Thiagarajar College Of Engineering, Madurai.

Department of Computer Applications

**Laboratory Manual**

For

**20CA390 - Data Warehousing and Data Mining Lab**

HDCA Prepared by,

Dr.S.Vijayalakshmi

**Table of Contents**

**EXPERIMENT-1**

To perform various commands given in PL/SQL in Oracle (For brushing up.)

**EXPERIMENT-2**

To perform multi-dimensional data model using SQL queries. E.g. Star, snowflake and Fact constellation schemas

**EXPERIMENT-3**

To perform various OLAP operations such slice, dice, roll up, drill up, pivot etc.

**EXPERIMENT-4**

To perform Preprocessing in data mining using Weka mining tool.

**EXPERIMENT-5**

To perform Clustering using Weka mining tool

**EXPERIMENT-6**

To perform Association rule analysis in Weka tool.

**EXPERIMENT-7**

To Develop Data Analytics Project using

Weka tool.

**EXPERIMENT-1**

**Aim**: To perform various commands given in PL/SQL in Oracle 8.0(For brushing up.)

**Tools/ Apparatus**: PL/SQL programming tool.

**Procedure**:

1) Open SQL 8.0 tool and it will display the login window.

2) Enter the username = ‘SCOTT’ and password = ‘TIGER’

3) After the successfully login try to write down the SQL queries in correct syntax

and run them successfully.

**EXPERIMENT-2**

**Aim**: To perform multi-dimensional data model using SQL queries. E.g. Star, snowflake and Fact constellation schemas.

**Tools/ Apparatus**: Oracle 8i/9i/10G

**Procedure**:

1) Open Oracle 8i/9i/10G tool and it will display the login window where one has to enter the login details.

2) Create the fact and the required dimensions tables as per the given business problem. There are three basic types of the multidimensional data model. They are Star, snowflake and Fact constellation schemas

3) As per the guidelines given in the theory draw all the three dimensional models.

Use the following queries : *Define cube sales\_star[time, item, branch, location];*

*Dollars\_sold = sum(sales\_in\_dollars), units\_sold = count(\*).*

*Define dimension time as(time\_key,day,day\_of\_week, month, quarter, year).*

Write the same queries for all other dimensions resp. Then run the following query:

Select s.time\_key,s.item\_key, s..branch\_key, s.location\_key,

Sum(s.number\_of\_units\_sold\*s.price), sum(s.number\_of\_units\_sold)

From time t,item I, branch b, location l, sales s,

Where s.time\_key = t.time\_key and s.item\_key = i.item\_key and s.branch\_key =

b.branch\_key and s.location\_key = l.location\_key

Group by s.time\_key, s.item\_key, s.branch\_key, s.location\_key.

Run the queries to create the various multi-dimensional models.

**EXPERIMENT-3**

**Aim**: To perform various OLAP operations such slice, dice, roll up, drill up, pivot etc.

**Tools/ Apparatus**: Oracle 8i/9i/10G, PL/SQL.

**Procedure**:

1) Open SQL tool and login successfully.

2) Write down the queries to perform slice. In which one should keep one of the

dimensions as constant and other dimensions should range from min to max.

3) Write down the queries to perform the dice. In which one has to keep two of the

dimensions constant.

4) Write down the queries to perform roll-up by keeping one dimension constant and others should range from min to max. It is more like a generalization.

5) Write down the queries to perform roll-up by keeping one dimension constant and others should range from min to max. It is more like a specialization.

**EXPERIMENT-4**

**Aim**: To perform the Data Processing for data mining using WEKA Tool

**Procedure**:

Introduce the datasets *sick*, *vote*, *mushroom* and *letter*.

**Apply discretization:**

* load the *sick* dataset and look at the attributes
* classify using NB, evaluating with cross-validation
* apply the supervised discretization filter and look at the effect (in the Preprocess panel)
* apply unsupervised discretization with different numbers of bins and look at the effect
* use the FilteredClassifier with NB and supervised discretization, evaluating with cross-validation
* repeat using unsupervised discretization with different numbers of bins
* compare and interpret the results.

**Apply feature selection using CfsSubsetEval**

* load the *mushroom* dataset and apply J48, IBk and NB, evaluating with crossvalidation
* select attributes using CfsSubsetEval and GreedyStepwise search
* interpret the results
* use AttributeSelectedClassifier (with CfsSubsetEval and GreedyStepwise
* search) for classifiers J48, IBk and NB, evaluating with cross-validation
* interpret the results.

**Apply feature selection using WrapperSubsetEval:**

* load the *vote* dataset and apply J48, IBk and NB, evaluating with crossvalidation
* select attributes using WrapperSubsetEval with InfoGainAttributeEval and

RankSearch, with the J48 classifier

* interpret the results
* use AttributeSelectedClassifier (with WrapperSubsetEval, InfoGainAttributeEval and RankSearch) with classifiers J48, IBk and NB, evaluating with cross-validation
* interpret the results.

**Sampling a dataset:**

* load the *letter* dataset and examine a particular (numeric) attribute
* apply the Resample filter to select half the dataset
* examine the same attribute and comment on the results.

**EXPERIMENT-5**

**Aim**: To perform Clustering using WEKA Tool

**Procedure**:

1. Go to the WEKA Explorer environment and load the training file “iris.arff”. Remove the class attribute using the preprocessing dialog. Go to the clustering dialog.

* Cluster the iris dataset using the k-Means Clustering algorithm with k=5. Hand in the result given by WEKA (Cluster mean and standard deviation).
* Cluster the iris dataset using the k-Means Clustering algorithm with k=3 and k=4 and with ten different values of the seed parameter, using the option: Classes to cluster evaluation and and store the results on an excel file. Compute the mean of the two different k values for the k-Means
* Visualize the cluster mean values and standard deviation for - sepallength versus sepalwidth – petallength versus petalwidth. Don’t erase the result of the k-Means algorithm.

2. Using the EM algorithm with the iris dataset, try to run the EM algorithm, with the automatic estimation of the number of clusters. Then, try to repeat the experiments of the previous section with the EM algorithm. Compare the results.

3. Create an “arff”-file containing the data points as given below.

t1 = (4, 2, 3, 5, 2, 2, 2, 1)

t2 = (3, 2, 5, 4, 3, 2, 1, 4)

t3 = (1, 3, 3, 5, 2, 3, 2, 1)

t4 = (4, 2, 0, 5, 2, 2, 2, 1)

t5 = (3, 2, 3, 4, 3, 2, 1, 4)

t6 = (2, 5, 3, 5, 2, 2, 2, 1)

t7 = (4, 1, 3, 7, 2, 1, 2, 1)

t8 = (3, 1, 5, 4, 3, 2, 1, 4)

t9 = (2, 5, 2, 5, 2, 5, 2, 1)

Cluster the above data file using K-means and EM with k=2 and k=3 clusters.

4. Using Cobweb algorithm, run the algorithm on the iris dataset. First you must use the option: Using training set and then using the option: **Classes to cluster evaluation**. For both the options visualize the tree obtained. What is the difference? How is the behaviour of the algorithm? In your opinion, what is the factor that influences the algorithm?

5. Download the dataset http://www.upo.es/eps/bigs/dataSet/Lymphoma96x4026+9classes.arff.

k-means and EM (with Classes to cluster evaluation option) change the number of cluster from

10 to 20, step equal to 1, and store the Incorrectly clustered instances for both the algorithms.

**EXPERIMENT-6**

**Aim**: To perform Association Rule Analysis using WEKA Tool

**Procedure**:

1. Implement the Apriori algorithm. Your implementation should allow the user to specify a minimum support threshold (*minsup*), a minimum confidence threshold (*minconf*), and a maximum number of rules to display at a time (*maxrules*).
2. Use your algorithm on the Binarized Lenses problem.
   * Run Apriori for 0.1 <= *minsup* <= 0.8 and 0.1 <= *minconf* <= 0.6, using increments of 0.1 (i.e., this means you should run the algorithm 48 times).
   * Summarize your findings.
   * Given this additional information about the data, what do your (discovered) associations tell us?
3. Use your algorithm on the Mirror Symmetry problem.
   * Run Apriori for various combinations of *minsup* and *minconf* values.
   * Summarize your findings.
   * This is an artificial problem. Each attribute represents a bit position in a string of 30 bits: *Lmost, Lmost1, ..., Lmost14, Rmost14, Rmost13, ..., Rmost1, Rmost* and the attribute *Symm* is 1 if the pattern is symmetric about its center, and 0 otherwise. Given this interpretation, do any of the rules discovered by your Apriori algorithm make sense?
4. Build your own association task.
   * Design your task so that it contains some simple associations you can check your algorithm against. List these associations.
   * Run Apriori for various combinations of *minsup* and *minconf* values.
   * Verify that the associations you designed into the task are discovered by your algorithm.
   * Are there any surprises?