production.ProductSubcategory.ProductCategoryID

Obraz zawierający stół

Opis wygenerowany automatycznie

## E

SELECT ProductCategory.Name AS Kategoria, ProductSubcategory.Name AS Podkategoria, AVG(ListPrice) AS Srednia\_cena, COUNT(Product.Name) AS Liczba\_produktow

FROM Production.Product

LEFT JOIN Production.ProductSubcategory ON Production.Product.ProductSubcategoryID = Production.ProductSubcategory.ProductSubcategoryID

LEFT JOIN Production.ProductCategory ON Production.ProductCategory.ProductCategoryID = Production.ProductSubcategory.ProductCategoryID

GROUP BY ProductSubcategory.Name, ProductCategory.Name

Obraz zawierający stół

Opis wygenerowany automatycznie

## F

SELECT

Production.Product.Name AS Nazwa,

Production.Product.ListPrice as Cena,

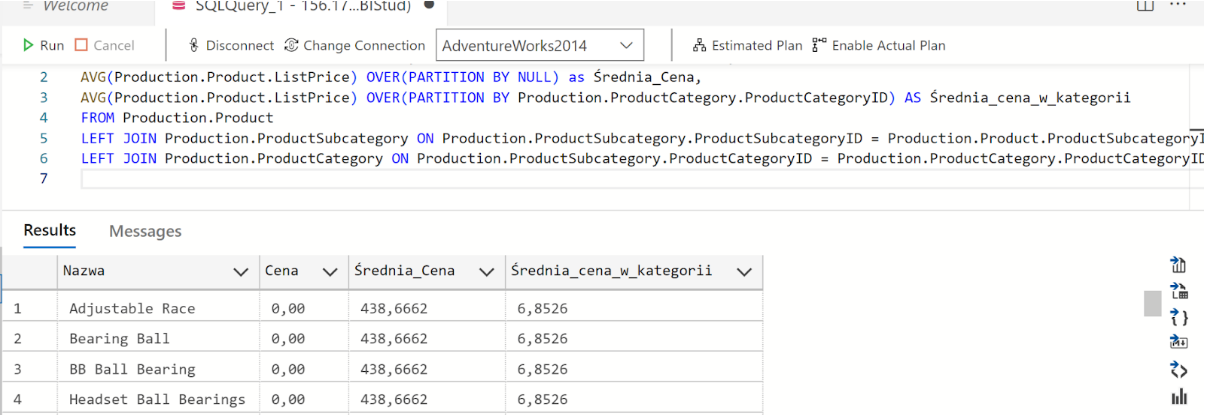
AVG(Production.Product.ListPrice) OVER(PARTITION BY NULL) AS Średnia\_Cena,

AVG(Production.Product.ListPrice) OVER(PARTITION BY Production.ProductCategory.ProductCategoryID) AS Średnia\_cena\_w\_kategorii

FROM Production.Product

LEFT JOIN Production.ProductSubcategory ON Production.ProductSubcategory.ProductSubcategoryID = Production.Product.ProductSubcategoryID

LEFT JOIN Production.ProductCategory ON Production.ProductSubcategory.ProductCategoryID = Production.ProductCategory.ProductCategoryID



## Konkluzje

Podczas łączenia tabel w podpunktach c, d, e, f użyty został LEFT JOIN zamiast INNER JOIN ze względu na to, że dla niektórych atrybutów baza danych zawiera wartości NULL, a dzięki łączeniu lewostronnemu (lub prawostronnemu jeśli łączylibyśmy tabele odwrotnie) tracimy mniej rekordów w wyniku złączenia tabel.

W podpunkcie f) użyta została klauzula OVER wraz z PARTITION BY w celu określenia kolumny, na której chcemy wykonać agregację (alternatywa dla klauzuli GROUP BY, która tutaj by nie zadziałała).

# Zadanie 2

import pymssql

import pandas as pd

server = '156.17.130.185:23333'

user = 'BIStud'

password = 'BIStudP@s'

conn = pymssql.connect(server, user, password, "AdventureWorks2014")

sql\_query = pd.read\_sql\_query('''

                              SELECT TerritoryID, Name, CountryRegionCode, SalesYTD, SalesLastYear

                              FROM Sales.SalesTerritory

                              '''

                              , conn)

df = pd.DataFrame(sql\_query)

print(df)

df.to\_csv('sales-territory.csv', index=False, sep=';')

conn.close()

## Konkluzje

Do wykonania zadania 2 użyte zostały dwie bilblioteki: biblioteka pymssql, dzięki której udało się połączyć z serwerem SQL oraz biblioteka pandas, która pozwoliła na zapisanie pliku .csv w odpowiedniej strukturze. Bez użycia biblioteki pandas dane w pliku zapisywały się w jednej kolumnie i było to mało czytelne, dlatego pomysł z użyciem “pandasa” był według nas najbardziej przejrzysty w uzyskaniu schludnego i przejrzystego rozwiązania, zarówno w pliku zewnętrznym .csv jak i w terminalu.

# Zadanie 3

## A

### Konkluzje

## B

### Konkluzje

## C

### Konkluzje

## D

### Konkluzje

## E

### Konkluzje