Collab Notebook Link : [https://colab.research.google.com/drive/1B5Xb7KWE9Vij8tBJavSRaCYTSgWCj6cq?usp=sharing)

Lab 06 - Data Cube Lattice Implementation

Name: $Raj\ Kariya$ Student ID 202103048

Date of Submission: $18/10/2023$

```
!pip install --quiet pyspark
from pyspark.sql import SparkSession
from pyspark.sql. functions import *
spark = SparkSession \
.builder \
.appName("Datacube") \
.getOrCreate()
```

```
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 316.9/316.9 MB 3.8 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
  Building wheel for pyspark (setup.py) ... done
```

```
from google.colab import drive
drive.mount("/content/gdrive")
```

```
Mounted at /content/gdrive
```

```
customers = spark.read.format("csv").option("header", "true").load("/content/gdrive/MyDrive/Colab Notebooks/datasets/dw/widom/customers.c
items = spark.read.format("csv").option("header", "true").load("/content/gdrive/MyDrive/Colab Notebooks/datasets/dw/widom/items.csv")
sales = spark.read.format("csv").option("header", "true").load("/content/gdrive/MyDrive/Colab Notebooks/datasets/dw/widom/sales.csv")
stores = spark.read.format("csv").option("header", "true").load("/content/gdrive/MyDrive/Colab Notebooks/datasets/dw/widom/stores.csv")
```

```
stores.show()
```

```
+-------+-------------+-----------+-----+
|storeid|         city|     county|state|
+-------+-------------+-----------+-----+
| store1|    Palo Alto|Santa Clara|   CA|
| store2|Mountain View|Santa Clara|   CA|
| store3|   Menlo Park|  San Mateo|   CA|
| store4|      Belmont|  San Mateo|   CA|
| store5|      Seattle|       King|   WA|
| store6|      Redmond|       King|   WA|
+-------+-------------+-----------+-----+
```

```
# SELECT storeID, itemID, custID, sum(price) from Sales
# GROUP BY storeID, itemID, custID
# WITH CUBE;
data_cube = sales.cube("storeid" , "itemid" , "custid").agg(sum("price").alias("Total")).sort("storeid","itemid", "custid")
data_cube.show()
# data_cube.write.parquet("/content/gdrive/My Drive/data_cube.parquet")
data_cube.write.parquet("data_cube.parquet", mode='overwrite')
```

```
+-------+------+------+------+
|storeid|itemid|custid| Total|
+-------+------+------+------+
|   NULL|  NULL|  NULL|3350.0|
|   NULL|  NULL|cust1 | 670.0|
|   NULL|  NULL|cust2 | 935.0|
|   NULL|  NULL|cust3 | 885.0|
|   NULL|  NULL|cust4 | 860.0|
|   NULL|item1 |  NULL| 135.0|
|   NULL|item1 |cust1 |  10.0|
|   NULL|item1 |cust2 |  80.0|
|   NULL|item1 |cust3 |  45.0|
|   NULL|item2 |  NULL|1325.0|
|   NULL|item2 |cust1 | 310.0|
|   NULL|item2 |cust2 | 335.0|
|   NULL|item2 |cust3 | 425.0|
|   NULL|item2 |cust4 | 255.0|
|   NULL|item3 |  NULL| 780.0|
|   NULL|item3 |cust1 | 170.0|
|   NULL|item3 |cust2 | 295.0|
|   NULL|item3 |cust3 | 150.0|
|   NULL|item3 |cust4 | 165.0|
|   NULL|item4 |  NULL| 655.0|
+-------+------+------+------+
only showing top 20 rows
```

a. Show Item axis

```
# a. Show Item axis
Item_axis = data_cube.select(col("itemid"),col("total")).filter(data_cube.storeid.isNull() & data_cube.custid.isNull())
Item_axis.show()
```

```
+------+------+
|itemid| total|
+------+------+
|  NULL|3350.0|
|item1 | 135.0|
|item2 |1325.0|
|item3 | 780.0|
|item4 | 655.0|
|item5 | 455.0|
+------+------+
```

b. Produce roll-up of (Item, Store, Customer)

```
# b. Produce roll-up of (Item, Store, Customer)
Roll = sales.rollup(col("itemid"), col("storeid") , col("custid")).count().orderBy("itemid","storeid" ,"custid")
Roll.show()
```

```
+------+-------+------+-----+
|itemid|storeid|custid|count|
+------+-------+------+-----+
|  NULL|   NULL|  NULL|   60|
|item1 |   NULL|  NULL|    5|
|item1 | store1|  NULL|    4|
|item1 | store1|cust1 |    1|
|item1 | store1|cust2 |    1|
|item1 | store1|cust3 |    2|
|item1 | store2|  NULL|    1|
|item1 | store2|cust2 |    1|
|item2 |   NULL|  NULL|   18|
|item2 | store1|  NULL|    4|
|item2 | store1|cust1 |    1|
|item2 | store1|cust2 |    2|
|item2 | store1|cust3 |    1|
|item2 | store2|  NULL|    8|
|item2 | store2|cust1 |    3|
|item2 | store2|cust2 |    1|
|item2 | store2|cust3 |    2|
|item2 | store2|cust4 |    2|
|item2 | store3|  NULL|    4|
|item2 | store3|cust2 |    2|
+------+-------+------+-----+
only showing top 20 rows
```

c. Show store-wise sales summary of blue Tshirt

```
# c. Show store-wise sales summary of blue Tshirt

ans2 = data_cube.join(items,items.itemid==data_cube.itemid).drop(items.itemid).where((items.category=="Tshirt") & (items.color == "blue")
ans1 = ans2.groupBy("storeid").agg(sum("Total").alias("sales_summary"))
ans = ans1.filter(ans1.storeid.isNotNull())
ans.show()
```

```
+-------+-------------+
|storeid|sales_summary|
+-------+-------------+
| store2|        130.0|
| store1|        140.0|
+-------+-------------+
```

d. List all 'Tshirts' (price <= 20) sold in 'California' to young people (age < 25).

```
# d.  List all 'Tshirts' (price <= 20) sold in 'California' to young people (age < 25)
ans3 = sales.join(items,items.itemid==sales.itemid).drop(items.itemid).where((items.category=="Tshirt"))
ans3 = ans3.join(customers,customers.custid ==ans3.custid).drop(customers.custid).where(customers.age < 25)
ans3  = ans3.join(stores,stores.storeid == ans3.storeid).drop(stores.storeid).where(stores.city == "California")
```

```
ans4 = ans3.filter(col("price") <= 20)
ans4.show()

# join1 = data_cube.join(items,items.itemid==data_cube.itemid).drop(items.itemid).where((items.category=="Tshirt"))
# join1 = join1.join(customers,customers.custid ==join1.custid).drop(customers.custid).where(customers.age < 25)
# join1  = join1.join(stores,stores.storeid == join1.storeid).drop(stores.storeid).where(stores.city == "California")
# join1 = join1.join((join1.itemid == sales.itemid) & (join1.storeid == sales.itemid) & (join1.custid == sales.custid)).drop(sales.itemid
# join1.show()
```

```
+-------+------+------+-----+--------+-----+-----+------+---+----+------+-----+
|storeid|itemid|custid|price|category|color|cname|gender|age|city|county|state|
+-------+------+------+-----+--------+-----+-----+------+---+----+------+-----+
+-------+------+------+-----+--------+-----+-----+------+---+----+------+-----+
```