CCS341 -DATA WAREHOUSING- LAB MANNUAL

EXPERIMENT NO:1.a.

Aim: Create a Weather Table with training data set which includes attributes like outlook, temperature, humidity, windy, play.

PROCEDURE:

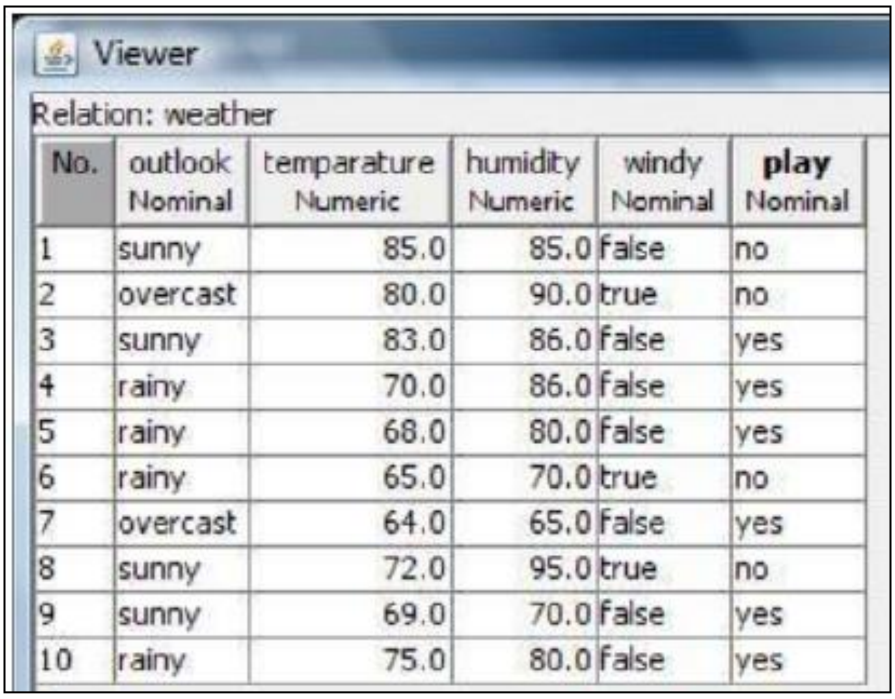
Steps:

1) Open Start → Programs → Accessories → Notepad
2) Type the following training data set with the help of Notepad for Weather Table.

```
@relation weather
        @attribute outlook {sunny,rainy,overcast}
        @attribute temparature numeric
        @attribute humidity numeric
        @attribute windy {true,false}
        @attribute play {yes,no}
@data
        sunny,85.0,85.0,false,no
        overcast,80.0,90.0,true,no
        sunny,83.0,86.0,false,yes
        rainy,70.0,86.0,false,yes
        rainy,68.0,80.0,false,yes
        rainy,65.0,70.0,true,no
        overcast,64.0,65.0,false,yes
        sunny,72.0,95.0,true,no
        sunny,69.0,70.0,false,yes
        rainy,75.0,80.0,false,yes
```

3) After that the file is saved with .arff file format.
4) Minimize the arff file and then open Start → Programs → weka-3-4.
5) Click on weka-3-4, then Weka dialog box is displayed on the screen.
6) In that dialog box there are four modes, click on explorer.
7) Explorer shows many options. In that click on 'open file' and select the arff file
8) Click on edit button which shows weather table on weka

TRAINING DATASET – WEATHER TABLE



Relation: weather

No.	outlook Nominal	temparature Numeric	humidity Numeric	windy Nominal	play Nominal
1	sunny	85.0	85.0	false	no
2	overcast	80.0	90.0	true	no
3	sunny	83.0	86.0	false	yes
4	rainy	70.0	86.0	false	yes
5	rainy	68.0	80.0	false	yes
6	rainy	65.0	70.0	true	no
7	overcast	64.0	65.0	false	yes
8	sunny	72.0	95.0	true	no
9	sunny	69.0	70.0	false	yes
10	rainy	75.0	80.0	false	yes

RESULT: This program has been successfully executed.

EXPERIMENT NO:1.b.

AIM: Apply Pre-Processing techniques to the training data set of Weather Table

PROCEDURE :

There are 3 pre-processing techniques they are:
1) Add 2) Remove 3) Normalization Creation of Weather Table

Add → Pre-Processing Technique:

1) Start → Programs → Weka-3-4 → Weka-3-4
2) Click on explorer.
3) Click on open file.
4) Select Weather.arff file and click on open.
5) Click on Choose button and select the Filters option.
6) In Filters, we have Supervised and Unsupervised data.
7) Click on Unsupervised data.
8) Select the attribute Add.
9) A new window is opened.
10) In that we enter attribute index, type, data format, nominal label values for Climate.
11) Click on OK.
12) Press the Apply button, then a new attribute is added to the Weather Table.
13) Save the file.
14) Click on the Edit button, it shows a new Weather Table on Weka.

Remove → Pre-Processing Technique:

1) Start → Programs → Weka-3-4 → Weka-3-4
2) Click on explorer.
3) Click on open file.
4) Select Weather.arff file and click on open.
5) Click on Choose button and select the Filters option.
6) In Filters, we have Supervised and Unsupervised data.
7) Click on Unsupervised data.
8) Select the attribute Remove.
9) Select the attributes windy, play to Remove.
10) Click Remove button and then Save.
11) Click on the Edit button, it shows a new Weather Table on Weka

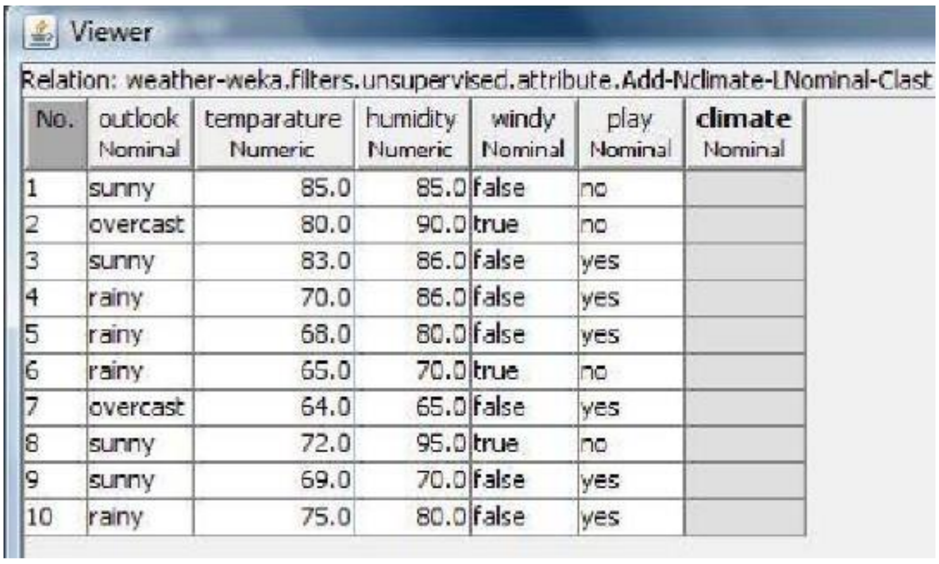
Normalize → Pre-Processing Technique:

1) Start → Programs → Weka-3-4 → Weka-3-4
2) Click on explorer.
3) Click on open file.
4) Select Weather.arff file and click on open.
5) Click on Choose button and select the Filters option.
6) In Filters, we have Supervised and Unsupervised data.
7) Click on Unsupervised data.
8) Select the attribute Normalize.
9) Select the attributes temperature, humidity to Normalize.
10) Click on Apply button and then Save.
11) Click on the Edit button, it shows a new Weather Table with normalized values on Weka.

RESULT: This program has been successfully executed.

Ex.1.b.

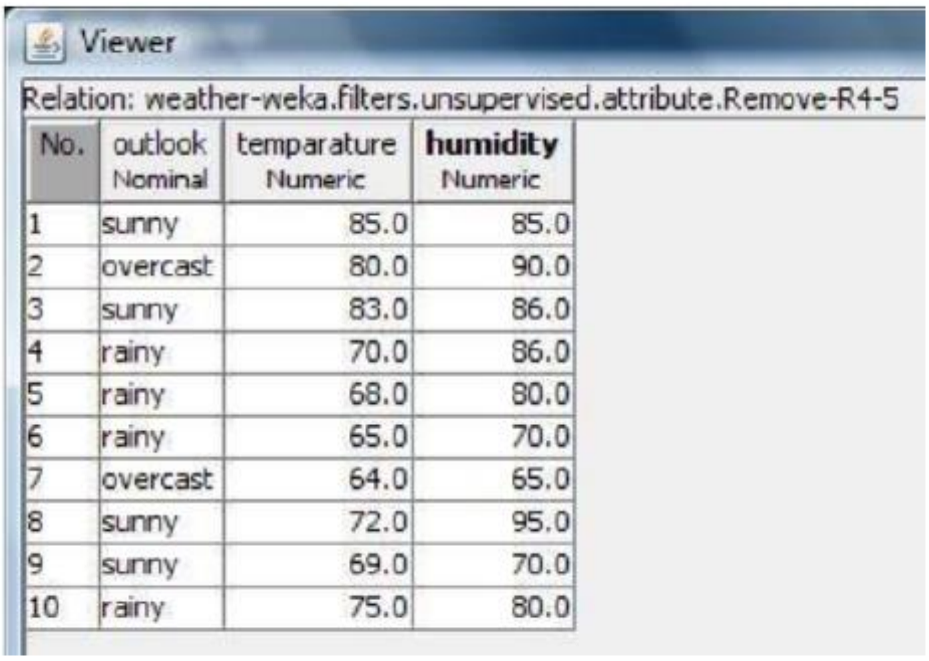
Weather Table after adding new attribute CLIMATE:



Relation: weather-weka.filters.unsupervised.attribute.Add-Nclimate-LNominal-Clast

No.	outlook Nominal	temparature Numeric	humidity Numeric	windy Nominal	play Nominal	climate Nominal
1	sunny	85.0	85.0	false	no	
2	overcast	80.0	90.0	true	no	
3	sunny	83.0	86.0	false	yes	
4	rainy	70.0	86.0	false	yes	
5	rainy	68.0	80.0	false	yes	
6	rainy	65.0	70.0	true	no	
7	overcast	64.0	65.0	false	yes	
8	sunny	72.0	95.0	true	no	
9	sunny	69.0	70.0	false	yes	
10	rainy	75.0	80.0	false	yes	

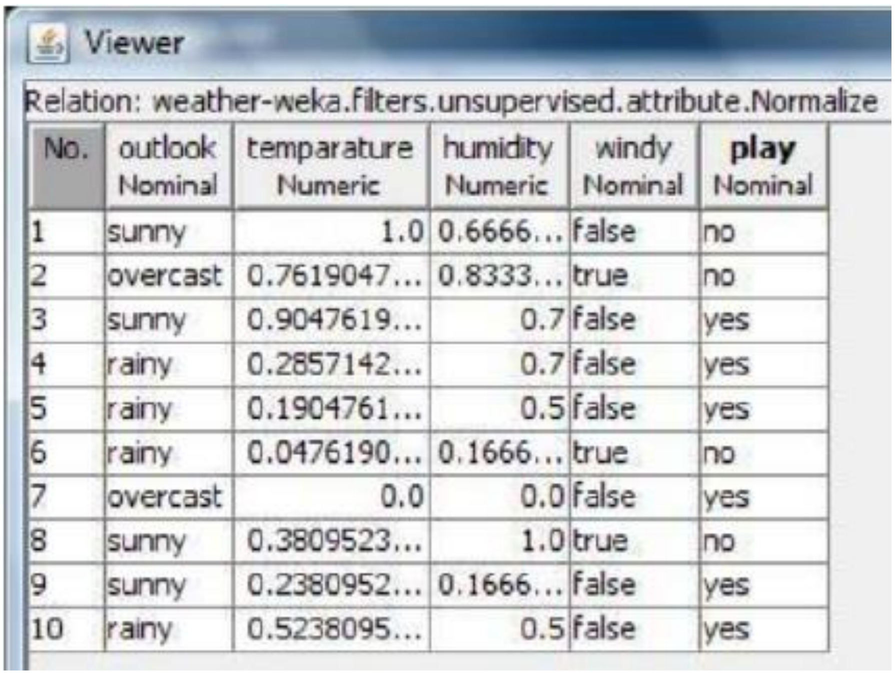
Weather Table removing attributes WINDY, PLAY:



Relation: weather-weka.filters.unsupervised.attribute.Remove-R4-5

No.	outlook Nominal	temparature Numeric	humidity Numeric
1	sunny	85.0	85.0
2	overcast	80.0	90.0
3	sunny	83.0	86.0
4	rainy	70.0	86.0
5	rainy	68.0	80.0
6	rainy	65.0	70.0
7	overcast	64.0	65.0
8	sunny	72.0	95.0
9	sunny	69.0	70.0
10	rainy	75.0	80.0

Normalize →Pre-Processing Technique:



Relation: weather-weka.filters.unsupervised.attribute.Normalize

No.	outlook Nominal	temparature Numeric	humidity Numeric	windy Nominal	play Nominal
1	sunny	1.0	0.6666...	false	no
2	overcast	0.7619047...	0.8333...	true	no
3	sunny	0.9047619...	0.7	false	yes
4	rainy	0.2857142...	0.7	false	yes
5	rainy	0.1904761...	0.5	false	yes
6	rainy	0.0476190...	0.1666...	true	no
7	overcast	0.0	0.0	false	yes
8	sunny	0.3809523...	1.0	true	no
9	sunny	0.2380952...	0.1666...	false	yes
10	rainy	0.5238095...	0.5	false	yes

EX. No: 2 Apply weka tool for data validation

AIM: To apply the concept of Linear Regression for evaluates the given dataset.

LINEAR REGRESSION: In statistics, Linear Regression is an approach for modeling a relationship between a scalar dependent variable Y and one or more explanatory variables denoted X.the case of explanatory variable is called Simple Linear Regression. Coefficient of Linear Regression is given by: Y=ax+b

PROBLEM: Consider the dataset below where x is the number of working expeince of a college graduate and y is the corresponding salary of the graduate. Build a regression equation and predict the salary of college graduate whose experience is 10 years

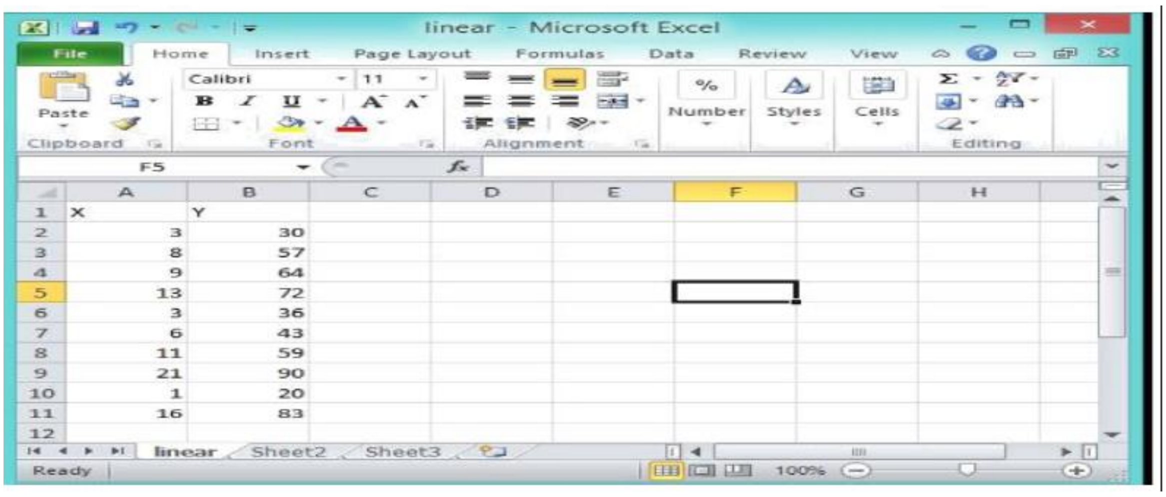
PROCEDURE:

STEPS:

1. Open the weka tool.
2. Download a dataset by using UCI.
3. Apply replace missing values.
4. Apply normalize filter.
5. Click the Classify Tab.
6. Choose the Simple Linear Regression option.
7. Select the training set of data.
8. Start the validation process.
9. Note the output.

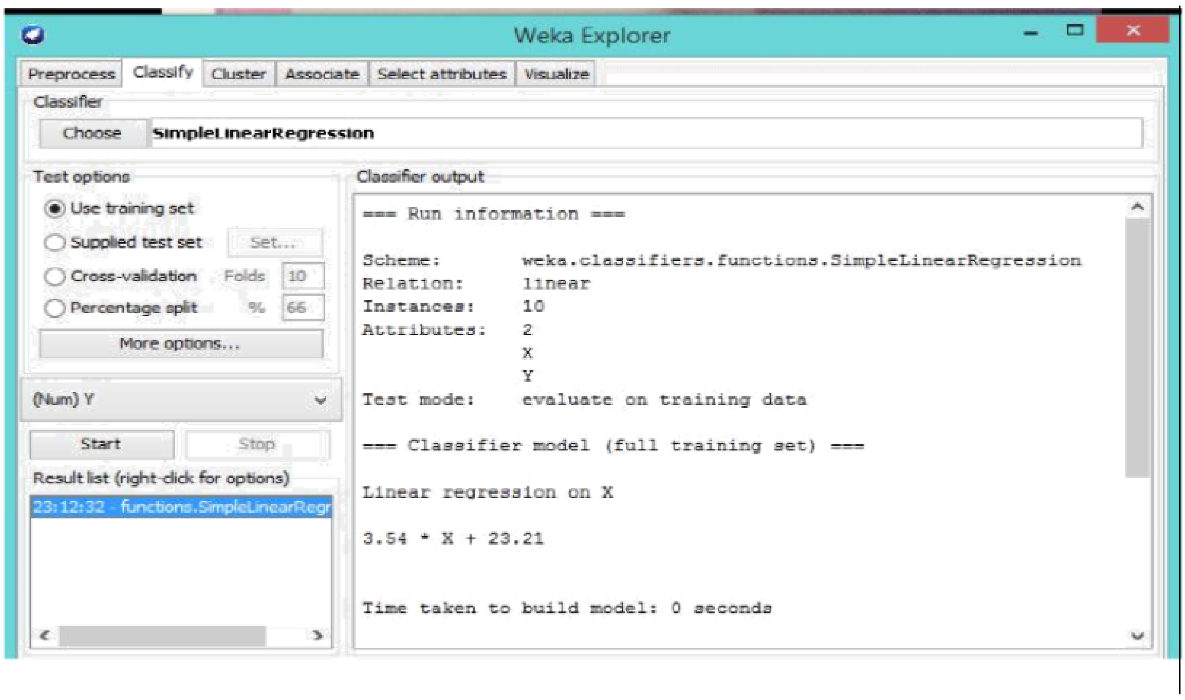
RESULT: Thus the concept of Linear Regression for training the given dataset is applied and implemented

Ex.No.2 - INPUT



	A	B
1	X	Y
2	3	30
3	8	57
4	9	64
5	13	72
6	3	36
7	6	43
8	11	59
9	21	90
10	1	20
11	16	83
12		

OUTPUT:



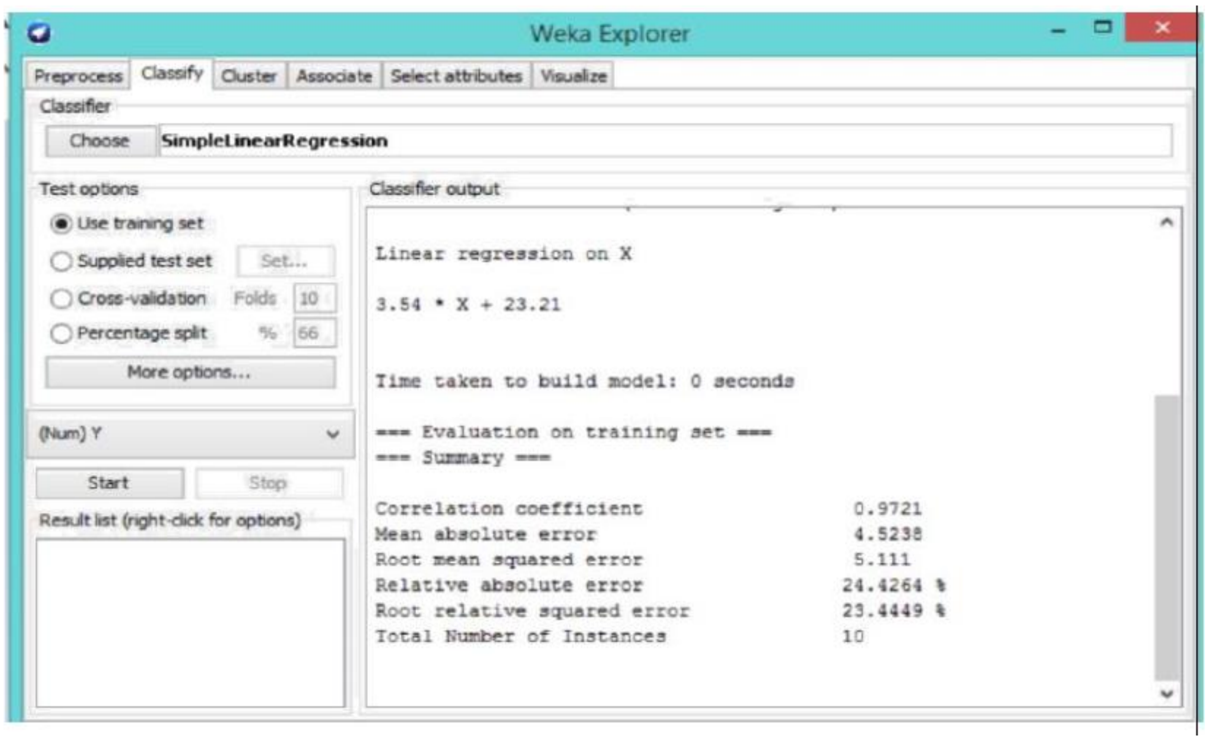
Classifier: SimpleLinearRegression

Test options:
- ● Use training set
- ○ Supplied test set
- ○ Cross-validation Folds 10
- ○ Percentage split % 66

(Num) Y

Result list: 23:12:32 - functions.SimpleLinearRegr

```
=== Run information ===

Scheme:       weka.classifiers.functions.SimpleLinearRegression
Relation:     linear
Instances:    10
Attributes:   2
              X
              Y
Test mode:    evaluate on training data

=== Classifier model (full training set) ===

Linear regression on X

3.54 * X + 23.21


Time taken to build model: 0 seconds
```



Classifier: SimpleLinearRegression

Test options:
- ● Use training set
- ○ Supplied test set
- ○ Cross-validation Folds 10
- ○ Percentage split % 66

(Num) Y

```
Linear regression on X

3.54 * X + 23.21


Time taken to build model: 0 seconds

=== Evaluation on training set ===
=== Summary ===

Correlation coefficient                  0.9721
Mean absolute error                      4.5238
Root mean squared error                  5.111
Relative absolute error                 24.4264 %
Root relative squared error             23.4449 %
Total Number of Instances               10
```

Ex.No.3.

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

Problem Definition: When a database contains a large number of attributes, there will be several attributes which do not become significant in the analysis that you are currently seeking. Thus, removing the unwanted attributes from the dataset becomes an important task in developing a good machine learning model. You may examine the entire dataset visually and decide on the irrelevant attributes. This could be a huge task for databases containing a large number of attributes like the supermarket case. WEKA provides an automated tool for feature selection.

PROCEDURE:

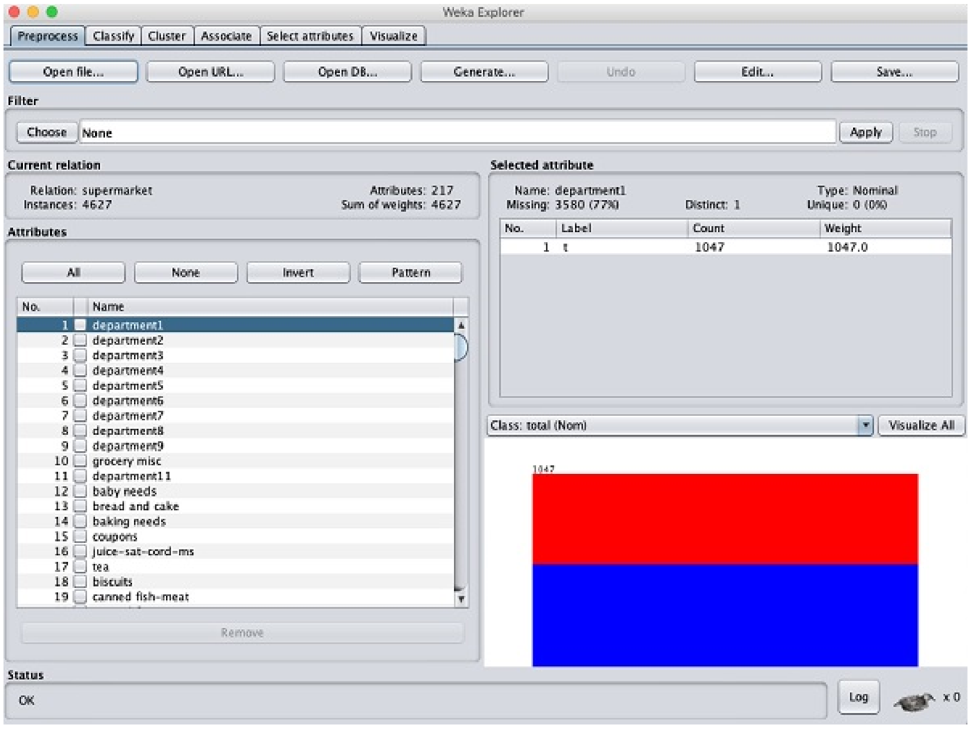
Steps :

1. Open the weka tool.
2. Download a dataset labor.arff by using UCI.
3. Click on the Select attributesTAB for Features Extraction..
4. In the Attribute Selection Mode, use full training set option. Click on the Start button to process the dataset. At the bottom of the result window, it will get the list
   of Selected attributes
5. To get the visual representation, right click on the result in the Result list.
6. Clicking on any of the squares will produce the data plot for further analysis. A typical data plot is shown in screen as result.

Result: Thus the application has been successfully implemented and executed

OUTPUT :

DATASET – LABOR.ARFF



Filter: None

Current relation
Relation: supermarket
Instances: 4627
Attributes: 217
Sum of weights: 4627

Attributes

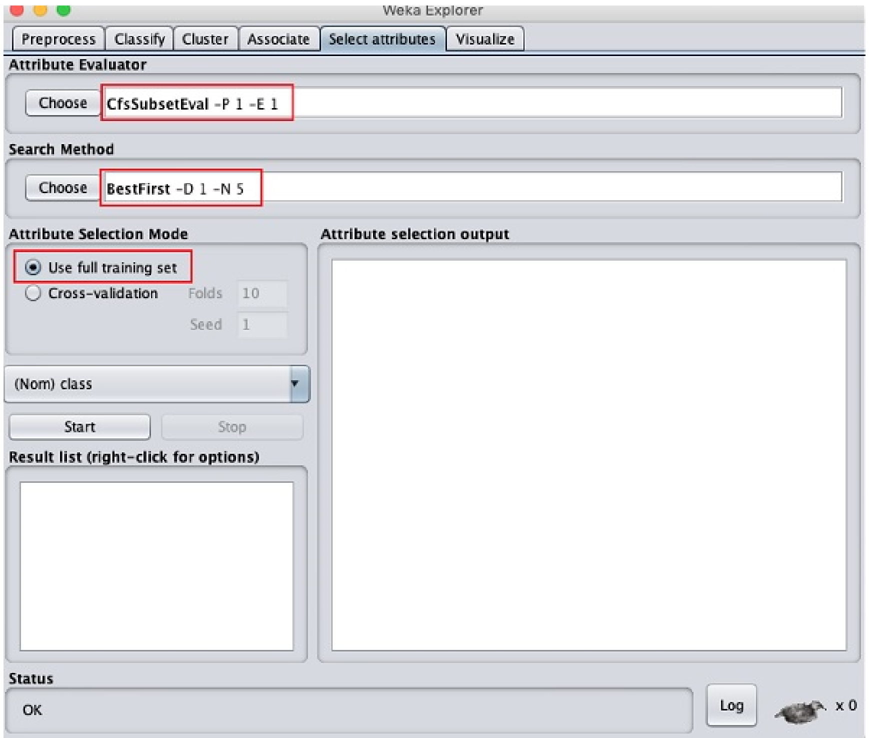
No.	Name
1	department1
2	department2
3	department3
4	department4
5	department5
6	department6
7	department7
8	department8
9	department9
10	grocery misc
11	department11
12	baby needs
13	bread and cake
14	baking needs
15	coupons
16	juice-sat-cord-ms
17	tea
18	biscuits
19	canned fish-meat

Selected attribute
Name: department1
Missing: 3580 (77%)
Distinct: 1
Type: Nominal
Unique: 0 (0%)

No.	Label	Count	Weight
1	t	1047	1047.0

Class: total (Nom)

Status
OK



Attribute Evaluator: CfsSubsetEval -P 1 -E 1

Search Method: BestFirst -D 1 -N 5

Attribute Selection Mode

● Use full training set

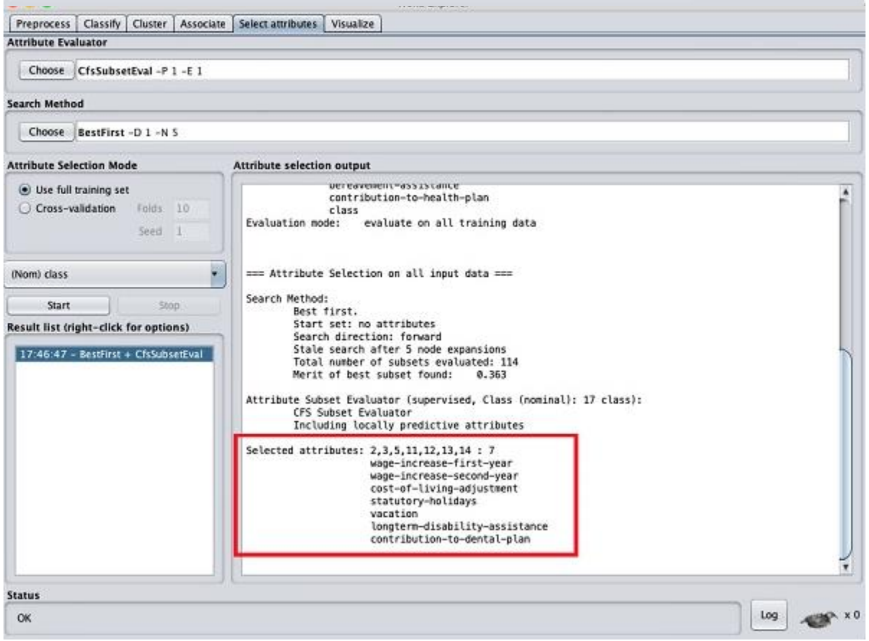
○ Cross-validation Folds 10 Seed 1

(Nom) class

Result list (right-click for options)

Attribute selection output

Status: OK



Attribute Evaluator: CfsSubsetEval -P 1 -E 1

Search Method: BestFirst -D 1 -N 5

Attribute Selection Mode

● Use full training set

○ Cross-validation Folds 10 Seed 1

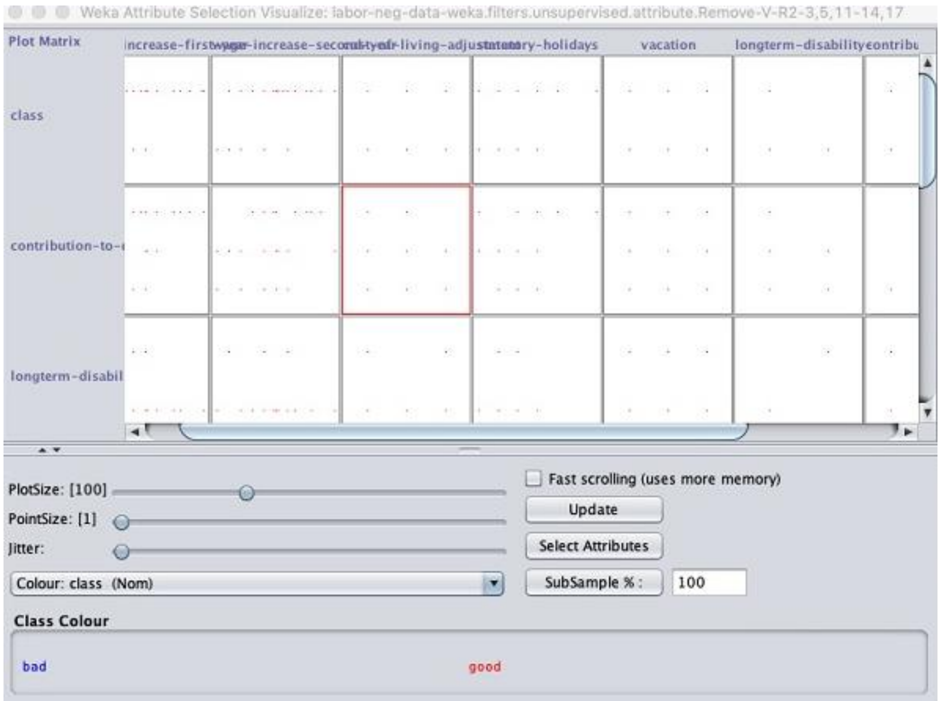
(Nom) class

Result list (right-click for options): 17:46:47 - BestFirst + CfsSubsetEval

Attribute selection output

```
                bereavement-assistance
                contribution-to-health-plan
                class
Evaluation mode:    evaluate on all training data



=== Attribute Selection on all input data ===

Search Method:
        Best first.
        Start set: no attributes
        Search direction: forward
        Stale search after 5 node expansions
        Total number of subsets evaluated: 114
        Merit of best subset found:    0.363

Attribute Subset Evaluator (supervised, Class (nominal): 17 class):
        CFS Subset Evaluator
        Including locally predictive attributes

Selected attributes: 2,3,5,11,12,13,14 : 7
                     wage-increase-first-year
                     wage-increase-second-year
                     cost-of-living-adjustment
                     statutory-holidays
                     vacation
                     longterm-disability-assistance
                     contribution-to-dental-plan
```

Status: OK



Weka Attribute Selection Visualize: labor-neg-data-weka.filters.unsupervised.attribute.Remove-V-R2-3,5,11-14,17

PlotSize: [100]

PointSize: [1]

Jitter:

Colour: class (Nom)

□ Fast scrolling (uses more memory)

SubSample % : 100

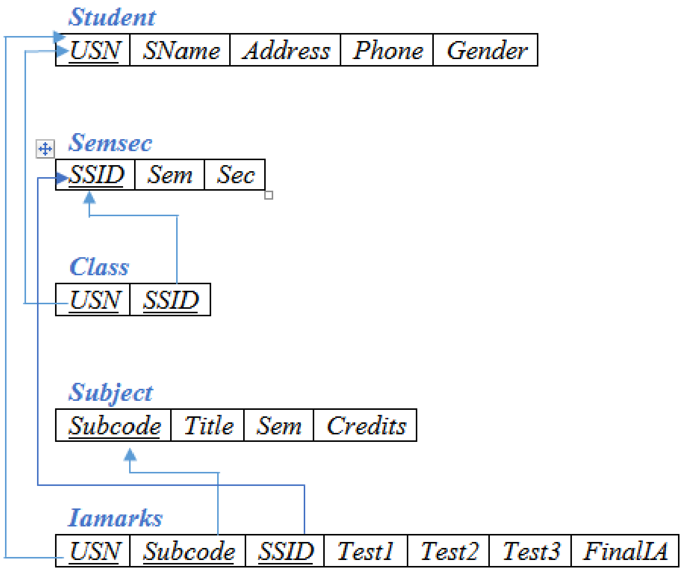
Class Colour

bad good

Ex.4

Aim: To write the query for schema definition

A **schema** is a collection of database objects like tables, triggers, stored procedures, etc. A schema is connected with a user which is known as the schema owner. Database may have one or more schema.



PROCEDURE:

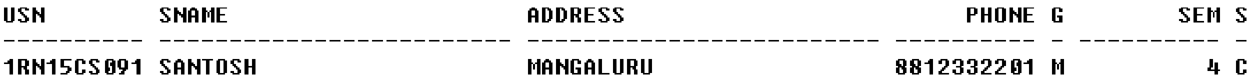
Step 1: Create the table and insert datas

- STUDENT (*USN, SName, Address, Phone, Gender*)
- SEMSEC (*SSID, Sem, Sec*)
- CLASS (*USN, SSID*)
- SUBJECT (*Subcode, Title, Sem, Credits*)
- IAMARKS (*USN, Subcode, SSID, Test1, Test2, Test3, FinalIA*)

Step 2: Write SQL queries to

1. List all the student details studying in fourth semester 'C' section.

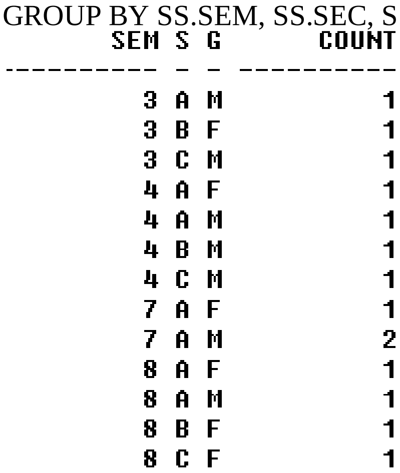
SELECT S.*, SS.SEM, SS.SEC FROM STUDENT S, SEMSEC SS, CLASS C WHERE S.USN = C.USN AND SS.SSID = C.SSID AND SS.SEM = 4 AND SS.SEc='C';



```
USN        SNAME                     ADDRESS                        PHONE G        SEM S
---------- ------------------------- ------------------------- ---------- - ---------- -
1RN15CS091 SANTOSH                   MANGALURU                 8812332201 M          4 C
```

2. Compute the total number of male and female students in each semester and in each section.

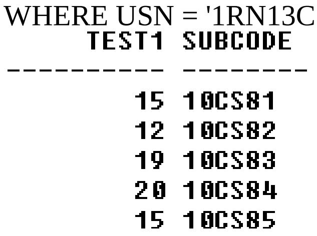
SELECT SS.SEM, SS.SEC, S.GENDER, COUNT (S.GENDER) AS COUNT FROM STUDENT S, SEMSEC SS, CLASS C WHERES.USN = C.USN AND SS.SSID = C.SSID
GROUP BY SS.SEM, SS.SEC, S.GENDER ORDER BY SEM;



```
       SEM S G      COUNT
---------- - - ----------
         3 A M          1
         3 B F          1
         3 C M          1
         4 A F          1
         4 A M          1
         4 B M          1
         4 C M          1
         7 A F          1
         7 A M          2
         8 A F          1
         8 A M          1
         8 B F          1
         8 C F          1
```

3. Create a view of Test1 marks of student USN '1BI15CS101' in all subjects.

CREATE VIEW STU_TEST1_MARKS_VIEW AS SELECT TEST1, SUBCODE FROM IAMARKS WHERE USN = '1RN13CS091';



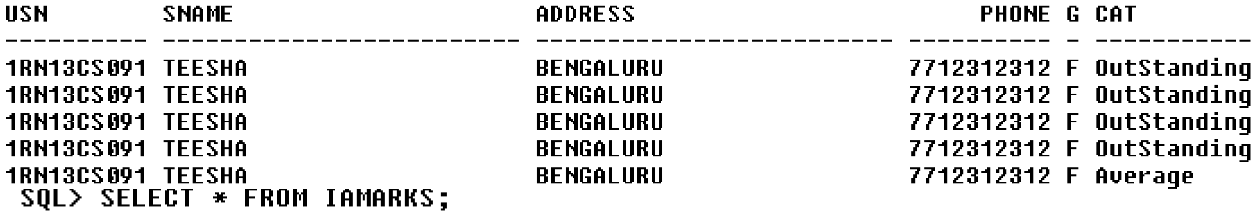
```
     TEST1 SUBCODE
---------- --------
        15 10CS81
        12 10CS82
        19 10CS83
        20 10CS84
        15 10CS85
```

4. Calculate the FinalIA (average of best two test marks) and update the corresponding table for all students.

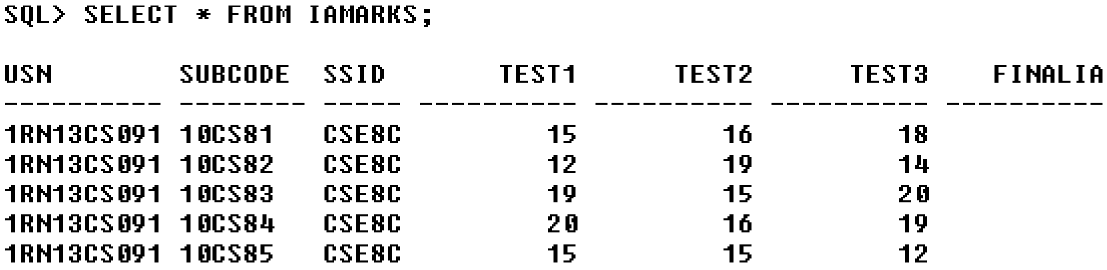
CREATE OR REPLACE PROCEDURE AVGMARKS IS CURSOR C_IAMARKS IS
SELECT GREATEST(TEST1,TEST2) AS A, GREATEST(TEST1,TEST3) AS B,
GREATEST(TEST3,TEST2) AS C FROM IAMARKS WHERE FINALIA IS NULL
FOR UPDATE;

Before execution of PL/SQL procedure, IAMARKS table contents are:

SELECT * FROM IAMARKS;



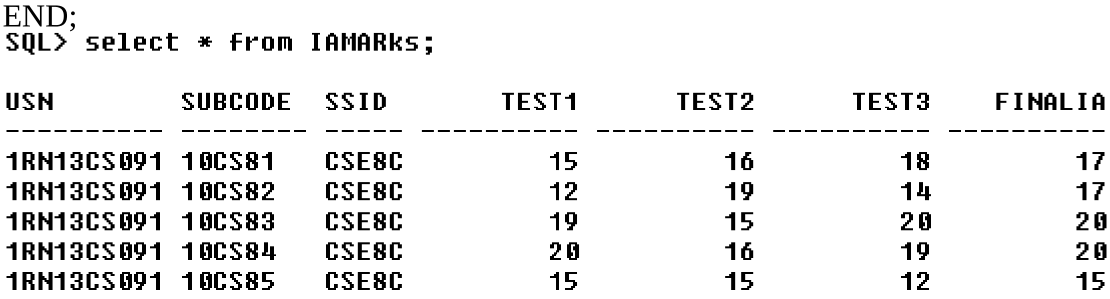
```
USN        SNAME                     ADDRESS                        PHONE G CAT
---------- ------------------------- ------------------------- ---------- - -----------
1RN13CS091 TEESHA                    BENGALURU                 7712312312 F OutStanding
1RN13CS091 TEESHA                    BENGALURU                 7712312312 F OutStanding
1RN13CS091 TEESHA                    BENGALURU                 7712312312 F OutStanding
1RN13CS091 TEESHA                    BENGALURU                 7712312312 F OutStanding
1RN13CS091 TEESHA                    BENGALURU                 7712312312 F Average
```



```
SQL> SELECT * FROM IAMARKS;

USN        SUBCODE  SSID       TEST1      TEST2      TEST3    FINALIA
---------- -------- ----- ---------- ---------- ---------- ----------
1RN13CS091 10CS81   CSE8C         15         16         18
1RN13CS091 10CS82   CSE8C         12         19         14
1RN13CS091 10CS83   CSE8C         19         15         20
1RN13CS091 10CS84   CSE8C         20         16         19
1RN13CS091 10CS85   CSE8C         15         15         12
```

SQL code is to invoke the PL/SQL stored procedure from the command line:

BEGIN
AVGMARKS;
END;



```
SQL> select * from IAMARks;

USN        SUBCODE  SSID       TEST1      TEST2      TEST3    FINALIA
---------- -------- ----- ---------- ---------- ---------- ----------
1RN13CS091 10CS81   CSE8C         15         16         18         17
1RN13CS091 10CS82   CSE8C         12         19         14         17
1RN13CS091 10CS83   CSE8C         19         15         20         20
1RN13CS091 10CS84   CSE8C         20         16         19         20
1RN13CS091 10CS85   CSE8C         15         15         12         15
```

5. Categorize students based on the following criterion:
If FinalIA = 17 to 20 then CAT = ‘Outstanding’
If FinalIA = 12 to 16 then CAT = ‘Average’
If FinalIA< 12 then CAT = ‘Weak’

Give these details only for 8th semester A, B, and C section students.

SELECT S.USN,S.SNAME,S.ADDRESS,S.PHONE,S.GENDER, (CASE WHEN IA.FINALIA BETWEEN 17 AND 20 THEN 'OUTSTANDING' WHEN IA.FINALIA BETWEEN 12 AND 16 THEN 'AVERAGE' ELSE 'WEAK' END) AS CAT FROM STUDENT S, SEMSEC SS, IAMARKS IA, SUBJECT SUB WHERE S.USN = IA.USN AND SS.SSID = IA.SSID AND SUB.SUBCODE = IA.SUBCODE AND SUB.SEM = 8;

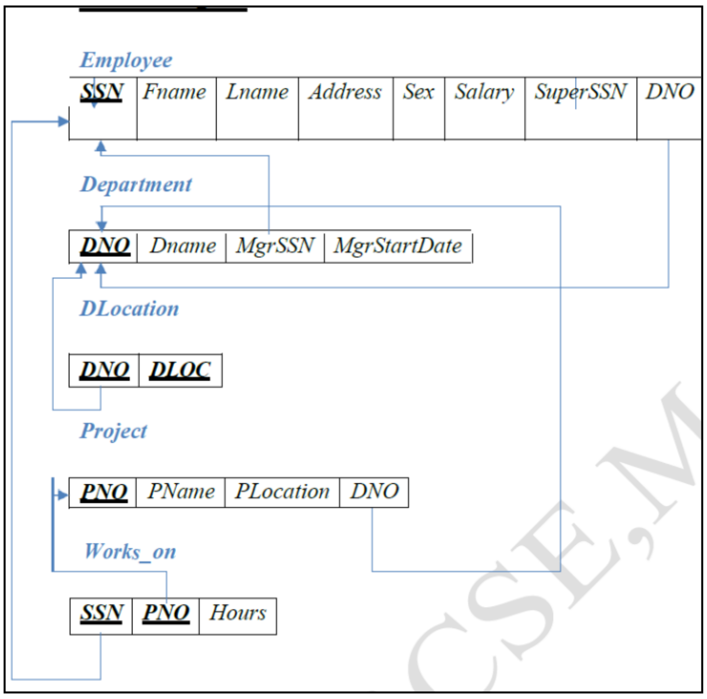
Result : Thus the result has be computed

Ex.No.5:

AIM : To design data ware house for real time applications – Company Database

PROCEDURE :

Create a schema for Company Database:



Step 1:

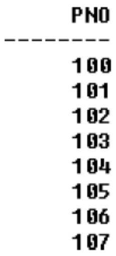
Create the table and insert datas

EMPLOYEE (*SSN, Name, Address, Sex, Salary, SuperSSN, DNo*)
DEPARTMENT (*DNo, DName, MgrSSN, MgrStartDate*)
DLOCATION (*DNo,DLoc*)
PROJECT (*PNo, PName, PLocation, DNo*)
WORKS_ON (*SSN, PNo, Hours*)

Step 2: Write SQL queries to

1. Make a list of all project numbers for projects that involve an employee whose last name is 'Scott', either as a worker or as a manager of the department that controls the project.

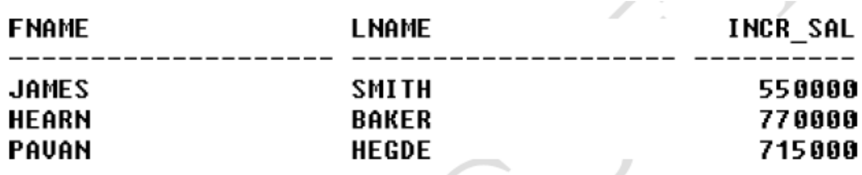
(SELECT DISTINCT P.PNO FROM PROJECT P, DEPARTMENT D, EMPLOYEE E
WHERE E.DNO=D.DNO AND D.MGRSSN=E.SSN AND E.LNAME='SCOTT')
UNION (SELECT DISTINCT P1.PNO FROM PROJECT P1, WORKS_ON W, EMPLOYEE E1 WHERE P1.PNO=W.PNO AND E1.SSN=W.SSN AND
E1.LNAME='SCOTT');



```
     PNO
--------
     100
     101
     102
     103
     104
     105
     106
     107
```

2. Show the resulting salaries if every employee working on the 'IoT' project is given a 10 percent raise.

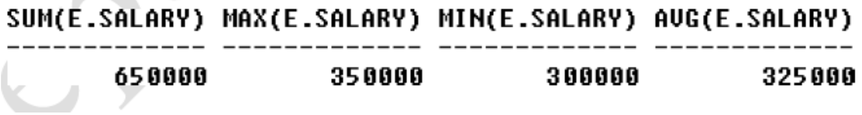
SELECT E.FNAME, E.LNAME, 1.1*E.SALARY AS INCR_SAL FROM EMPLOYEE E, WORKS_ON W, PROJECT P WHERE E.SSN=W.SSN AND W.PNO=P.PNO
AND P.PNAME='IOT';



```
FNAME                LNAME                INCR_SAL
-------------------- -------------------- ----------
JAMES                SMITH                    550000
HEARN                BAKER                    770000
PAVAN                HEGDE                    715000
```

3. Find the sum of the salaries of all employees of the ‘Accounts’ department, as well
as the maximum salary, the minimum salary, and the average salary in this department

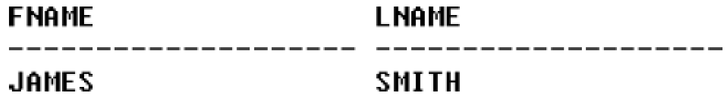
SELECT SUM (E.SALARY), MAX (E.SALARY), MIN (E.SALARY), AVG (E.SALARY)
FROM EMPLOYEE E, DEPARTMENT D WHERE E.DNO=D.DNO AND D.DNAME=’ACCOUNTS’;



```
SUM(E.SALARY) MAX(E.SALARY) MIN(E.SALARY) AVG(E.SALARY)
------------- ------------- ------------- -------------
       650000        350000        300000        325000
```

4. Retrieve the name of each employee who works on all the projects controlled by department number

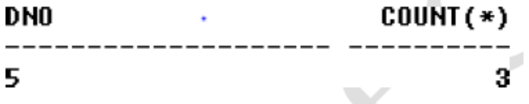
SELECT E.FNAME, E.LNAME FROM EMPLOYEE EWHERE NOT EXISTS((SELECT PNOFROM PROJECT WHERE DNO=’5’) MINUS (SELECT PNO FROM WORKS_ON
WHERE E.SSN=SSN));



```
FNAME                LNAME
-------------------- --------------------
JAMES                SMITH
```

5 (use NOT EXISTS operator). For each department that has more than five employees, retrieve the department number and the number of its employees who are making more than Rs. 6,00,000.

SELECT D.DNO, COUNT (*)FROM DEPARTMENT D, EMPLOYEE E WHERE D.DNO=E.DNOAND E.SALARY>600000AND D.DNO IN (SELECT E1.DNO
FROM EMPLOYEE E1 GROUP BY E1.DNO HAVING COUNT (*)>5)GROUP BY D.DNO;



```
DNO                    COUNT(*)
-------------------- ----------
5                             3
```

Result : Thus the result has be computed