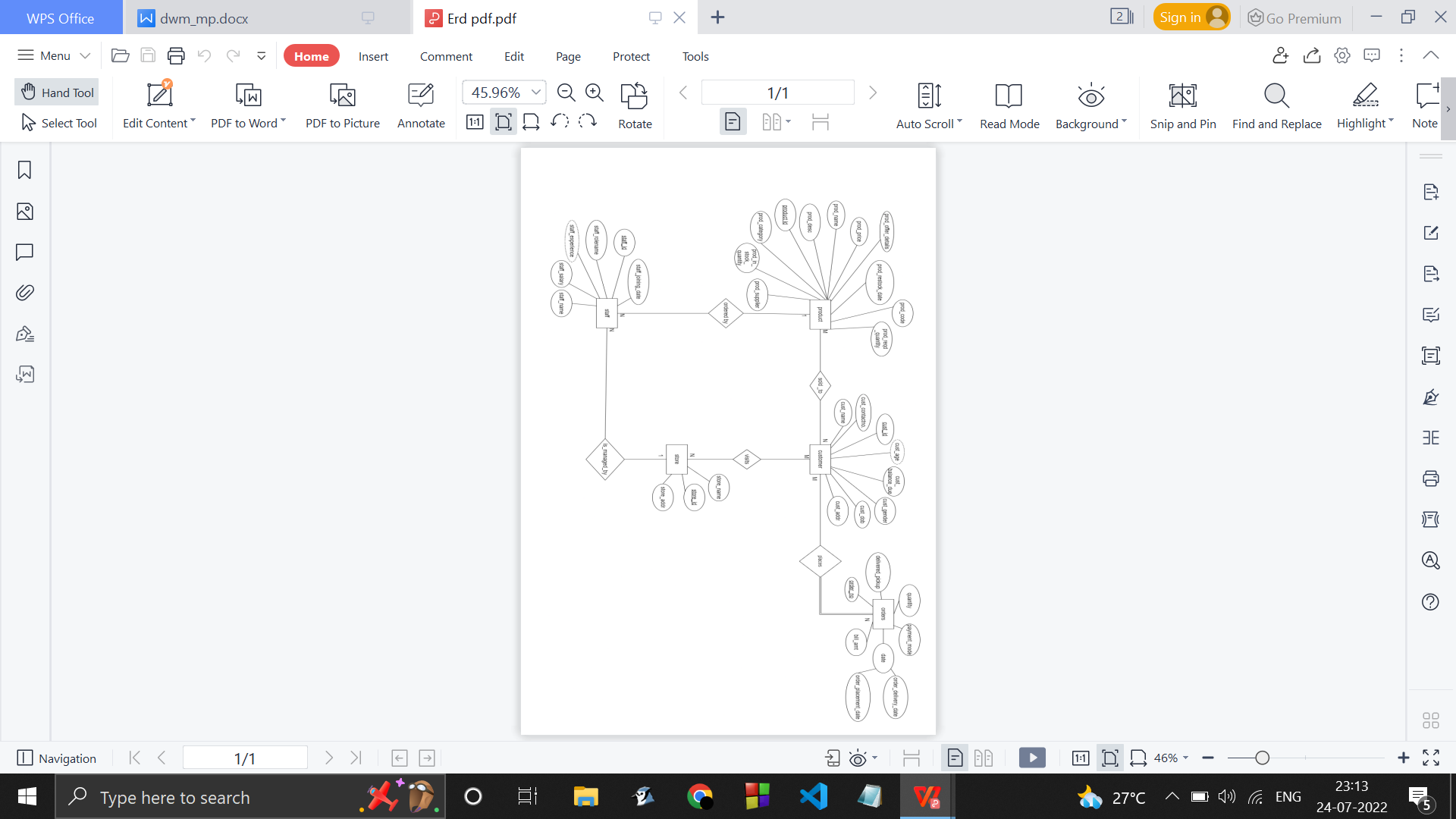
**Lab 01**

* **PROBLEM STATEMENT :**

The objective of this Retail Store Management System is to maintain transaction records and details of the orders placed for different products by the customers in different stores managed by the store staff such that if products go out of stock , the staff orders them from the supplier.This helps to analyze the facts or measures in the departmental store thereby helping to improve the business activities.

The ultimate objective is to predict the trends of sales the books in the stores. These trend predictions can then be used by the store’s and the supplier’s management to make decisions regarding the measures(facts).

* **ERD FOR RETAIL STORE MANAGEMENT SYSTEM :**



The ER diagram above represents the model of a retail management system showing all the database tables and relations between the entities PRODUCT, CUSTOMER, STAFF, STORE and ORDERs.

Entities and their attributes :

PRODUCT : Each product has a name ,code , id, category and a description associated with it. When products goes out of stock , the staff’s responsibility is to order them from the supplier and get them delievered to the corresponding store .

STAFF : Each staff has a role for eg. manager staff , the cleaning staff , etc. Experience of the staff is calculated using their joining dates .

CUSTOMER : when a customer visits a store to buy products , address is recorded if in case the customer wants the order to be delievered to his/her house . Customer’s balance payment of the products is also kept track of .

ORDERS : An order placed by customer can either be delievered or picked upby the customers themselves. The delievery time is recorded . A bill is created based on the all the products bought by them and their price total. The mode of payment for the products is also recorded.

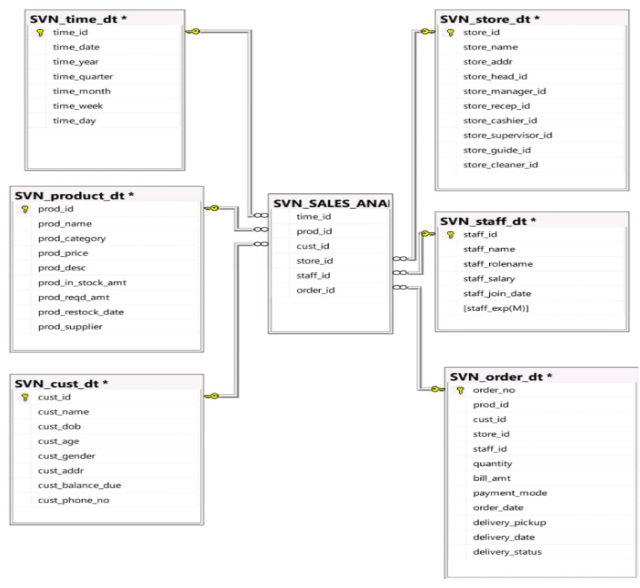
( Primary keys , Derived attributes , Foreign keys )

|  |  |
| --- | --- |
| ENTITIES | ATTRIBUTES |
| PRODUCT | * Prod\_id (PK) * Prod\_Name * Prod\_Category * Prod\_code * Prod\_price * Prod\_offer\_details * Prod\_Desc * Prod\_In\_stock\_quantity * Prod\_Reqd\_quantity * Prod\_Restock\_date * Prod\_Supplier |
| STAFF | * Staff\_id (PK) * Staff\_Rolename * Staff\_Name * Staff\_Salary * Staff\_Joining\_date * Staff\_experience * Staff\_Appointed\_in\_store(FK store\_id) |
| CUSTOMER | * Cust\_id (PK) * Cust\_Name * Cust\_Dob * Cust\_Age * Cust\_Gender * Cust\_Balancedue * Cust\_Phoneno * Cust\_addr |
| STORE | * Store\_id (PK) * Store\_Name * Store\_addr |
| ORDERS | * Order\_no (Auto-increment) * Prod\_id( FK Prod\_id) * Quantity * Bill\_amt * Payment\_mode * Delieverd\_pickup * Order\_placed\_on\_date * Order\_delieverd\_on\_date * To\_be\_delievered\_at( FK Cust\_address) * Cust\_id ( FK Cust\_id ) * Store\_id ( FK Store\_id ) * PK : Order\_no , Prod\_id |

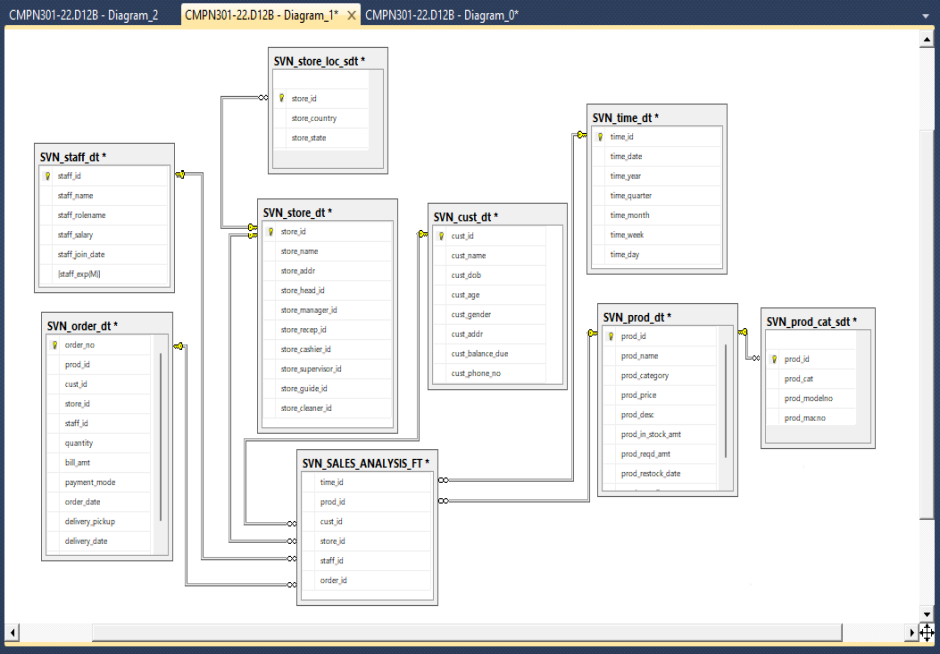
Information package diagram :

**Lab 02**

Dimension and fact table :

Star schema : 

Snowflake schema :

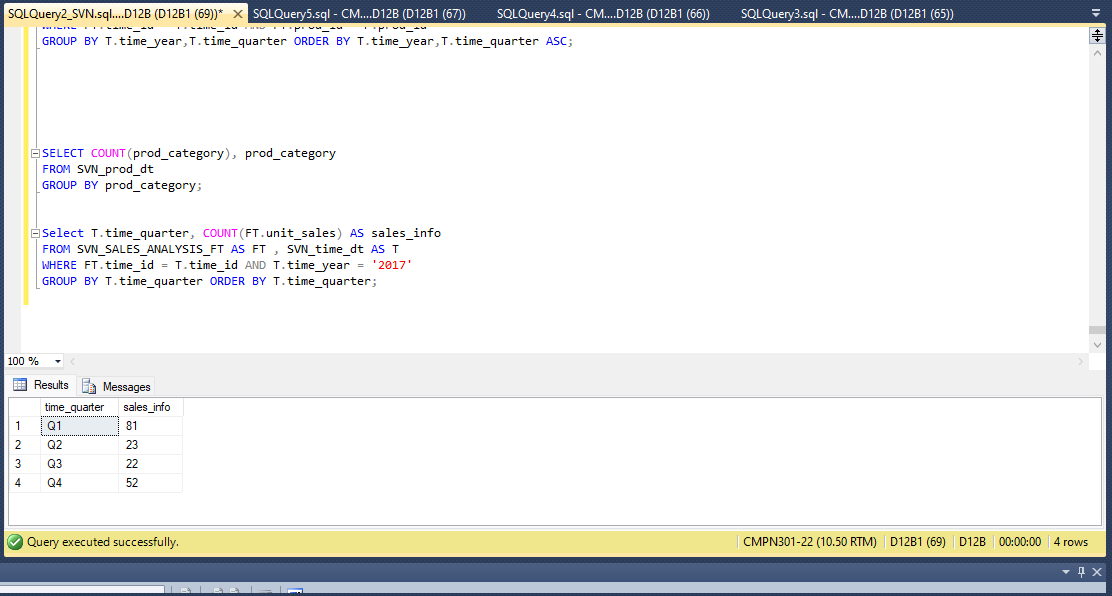


Galaxy schema :

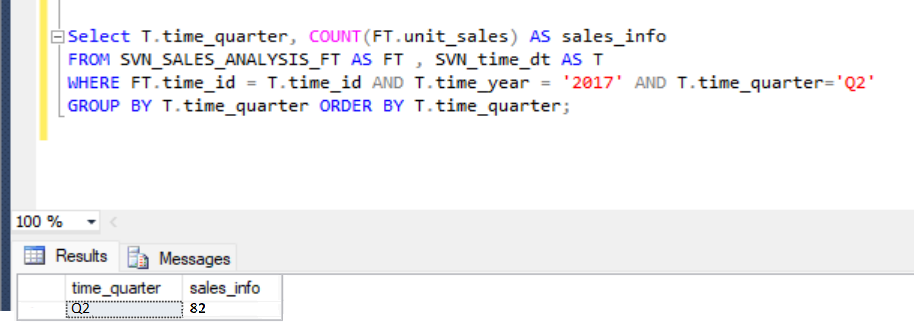
**Lab 03**

OLAP OPNS

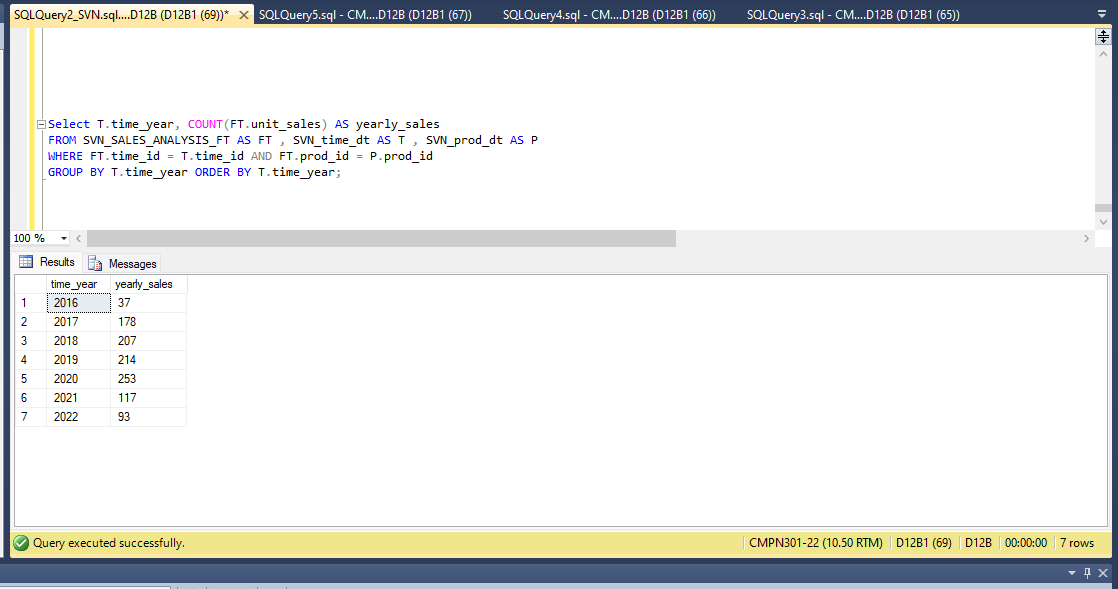
1. **Slice :**



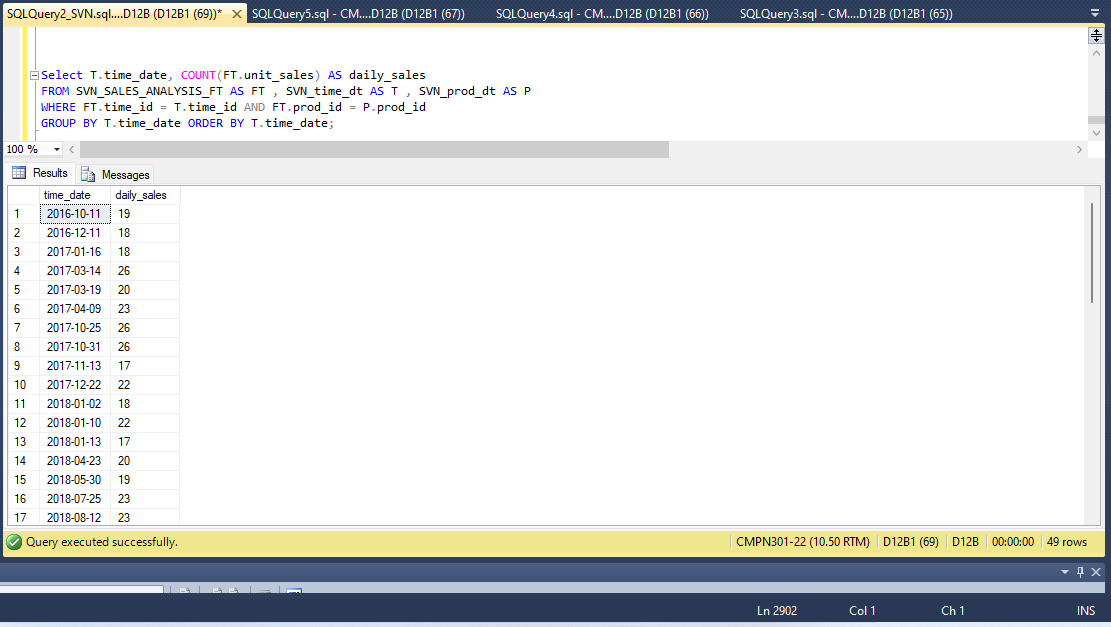
1. **Dice :**



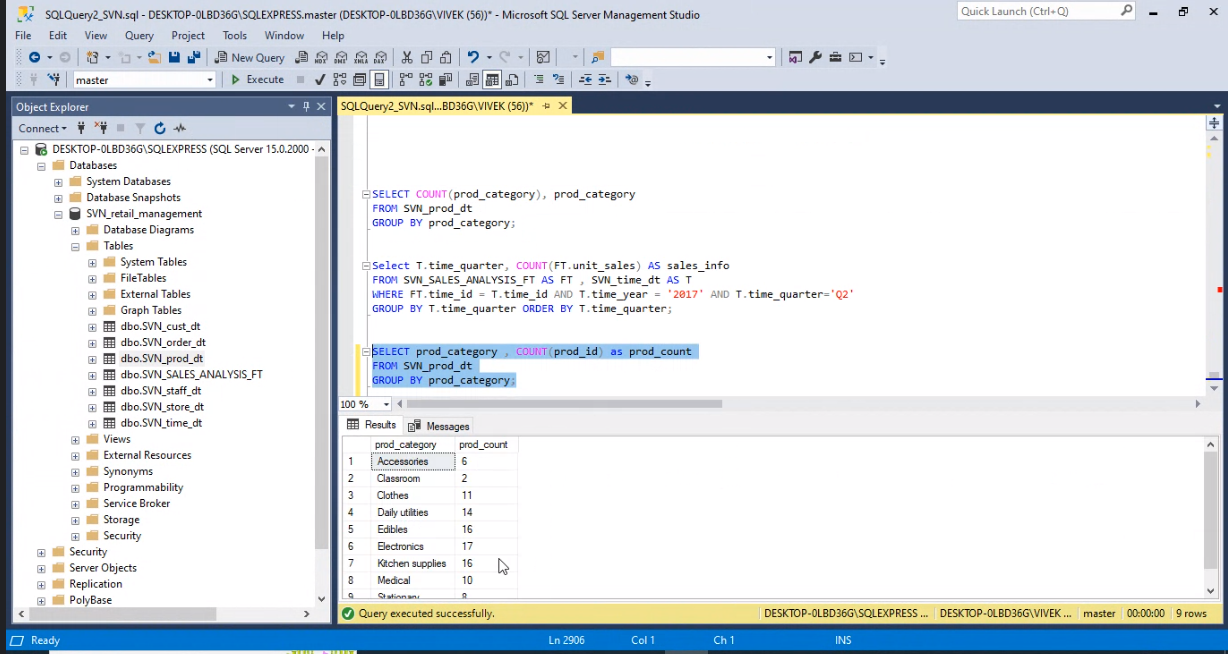
1. **Roll up ….wrt YEARLY**



1. **Drill Down…..wrt DAILY**



1. **Pivot…rotate operation**



SELECT \* FROM

(

SELECT prod\_category, prod\_id FROM SVN\_prod\_dt as p

) temp\_table

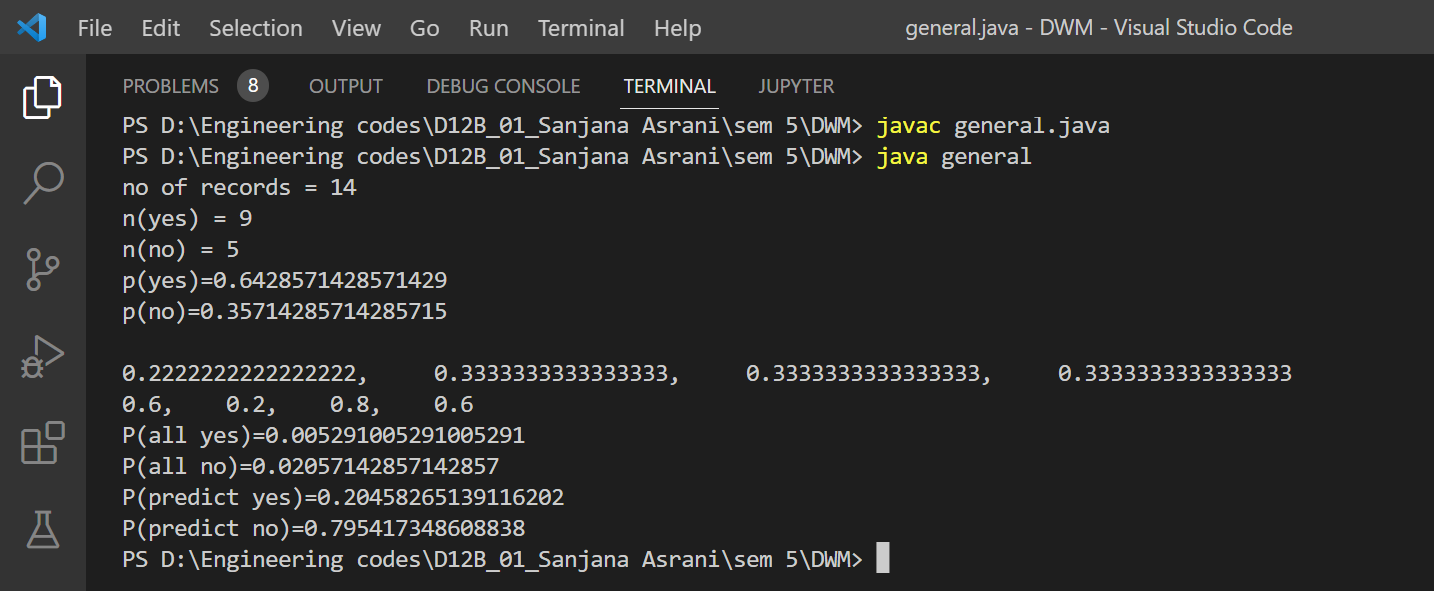
PIVOT (

COUNT (prod\_id)

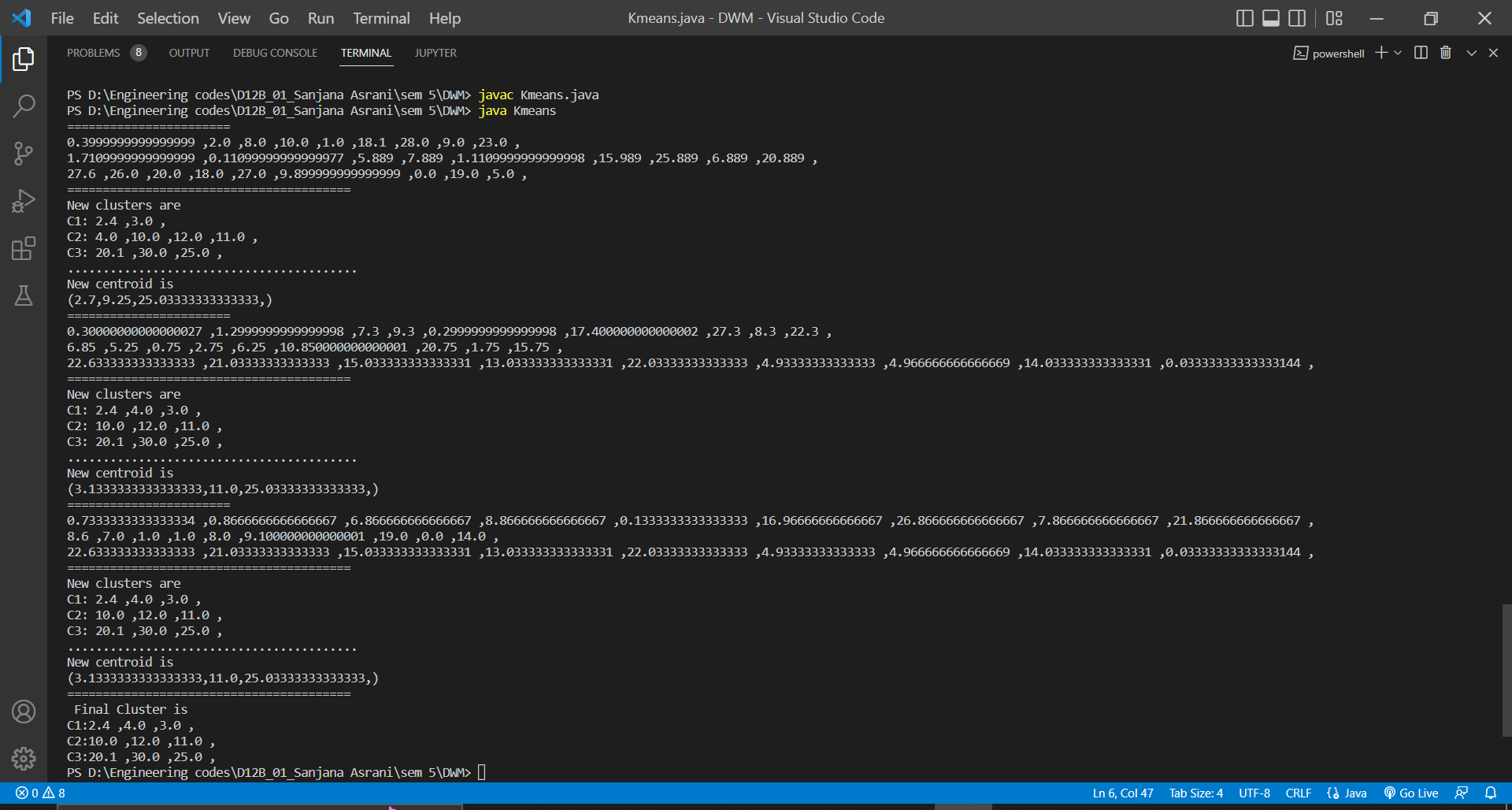
FOR prod\_category IN (Clothes, Edibles,Electronics)

Pivot\_table;

**LAB 4**

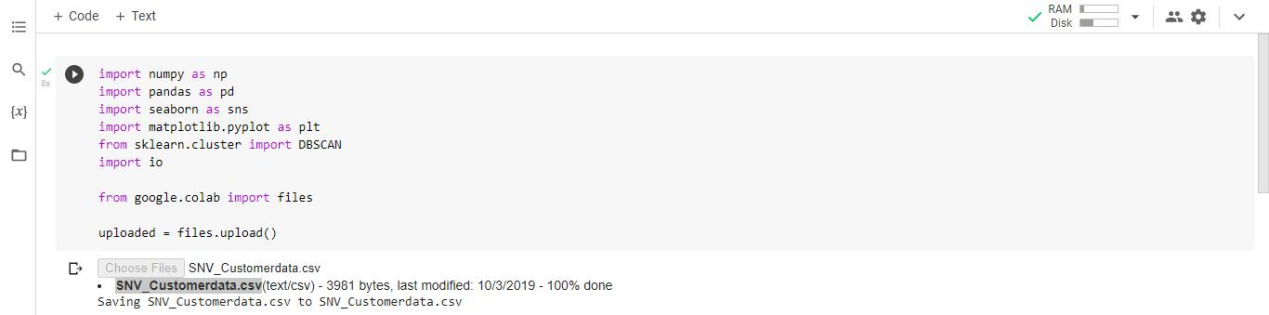


Classification

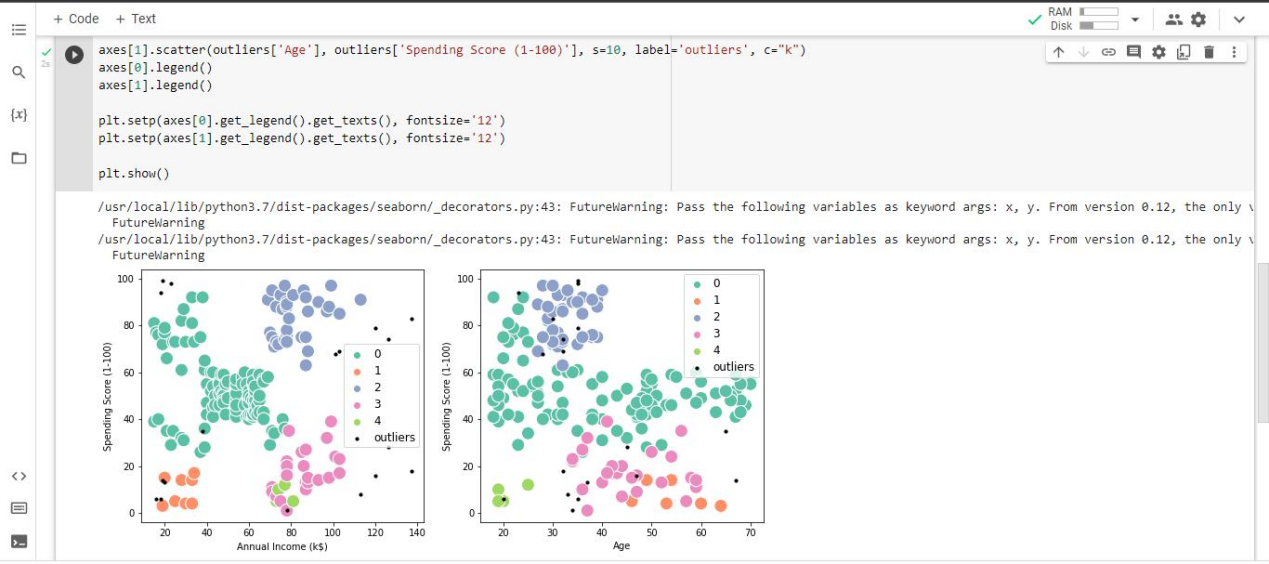


clustering

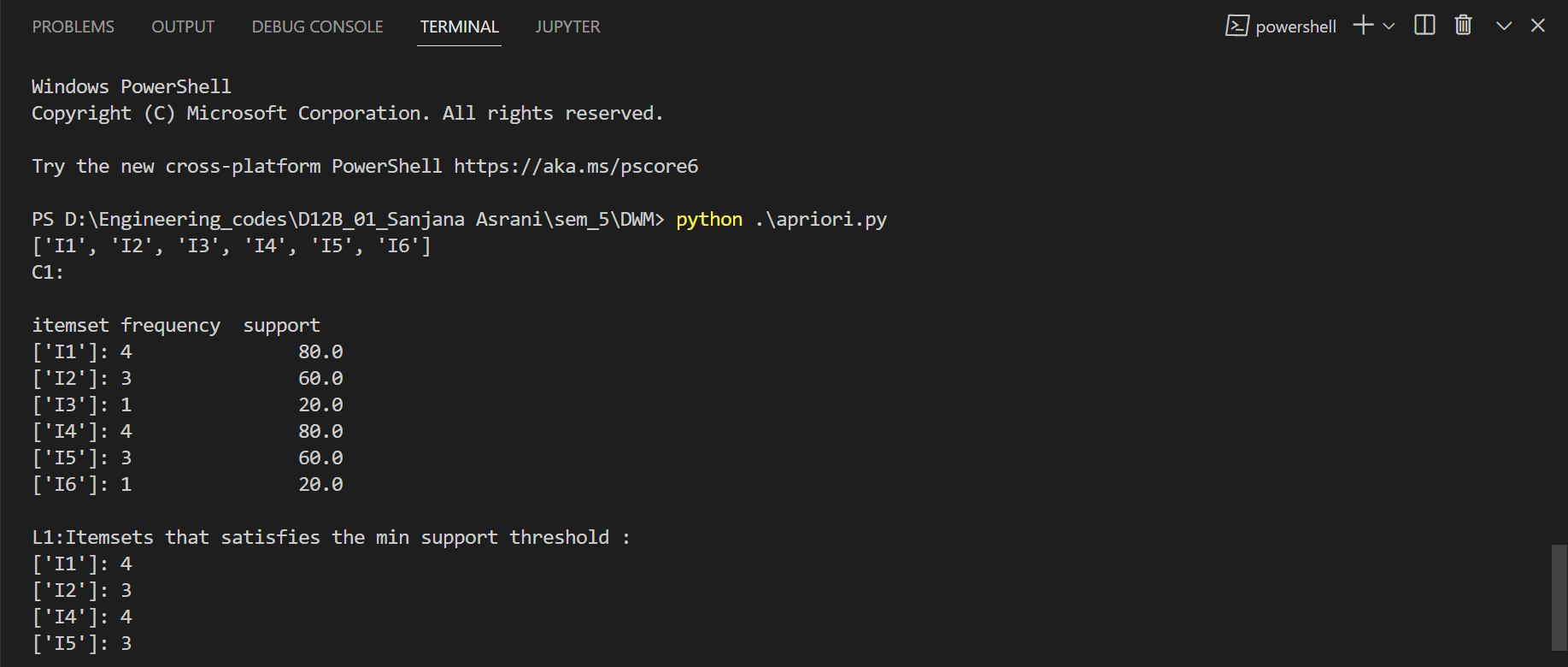
**LAB 5**

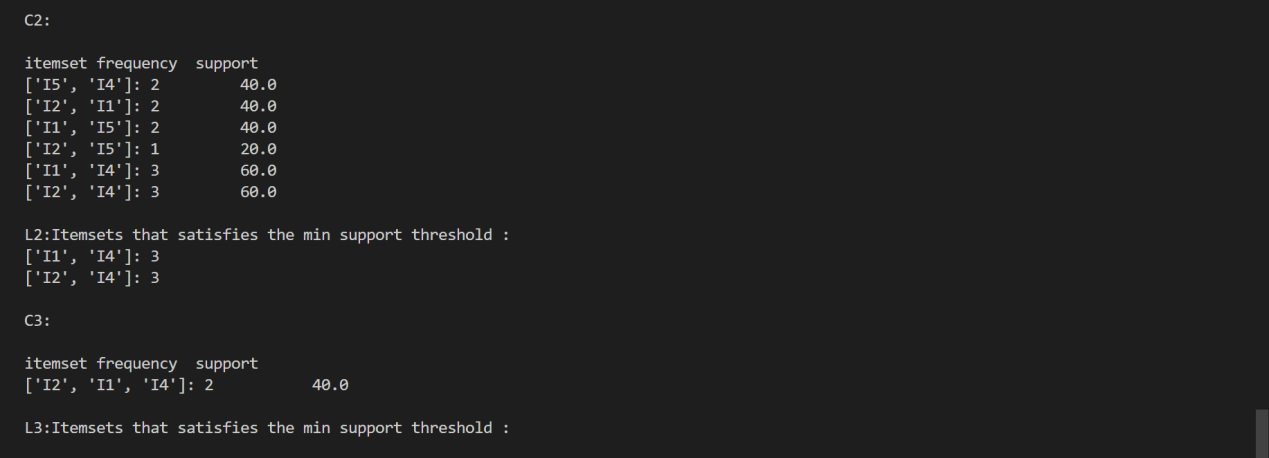


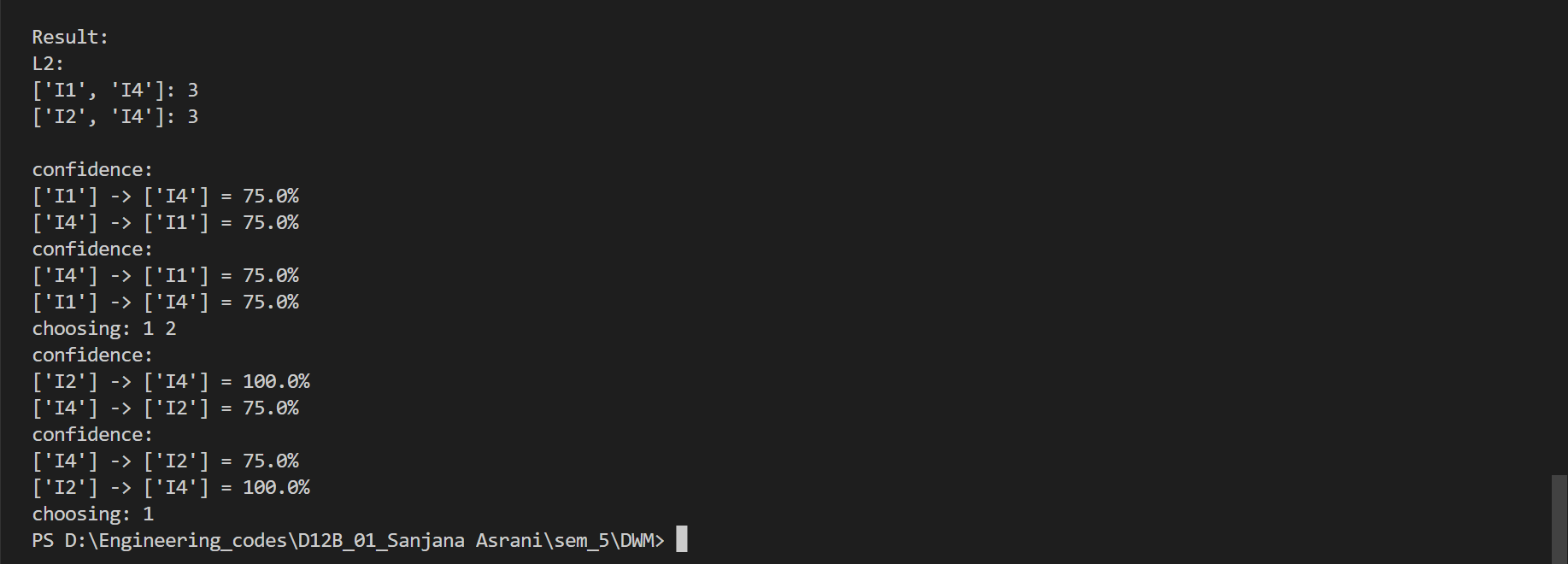




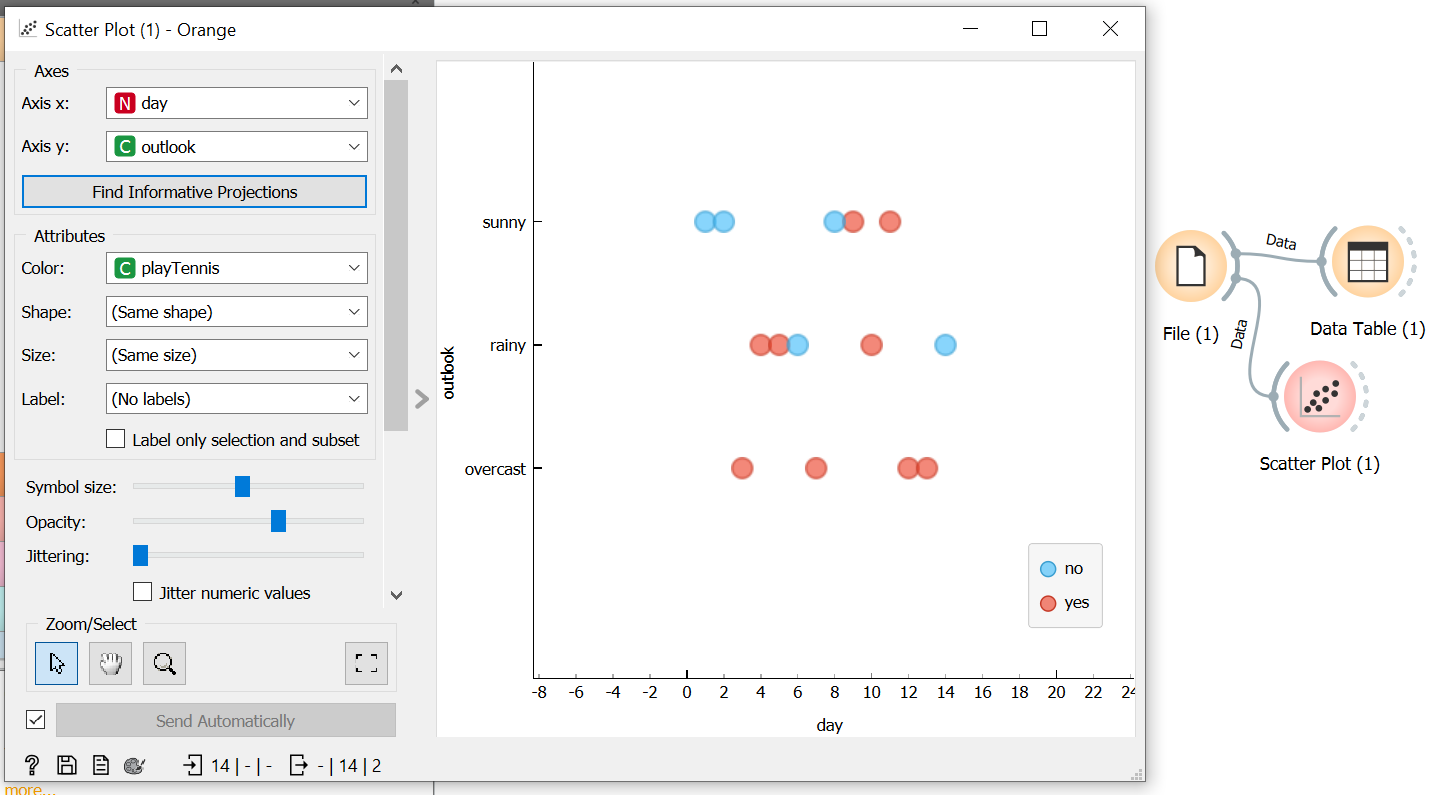
**LAB 6**



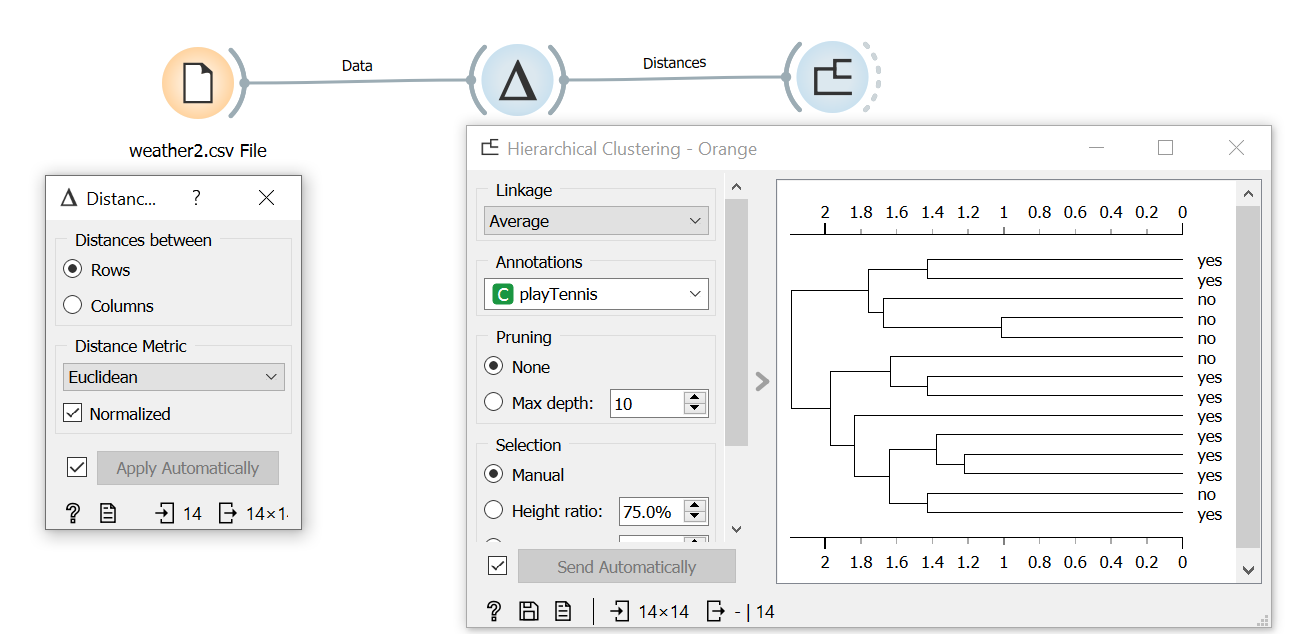


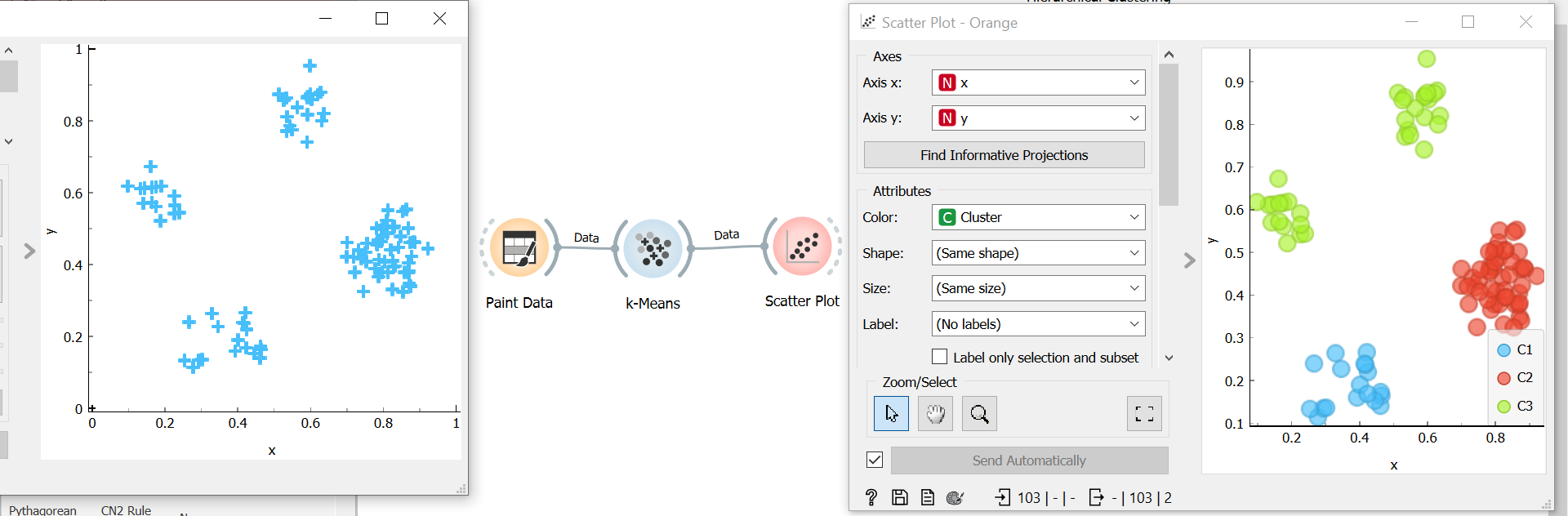
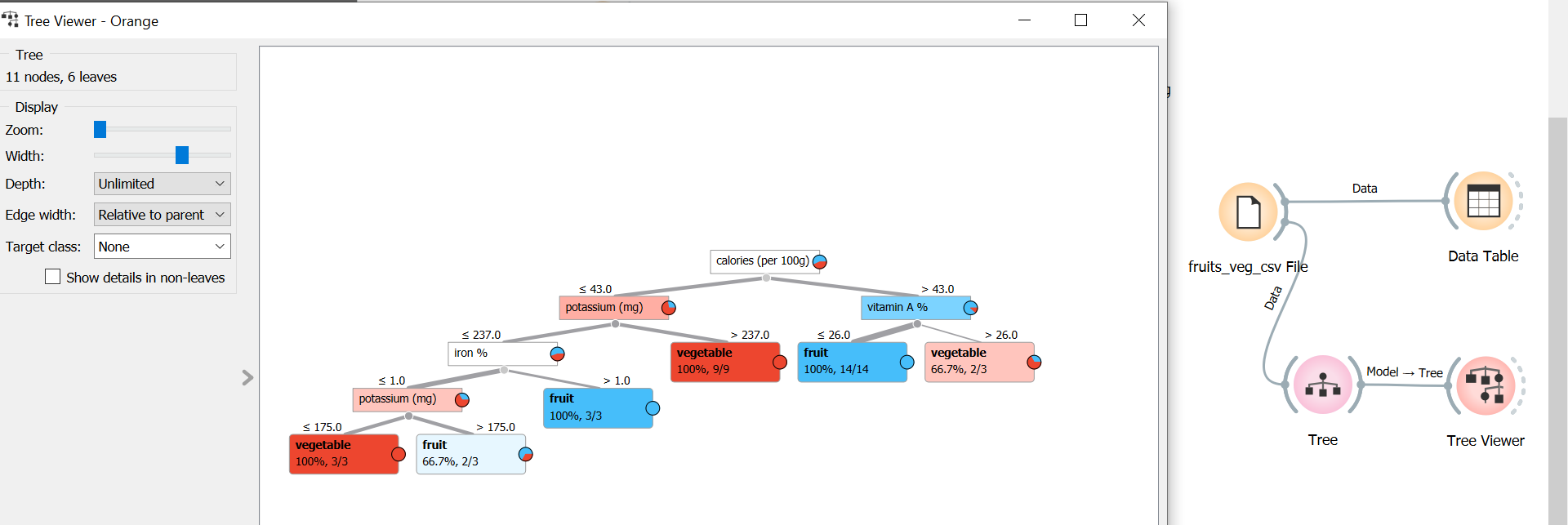


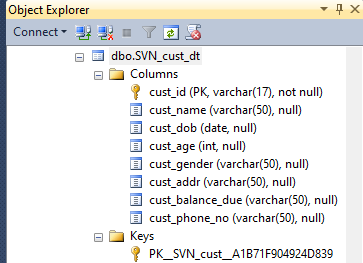
**LAB 08**

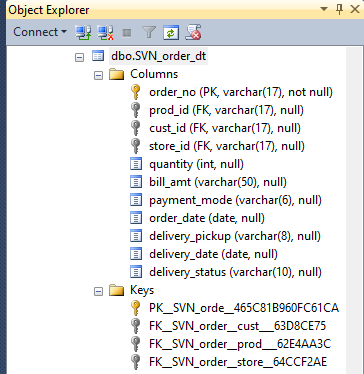


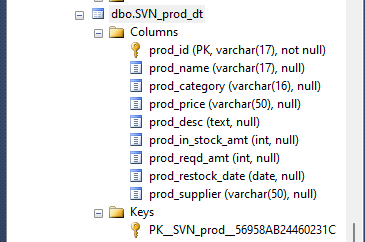
SCATTER PLOT

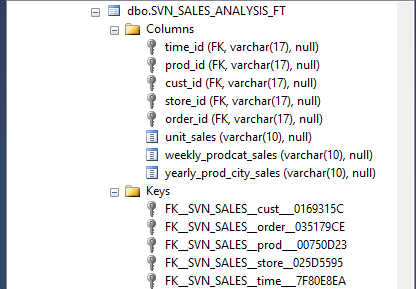


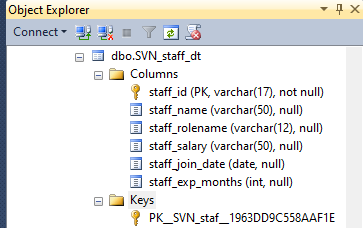


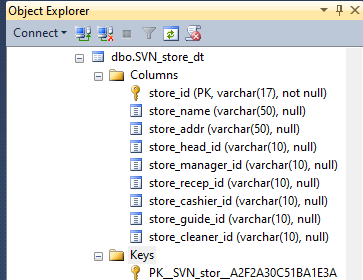


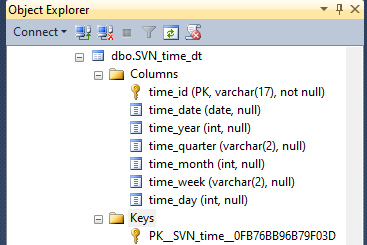












Code to populate tables :

from operator import concat

import random

list = []

def item\_create():

list\_item = ["T\_"+str(random.randint(1, 50)),"P\_"+str( random.randint(1, 105)), "C\_"+str(random.randint(1, 500)),

"St\_"+str(random.randint(1, 30)), "ON\_"+str(random.randint(1, 1000))]

# no of foreign keys jitne list items with their number of rows

if list\_item in list:

item\_create()

return list\_item

for i in range(1, 100):

# 100 records for fact table will be generated

list.append(item\_create())

for i in list:

print(

f'insert into SVN\_SALES\_ANALYSIS\_FT(time\_id, prod\_id, cust\_id, store\_id,order\_id,'

f' unit\_sales,weekly\_prodcat\_sales,yearly\_prod\_city\_sales ) values (\'{i[0]}\', \'{i[1]}\', \'{i[2]}\', '

f'\'{i[3]}\', \'{i[4]}\', \'{"$"+str(random.randint(1, 600))}\', \'{"$"+str(random.randint(1, 2000))}\', '

f'\'{"$"+str(random.randint(1, 10000))}\');')

# take all fk from the list and measures between the specified range.