**Anna Plust, 36148**

**Duże zbiory danych, 2022-23**

**Lab 7, Sprawozdanie**

public class Main {

public static void main(String[] args) {

SparkConf conf = new SparkConf()

.setAppName("")

.setMaster("local");

JavaSparkContext sc = new JavaSparkContext(conf);

SparkSession spark = SparkSession

.builder()

.sparkContext(sc.sc())

.getOrCreate();

Dataset<Row> orders = spark.read()

.option("header", true)

.option("inferSchema", true)

.csv("instacart/orders.csv");

**// 1) Policzenie liczby zamówień wg godziny ich wykonania**

Dataset x1 = orders.groupBy(col("order\_hour\_of\_day"))

.count()

.orderBy("order\_hour\_of\_day");

X1.show(((int) x1.count()), false);

**// 2) Policzenie zamówień wg dnia tygodnia ich wykonania (wyświetlenie w kolejności rosnącej).**

List<Row> list = new ArrayList<Row>();

list.add(RowFactory.create("Sunday", "0"));

list.add(RowFactory.create("Monday", "1"));

list.add(RowFactory.create("Tuesday", "2"));

list.add(RowFactory.create("Wednesday", "3"));

list.add(RowFactory.create("Thursday", "4"));

list.add(RowFactory.create("Friday", "5"));

list.add(RowFactory.create("Saturday", "6"));

List<org.apache.spark.sql.types.StructField> listOfStructField = new ArrayList<org.apache.spark.sql.types.StructField>();

listOfStructField.add(DataTypes.createStructField("name", DataTypes.StringType, true));

listOfStructField.add(DataTypes.createStructField("number", DataTypes.StringType, true));

StructType structType = DataTypes.createStructType(listOfStructField);

Dataset<Row> days = spark.createDataFrame(list,structType);

days.show();

Dataset x21 = orders.groupBy(col("order\_dow")).count();

Dataset<Row> x22 = days

.join(x21, orders.col("order\_dow").equalTo(days.col("number")))

.orderBy("count")

.select("name", "count");

x22.show(((int) x22.count()), false);

**// 3) Policzenie liczby oraz procentowego rozkładu produktów wg działu sklepu (wyświetlenie w kolejności rosnącej)**

Dataset<Row> products = spark.read()

.option("header", true)

.option("inferSchema", true)

.csv("instacart/products.csv");

Dataset<Row> departments = spark.read()

.option("header", true)

.option("inferSchema", true)

.csv("instacart/departments.csv");

long sum = products.count();

Dataset x31 = products.groupBy("department\_id")

.count()

.join(departments, "department\_id")

.select("department", "count");

Dataset x32 = x31.withColumn("percentage", col("count").divide(sum).multiply(100))

.orderBy("percentage");

x32.show(((int) x32.count()), false);

}

}

**Aplikacja została uruchomiona w środowisku IntelliJ i otrzymano następujące wyniki.**

**1) Policzenie liczby zamówień wg godziny ich wykonania**

A picture containing table

Description automatically generated

**2) Do wyświetlenia wyniku należy użyć nazw dni. W tym celu należy stworzyć kolekcję dataframe (Dataset<Row>) zawierającą numery i nazwy dni tygodnia, a następnie połączyć ją z wynikiem zliczania zamówień (proszę jako pierwszy dzień tygodnia przyjąć niedzielę).**

**A picture containing text

Description automatically generated**

**Policzenie zamówień wg dnia tygodnia ich wykonania (wyświetlenie w kolejności rosnącej).**

**A picture containing table

Description automatically generated**

**3) Policzenie liczby oraz procentowego rozkładu produktów wg działu sklepu (wyświetlenie w kolejności rosnącej)**

**A screenshot of a computer

Description automatically generated with low confidence**