VV COLLEGE OF ENGINEERING
TISAIYANVILAI

LAB MANUAL

STUDENT NAME : .....

REGISTER NUMBER : .....

SUBJECT CODE : CCS341

SUBJECT NAME : DATA WAREHOUSING

DEGREE /BRANCH : BE / CSE

YEAR / SEM : III / 06

ACADEMIC YEAR : 2023 – 2024

V V COLLEGE OF ENGINEERING
(Approved By AICTE, New Delhi and Affiliated To Anna University Chennai)
V V Nagar, Arasoor, Tisaiyanvilai

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

College Vision and Mission Statement

Vision

"Emerge as a premier technical institution of global standards, producing enterprising, knowledgeable engineers and entrepreneurs."

Mission

- Impart quality and contemporary technical education for rural students.
- Have the state of the art infrastructure and equipment for quality learning.
- Enable knowledge with ethics, values and social responsibilities.
- Inculcate innovation and creativity among students for contribution to society.

Vision and Mission of the Department of Computer Science and Engineering

Vision

"Produce competent and intellectual computer science graduates by empowering them to compete globally towards professional excellence".

Mission

- Provide resources, environment and continuing learning processes for better exposure in latest and contemporary technologies in Computer Science and Engineering.
- Encourage creativity and innovation and the development of self-employment through knowledge and skills, for contribution to society
- Provide quality education in Computer Science and Engineering by creating a platform to enable coding, problem solving, design, development, testing and implementation of solutions for the benefit of society.

I. PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

Graduates of Computer Science Engineering can

- Apply their technical competence in computer science to solve real world problems, with technical and people leadership.
- Conduct cutting edge research and develop solutions on problems of social relevance.
- Work in a business environment, exhibiting team skills, work ethics, adaptability and lifelong learning.

II. PROGRAM OUTCOMES (POs)

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

III. PROGRAM SPECIFIC OUTCOMES (PSOs)

- Exhibit design and programming skills to build and automate business solutions using cutting edge technologies.
- Strong theoretical foundation leading to excellence and excitement towards research, to provide elegant solutions to complex problems.
- Ability to work effectively with various engineering fields as a team to design, build and develop system applications.

V V COLLEGE OF ENGINEERING
(Approved By AICTE, New Delhi and Affiliated To Anna University Chennai)
V V Nagar, Arasoor, Tisaiyanvilai

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

LABORATORY

LIST OF EXPERIMENTS – R2021

PRACTICAL SUBJECT NAME	DATA WAREHOUSING
PRACTICAL SUBJECT CODE	CCS341
SEMESTER/ YEAR	06 / THIRD
TOTAL HOURS	30
DEGREE / DEPARTMENT	BE / COMPUTER SCIENCE AND ENGINEERING
STAFF IN-CHARGE	Mr.A.SUBASH DAVID
LAB INSTRUCTOR	Ms. ANITHA
REGULATION	2021

CO1	Design data warehouse architecture for various Problems
CO2	Apply the OLAP Technology
CO3	Analyse the partitioning strategy
CO4	Critically analyze the differentiation of various schema for given problem
CO5	Frame roles of process manager & system manager

V V COLLEGE OF ENGINEERING
(Approved By AICTE, New Delhi and Affiliated To Anna University Chennai)
V V Nagar, Arasoor, Tisaiyanvilai

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

S.No	Name of the Experiment	CO Mapping	PO Mapping
1	Data exploration and integration with WEKA	CO1	PO1,PO2,PO3, PO4,PO5,PO9, PO12.
2	Apply weka tool for data validation	CO1	PO1,PO2,PO3, PO4,PO5,PO9, PO12
3	Plan the architecture for real time application	CO1	PO1,PO2,PO3, PO4,PO5,PO9, PO12.
4	Write the query for schema definition	CO2	PO1,PO2,PO3, PO4,PO5,PO9, PO11,PO12.
5	Design data ware house for real time applications	CO2	PO1,PO2,PO3, PO4,PO5,PO9, PO11,PO12
6	Analyse the dimensional Modeling	CO3	PO1,PO2,PO3, PO4,PO12
7	Case study using OLAP	CO4	PO1,PO2,PO3, PO4,PO12
8	Case study using OTLP	CO5	PO1,PO2,PO3, PO4,PO6,PO11, PO12
9	Implementation of warehouse testing.	CO5	PO1,PO2,PO3, PO4,PO6,PO11, PO12

V V COLLEGE OF ENGINEERING
(Approved By AICTE, New Delhi and Affiliated To Anna University Chennai)
V V Nagar, Arasoor, Tisaiyanvilai

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

RUBRICS FOR ASSESSING
LABORATORY

SI. No.	Criteria	Total Marks	Excellent (25)	Good (20)	Average (10)	Poor (5)
			91% - 100%	71% - 90%	50% - 70%	<50%
1	Preparation	25	Gives clear idea about the aim and having good capability of executing experiments.	Capability of executing experiments but no proper clarification about the objective.	Gives clear idea about the target and has less capability of executing experiments.	Gives indistinct idea about the target and has less capability of executing experiments & who feel difficult to follow the objectives.
2	Viva	25	Have executed the experiments in an efficient way & make credible and unbiased judgments regarding the experiments.	Executed the experiments with less efficient & has partial judgments regarding the experiments.	Executed the experiments with less efficiency and has no judgments regarding experiments.	Incomplete experiments & lack of judgments regarding experiments.
3	Performance	25	Followed all the instructions given in the procedure and submitted the manual on time.	Followed all the instructions given in the procedure with some assisting.	Followed some of the instructions given in the procedure & late in submission of manual.	Unable to follow the instructions given in the procedure & late in submission of manual.

V V COLLEGE OF ENGINEERING
(Approved By AICTE, New Delhi and Affiliated To Anna University Chennai)
V V Nagar, Arasoor, Tisaiyanvilai

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Department of Computer Science and Engineering		
Preparation	25	
Viva	25	
Performance	25	
Total	75	
Lab Incharge	Date	

INDEX

.No	DATE	NAME OF THE EXPERIMENT	SIGN
1		Data exploration and integration with WEKA	
2		Apply weka tool for data validation	
3		Plan the architecture for real time application	
4		Write the query for schema definition	
5		Design data ware house for real time applications	
6		Analyse the dimensional Modeling	
7		Case study using OLAP	
8		Case study using OTLP	
9		Implementation of warehouse testing.	

EX.NO:1 DATA EXPLORATION AND INTEGRATION WITH WEKA

PROCEDURE:

STEP 1: Install the weka tool

EXPLORER- An environment for exploring data with WEKA

a) Click on —explorer button to bring up the explorer window.

b) Make sure the —preprocess tab is highlighted.

c) Open a new file by clicking on —Open New file and choosing a file with —.arff

etension from the —Data directory.

d) Attributes appear in the window below.

e) Click on the attributes to see the visualization on the right.

f) Click —visualize all to see them all

1.PREPROCESSING:

Loading Data:

The first four buttons at the top of the preprocess section enable you to loaddata into WEKA:

Data Ware Housing & Mining Lab Dept of CSE VEMUIT

1. Open file.... Brings up a dialog box allowing you to browse for the datafile on the local file

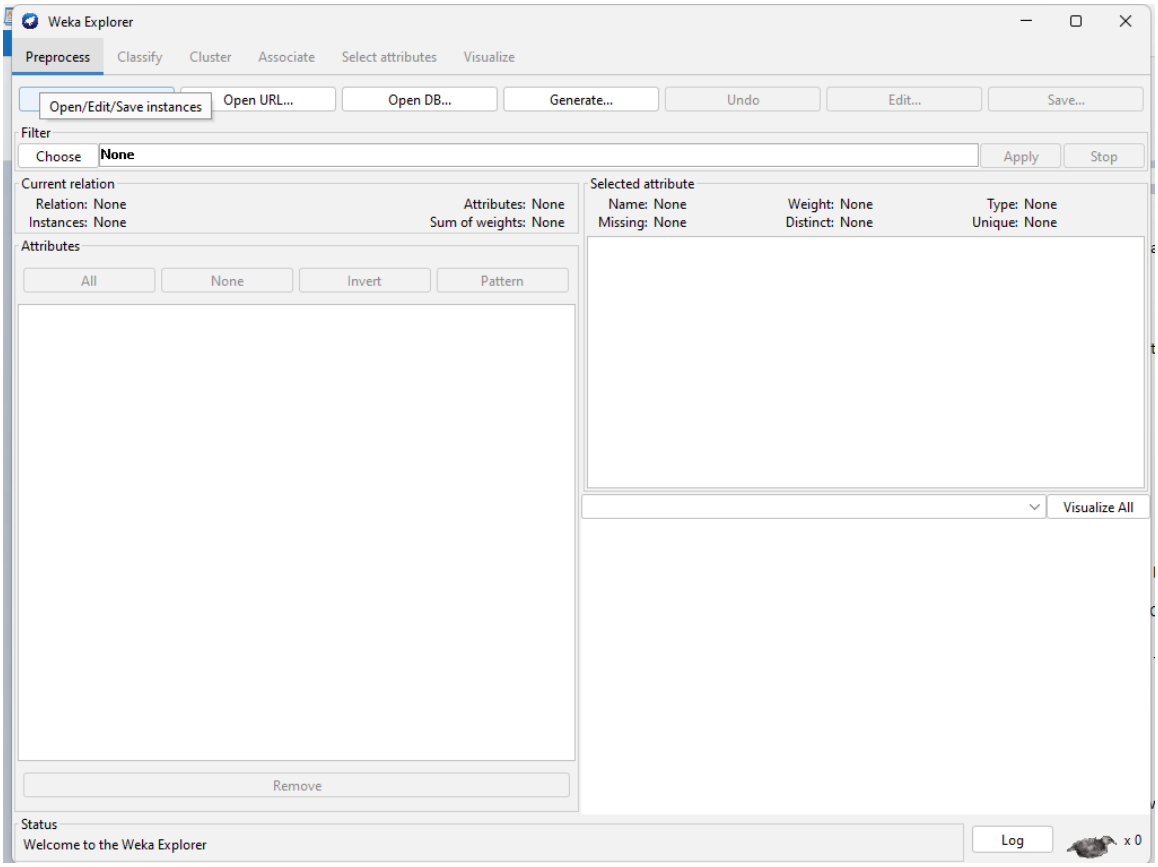
system.

2. Open URL.... Asks for a Uniform Resource Locator address for wherethe data is stored.

3. Open DB.... Reads data from a database. (Note that to make this workyou might have to edit

the fill in weka/experiment/DatabaseUtils.props.)

4. Generate.... Enables you to generate artificial data from a variety ofDataGenerators.



2. Classification:

Selecting a Classifier

At the top of the classify section is the Classifier box. This box has a text fieldthat gives the name of the currently selected classifier, and its options. Clickingon the text box with the left mouse button brings up a GenericObjectEditordialog box, just the same as for filters, that you can use to configure the optionsof the current classifier.

Test Options

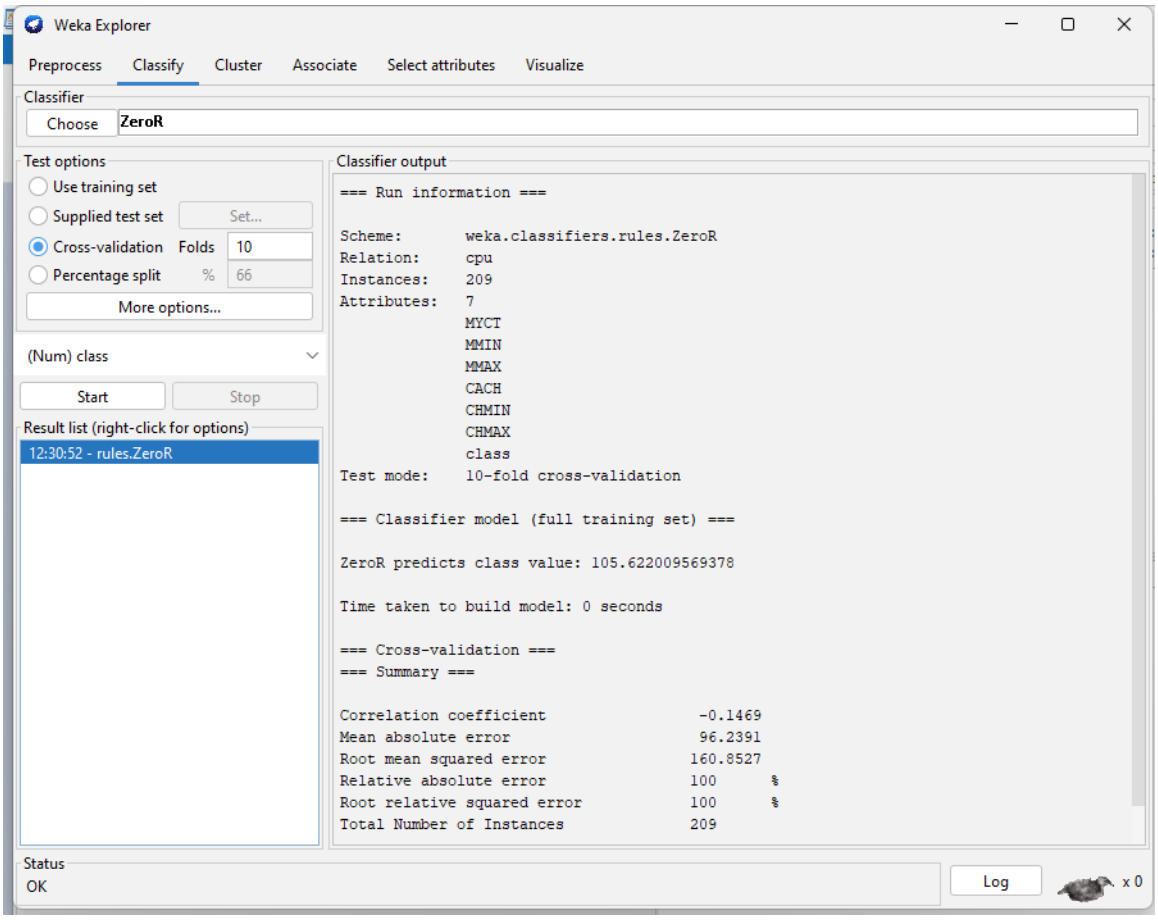
There are four test modes:

Use training set: The classifier is evaluated on how well it predicts theclass of the instances it was trained on.

Supplied test set: The classifier is evaluated on how well it predicts theclass of a set of instances loaded from a file. Clicking the Set... button brings up a dialog allowing you to choose the file to test on.

Cross-validation: The classifier is evaluated by cross-validation, usingthe number of folds that are entered in the Folds text field.

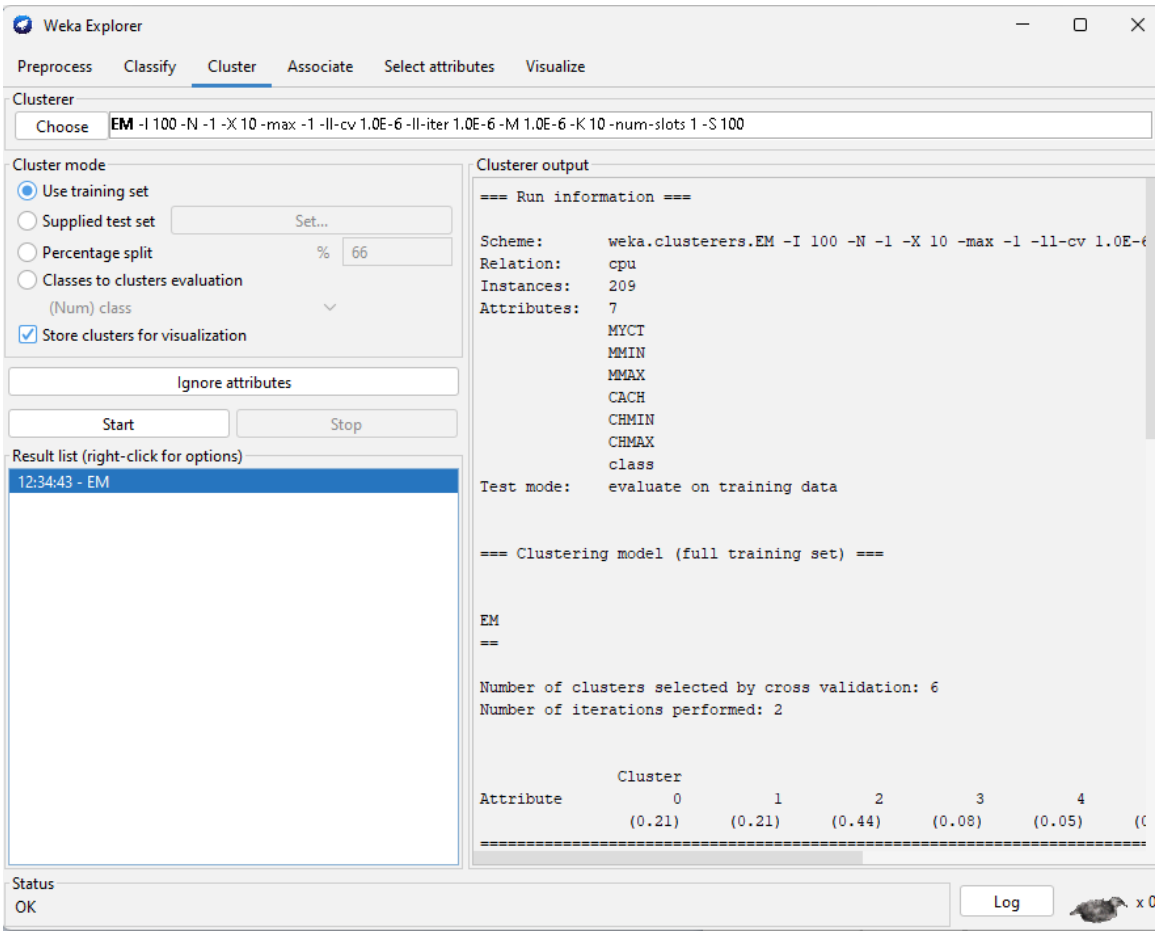
Percentage split : The classifier is evaluated on how well it predicts acertain percentage of the data which is held out for testing. The amountof data held out depends on the value entered in the % field.



3. Clustering:

Cluster Modes:

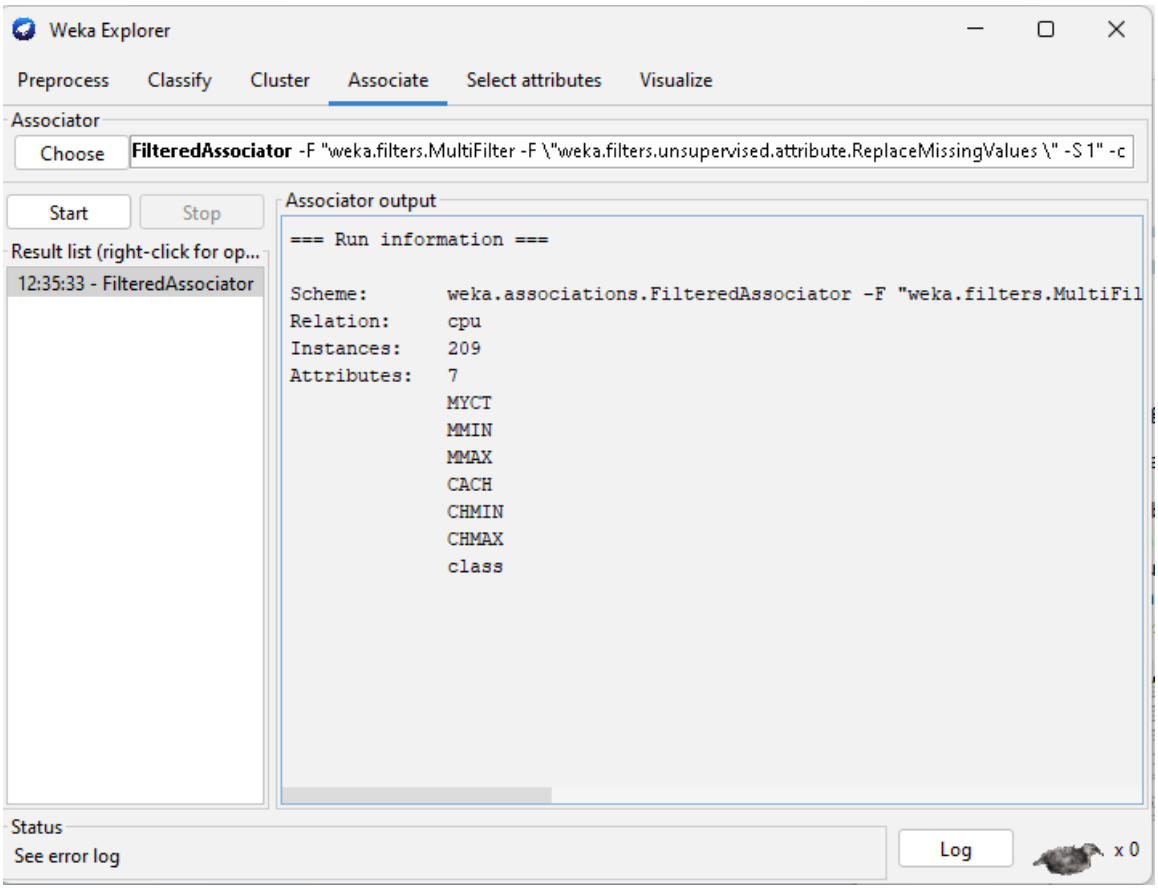
The Cluster mode box is used to choose what to cluster and how to evaluatethe results. The first three options are the same as for classification: Use training set, Supplied test set and Percentage split.



4. Associating:

Setting Up

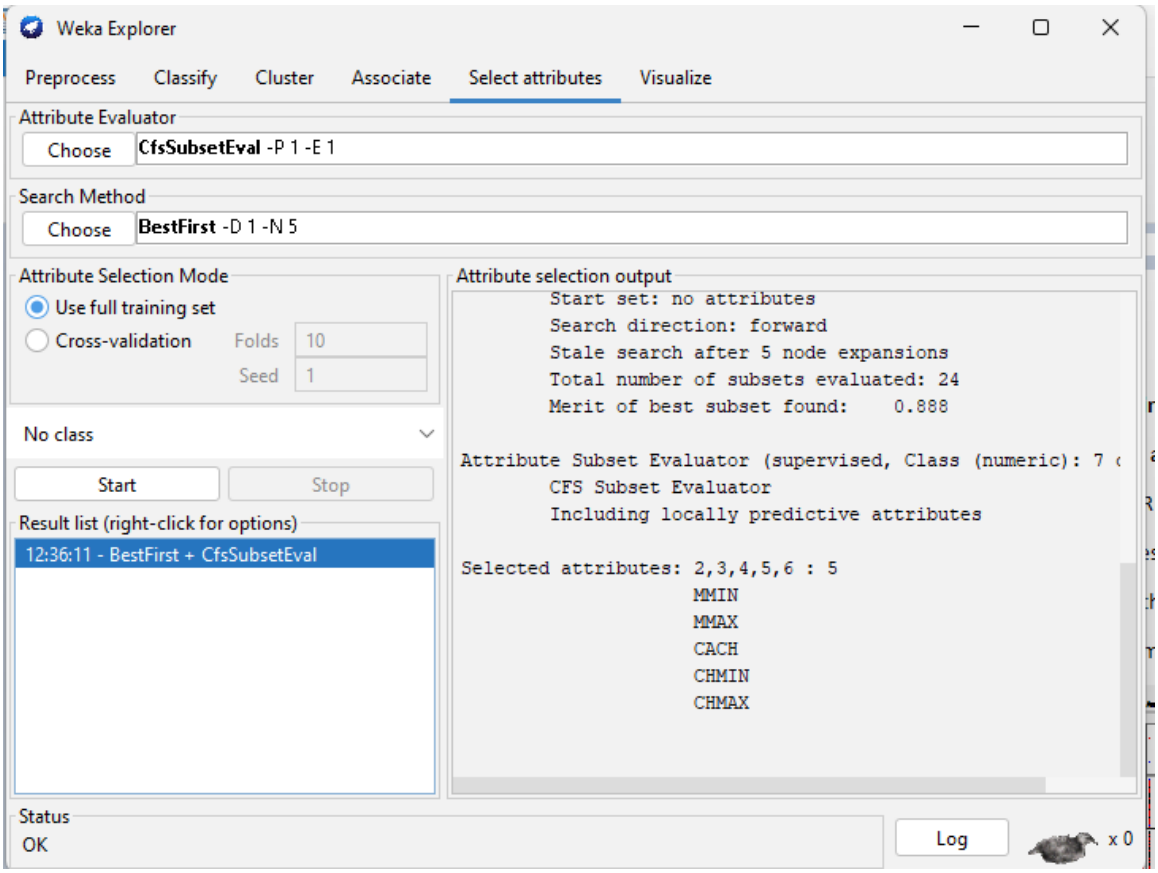
This panel contains schemes for learning association rules, and the learners are chosen and configured in the same way as the clusterers, filters, and classifiers in the other panels.



5. Selecting Attributes:

Searching and Evaluating

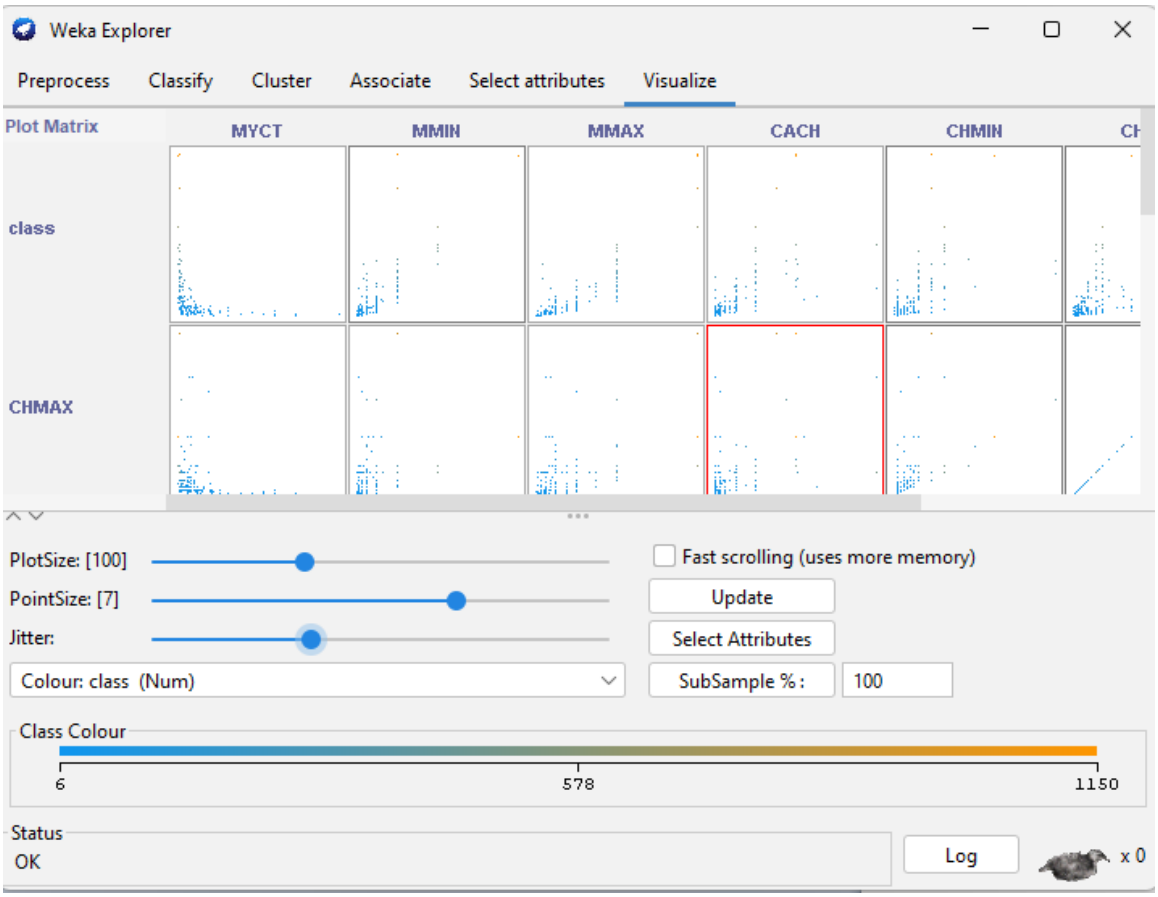
Attribute selection involves searching through all possible combinations of attributes in the database to find which subset of attributes works best for prediction. To do this, two objects must be set up: an attribute evaluator and a searchmethod. The evaluator determines what method is used to assign a worth toeach subset of attributes. The search method determines what style of search is performed.



6. Visualizing:

Study the arff file format

An ARFF (Attribute-Relation File Format) file is an ASCII text file that describes a list of instances sharing a set of attributes. ARFF files were developed by the Machine Learning Project at the Department of Computer Science of The University of Waikato for use with the Weka machine learning software.


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Plot Matrix
MYCT
MMIN
MMAX
CACH
CHMIN
class
CHMAX
PlotSize: [100]
PointSize: [7]
Jitter:
Fast scrolling (uses more memory)
Update
Select Attributes
Colour: class (Num)
SubSample % :
100
Class Colour
6
578
1150
Status
OK
Log
x 0

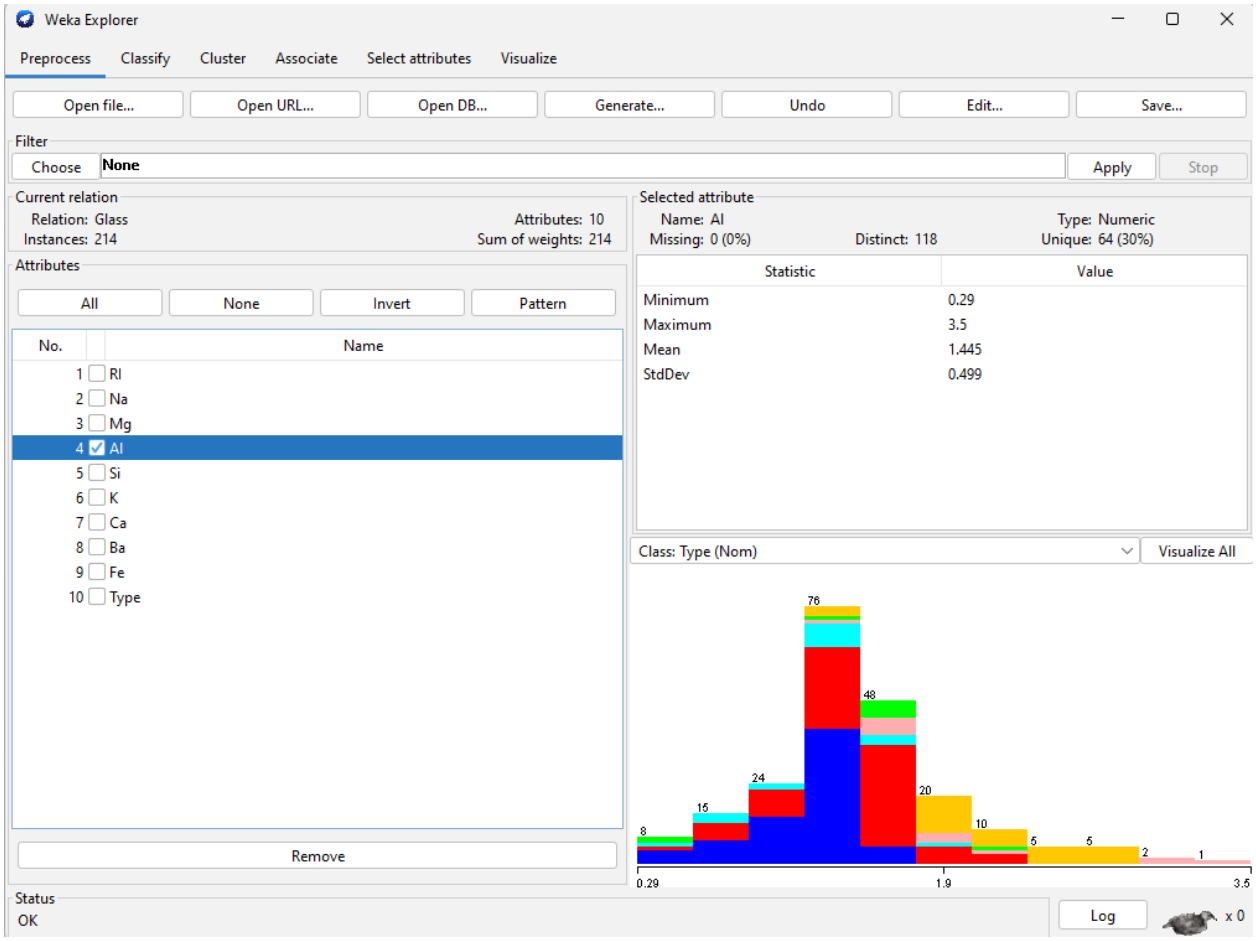
EX.NO:2 APPLY WEKA TOOL FOR DATA VALIDATION

STEP 1:PREPROCESSING

1. For preprocessing the data after selecting the dataset

2. Select Filter option & apply the resample filter & see the below results.

3. Select another filter option & apply the discretization filter start the attribute selection process by clicking on “Start” button.



STEP 2:CLASSIFY

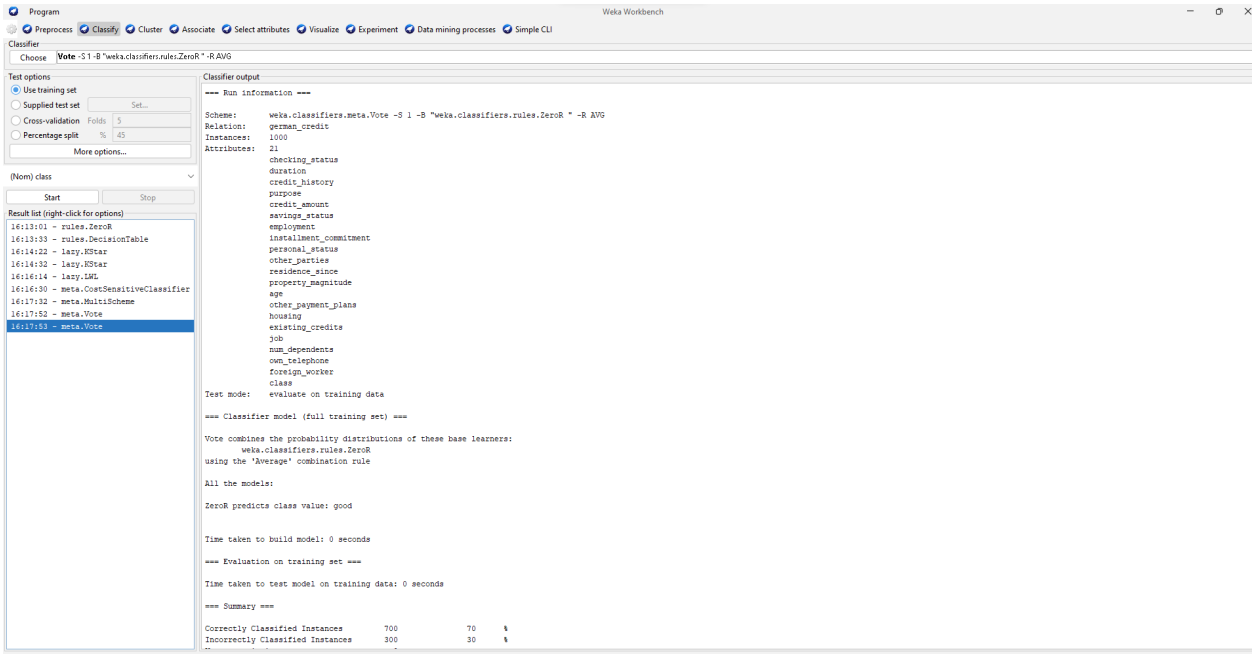
PROCEDURE:

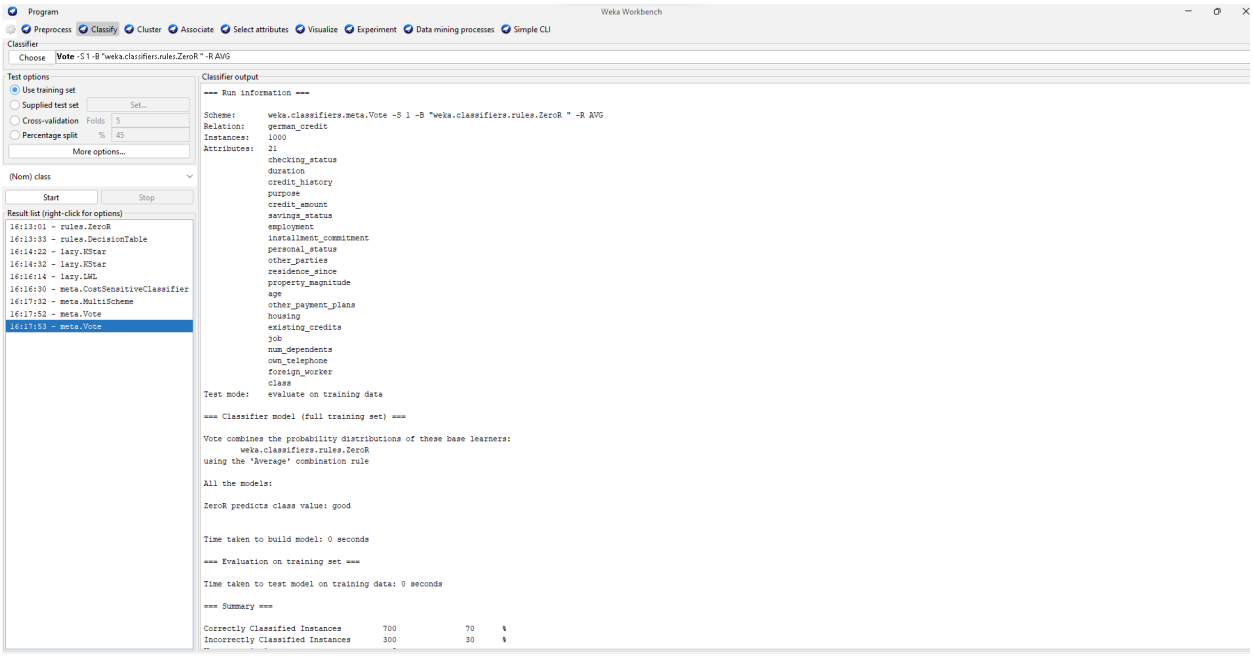
1. Load the dataset (Contact-lenses.arff) into weka

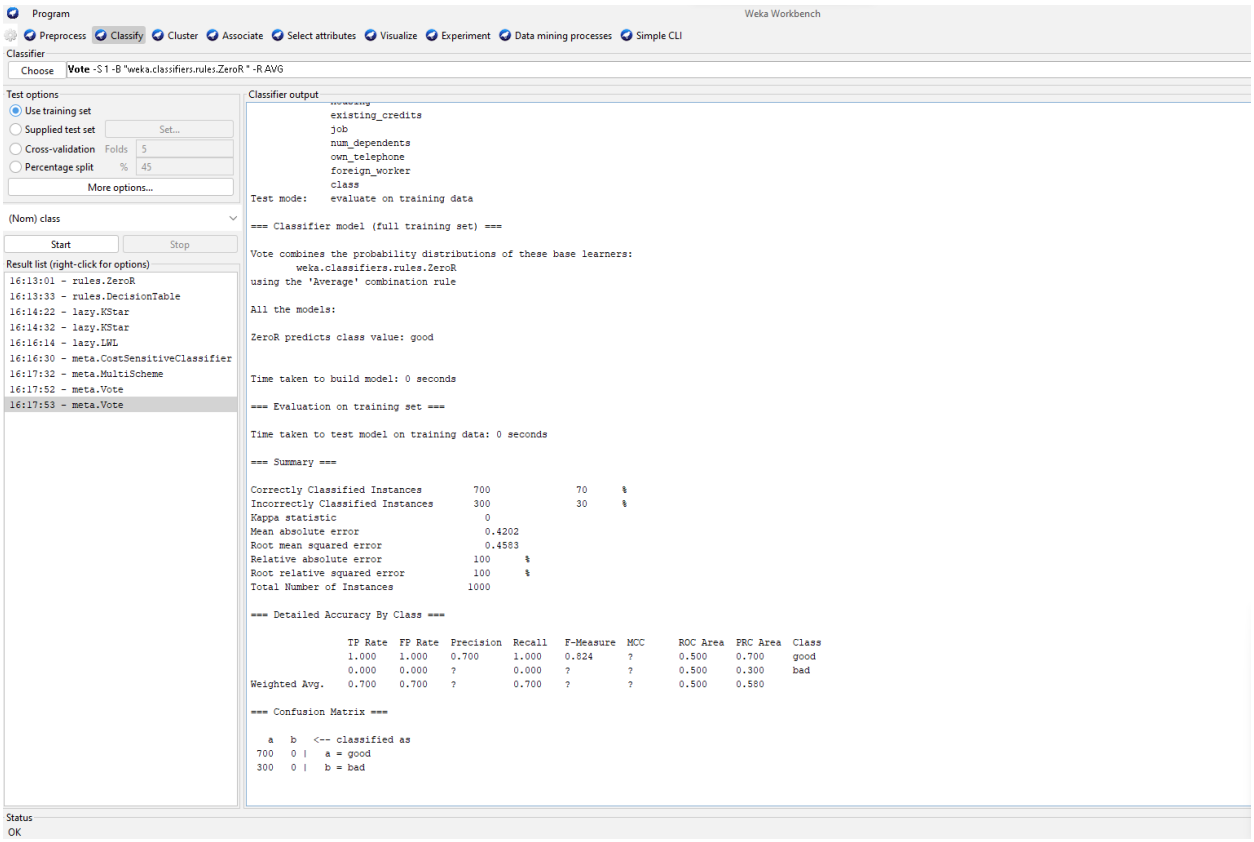
2. Go to classify option & in left-hand navigation bar we can see different classification algorithms under tree section

3. In which we selected Id3 algorithm, in more options select the output entropy evaluation measures& click on start option.

4. Then we will get classifier output, entropy values& Kappa Statistic as represented below.







STEP3:ASSOCIATE

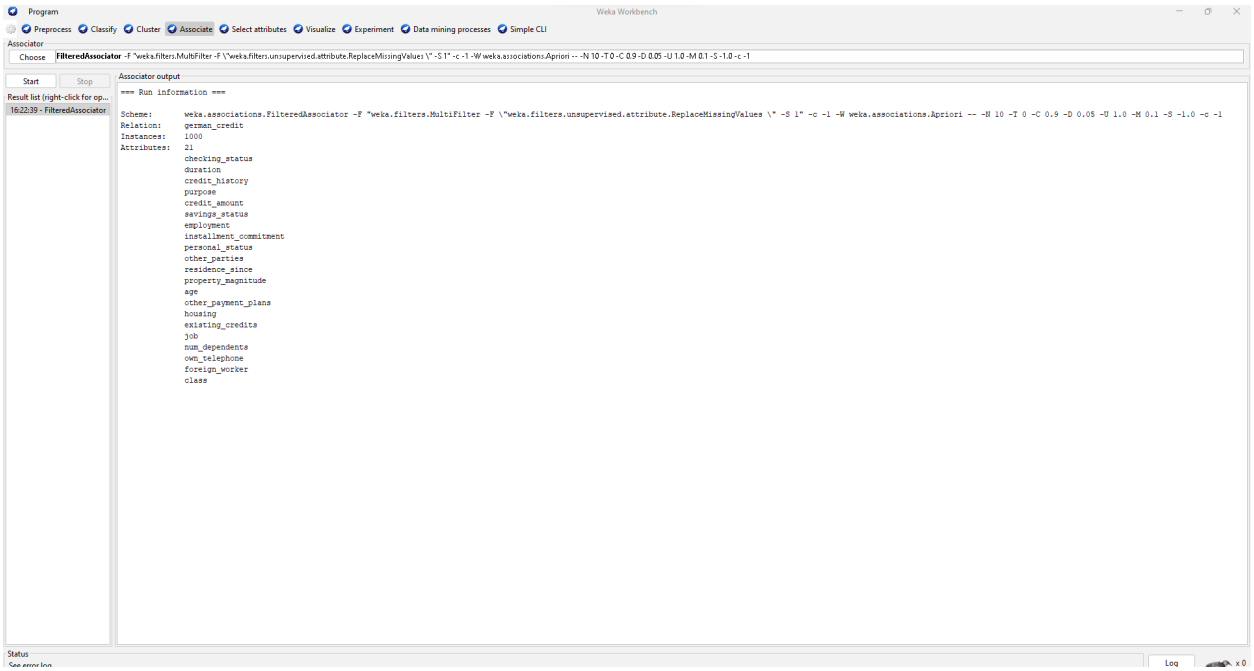
PROCEDURE:

1. Load the dataset (Breast-Cancer.arff) into weka tool& select the discretize filter and apply it.

2. Go to associate option & in left-hand navigation bar we can see different associat algorithms.

3. In which we can select Aprori algorithm & click on select option.

4. Below we can see the rules generated with different support & confidence values for that selected dataset.



STEP 4:CLUSTER

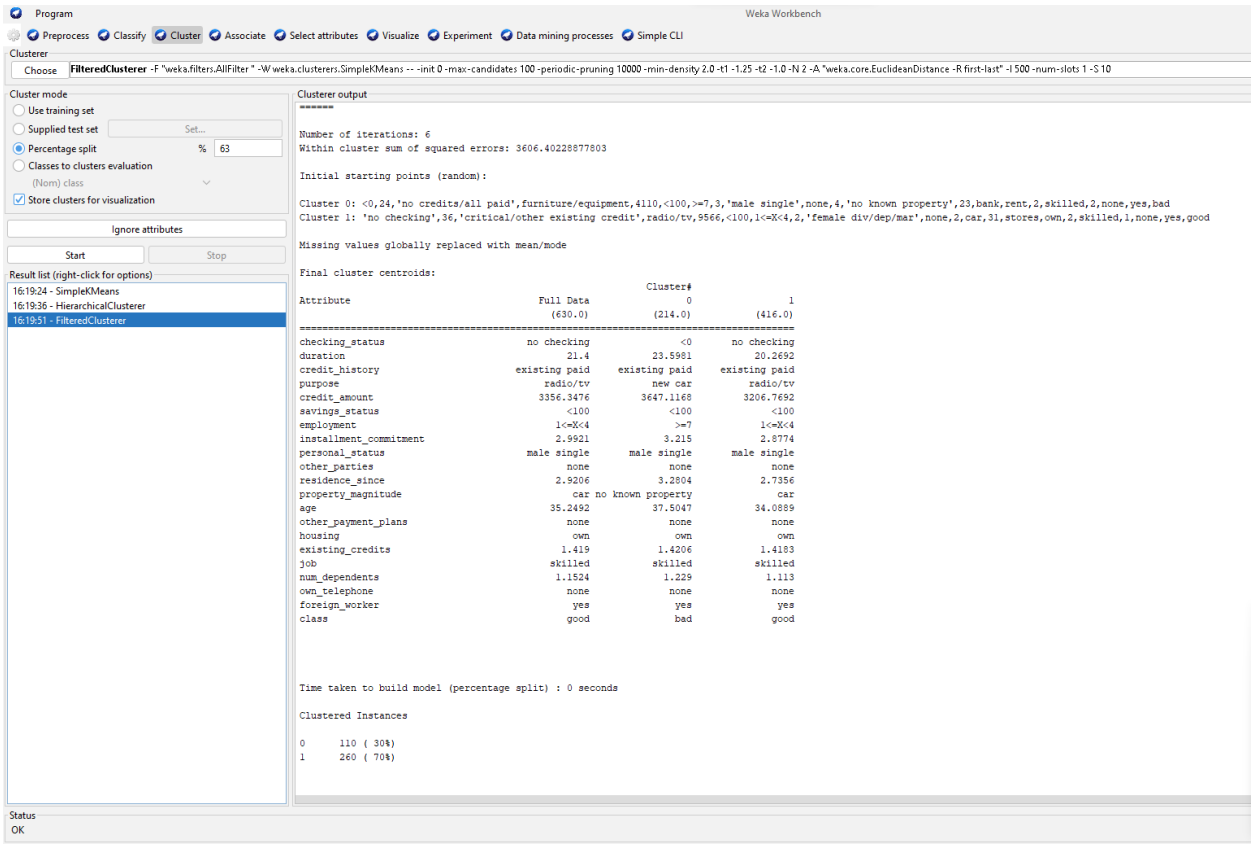
PROCEDURE:

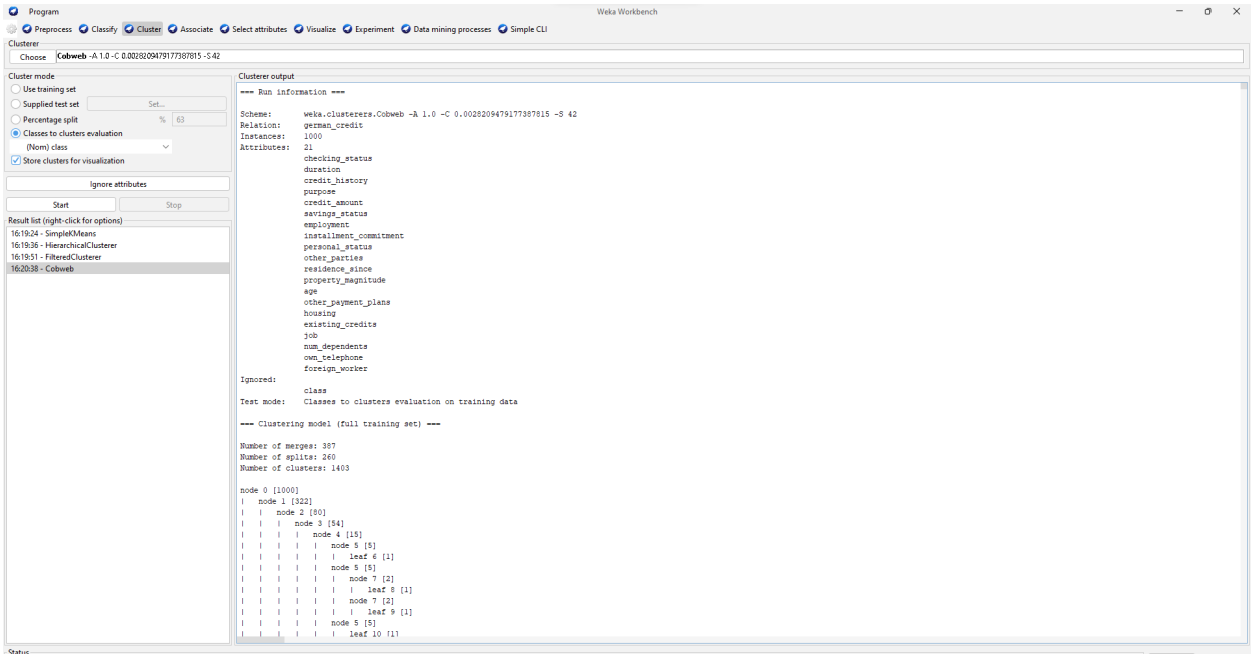
1. Load the dataset (Iris.arff) into weka tool

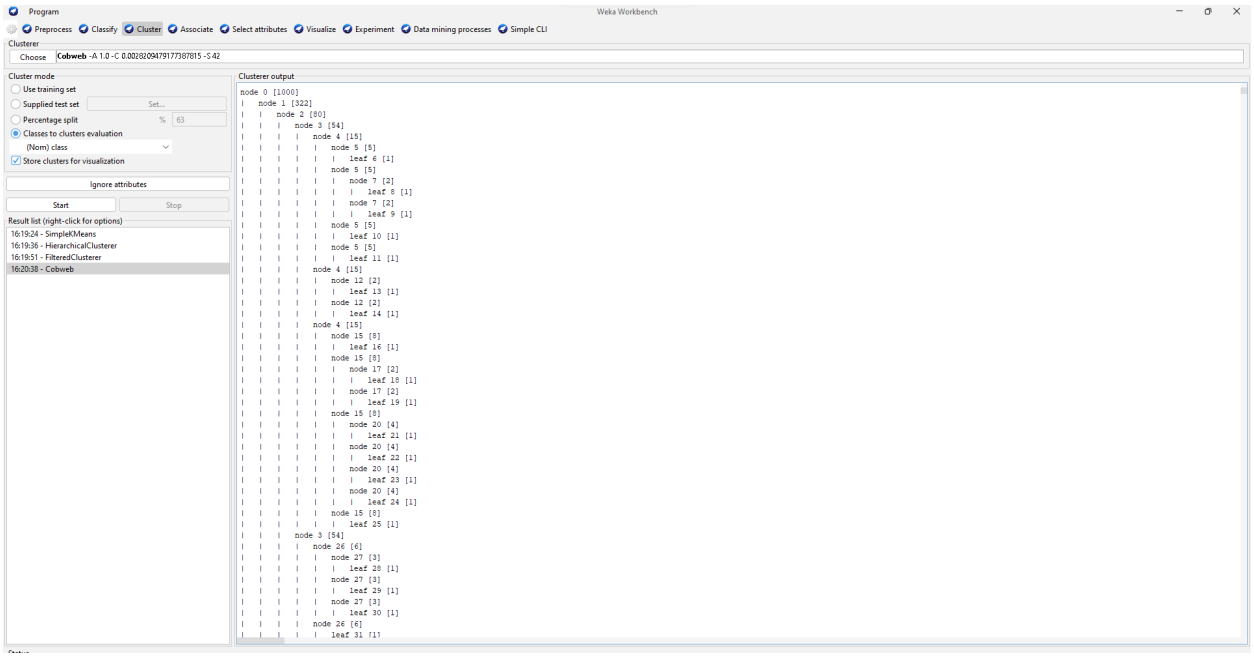
2. Go to classify option & in left-hand navigation bar we can see differentclustering algorithms under lazy section.

3. In which we selected Simple K-Means algorithm & click on start option with —use training set‖ test option enabled.

4. Then we will get the sum of squared errors, centroids, No. of iterations & clustered instances as represented below.


Program
Weka Workbench
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Experiment
Data mining processes
Simple CLI
Clusterer
Choose
FilteredClusterer -F "weka.filters.AllFilter " -W weka.clusterers.SimpleKMeans -- -init 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 2 -A "weka.core.EuclideanDistance -R first-last" -I 500 -num-slots 1 -S 10
Cluster mode
Use training set
Supplied test set
Set...
Percentage split
%
63
Classes to clusters evaluation
(Nom) class
Store clusters for visualization
Ignore attributes
Start
Stop
Result list (right-click for options)
16:19:24 - SimpleKMeans
16:19:36 - HierarchicalClusterer
16:19:51 - FilteredClusterer
Clusterer output
=======
Number of iterations: 6
Within cluster sum of squared errors: 3606.40228877803
Initial starting points (random):
Cluster 0: <0,24,'no credits/all paid',furniture/equipment,4110,<100,>=7,3,'male single',none,4,'no known property',23,bank,rent,2,skilled,2,none,yes,bad
Cluster 1: 'no checking',36,'critical/other existing credit',radio/tv,9566,<100,1<=X<4,2,'female div/dep/mar',none,2,car,31,stores,own,2,skilled,1,none,yes,good
Missing values globally replaced with mean/mode
Final cluster centroids:
Cluster#
Attribute Full Data 0 1
(630.0) (214.0) (416.0)
checking_status no checking <0 no checking
duration 21.4 23.5981 20.2692
credit_history existing paid existing paid existing paid
purpose radio/tv new car radio/tv
credit_amount 3356.3476 3647.1168 3206.7692
savings_status <100 <100 <100
employment 1<=X<4 >=7 1<=X<4
installment_commitment 2.9921 3.215 2.8774
personal_status male single male single male single
other_parties none none none
residence_since 2.9206 3.2804 2.7356
property_magnitude car no known property car
age 35.2492 37.5047 34.0889
other_payment_plans none none none
housing own own own
existing_credits 1.419 1.4206 1.4183
job skilled skilled skilled
num_dependents 1.1524 1.229 1.113
own_telephone none none none
foreign_worker yes yes yes
class good bad good
Time taken to build model (percentage split) : 0 seconds
Clustered Instances
0 110 (30%)
1 260 (70%)
Status
OK

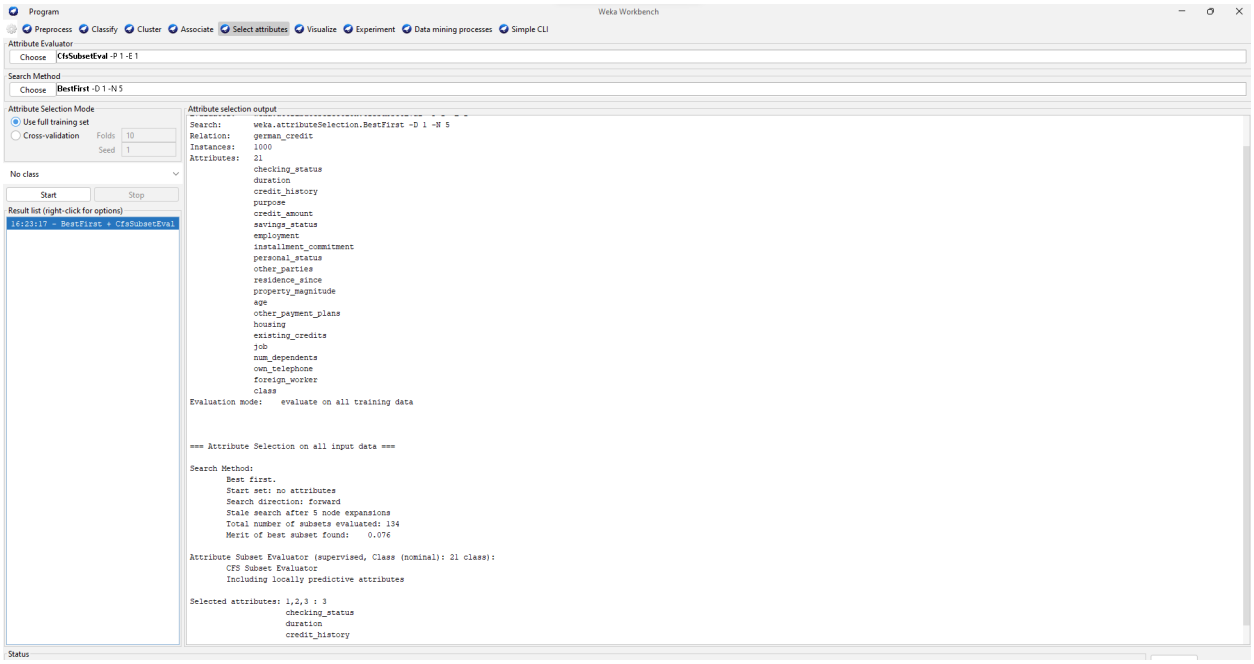

Program
Weka Workbench
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Experiment
Data mining processes
Simple CLI
Clusterer
Choose
Cobweb -A 1.0 -C 0.0028209479177387815 -S 42
Cluster mode
Use training set
Supplied test set
Set...
Percentage split
%
63
Classes to clusters evaluation
(Nom) class
Store clusters for visualization
Ignore attributes
Start
Stop
Result list (right-click for options)
16:19:24 - SimpleKMeans
16:19:36 - HierarchicalClusterer
16:19:51 - FilteredClusterer
16:20:38 - Cobweb
Clusterer output
=== Run information ===
Scheme: weka.clusterers.Cobweb -A 1.0 -C 0.0028209479177387815 -S 42
Relation: german_credit
Instances: 1000
Attributes: 21
checking_status
duration
credit_history
purpose
credit_amount
savings_status
employment
installment_commitment
personal_status
other_parties
residence_since
property_magnitude
age
other_payment_plans
housing
existing_credits
job
num_dependents
own_telephone
foreign_worker
Ignored:
class
Test mode: Classes to clusters evaluation on training data
=== Clustering model (full training set) ===
Number of merges: 387
Number of splits: 260
Number of clusters: 1403
node 0 [1000]
| node 1 [322]
| | node 2 [80]
| | | node 3 [54]
| | | | node 4 [15]
| | | | | node 5 [5]
| | | | | | leaf 6 [1]
| | | | | node 5 [5]
| | | | | | node 7 [2]
| | | | | | | leaf 8 [1]
| | | | | | node 7 [2]
| | | | | | | leaf 9 [1]
| | | | | node 5 [5]
| | | | | | leaf 10 [1]
Status

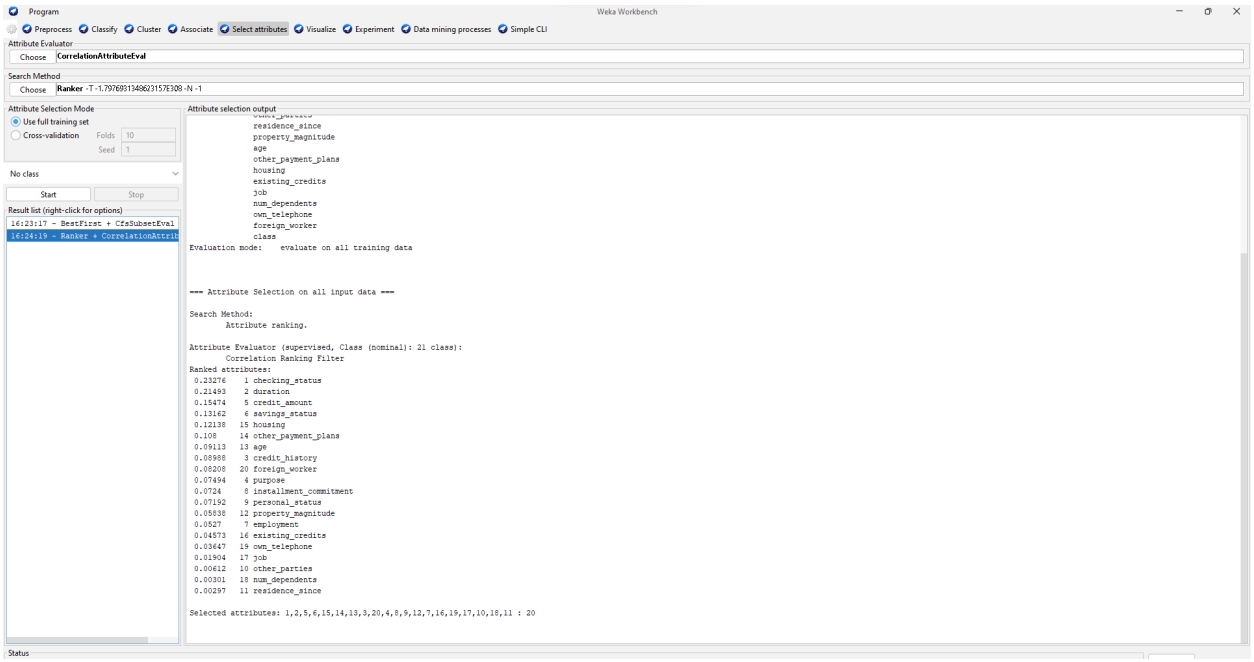


STEP 5:SELECT ATTRIBUTES

PROCEDURE:

1.Open the Weka GUI Chooser.
2 Click the “Explorer” button to launch the Explorer.
3.Open the dataset.
4. Click the “Select attributes” tab to access the feature selection methods.





STEP 6:VISUALIZE

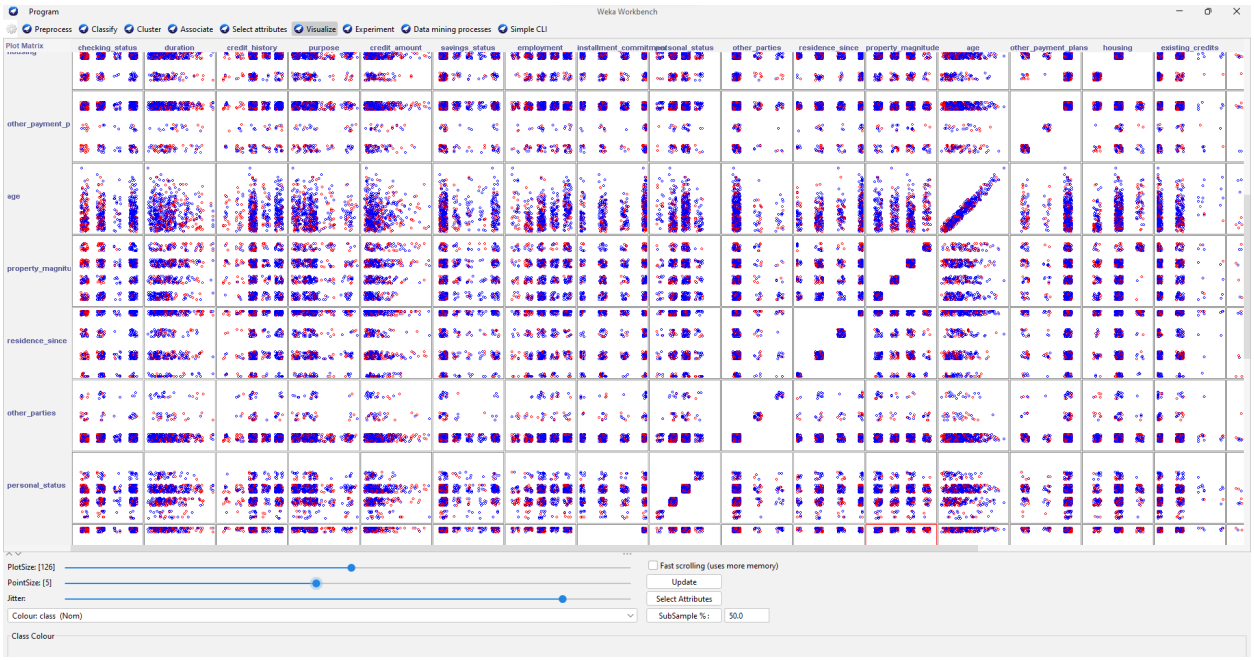
PROCEDURE:

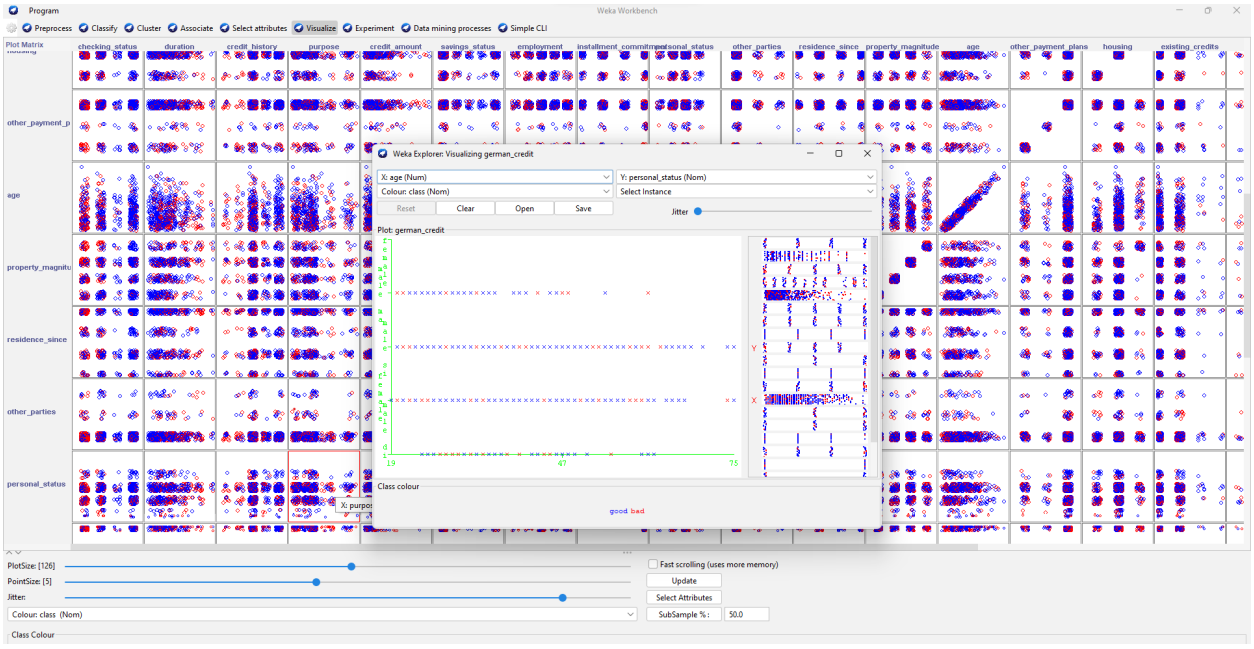
1. To visualize the dataset, go to the Visualize tab.

2.The tab shows the attributes plot matrix.

3.The dataset attributes are marked on the x-axis and y-axis while the instances are plotted.

4. The box with the x-axis attribute and y-axis attribute can be enlarged.


Program
Weka Workbench
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Experiment
Data mining processes
Simple CLI
Plot Matrix
checking_status
duration
credit_history
purpose
credit_amount
savings_status
employment
installment_commitment
personal_status
other_parties
residence_since
property_magnitude
age
other_payment_plans
housing
existing_credits
PlotSize: [126]
PointSize: [5]
Jitter:
Colour: class (Nom)
Fast scrolling (uses more memory)
Update
Select Attributes
SubSample %:
50.0
Class Colour


Program
Weka Workbench
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Experiment
Data mining processes
Simple CLI
Plot Matrix
Weka Explorer: Visualizing german_credit
X: age (Num)
Y: personal_status (Nom)
Colour: class (Nom)
Select Instance
Reset
Clear
Open
Save
Jitter
Plot: german_credit
19
47
75
Class colour
good
bad
PlotSize: [126]
PointSize: [5]
Jitter:
Colour: class (Nom)
Fast scrolling (uses more memory)
Update
Select Attributes
SubSample %:
50.0
Class Colour

EX.NO : 3 DATE:	**Planning the architecture for real-time application**

Aim:

To plan the architecture for a real-time application using Weka

Algorithms:

Step-1 : Identify the specific real-time processing needs and objectives of the application.

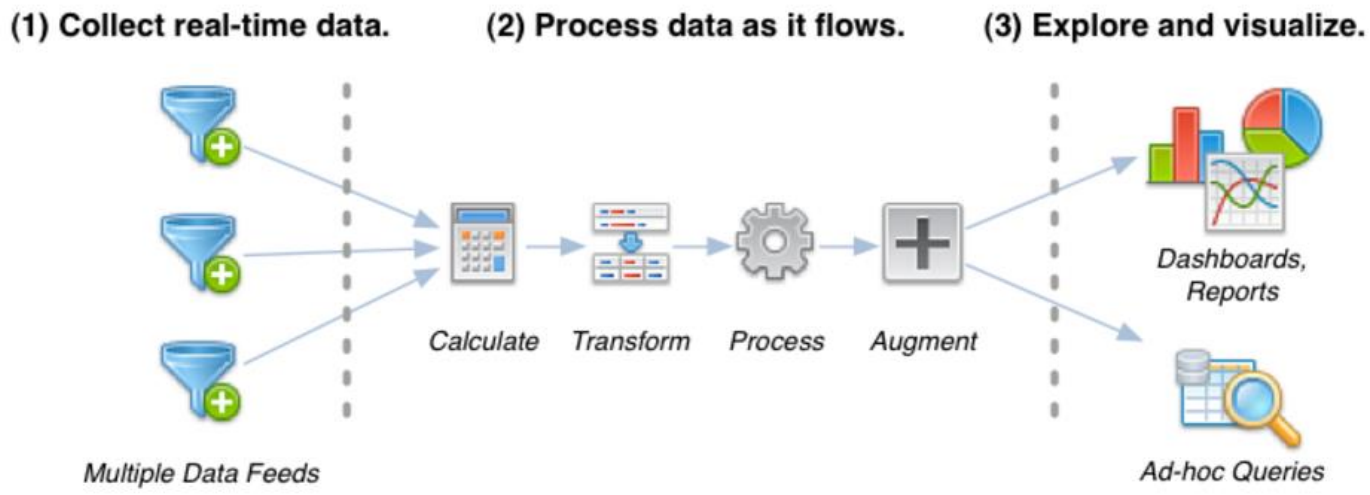
Step-2 : Explore Weka's algorithms and models suitable for real-time applications, considering factors like speed and accuracy

Step-3 : Design the overall architecture, including components for data ingestion, processing, storage, and visualization.

Step-4 : Plan for scalability by designing the architecture to handle increasing data volume and processing demands.

Step-5 : Develop testing strategies to validate the effectiveness and efficiency of the planned architecture.

Procedure:



1.**Scalability Considerations :**

As your real-time application grows in usage and data volume, scalability becomes crucial. Design your architecture to scale horizontally by adding more resources or

distributing workload across multiple servers. Weka's algorithms should be able to scale efficiently with increased computational demands.

2.Fault Tolerance and Resilience:

Implement mechanisms to handle failures gracefully, ensuring that your real-time application remains operational even in the face of component failures. This might involve replicating critical components, implementing retry logic, or using fault-tolerant data storage solutions.

3.Data Governance and Compliance:

Ensure that your real-time application adheres to data governance policies and regulatory requirements. This includes maintaining data privacy, ensuring data security, and complying with relevant regulations such as GDPR or HIPAA. Weka's functionalities should align with your data governance framework.

4.Cost Optimization:

Monitor resource usage and optimize costs associated with running your real-time application. This involves efficiently utilizing computational resources, optimizing data storage strategies, and leveraging cost-effective cloud services. Weka's algorithms should be chosen and configured to minimize unnecessary resource consumption.

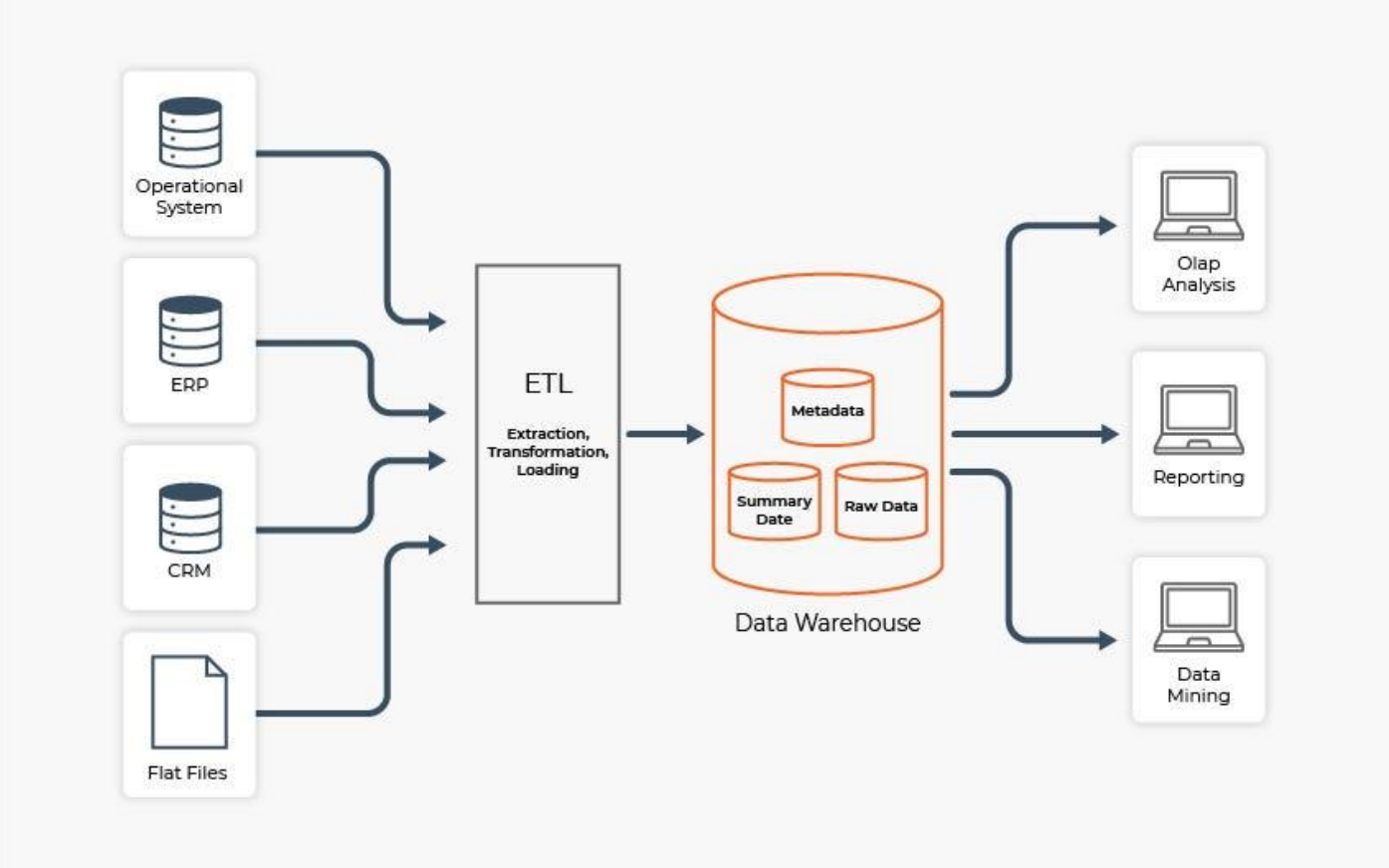
5.User Experience and Feedback Loop:

Incorporate mechanisms to gather feedback from users and monitor user interactions with your real-time application. This feedback loop can inform iterative improvements to the application's functionality, performance, and usability. Weka's models can be continuously refined based on user feedback and changing requirements.

6.Cross-Platform Compatibility:

Ensure compatibility and interoperability with different platforms and environments where your real-time application may be deployed. This includes compatibility

with various operating systems, cloud platforms, and programming languages. Weka's libraries and APIs should be compatible with your chosen deployment environment.



Result:

Thus, the study of planning architecture for real-time applications in Weka was successful.

EX.NO: 04	WRITE THE QUERY FOR SCHEMA DEFINITION
DATE:	

AIM:

To write the query for schema definition.

ALGORITHM:

STEP 1: Start mysql.

STEP 2: Determine the data warehousing architecture (e.g., star schema, snowflake schema).

STEP 3: Identify the main subject of analysis (e.g., sales data) to define the fact table.

STEP 4: List down the dimensions (e.g., time, product, customer, store) that relate to the fact table.

STEP 5: Use the 'CREATE TABLE' SQL statement to define the fact table with necessary attributes and constraints (data types, primary keys).

STEP 6: Use 'CREATE TABLE' SQL statements to define dimension tables with relevant attributes, linking them to the fact table using foreign keys.

STEP 7: Execute the SQL statements in the DBMS to create the schema within the data warehouse.

STEP 8: Validate the schema creation by querying the tables to ensure they have been set up correctly.

STEP 9: Document the schema definitions for future reference and maintenance.

STEP 10: Close the DBMS software upon completion of the schema definition.

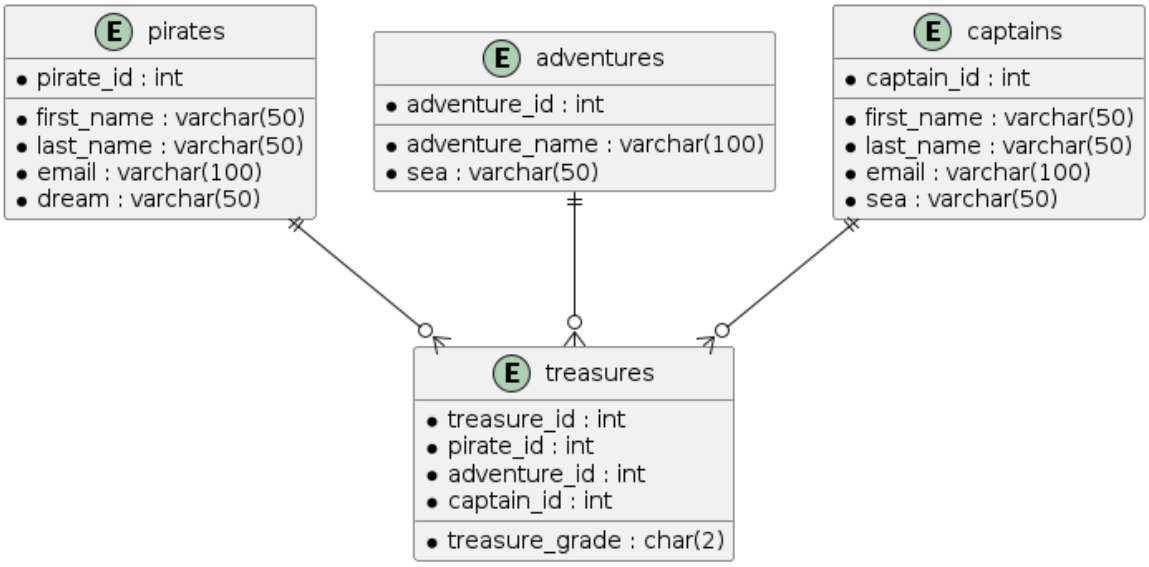
STEP 11: Stop.

PROGRAM:

```
CREATE TABLE pirates (
    pirate_id INT PRIMARY KEY,
    first_name VARCHAR(50),
    last_name VARCHAR(50),
    email VARCHAR(100),
    dream VARCHAR(50)
);

CREATE TABLE adventures (
    adventure_id INT PRIMARY KEY,
    adventure_name VARCHAR(100),
    sea VARCHAR(50)
);

CREATE TABLE captains (
    captain_id INT PRIMARY KEY,
    first_name VARCHAR(50),
    last_name VARCHAR(50),
    email VARCHAR(100),
    sea VARCHAR(50)
);

CREATE TABLE treasures (
    treasure_id INT PRIMARY KEY,
    pirate_id INT,
    adventure_id INT,
    captain_id INT,
    treasure_grade CHAR(2),
    FOREIGN KEY (pirate_id) REFERENCES pirates(pirate_id),
    FOREIGN KEY (adventure_id) REFERENCES adventures(adventure_id),
    FOREIGN KEY (captain_id) REFERENCES captains(captain_id)
);
```

```
INSERT INTO pirates (pirate_id, first_name, last_name, email, dream)
VALUES
(1, 'Monkey D.', 'Luffy', 'luffy@one.piece', 'Pirate King'),
(2, 'Roronoa', 'Zoro', 'zoro@one.piece', 'Greatest Swordsman');

INSERT INTO adventures (adventure_id, adventure_name, sea)
VALUES
(101, 'Search for One Piece', 'Grand Line'),
(102, 'Master the Way of the Sword', 'East Blue');

INSERT INTO captains (captain_id, first_name, last_name, email, sea)
VALUES
(201, 'Red-Haired', 'Shanks', 'shanks@one.piece', 'Grand Line'),
(202, 'Dracule', 'Mihawk', 'mihawk@one.piece', 'Grand Line');

INSERT INTO treasures (treasure_id, pirate_id, adventure_id, captain_id, treasure_grade)
VALUES
(301, 1, 101, 201, 'S'),
(302, 2, 102, 202, 'A');

SELECT
  p.pirate_id,
  p.first_name,
  p.last_name,
  t.treasure_grade,
  a.adventure_name
FROM
  pirates p
JOIN
  treasures t ON t.pirate_id = p.pirate_id
JOIN
  adventures a ON t.adventure_id = a.adventure_id;

SELECT
  a.adventure_name,
  c.first_name,
  c.last_name,
  t.treasure_grade
FROM
  adventures a
JOIN
  treasures t ON a.adventure_id = t.adventure_id
JOIN
  captains c ON t.captain_id = c.captain_id;
```

RELATIONSHIPS BETWEEN ENTITIES:



Pirates and Treasures:

A one-to-many relationship exists between pirates and treasures. This means a single pirate can be associated with multiple treasures. This relationship is depicted by the line connecting pirates to treasures with a crow's foot at the treasures end, indicating multiple records.

Adventures and Treasures:

Adventures also have a one-to-many relationship with treasures. An adventure can result in multiple grades (or treasures) being awarded. Similar to the pirates, the crow's foot at the treasures end shows the potential for multiple treasures in one adventure.

Captains and Treasures:

Lastly, captains, who can be thought of as teachers or guides in the "One Piece" world, have a one-to-many relationship with treasures. Each captain may award several different treasure grades across multiple adventures.

Each treasures record relates back to exactly one pirates, one adventures, and one captains entry, which are the foreign keys referring to their respective pirate_id, adventure_id, and captain_id primary keys. These relationships are foundational in tracking which pirate earned what treasure grade under which captain's guidance on which adventure. This setup allows for a rich and complex story to be captured within the data, much like the epic tales of the "One Piece" universe.

RESULT:

EX.NO: 05	DESIGN DATA WARE HOUSE FOR REAL TIME APPLICATION
DATE:	

AIM:

To design data ware house for real time application.

ALGORITHM:

STEP 1: Start the program.

STEP 2: Prompt user for MySQL database credentials and desired database name.

STEP 3: Establish a connection to MySQL using the provided credentials.

STEP 4: If the connection is successful, inform the user. If not, display an error message.

STEP 5: Execute a query to create a new database with the provided name.

STEP 6: Re-establish connection including the new database.

STEP 7: Execute queries to create 'student', 'professor', 'course', 'time', and 'enrollment' tables.

STEP 8: Inform the user upon successful creation of each table. If not, display an error message.

STEP 9: Write a configuration file with the database connection details for a web application deployment.

STEP 10: Retrieve deployment script for the web application from a remote source.

STEP 11: Write deployment script content to a local file and execute the script to deploy the web application.

STEP 12: Stop the program.

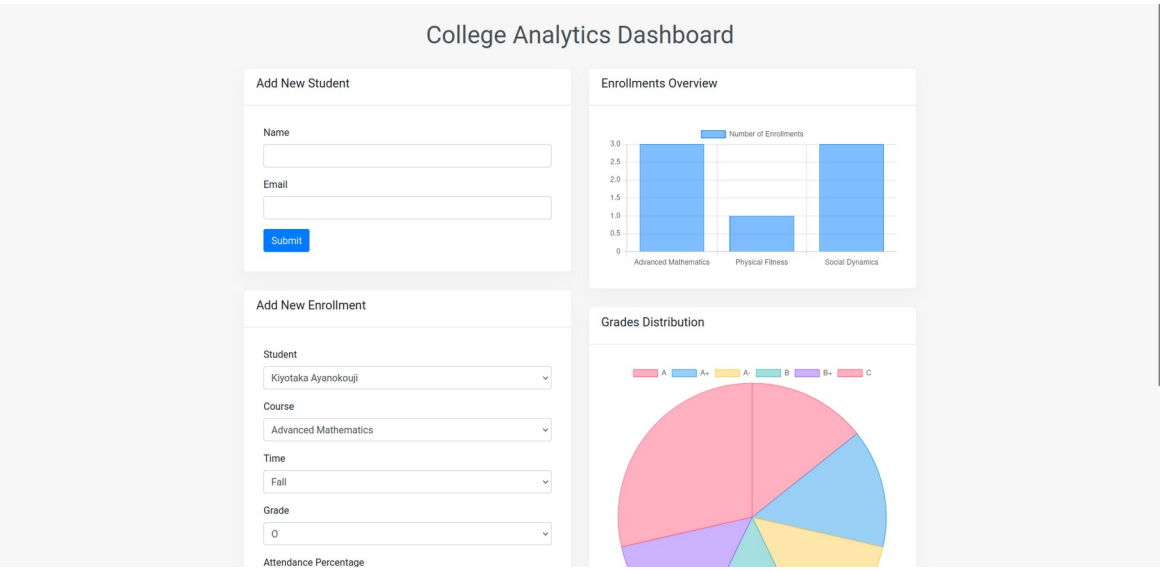
PROGRAM:

```
import mysql.connector
from mysql.connector import Error
import requests
import os
import subprocess

import subprocess

def create_mysql_connection(host_name, user_name, user_password,
database_name=None):
    connection = None
    try:
        connection = mysql.connector.connect(
            host=host_name,
            user=user_name,
            passwd=user_password,
            database=database_name
        )
        print("MySQL Database connection successful")
    except Error as err:
        print(f"Error: '{err}'")

    return connection

def create_database(connection, query):
    cursor = connection.cursor()
    try:
        cursor.execute(query)
        print("Database created successfully")
    except Error as err:
        print(f"Error: '{err}'")

def execute_query(connection, query):
    cursor = connection.cursor()
    try:
        cursor.execute(query)
        connection.commit()
        print("Query successful")
    except Error as err:
        print(f"Error: '{err}'")

host_name = input("MySQL hostname > ")
```

```
user_name = input("MySQL username > ")
passwd    = input("MySQL password > ")
database_name = input("Enter new database name to create > ")

connection = create_mysql_connection(host_name, user_name, passwd)
create_database_query = f"CREATE DATABASE {database_name}"
create_database(connection, create_database_query)

db_connection = create_mysql_connection(host_name, user_name, passwd,
database_name)
create_student_table_query = "CREATE TABLE student (id INTEGER PRIMARY KEY
AUTO_INCREMENT, name VARCHAR(50) NOT NULL, email VARCHAR(120)
UNIQUE NOT NULL);"
create_professors_table_query = "CREATE TABLE professor (id INTEGER PRIMARY
KEY AUTO_INCREMENT, name VARCHAR(50) NOT NULL, department
VARCHAR(100) NOT NULL);"
create_course_table_query = "CREATE TABLE course (id INTEGER PRIMARY KEY
AUTO_INCREMENT, title VARCHAR(100) NOT NULL, department VARCHAR(100)
NOT NULL, professor_id INTEGER, FOREIGN KEY (professor_id) REFERENCES
professor(id));"
create_time_table_query = "CREATE TABLE time (id INTEGER PRIMARY KEY
AUTO_INCREMENT, academic_year VARCHAR(20) NOT NULL, semester
VARCHAR(20) NOT NULL);"
create_enrollment_table_query = "CREATE TABLE enrollment (id INTEGER PRIMARY
KEY AUTO_INCREMENT, student_id INTEGER, course_id INTEGER, time_id
INTEGER, grade VARCHAR(2), attendance_percentage FLOAT, FOREIGN KEY
(student_id) REFERENCES student(id), FOREIGN KEY (course_id) REFERENCES
course(id), FOREIGN KEY (time_id) REFERENCES time(id));"

execute_query(db_connection, create_student_table_query)
execute_query(db_connection, create_professors_table_query)
execute_query(db_connection, create_course_table_query)
execute_query(db_connection, create_time_table_query)
execute_query(db_connection, create_enrollment_table_query)


# Web Application Auto deploy

config = f"""DB_USERNAME="{user_name}"
DB_PASSWORD="{passwd}"
DB_DATABASE="{database_name}"
DB_HOSTNAME="{host_name}"
"""
config_file = open("config.py", "w")
```

```
config_file.write(config)
config_file.close()

content = requests.get("https://pastebin.com/raw/nkBxBu9q").content.decode()
web_auto_file = open("web_auto.py", "w")
web_auto_file.write(content)
web_auto_file.close()
subprocess.call(["python3 web_auto.py"], shell=True)
```

OUTPUT:


College Analytics Dashboard
Add New Student
Name
Email
Submit
Add New Enrollment
Student
Kiyotaka Ayanokouji
Course
Advanced Mathematics
Time
Fall
Grade
0
Attendance Percentage
Enrollments Overview
Number of Enrollments
3.0
2.5
2.0
1.5
1.0
0.5
0
Advanced Mathematics
Physical Fitness
Social Dynamics
Grades Distribution
A
A+
A-
B
B+
C

RESULT:

EX.NO: 06	ANALYZE THE DIMENSIONAL MODELING
DATE:	

AIM:

To explore data and integrate with weka.

ALGORITHM:

STEP 1: Connect to your MYSQL database instance.

STEP 2: Create dimension tables for Date, Product, Customer, and Store using the SQL 'CREATE TABLE' command.

STEP 3: Populate each dimension table with sample data ensuring data integrity and consistency.

STEP 4: Create a central fact table for Sales that includes foreign keys to each dimension table.

STEP 5: Insert sample transactional data into your Sales fact table, corresponding to the foreign keys in the dimension tables.

STEP 6: Query your Star Schema to analyze the data with SQL commands such as 'SELECT', 'GROUP BY', and 'JOIN'.

STEP 7: Visualize or report the analysis results to understand the business process insights.

STEP 8: Validate the data model and optimize the schema for performance as needed, such as indexing foreign keys.

STEP 9: Document your findings and the schema design for reference and presentation.

STEP 10: Clean up the environment if necessary by removing or archiving the dataset.

PROGRAM

STAR SCHEMA:

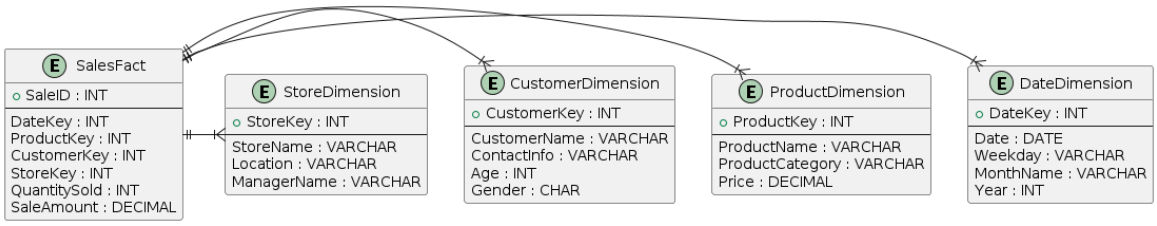
A Star Schema is a type of database schema that is a popular choice for data warehouse designs because of its simplicity and performance benefits.

Define Business Process: Choose a business process like 'Sales.' The fact table will record sales transactions.

Identify Fact Table: The fact table will be central in Star Schema, and it will contain quantitative data about sales, like sale_amount, quantity_sold, etc. Don't forget to include keys that will connect to the dimension tables.

Identify Dimension Tables: The dimension tables provide context to facts. For a Sales business process, common dimensions include Date, Product, Customer, and Store.

Design Schema: In MariaDB, create tables using the CREATE TABLE statement with appropriate data types for columns.



- SalesFact.DateKey references DateDimension.DateKey: This means that each sale record corresponds to a single date entry, providing the date context for the sale.
- SalesFact.ProductKey references ProductDimension.ProductKey: This relationship tells us what product the sale record is associated with, giving details like product name and category.
- SalesFact.CustomerKey references CustomerDimension.CustomerKey: Through this key, we can identify who the customer was for each sale.
- SalesFact.StoreKey references StoreDimension.StoreKey: This informs us which store the sale took place in and gives us additional store details.

```
CREATE TABLE DateDimension (
  DateKey INT AUTO_INCREMENT,
  Date DATE,
  Weekday VARCHAR(9),
  MonthName VARCHAR(9),
  Year INT,
  PRIMARY KEY (DateKey)
);

CREATE TABLE ProductDimension (
  ProductKey INT AUTO_INCREMENT,
  ProductName VARCHAR(255),
  ProductCategory VARCHAR(255),
  Price DECIMAL(10,2),
  PRIMARY KEY (ProductKey)
);

CREATE TABLE CustomerDimension (
  CustomerKey INT AUTO_INCREMENT,
  CustomerName VARCHAR(255),
  ContactInfo VARCHAR(255),
  Age INT,
  Gender CHAR(1),
  PRIMARY KEY (CustomerKey)
);

CREATE TABLE StoreDimension (
  StoreKey INT AUTO_INCREMENT,
  StoreName VARCHAR(255),
  Location VARCHAR(255),
  ManagerName VARCHAR(255),
  PRIMARY KEY (StoreKey)
);

CREATE TABLE SalesFact (
  SaleID INT AUTO_INCREMENT,
  DateKey INT,
  ProductKey INT,
  CustomerKey INT,
  StoreKey INT,
  QuantitySold INT,
  SaleAmount DECIMAL(10,2),
```

```
  PRIMARY KEY (SaleID),
  FOREIGN KEY (DateKey) REFERENCES DateDimension(DateKey),
  FOREIGN KEY (ProductKey) REFERENCES ProductDimension(ProductKey),
  FOREIGN KEY (CustomerKey) REFERENCES CustomerDimension(CustomerKey),
  FOREIGN KEY (StoreKey) REFERENCES StoreDimension(StoreKey)
);
```

Testing:

```
INSERT INTO DateDimension (Date, Weekday, MonthName, Year) VALUES
  ('2022-01-10', 'Monday', 'January', 2022),
  ('2022-01-11', 'Tuesday', 'January', 2022),
  ('2022-01-12', 'Wednesday', 'January', 2022);

INSERT INTO ProductDimension (ProductName, ProductCategory, Price) VALUES
  ('Potato kettle', 'Kitchenware', 2499.99),
  ('Medvedeva light', 'Electronics', 4599.50),
  ('Pavlovsk tablet', 'Electronics', 15999.90);

INSERT INTO CustomerDimension (CustomerName, ContactInfo, Age, Gender)
VALUES
  ('Alexey Navalny', 'navalny@gmail.com', 45, 'M'),
  ('Nadezhda Savchenko', 'nadiya@moscow.ru', 40, 'F'),
  ('Ivan Urgant', 'ivanurgant@moscow.ru', 45, 'M');

INSERT INTO StoreDimension (StoreName, Location, ManagerName) VALUES
  ('Moscow Store', 'Moscow', 'Dmitry Anatolyevich'),
  ('Siberian Center', 'Novosibirsk', 'Anna Sergeyevna'),
  ('Donskoy Kiosk', 'Rostov-on-Don', 'Yakov Alexandrovich');

INSERT INTO SalesFact (DateKey, ProductKey, CustomerKey, StoreKey, QuantitySold,
SaleAmount) VALUES
  (1, 1, 1, 1, 2, 4999.98),
  (2, 2, 2, 2, 1, 4599.50),
  (3, 3, 3, 3, 1, 15999.90);

SELECT StoreDimension.StoreName, SUM(SalesFact.SaleAmount) AS TotalSales
FROM SalesFact
JOIN StoreDimension ON SalesFact.StoreKey = StoreDimension.StoreKey
GROUP BY StoreDimension.StoreName;
```

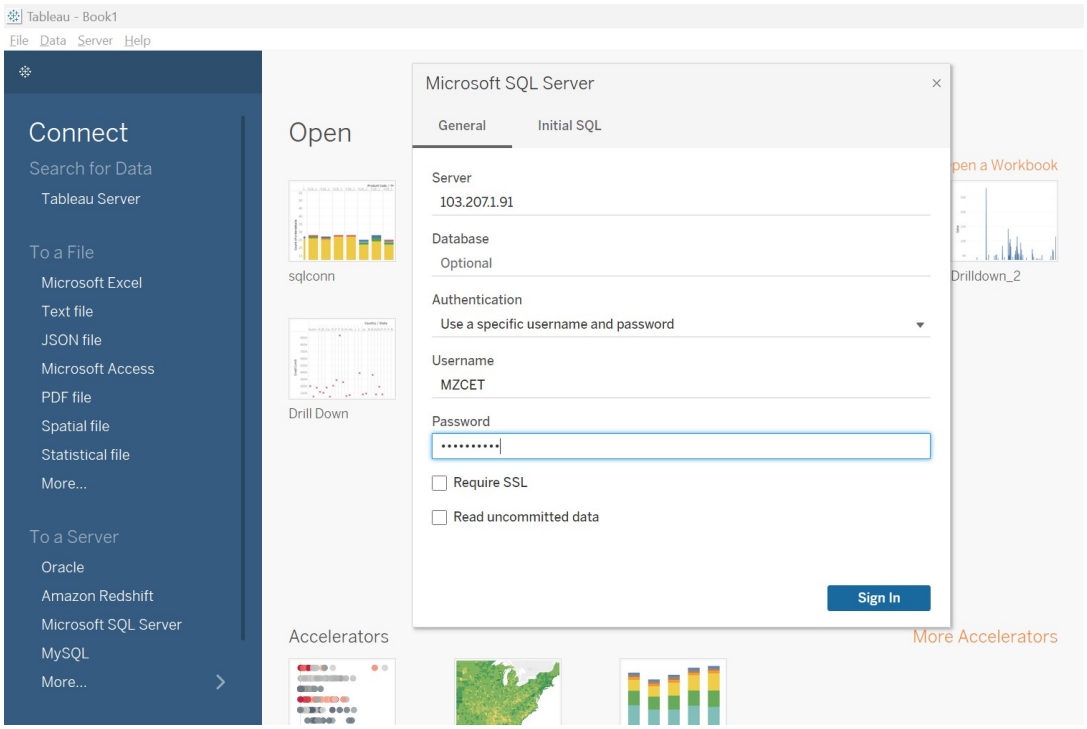
RESULT:

Ex.No.7 – Case Study on OLAP

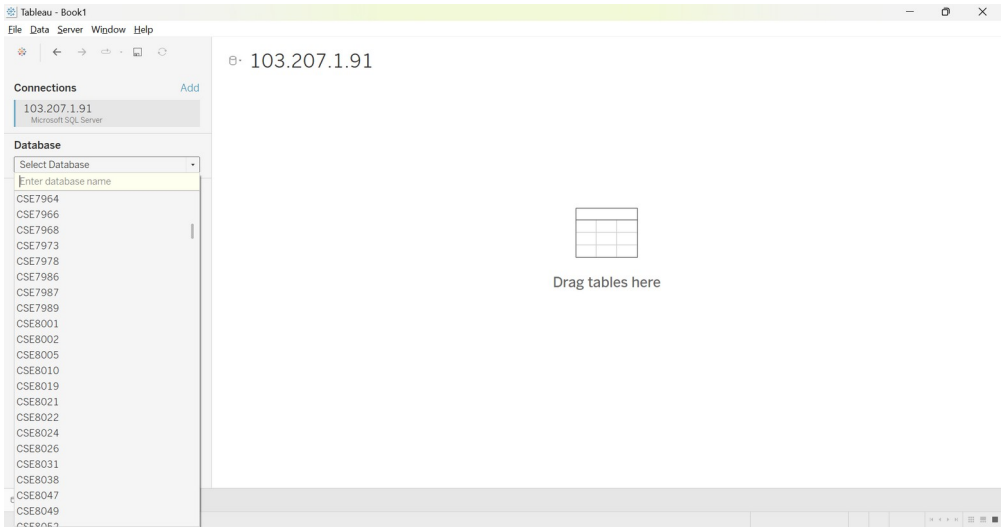
Aim: To perform Online Analytical processing using Tableau

Procedure:

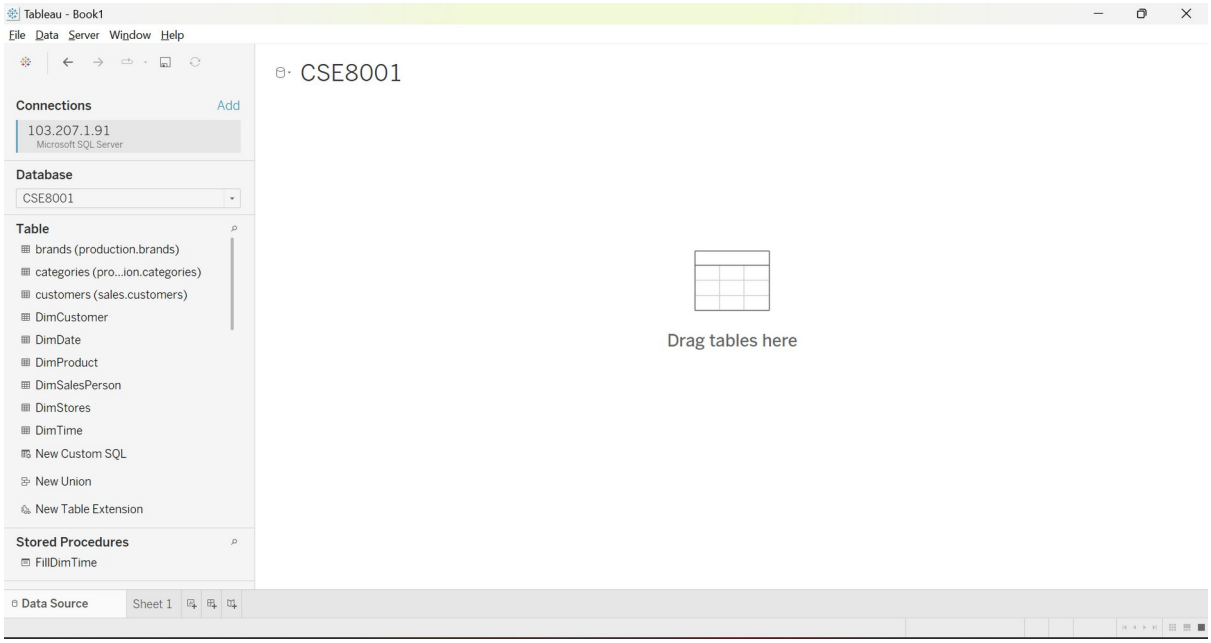
Step 1 : Connect to MSSQL Server or any server using connect option in the Tableau.



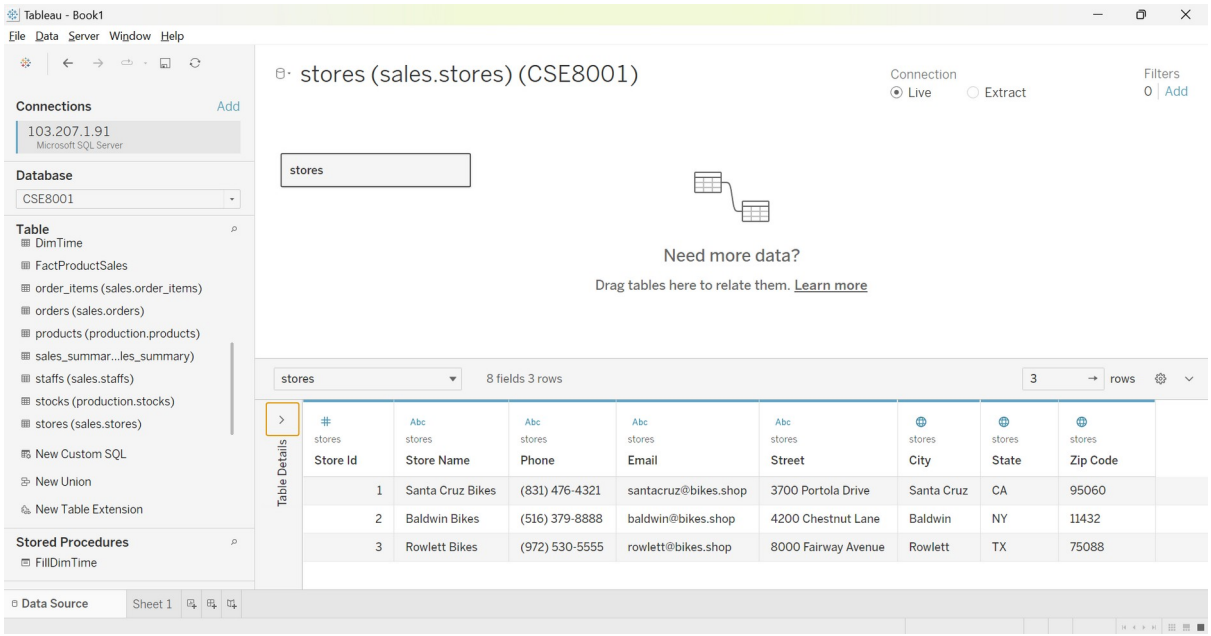
Step 2: Select the database from the list of database

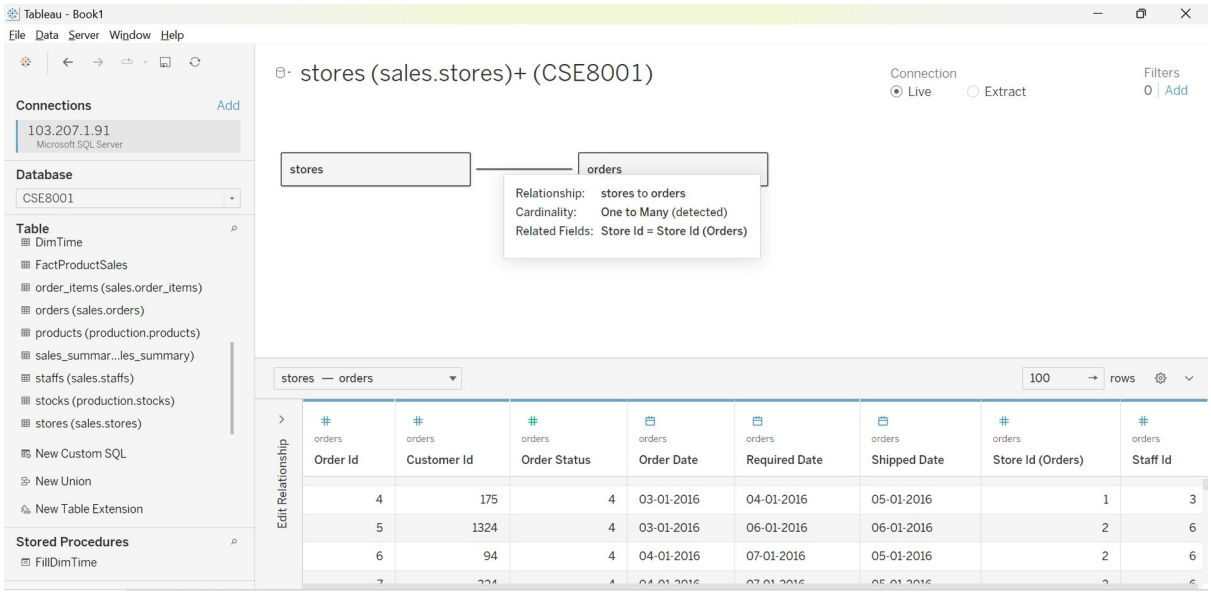


Step 3: Drag and drop the tables from the database for OLAP operations

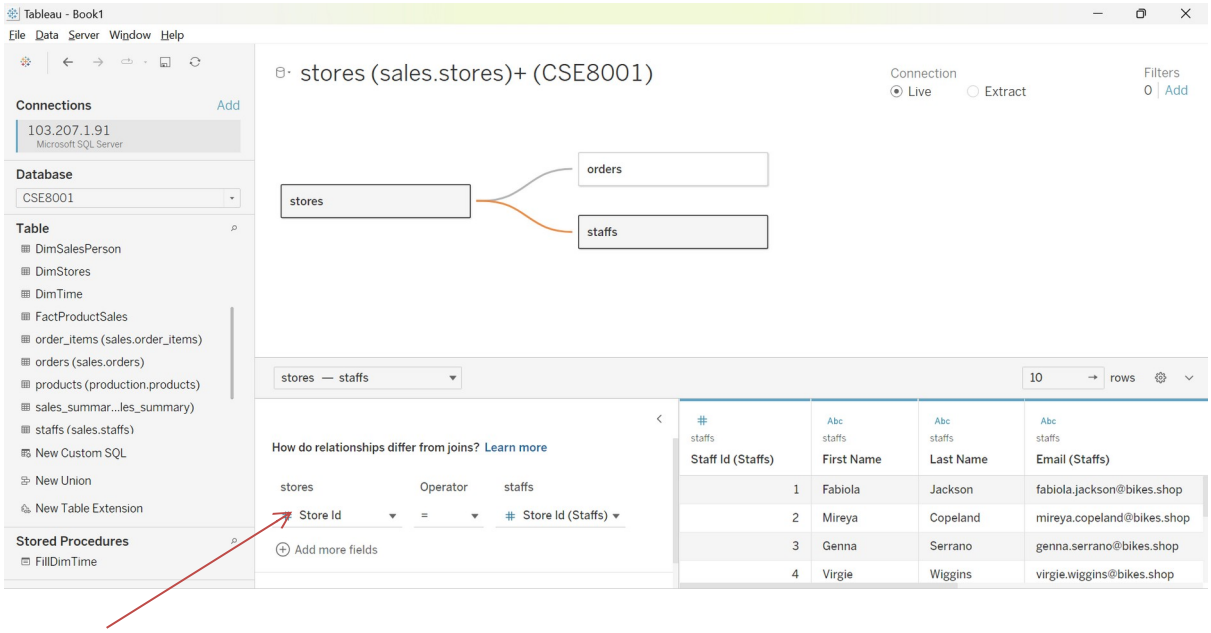


Step 4: In our case we are selecting Stores, Orders and Staff tables from Sales Schema.

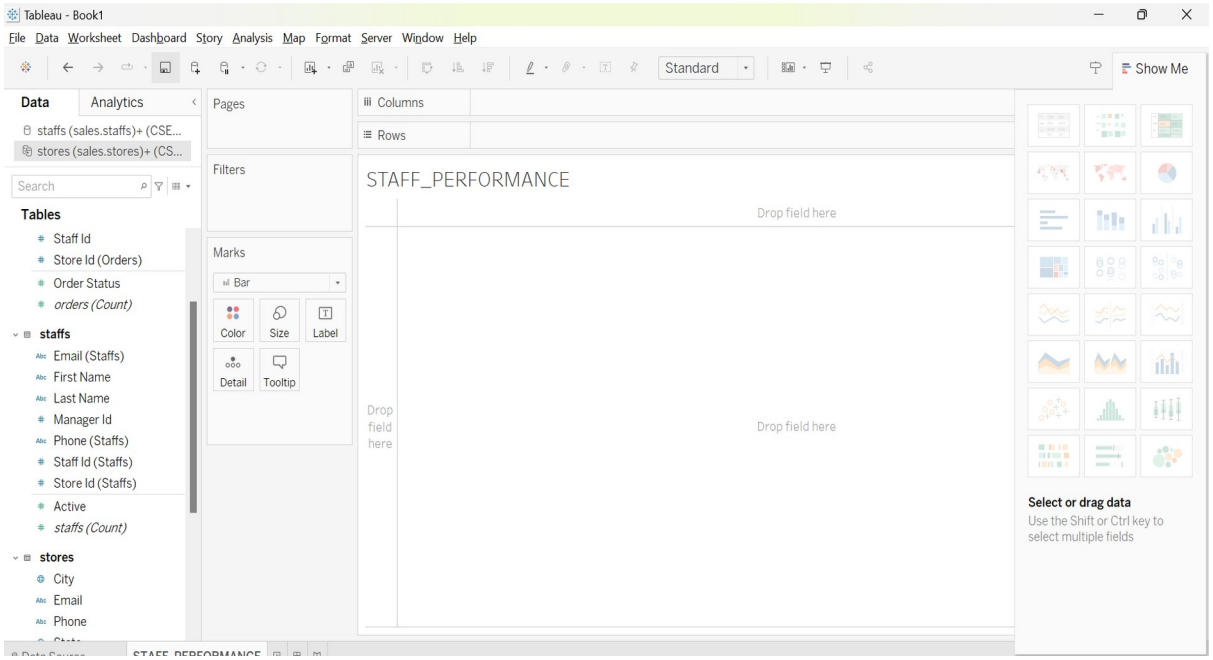




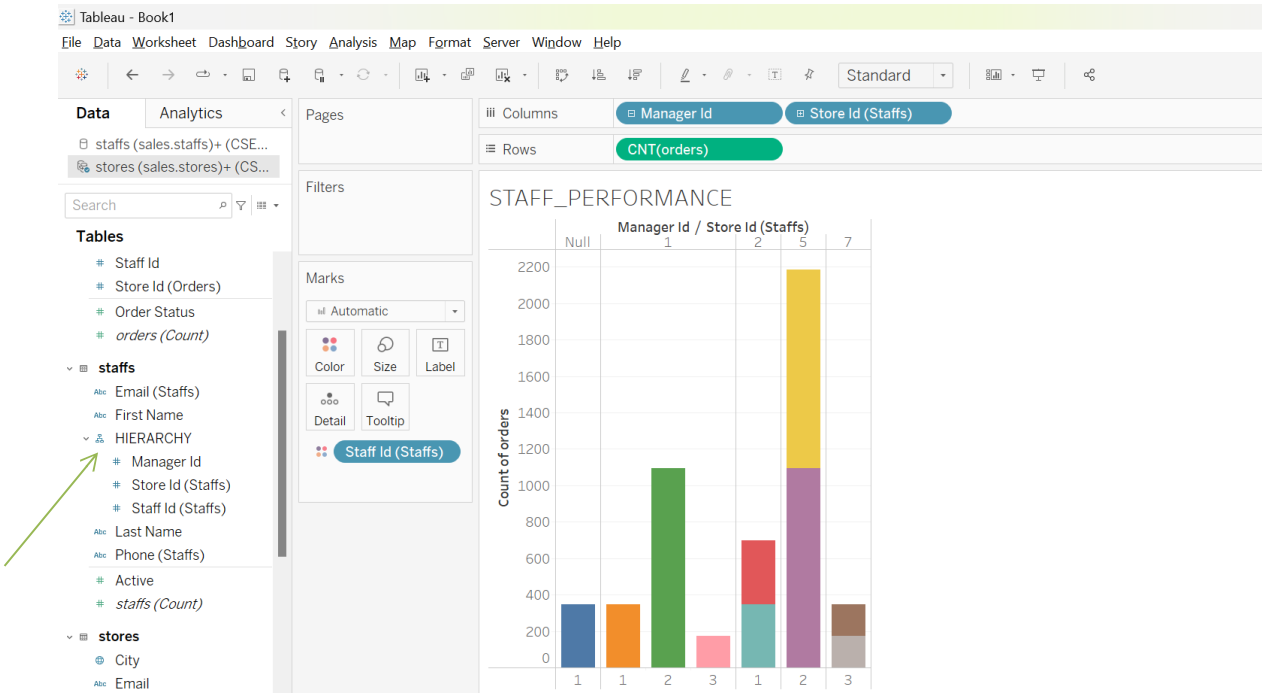
Step 5: Set the relationship between the tables.



Step 6: Dimensions and measures from the selected tables are displayed in Data tab

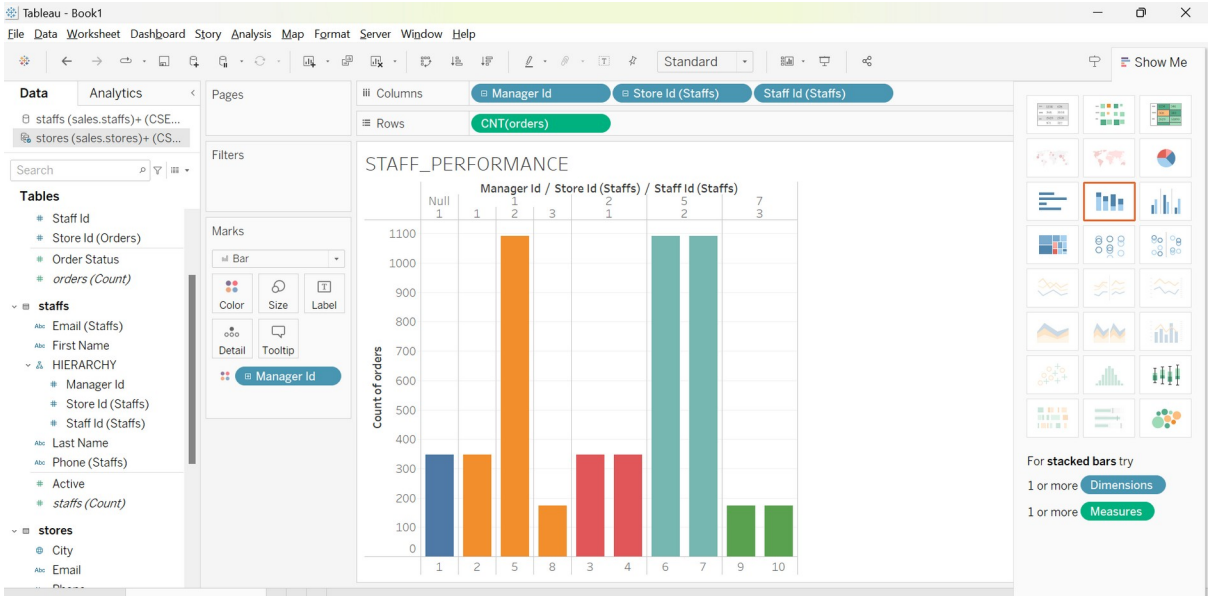


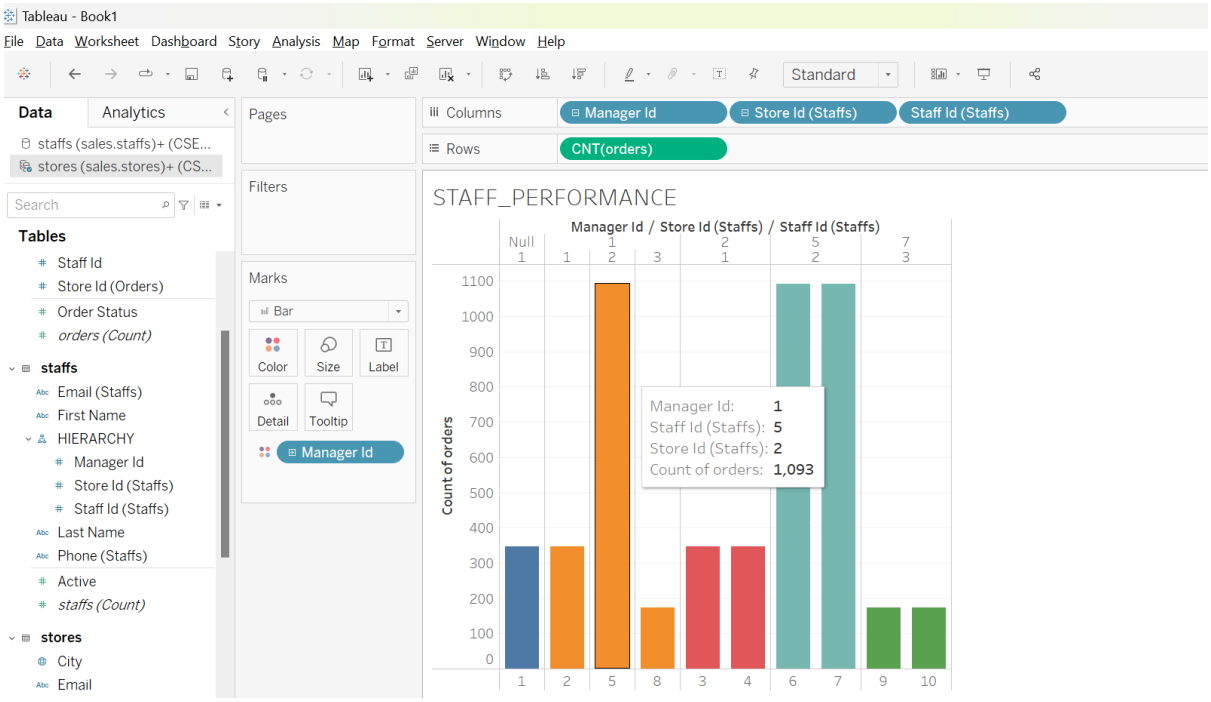
Step 7: Create a Roll up operation Manager Id-> Store Id-> Staff Id



Step 8: Drag and drop the column and rows to get the analysis

Bar Chart





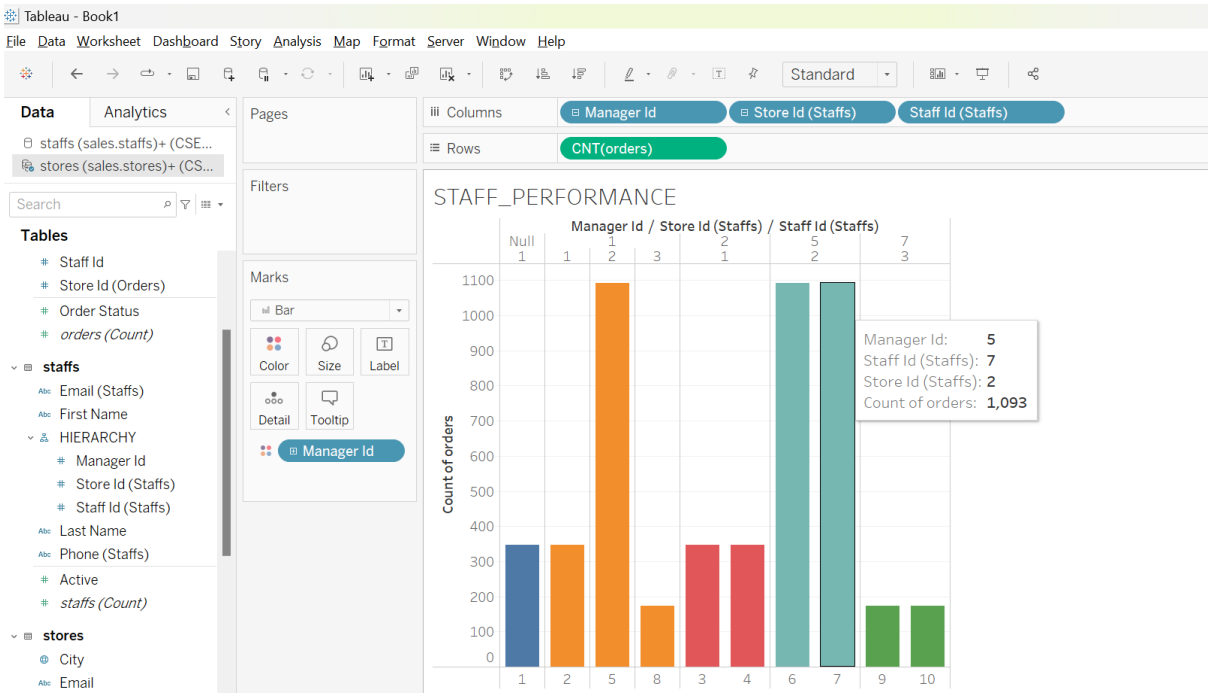


Tableau - Book1
File Data Worksheet Dashboard Story Analysis Map Format Server Window Help
Standard
Data
Analytics
staffs (sales.staffs)+ (CSE...
stores (sales.stores)+ (CS...
Search
Tables
Staff Id
Store Id (Orders)
Order Status
orders (Count)
staffs
Email (Staffs)
First Name
HIERARCHY
Manager Id
Store Id (Staffs)
Staff Id (Staffs)
Last Name
Phone (Staffs)
Active
staffs (Count)
stores
City
Email
Pages
Filters
Marks
Bar
Color
Size
Label
Detail
Tooltip
Manager Id
Columns
Manager Id
Store Id (Staffs)
Staff Id (Staffs)
Rows
CNT(orders)
STAFF_PERFORMANCE
Manager Id / Store Id (Staffs) / Staff Id (Staffs)
Count of orders
Manager Id: 5
Staff Id (Staffs): 7
Store Id (Staffs): 2
Count of orders: 1,093

Heap Map

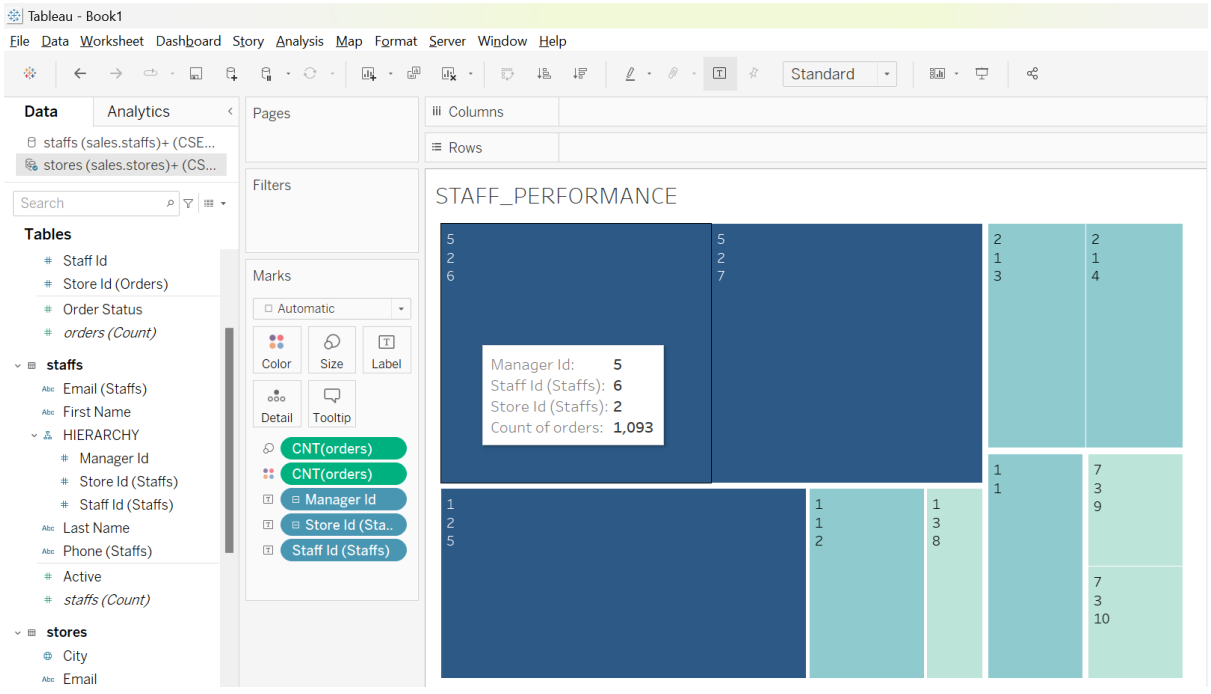


Tableau - Book1
File Data Worksheet Dashboard Story Analysis Map Format Server Window Help
Standard
Data
Analytics
staffs (sales.staffs)+ (CSE...
stores (sales.stores)+ (CS...
Search
Tables
Staff Id
Store Id (Orders)
Order Status
orders (Count)
staffs
Email (Staffs)
First Name
HIERARCHY
Manager Id
Store Id (Staffs)
Staff Id (Staffs)
Last Name
Phone (Staffs)
Active
staffs (Count)
stores
City
Email
Pages
Filters
Marks
Automatic
Color
Size
Label
Detail
Tooltip
CNT(orders)
CNT(orders)
Manager Id
Store Id (Sta..
Staff Id (Staffs)
Columns
Rows
STAFF_PERFORMANCE
Manager Id: 5
Staff Id (Staffs): 6
Store Id (Staffs): 2
Count of orders: 1,093

Bar chart

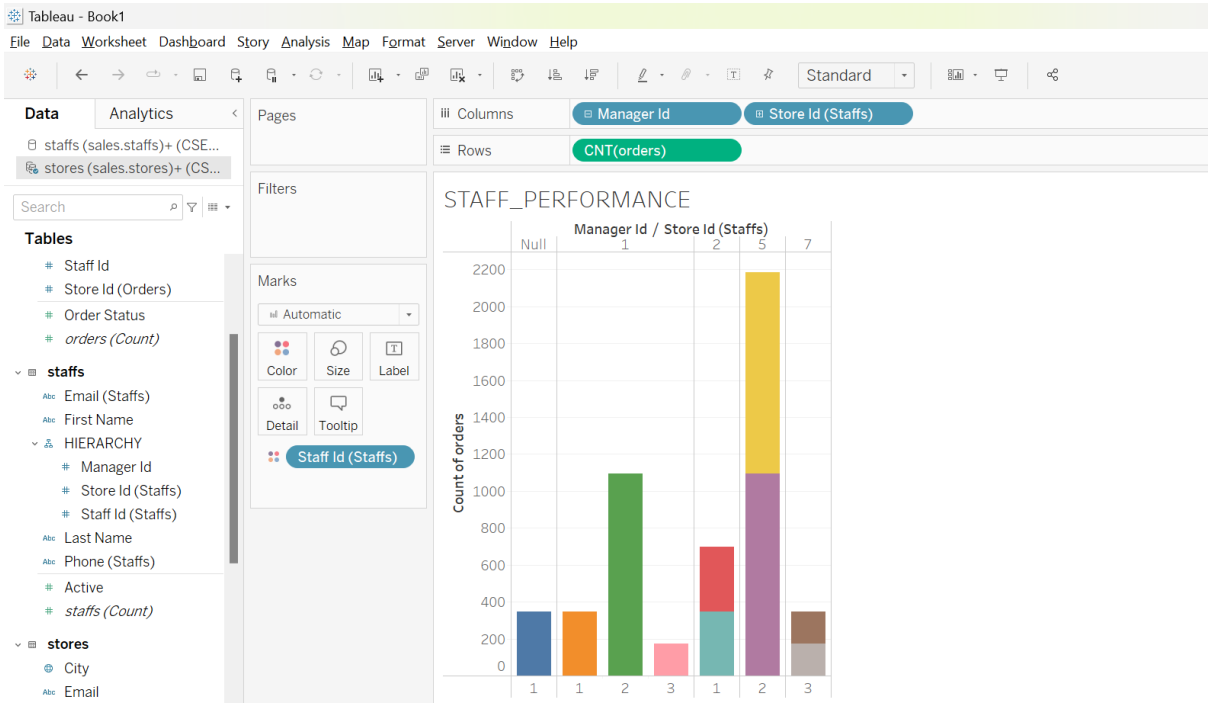


Tableau - Book1
File Data Worksheet Dashboard Story Analysis Map Format Server Window Help
Standard
Data
Analytics
staffs (sales.staffs)+ (CSE...
stores (sales.stores)+ (CS...
Search
Tables
Staff Id
Store Id (Orders)
Order Status
orders (Count)
staffs
Email (Staffs)
First Name
HIERARCHY
Manager Id
Store Id (Staffs)
Staff Id (Staffs)
Last Name
Phone (Staffs)
Active
staffs (Count)
stores
City
Email
Pages
Filters
Marks
Automatic
Color
Size
Label
Detail
Tooltip
Staff Id (Staffs)
Columns
Manager Id
Store Id (Staffs)
Rows
CNT(orders)
STAFF_PERFORMANCE
Manager Id / Store Id (Staffs)
Null 1 2 5 7
Count of orders
2200 2000 1800 1600 1400 1200 1000 800 600 400 200 0
1 1 2 3 1 2 3

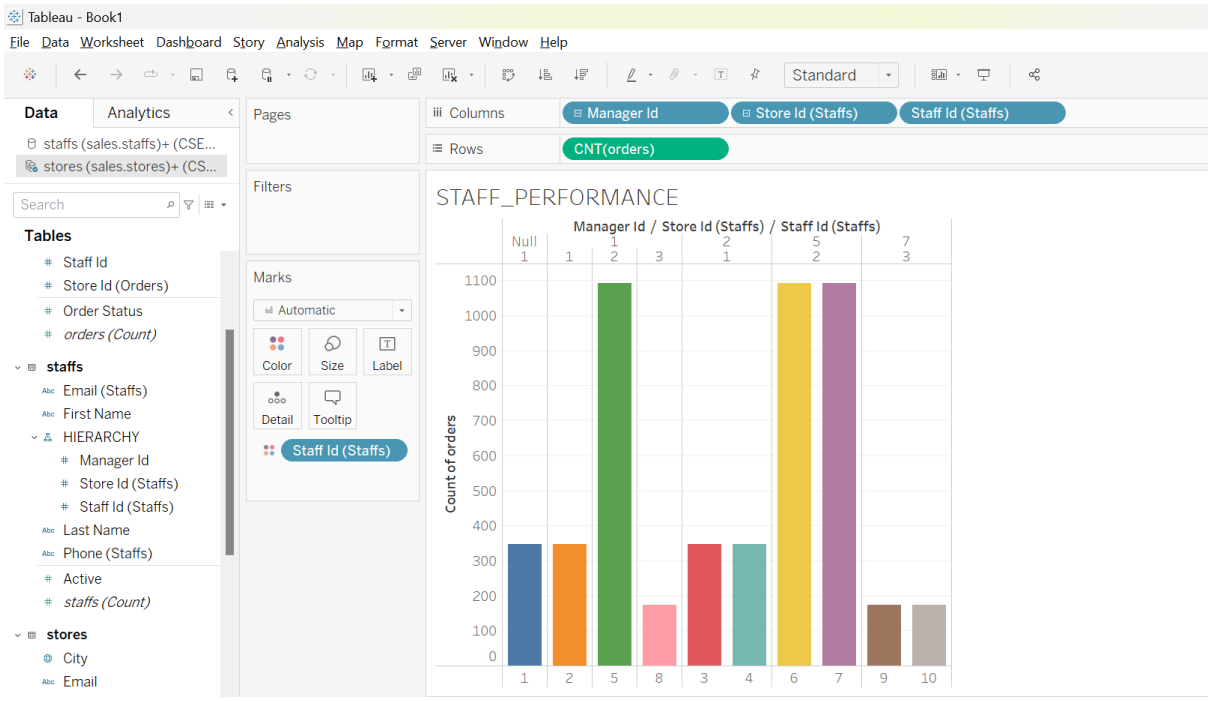
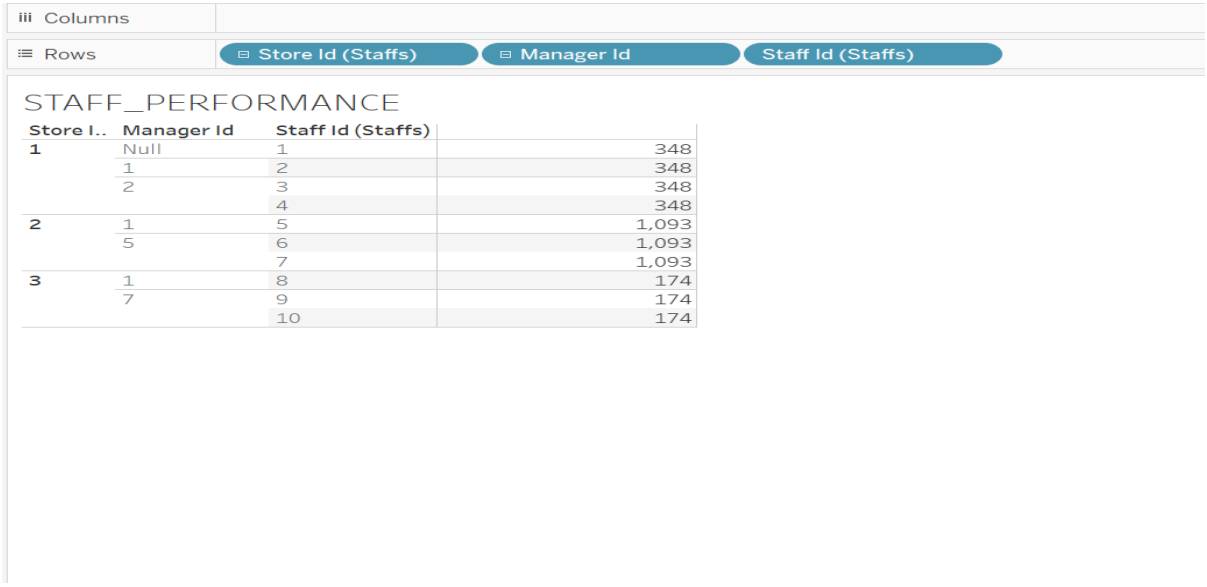
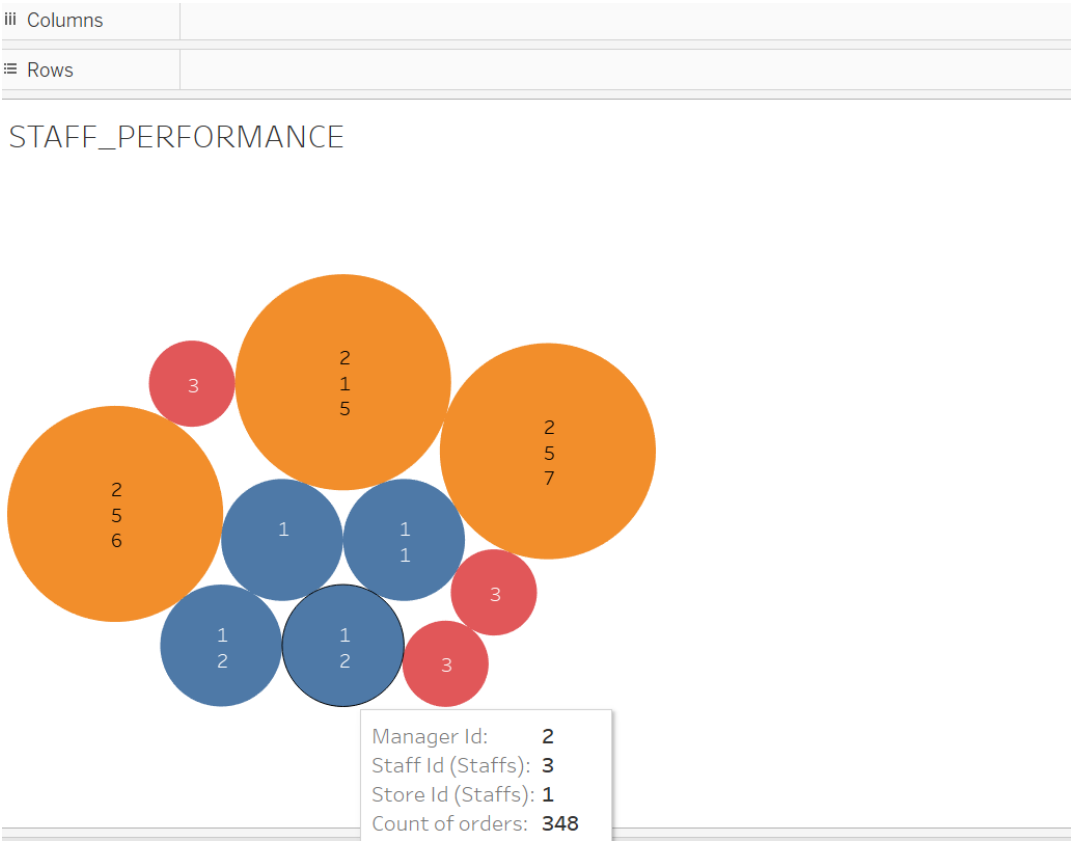


Tableau - Book1
File Data Worksheet Dashboard Story Analysis Map Format Server Window Help
Standard
Data
Analytics
staffs (sales.staffs)+ (CSE...
stores (sales.stores)+ (CS...
Search
Tables
Staff Id
Store Id (Orders)
Order Status
orders (Count)
staffs
Email (Staffs)
First Name
HIERARCHY
Manager Id
Store Id (Staffs)
Staff Id (Staffs)
Last Name
Phone (Staffs)
Active
staffs (Count)
stores
City
Email
Pages
Filters
Marks
Automatic
Color
Size
Label
Detail
Tooltip
Staff Id (Staffs)
Columns
Manager Id
Store Id (Staffs)
Staff Id (Staffs)
Rows
CNT(orders)
STAFF_PERFORMANCE
Manager Id / Store Id (Staffs) / Staff Id (Staffs)
Null 1 1 1 2 3 2 1 5 2 7 3
Count of orders
1100 1000 900 800 700 600 500 400 300 200 100 0
1 2 5 8 3 4 6 7 9 10

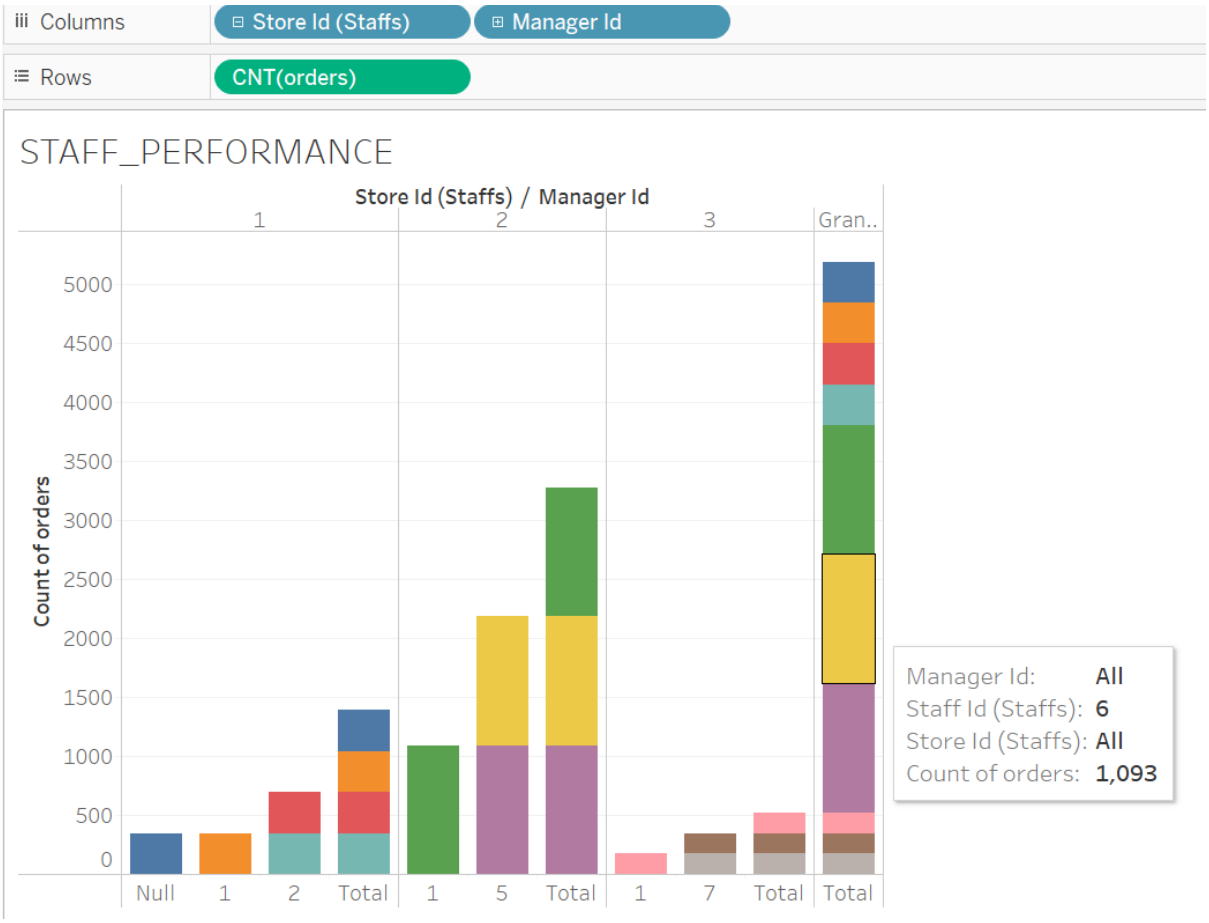
Table Format



Store I..	Manager Id	Staff Id (Staffs)	
1	Null	1	348
	1	2	348
	2	3	348
		4	348
2	1	5	1,093
	5	6	1,093
		7	1,093
3	1	8	174
	7	9	174
		10	174



Step 9: In analytics tab click total to show more discrete dimensions



EX.NO:8 **CASE STUDY USING OTLP**

Aim:

To learn about case study using OTLP.

Example:

An example of a case study using OTLP (Online Transactional Processing). Let's consider a fictional company called "EcoMart," an online grocery store that utilizes OTLP for managing its transactions.

Case Study: EcoMart's Online Transactional Processing System

Overview:

EcoMart is an online grocery store that sells organic and environmentally friendly products. The company operates solely through its website and mobile application, offering a wide range of products including fresh produce, pantry items, household goods, and personal care products.

Challenges:

EcoMart faces several challenges in managing its online transactions efficiently:

High Transaction Volume:

With a growing customer base, EcoMart experiences a high volume of transactions daily, including purchases, refunds, and inventory updates.

Real-time Inventory Management:

It's crucial for EcoMart to maintain accurate inventory levels to prevent stockouts and ensure timely order fulfillment.

Customer Satisfaction:

Customers expect seamless and fast transactions when shopping online. Any delays or errors in processing transactions can lead to dissatisfaction and loss of business.

Implementation:

Database Architecture:

EcoMart sets up a robust relational database management system (RDBMS) to store transactional data. The database schema includes tables for orders, products, customers, inventory, and transactions.

Transaction Processing:

When a customer places an order on the EcoMart website or mobile app, the OTLP system immediately processes the transaction. It verifies product availability, updates inventory levels, calculates the total order value, and generates an order confirmation for the customer.

Real-time Inventory Updates:

As transactions are processed, the OTLP system updates the inventory levels in real-time to reflect the changes. This ensures that customers can only purchase products that are currently in stock, reducing the risk of overselling.

Fault Tolerance:

To ensure reliability, EcoMart implements fault-tolerant mechanisms within its OTLP system. Redundant servers, data backups, and failover mechanisms are in place to minimize downtime and data loss in case of hardware failures or system crashes.

Performance Monitoring:

EcoMart continuously monitors the performance of its OTLP system to identify any bottlenecks or issues. Performance metrics such as transaction throughput, response times, and error rates are monitored regularly, and adjustments are made as needed to optimize performance.

Benefits:

Efficiency:

By processing transactions in real-time, EcoMart improves operational efficiency and reduces the risk of errors or delays.

Accurate Inventory Management:

The OTLP system ensures accurate inventory management, minimizing stockouts and improving overall customer satisfaction.

Scalability:

As EcoMart continues to grow, its OTLP system can easily scale to accommodate increasing transaction volumes without sacrificing performance or reliability.

Customer Satisfaction:

Fast and reliable transaction processing leads to enhanced customer satisfaction and loyalty, driving repeat business and positive word-of-mouth.

Conclusion: By implementing an OTLP system, EcoMart has overcome its challenges related to online transaction processing. The system enables the company to manage transactions efficiently, maintain accurate inventory levels, and deliver an exceptional shopping experience to its customers. As EcoMart continues to grow, its OTLP system will play a crucial role in supporting its ongoing success in the competitive online grocery market.

EX.NO: 09	IMPLEMENTATION OF DATA WAREHOUSING
DATE:	

AIM:

To explore data warehousing concepts and integrate with practical implementation.

ALGORITHM:

STEP 1: Start mysql.

STEP 2: Understand the organizational requirements and business objectives).

STEP 3: Design and create dimension tables for essential dimensions such as Time, Product, Customer, and Location using SQL .

STEP 4: Populate each dimension table with sample data, ensuring data integrity and consistency

STEP 5: Create a central fact table for capturing relevant metrics

STEP 6: Insert sample transactional data into the fact table, corresponding to the keys

STEP 7: Construct SQL queries to perform analysis

STEP 8: Validate the data model and optimize the schema

STEP 9: Stop.

PROGRAM:

```
CREATE TABLE ProductDimension (
ProductKey INT AUTO_INCREMENT,
ProductName VARCHAR(255),
ProductCategory VARCHAR(255),
Price DECIMAL(10,2),
PRIMARY KEY (ProductKey)
);

CREATE TABLE CustomerDimension (
CustomerKey INT AUTO_INCREMENT,
CustomerName VARCHAR(255),
ContactInfo VARCHAR(255),
Age INT,
Gender CHAR(1),
PRIMARY KEY (CustomerKey)
);

CREATE TABLE LocationDimension (
LocationKey INT AUTO_INCREMENT,
LocationName VARCHAR(255),
City VARCHAR(255),
Region VARCHAR(255),
PRIMARY KEY (LocationKey)
);
```

```
INSERT INTO TimeDimension (Date, Weekday, MonthName, Year) VALUES
('2024-01-10', 'Monday', 'January', 2024),
('2024-01-11', 'Tuesday', 'January', 2024),
('2024-01-12', 'Wednesday', 'January', 2024);


INSERT INTO ProductDimension (ProductName, ProductCategory, Price)
VALUES
('Laptop', 'Electronics', 1299.99),
('Smartphone', 'Electronics', 899.50),
('Chair', 'Furniture', 299.90);


INSERT INTO CustomerDimension (CustomerName, ContactInfo, Age, Gender)
VALUES
('John Doe', 'john.doe@example.com', 35, 'M'),
('Jane Smith', 'jane.smith@example.com', 28, 'F'),
('David Johnson', 'david.johnson@example.com', 42, 'M');


INSERT INTO LocationDimension (LocationName, City, Region) VALUES
('Headquarters', 'New York', 'East Coast'),
('Branch Office', 'Los Angeles', 'West Coast'),
('Retail Store', 'Chicago', 'Midwest');


INSERT INTO InventoryFact (TimeKey, ProductKey, CustomerKey,
LocationKey, QuantitySold, SaleAmount) VALUES
(1, 1, 1, 1, 2, 2599.98),
(2, 2, 2, 2, 1, 899.50),
(3, 3, 3, 3, 3, 899.70);
```

RESULT:

VIVA QUESTION

1. What is Datawarehousing?

A Datawarehouse is the repository of a data and it is used for Management decision support system. Datawarehouse consists of wide variety of data that has high level of business conditions at a single point in time.

In single sentence, it is repository of integrated information which can be available for queries and analysis.

2. What is Business Intelligence?

Business Intelligence is also known as DSS – Decision support system which refers to the technologies, application and practices for the collection, integration and analysis of the business related information or data. Even, it helps to see the data on the information itself.

3. What is Dimension Table?

Dimension table is a table which contain attributes of measurements stored in fact tables. This table consists of hierarchies, categories and logic that can be used to traverse in nodes.

4. What is Fact Table?

Fact table contains the measurement of business processes, and it contains foreign keys for the dimension tables.

Example – If the business process is manufacturing of bricks

Average number of bricks produced by one person/machine – measure of the business process

5. What is Data Mining?

Data Mining is set to be a process of analyzing the data in different dimensions or perspectives and summarizing into a useful information. Can be queried and retrieved the data from database in their own format.

6. What is OLTP?

OLTP is abbreviated as On-Line Transaction Processing, and it is an application that modifies the data whenever it received and has large number of simultaneous users.

7. What is OLAP?

OLAP is abbreviated as Online Analytical Processing, and it is set to be a system which collects, manages, processes multi-dimensional data for analysis and management purposes.

8. What is the difference between OLTP and OLAP?

Following are the differences between OLTP and OLAP:

OLTP	OLAP
Data is from original data source Simple queries by users Normalized small database Fundamental	Data is from various data sources Complex queries by system De-normalized Large Database

9. What is ODS?

ODS is abbreviated as Operational Data Store and it is a repository of real time operational data rather than long term trend data.

10. What is the difference between View and Materialized View?

A view is nothing but a virtual table which takes the output of the query and it can be used in place of tables. A materialized view is nothing but an indirect access to the table data by storing the results of a query in a separate schema.

11.What is ETL?

ETL is abbreviated as Extract, Transform and Load. ETL is a software which is used to reads the data from the specified data source and extracts a desired subset of data. Next, it transform the data using rules and lookup tables and convert it to a desired state.

Then, load function is used to load the resulting data to the target database.

12.What is VLDB?

VLDB is abbreviated as Very Large Database and its size is set to be more than one terabyte database. These are decision support systems which is used to server large number of users.

13.What is real-time datawarehousing?

Real-time datawarehousing captures the business data whenever it occurs. When there is business activity gets completed, that data will be available in the flow and become available for use instantly.

14.What are Aggregate tables?

Aggregate tables are the tables which contain the existing warehouse data which has been grouped to certain level of dimensions. It is easy to retrieve data from the aggregated tables than the original table which has more number of records.

This table reduces the load in the database server and increases the performance of the query.

15.What is factless fact tables?

A factless fact tables are the fact table which doesn't contain numeric fact column in the fact table.

16.How can we load the time dimension?

Time dimensions are usually loaded through all possible dates in a year and it can be done through a program. Here, 100 years can be represented with one row per day.

17.What are Non-additive facts?

Non-Addictive facts are said to be facts that cannot be summed up for any of the dimensions present in the fact table. If there are changes in the dimensions, same facts can be useful.

18.What is conformed fact?

Conformed fact is a table which can be used across multiple data marts in combined with the multiple fact tables.

19.What is Datamart?

A Datamart is a specialized version of Datawarehousing and it contains a snapshot of operational data that helps the business people to decide with the analysis of past trends and experiences. A data mart helps to emphasizes on easy access to relevant information.

20.What is Active Datawarehousing?

An active datawarehouse is a datawarehouse that enables decision makers within a company or organization to manage customer relationships effectively and efficiently.

21.What is the difference between Datawarehouse and OLAP?

Datawarehouse is a place where the whole data is stored for analyzing, but OLAP is used for analyzing the data, managing aggregations, information partitioning into minor level information.

22.What is ER Diagram?

ER diagram is abbreviated as Entity-Relationship diagram which illustrates the interrelationships between the entities in the database. This diagram shows the structure of each tables and the links between the tables.

23.What are the key columns in Fact and dimension tables?

Foreign keys of dimension tables are primary keys of entity tables. Foreign keys of fact tables are the primary keys of the dimension tables. Conformed fact is a table which can be used across multiple data marts in combined with the multiple fact tables.

24.What are the key columns in Fact and dimension tables?

Foreign keys of dimension tables are primary keys of entity tables. Foreign keys of fact tables are the primary keys of the dimension tables.

25.What is SCD?

SCD is defined as slowly changing dimensions, and it applies to the cases where record changes over time.

26.What are the types of SCD?

There are three types of SCD and they are as follows:

SCD 1 – The new record replaces the original record

SCD 2 – A new record is added to the existing customer dimension table

SCD 3 – A original data is modified to include new data

27.What is BUS Schema?

BUS schema consists of suite of confirmed dimension and standardized definition if there is a fact tables.

28.What is Star Schema?

Star schema is nothing but a type of organizing the tables in such a way that result can be retrieved from the database quickly in the data warehouse environment.

29.What is Snowflake Schema?

Snowflake schema which has primary dimension table to which one or more dimensions can be joined. The primary dimension table is the only table that can be joined with the fact table.

30.What is a core dimension?

Core dimension is nothing but a Dimension table which is used as dedicated for single fact table or datamart.