# DUBLIN INSTITUTE OF TECHNOLOGY

**Working with Data (DATA9910)**

**(Assignment)**

**Professor: Brendan Tierney**

NAME: **ASHIS SAHU**

STUDENT ID: **D17129721**

COURSE: **DT-228A ( DATA ANALYTICS ) 2018-19**

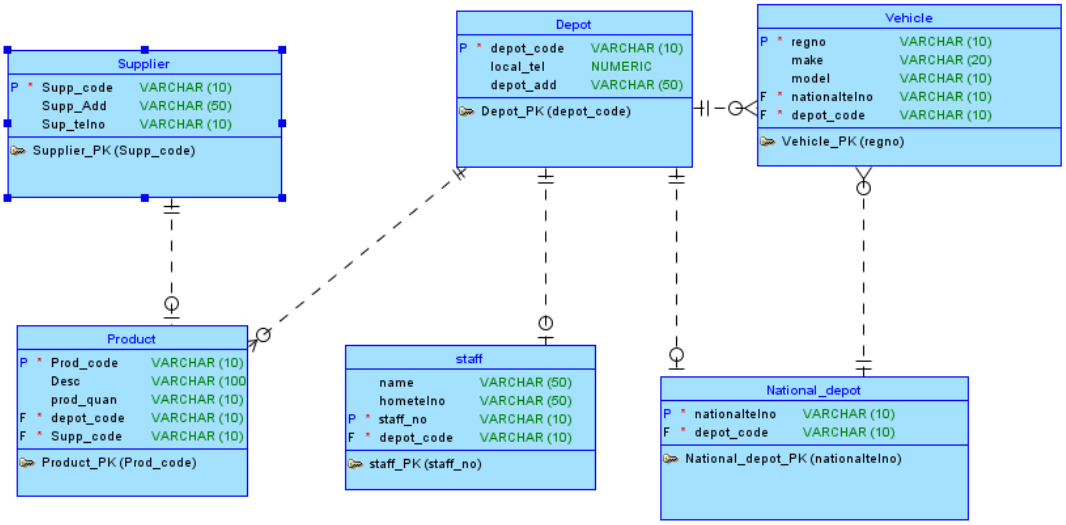
## Table of Contents

1. PART A (Exercise 2 - Dublin Logistics) .......... 4
ER Diagram .......... 4
DDL COMMANDS .......... 4
INSERTING VALUES TO TABLES .......... 11
SQL QUERIES .......... 15
QUERY #1 .......... 15
QUERY #2 .......... 15
QUERY #3 .......... 16
QUERY #4 .......... 16
QUERY #5 .......... 16
2. PART B SQL (Statistical and Analytical Functions on bank marketing dataset) .......... 17
Steps for loading data (BANK MARKETING DATASET) .......... 17
2.1 MAX .......... 23
2.2 MIN .......... 23
2.3 SPEARMAN CORRELATION .......... 25
2.4 MEDIAN .......... 25
2.5 RANK .......... 26
2.6 LAG .......... 27
2.7 LEAD .......... 28
2.8 NTILE .......... 29
3. Part C (Machine Learning using SQL) .......... 30
MODEL 1 (DECISION TREE) .......... 30
CREATING TRAIN DATASET .......... 30
CREATING TEST DATASET .......... 30
Create the settings table .......... 31
VARAIBLES IN USAGE FOR DECISION TREE .......... 32
CREATING CONFUSION MATRIX FOR DECISION TREE .......... 34
Model 2 – SUPPORT VECTOR MACHINE .......... 35
Create the settings table .......... 35
CREATING SVM MODEL .......... 36

ATTRIBUTE USED FOR SVM ..... 36
CONFUSION MATRIX FOR SVM ..... 38
Model 3 LINEAR REGRESSION ..... 39
Create the settings table ..... 39
BUILDING LINEAR REGRESSION MODEL ..... 40
ATTRIBUT USAGE IN LINEAR REGRESSION MODEL ..... 40
CONFUSION MATRIX FOR LINEAR REGRESSION ..... 41
RESULTS AND OBSERVATION ..... 42
4. Part D ( PL/SQL Code – CONFUSION MATRIX) ..... 43
APPENDIX FOR CONFUSION MATRIX ..... 43

# 1. PART A (Exercise 2 - Dublin Logistics)

## ER Diagram



## DDL COMMANDS

```
DROP TABLE depot CASCADE CONSTRAINTS;


DROP TABLE national_depot CASCADE CONSTRAINTS;


DROP TABLE product CASCADE CONSTRAINTS;


DROP TABLE relation_21 CASCADE CONSTRAINTS;
DROP TABLE staff CASCADE CONSTRAINTS;

DROP TABLE supplier CASCADE CONSTRAINTS;

DROP TABLE vehicle CASCADE CONSTRAINTS;

CREATE TABLE depot (
depot_code VARCHAR2(10) NOT NULL,
local_tel NUMBER,
depot_add VARCHAR2(50)
);

ALTER TABLE depot ADD CONSTRAINT depot_pk PRIMARY KEY ( depot_code );

CREATE TABLE national_depot (
nationaltelno VARCHAR2(10) NOT NULL,
depot_code VARCHAR2(10) NOT NULL
);

CREATE UNIQUE INDEX national_depot__idx ON
```

```
national_depot (

depot_code

ASC );


CREATE UNIQUE INDEX national_depot__idxv1 ON

national_depot (

depot_code

ASC );


ALTER TABLE national_depot ADD CONSTRAINT national_depot_pk
PRIMARY KEY ( nationaltelno );


CREATE TABLE product (

prod_code VARCHAR2(10) NOT NULL,

&quot;desc&quot; VARCHAR2(100),

prod_quan VARCHAR2(10),

supp_code VARCHAR2(10) NOT NULL

);


ALTER TABLE product ADD CONSTRAINT product_pk PRIMARY KEY (
prod_code );
```

```
CREATE TABLE relation_21 (
product_prod_code VARCHAR2(10) NOT NULL,
depot_depot_code VARCHAR2(10) NOT NULL
);

ALTER TABLE relation_21
ADD CONSTRAINT relation_21_pk PRIMARY KEY ( product_prod_code,
depot_depot_code);

CREATE TABLE staff (
name VARCHAR2(50),
hometelno VARCHAR2(50),
depot_code VARCHAR2(10) NOT NULL,
staff_no VARCHAR2(10) NOT NULL

);

ALTER TABLE staff ADD CONSTRAINT staff_pk PRIMARY KEY ( staff_no );

CREATE TABLE supplier (
supp_code VARCHAR2(10) NOT NULL,
supp_add VARCHAR2(50),
sup_telno VARCHAR2(10)
);

ALTER TABLE supplier ADD CONSTRAINT supplier_pk PRIMARY KEY ( supp_code );

CREATE TABLE vehicle (
regno VARCHAR2(10) NOT NULL,
make VARCHAR2(20),
model VARCHAR2(10),
nationaltelno VARCHAR2(10) NOT NULL,
depot_code VARCHAR2(10) NOT NULL
);

ALTER TABLE vehicle ADD CONSTRAINT vehicle_pk PRIMARY KEY ( regno );

*ALTER TABLE depot
ADD CONSTRAINT depot_product_fk FOREIGN KEY ( prod_code,
supp_code )
REFERENCES product ( prod_code );
```

```
*ALTER TABLE depot
ADD CONSTRAINT depot_product_fkv2 FOREIGN KEY ( product_prod_code )
REFERENCES product ( prod_code );

ALTER TABLE national_depot
ADD CONSTRAINT national_depot_depot_fk FOREIGN KEY ( depot_code )
REFERENCES depot ( depot_code );

*ALTER TABLE product
ADD CONSTRAINT product_supplier_fk FOREIGN KEY ( supp_code )
REFERENCES supplier ( supp_code );

ALTER TABLE vehicle
ADD CONSTRAINT relation_15 FOREIGN KEY ( depot_code )
REFERENCES depot ( depot_code );

ALTER TABLE staff
ADD CONSTRAINT staff_depot_fk FOREIGN KEY ( depot_code )
REFERENCES depot ( depot_code );
```

```
ALTER TABLE vehicle
ADD CONSTRAINT vehicle_national_depot_fk FOREIGN KEY ( nationaltelno )
REFERENCES national_depot ( nationaltelno );

ALTER TABLE national_depot
ADD CONSTRAINT national_depot_depot_fk FOREIGN KEY ( depot_code )
REFERENCES depot ( depot_code );

ALTER TABLE product
ADD CONSTRAINT product_supplier_fk FOREIGN KEY ( supp_code )
REFERENCES supplier ( supp_code );

ALTER TABLE relation_21
ADD CONSTRAINT relation_21_depot_fk FOREIGN KEY ( depot_depot_code )
REFERENCES depot ( depot_code );

ALTER TABLE vehicle
ADD CONSTRAINT vehicle_national_depot_fk FOREIGN KEY ( nationaltelno )
REFERENCES national_depot ( nationaltelno );
```

## INSERTING VALUES TO TABLES

```
select * from depot;

select * from national_depot;

select * from PRODUCT;

select * from STAFF;

select * from SUPPLIER;

select * from VEHICLE;


insert into depot( DEPOT_CODE, LOCAL_TEL, DEPOT_ADD)

VALUES ('111','1111','AA11');




insert into depot( DEPOT_CODE, LOCAL_TEL, DEPOT_ADD)

VALUES('222','2222','BB22');


insert into depot( DEPOT_CODE, LOCAL_TEL, DEPOT_ADD)

VALUES('333','3333','CC33');


insert into depot( DEPOT_CODE, LOCAL_TEL, DEPOT_ADD)

VALUES('444','4444','DD44');
```

```
insert into depot( DEPOT_CODE, LOCAL_TEL, DEPOT_ADD)
VALUES('555','5555','EE55');


--NATIONAL DEPOT
insert into NATIONAL_DEPOT (NATIONALTELNO,DEPOT_CODE)
VALUES ('2111','111');


insert into NATIONAL_DEPOT (NATIONALTELNO,DEPOT_CODE)
VALUES ('3111','222');
insert into NATIONAL_DEPOT (NATIONALTELNO,DEPOT_CODE)
VALUES ('4111','333');
insert into NATIONAL_DEPOT (NATIONALTELNO,DEPOT_CODE)
VALUES ('5111','444');
insert into NATIONAL_DEPOT (NATIONALTELNO,DEPOT_CODE)
VALUES ('6111','555');


--VEHICLE
insert into VEHICLE (REGNO, MAKE, MODEL,
NATIONALTELNO,DEPOT_CODE)
VALUES ('AAAA','2004','FORD','2111','111');
```

```
insert into vehicle (REGNO, MAKE, MODEL,
NATIONALTELNO,DEPOT_CODE)

VALUES ('BBBB','2005','FORDM','2111','444');

insert into vehicle (REGNO, MAKE, MODEL,
NATIONALTELNO,DEPOT_CODE)

VALUES ('CCCC','2006','FORD','2111','222');

insert into vehicle (REGNO, MAKE, MODEL,
NATIONALTELNO,DEPOT_CODE)

VALUES ('DDDD','2007','MERCEDES','5111','444');

insert into vehicle (REGNO, MAKE, MODEL,
NATIONALTELNO,DEPOT_CODE)

VALUES ('EEEE','2008','MERCEDES','5111','555');



--STAFF

INSERT INTO STAFF (NAME, HOMETELNO, DEPOT_CODE, STAFF_NO )

VALUES ('AAAAA', '24234', '111','1');

INSERT INTO STAFF (NAME, HOMETELNO, DEPOT_CODE, STAFF_NO )

VALUES ('BBBBB', '24234', '111','2');

INSERT INTO STAFF (NAME, HOMETELNO, DEPOT_CODE, STAFF_NO )

VALUES ('CCCCC', '74234', '222','3');

INSERT INTO STAFF (NAME, HOMETELNO, DEPOT_CODE, STAFF_NO )

VALUES ('DDDDD', '94234', '444','4');

INSERT INTO STAFF (NAME, HOMETELNO, DEPOT_CODE, STAFF_NO )
```

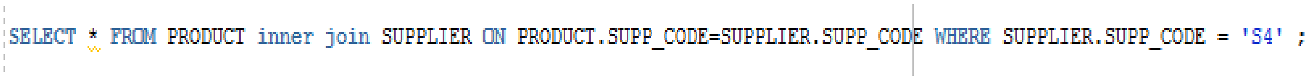
```
VALUES ('EEEEE', '24234', '555','5');


--SUPPLIER
INSERT INTO SUPPLIER (SUPP_CODE,SUPP_ADD,SUP_TELNO )
VALUES ('S1', 'SADD1', '77447');


INSERT INTO SUPPLIER (SUPP_CODE,SUPP_ADD,SUP_TELNO )
VALUES ('S2', 'SADD2', '774471');
INSERT INTO SUPPLIER (SUPP_CODE,SUPP_ADD,SUP_TELNO )
VALUES ('S3', 'SADD3', '774472');
INSERT INTO SUPPLIER (SUPP_CODE,SUPP_ADD,SUP_TELNO )
VALUES ('S4', 'SADD4', '774473');
INSERT INTO SUPPLIER (SUPP_CODE,SUPP_ADD,SUP_TELNO )
VALUES ('S5', 'SADD5', '774474');


--PRODUCT
INSERT INTO PRODUCT (PROD_CODE,"desc",PROD_QUAN, SUPP_CODE )
VALUES ('P1','TOY','441','S1');
INSERT INTO PRODUCT (PROD_CODE,"desc",PROD_QUAN, SUPP_CODE )
VALUES ('P2','PEN','442','S1');
INSERT INTO PRODUCT (PROD_CODE,"desc",PROD_QUAN, SUPP_CODE )
```

```
VALUES ('P3','SHAREBROC','443','S4');

INSERT INTO PRODUCT (PROD_CODE,"desc",PROD_QUAN, SUPP_CODE )

VALUES ('P4','PC','444','S4');

INSERT INTO PRODUCT (PROD_CODE,"desc",PROD_QUAN, SUPP_CODE )

VALUES ('P5','TELEPHONE','445','S4');
```

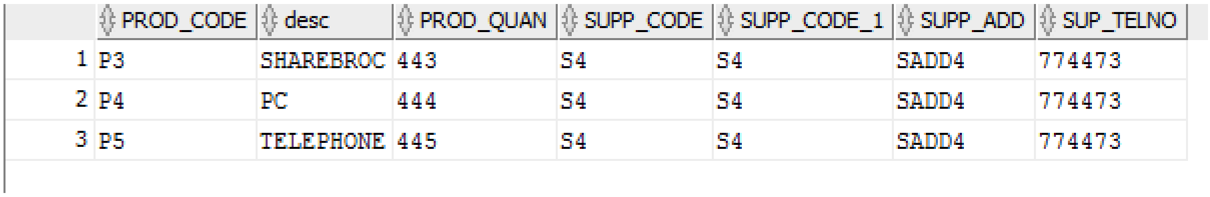
## SQL QUERIES

### QUERY #1

**PROVIDE THE DETAILS OF THE PRODUCTS WITH THE SUPPLIER S4.**



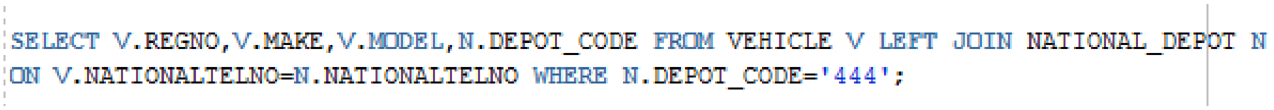
```
SELECT * FROM PRODUCT inner join SUPPLIER ON PRODUCT.SUPP_CODE=SUPPLIER.SUPP_CODE WHERE SUPPLIER.SUPP_CODE = 'S4' ;
```



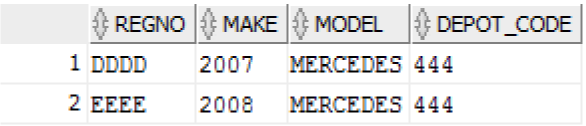
| | PROD_CODE | desc | PROD_QUAN | SUPP_CODE | SUPP_CODE_1 | SUPP_ADD | SUP_TELNO |
|---|---|---|---|---|---|---|---|
| 1 | P3 | SHAREBROC | 443 | S4 | S4 | SADD4 | 774473 |
| 2 | P4 | PC | 444 | S4 | S4 | SADD4 | 774473 |
| 3 | P5 | TELEPHONE | 445 | S4 | S4 | SADD4 | 774473 |

### QUERY #2

**PROVIDE THE VEHICLE INFORMATION WITH THE MAINTENANCE OR NATIONAL DEPOT (DEPOT CODE ‘444’).**



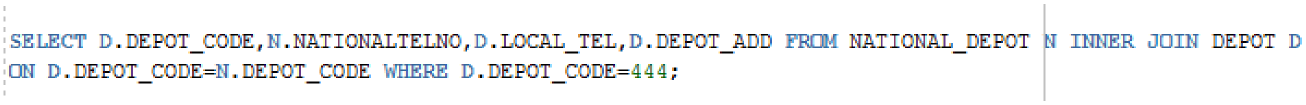
```
SELECT V.REGNO,V.MAKE,V.MODEL,N.DEPOT_CODE FROM VEHICLE V LEFT JOIN NATIONAL_DEPOT N
ON V.NATIONALTELNO=N.NATIONALTELNO WHERE N.DEPOT_CODE='444';
```



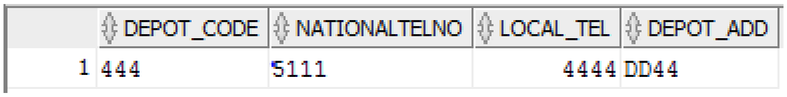
| | REGNO | MAKE | MODEL | DEPOT_CODE |
|---|---|---|---|---|
| 1 | DDDD | 2007 | MERCEDES | 444 |
| 2 | EEEE | 2008 | MERCEDES | 444 |

## *QUERY #3*

**PROVIDE THE NATIONAL DEPOT INFORMATION SUCH AS MAINTENANCE HELPLINETELEPHONE NO, LOCAL TELEPONE, ADDRESS OF THE DEPOT WITH THE DEPOT CODE '444'.**



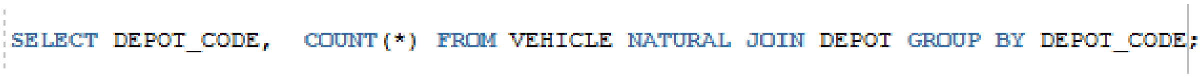
```
SELECT D.DEPOT_CODE,N.NATIONALTELNO,D.LOCAL_TEL,D.DEPOT_ADD FROM NATIONAL_DEPOT N INNER JOIN DEPOT D
ON D.DEPOT_CODE=N.DEPOT_CODE WHERE D.DEPOT_CODE=444;
```



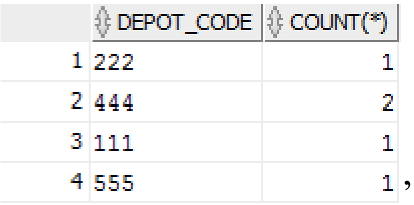
| | DEPOT_CODE | NATIONALTELNO | LOCAL_TEL | DEPOT_ADD |
|---|---|---|---|---|
| 1 | 444 | 5111 | 4444 | DD44 |

## *QUERY #4*

**EVALUATE THE NUMBER OF VEHICLES AT EACH DEPOT FOR DELIVERIES.**



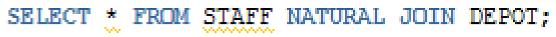
```
SELECT DEPOT_CODE,  COUNT(*) FROM VEHICLE NATURAL JOIN DEPOT GROUP BY DEPOT_CODE;
```



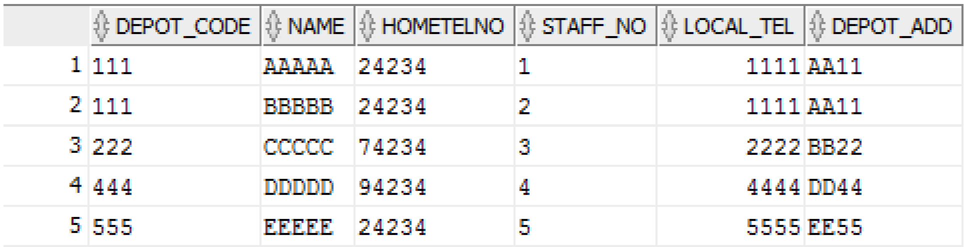
| | DEPOT_CODE | COUNT(*) |
|---|---|---|
| 1 | 222 | 1 |
| 2 | 444 | 2 |
| 3 | 111 | 1 |
| 4 | 555 | 1 |

## *QUERY #5*

**PROVIDE THE MANAGER OR STAFF INFORMATION AT EACH DEPOT.**



```
SELECT * FROM STAFF NATURAL JOIN DEPOT;
```

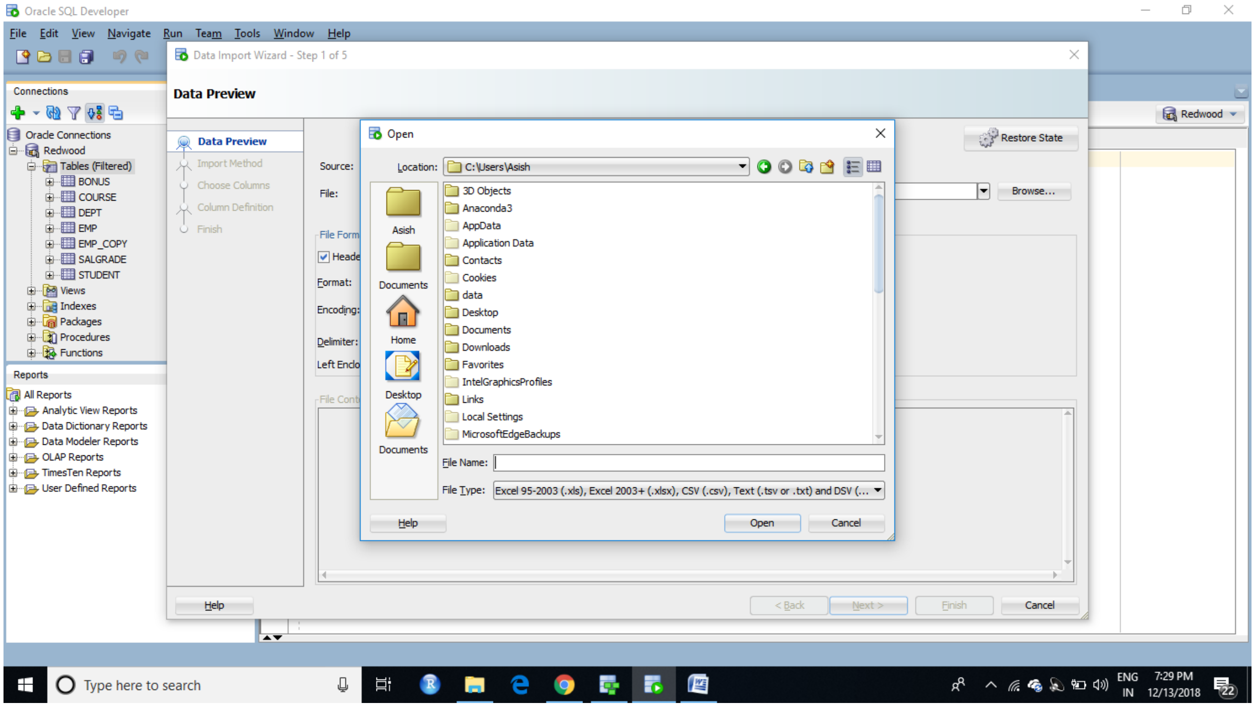


| | DEPOT_CODE | NAME | HOMETELNO | STAFF_NO | LOCAL_TEL | DEPOT_ADD |
|---|---|---|---|---|---|---|
| 1 | 111 | AAAAA | 24234 | 1 | 1111 | AA11 |
| 2 | 111 | BBBBB | 24234 | 2 | 1111 | AA11 |
| 3 | 222 | CCCCC | 74234 | 3 | 2222 | BB22 |
| 4 | 444 | DDDDD | 94234 | 4 | 4444 | DD44 |
| 5 | 555 | EEEEE | 24234 | 5 | 5555 | EE55 |

## 2. PART B SQL (Statistical and Analytical Functions on bank marketing dataset)

## *Steps for loading data (BANK MARKETING DATASET)*

### *1. In Redwood connection >> Right click table menu >> import table..*



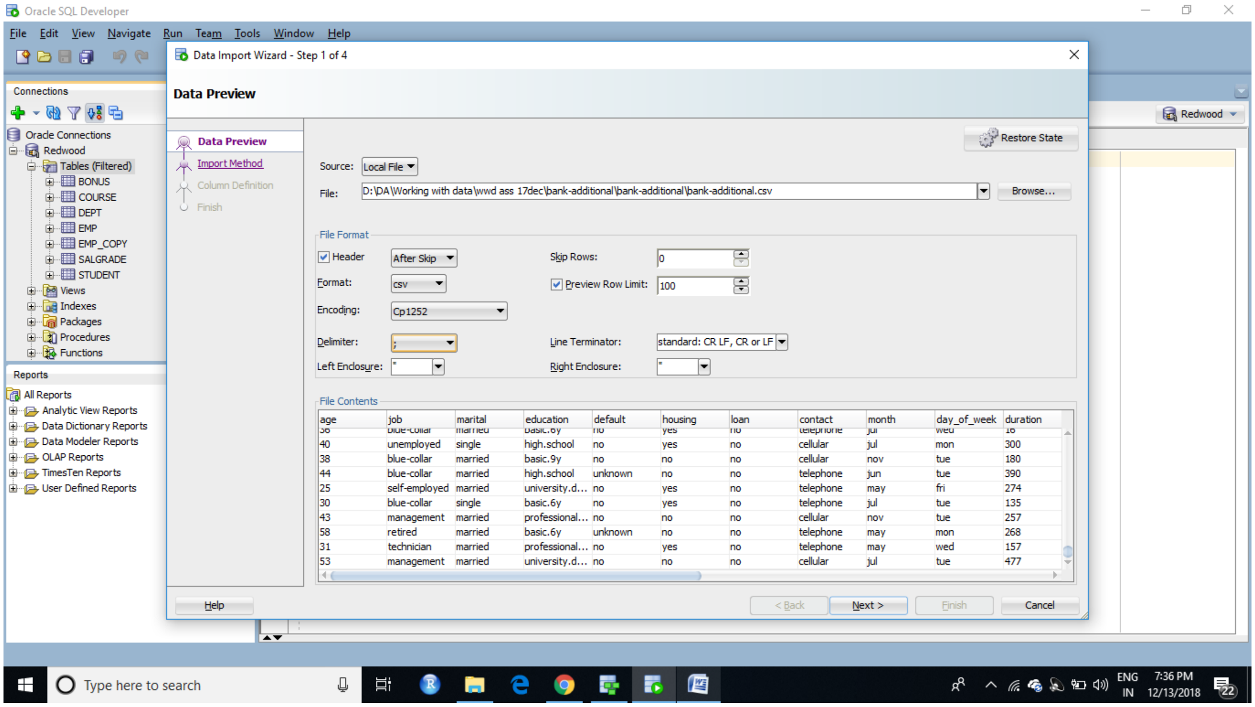
Location: C:\Users\Asish

3D Objects
Anaconda3
AppData
Application Data
Contacts
Cookies
data
Desktop
Documents
Downloads
Favorites
IntelGraphicsProfiles
Links
Local Settings
MicrosoftEdgeBackups

File Name:

File Type: Excel 95-2003 (.xls), Excel 2003+ (.xlsx), CSV (.csv), Text (.tsv or .txt) and DSV (...

2. *Change **delimiter to semi-colon** and click **Next**.*



Data Import Wizard - Step 1 of 4

**Data Preview**

Source: Local File

File: D:\DA\Working with data\wwd ass 17dec\bank-additional\bank-additional\bank-additional.csv

File Format

☑ Header After Skip

Skip Rows: 0

Format: csv

☑ Preview Row Limit: 100

Encoding: Cp1252

Delimiter: ;

Line Terminator: standard: CR LF, CR or LF

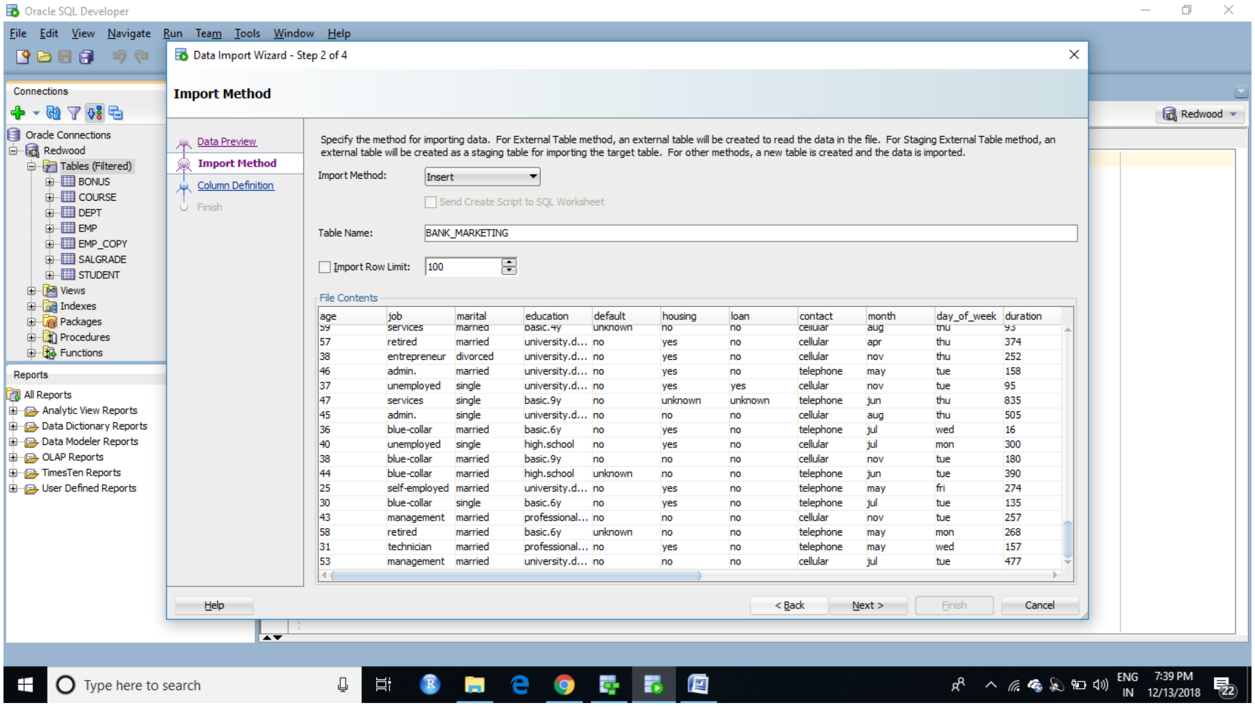
Left Enclosure: "

Right Enclosure: "

File Contents

| age | job | marital | education | default | housing | loan | contact | month | day_of_week | duration |
|---|---|---|---|---|---|---|---|---|---|---|
| 36 | blue-collar | married | basic.6y | no | yes | no | telephone | jul | wed | 16 |
| 40 | unemployed | single | high.school | no | yes | no | cellular | jul | mon | 300 |
| 38 | blue-collar | married | basic.9y | no | no | no | cellular | nov | tue | 180 |
| 44 | blue-collar | married | high.school | unknown | no | no | telephone | jun | tue | 390 |
| 25 | self-employed | married | university.d... | no | yes | no | telephone | may | fri | 274 |
| 30 | blue-collar | single | basic.6y | no | yes | no | telephone | jul | tue | 135 |
| 43 | management | married | professional... | no | no | no | cellular | nov | tue | 257 |
| 58 | retired | married | basic.6y | unknown | no | no | telephone | may | mon | 268 |
| 31 | technician | married | professional... | no | yes | no | telephone | may | wed | 157 |
| 53 | management | married | university.d... | no | no | no | cellular | jul | tue | 477 |

3. *In* ***Import method window,*** *change the table name and click next.*



Import Method

Specify the method for importing data. For External Table method, an external table will be created to read the data in the file. For Staging External Table method, an external table will be created as a staging table for importing the target table. For other methods, a new table is created and the data is imported.

Import Method: Insert

□ Send Create Script to SQL Worksheet

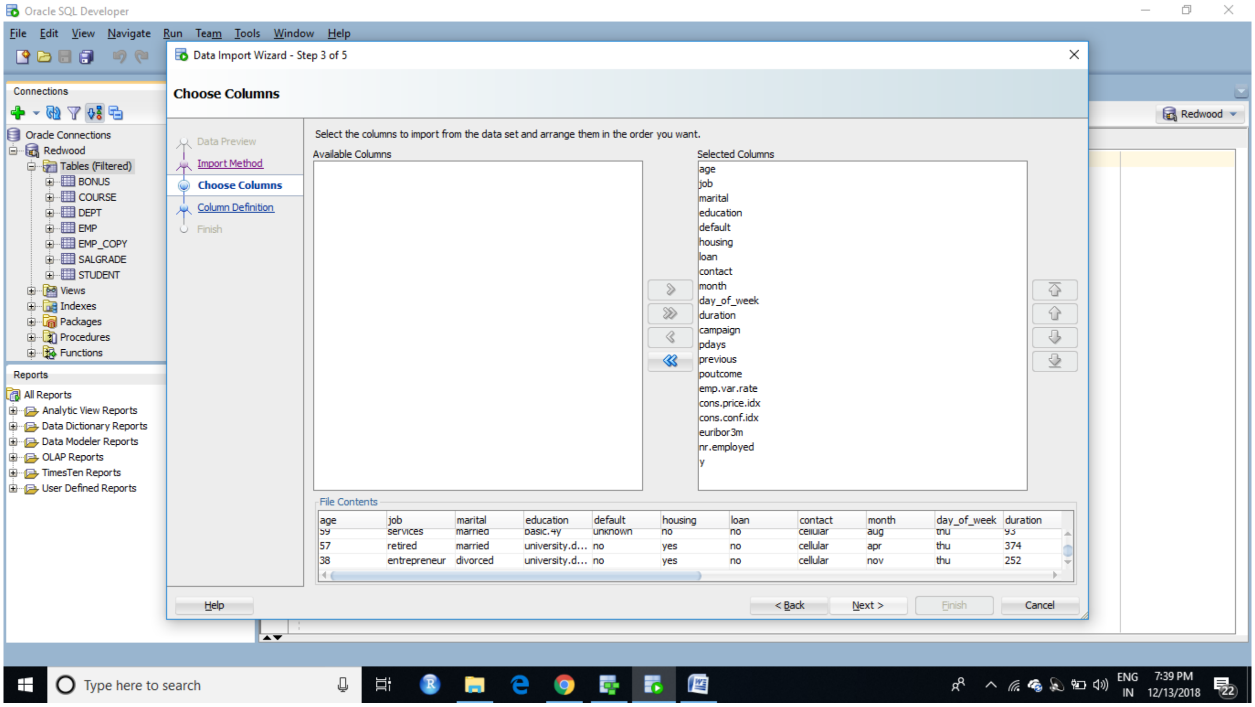
Table Name: BANK_MARKETING

□ Import Row Limit: 100

File Contents

| age | job | marital | education | default | housing | loan | contact | month | day_of_week | duration |
|---|---|---|---|---|---|---|---|---|---|---|
| 59 | services | married | basic.4y | unknown | no | no | cellular | aug | thu | 93 |
| 57 | retired | married | university.d... | no | yes | no | cellular | apr | thu | 374 |
| 38 | entrepreneur | divorced | university.d... | no | yes | no | cellular | nov | thu | 252 |
| 46 | admin. | married | university.d... | no | yes | no | telephone | may | tue | 158 |
| 37 | unemployed | single | university.d... | no | yes | yes | cellular | nov | tue | 95 |
| 47 | services | single | basic.9y | no | unknown | unknown | telephone | jun | thu | 835 |
| 45 | admin. | single | university.d... | no | no | no | cellular | aug | thu | 505 |
| 36 | blue-collar | married | basic.6y | no | yes | no | telephone | jul | wed | 16 |
| 40 | unemployed | single | high.school | no | yes | no | cellular | jul | mon | 300 |
| 38 | blue-collar | married | basic.9y | no | no | no | cellular | nov | tue | 180 |
| 44 | blue-collar | married | high.school | unknown | no | no | telephone | jun | tue | 390 |
| 25 | self-employed | married | university.d... | no | yes | no | telephone | may | fri | 274 |
| 30 | blue-collar | single | basic.6y | no | yes | no | telephone | jul | tue | 135 |
| 43 | management | married | professional... | no | no | no | cellular | nov | tue | 257 |
| 58 | retired | married | basic.6y | unknown | no | no | telephone | may | mon | 268 |
| 31 | technician | married | professional... | no | yes | no | telephone | may | wed | 157 |
| 53 | management | married | university.d... | no | no | no | cellular | jul | tue | 477 |

4. *Again next in **choose column** window.*



Data Import Wizard - Step 3 of 5

Choose Columns

Select the columns to import from the data set and arrange them in the order you want.

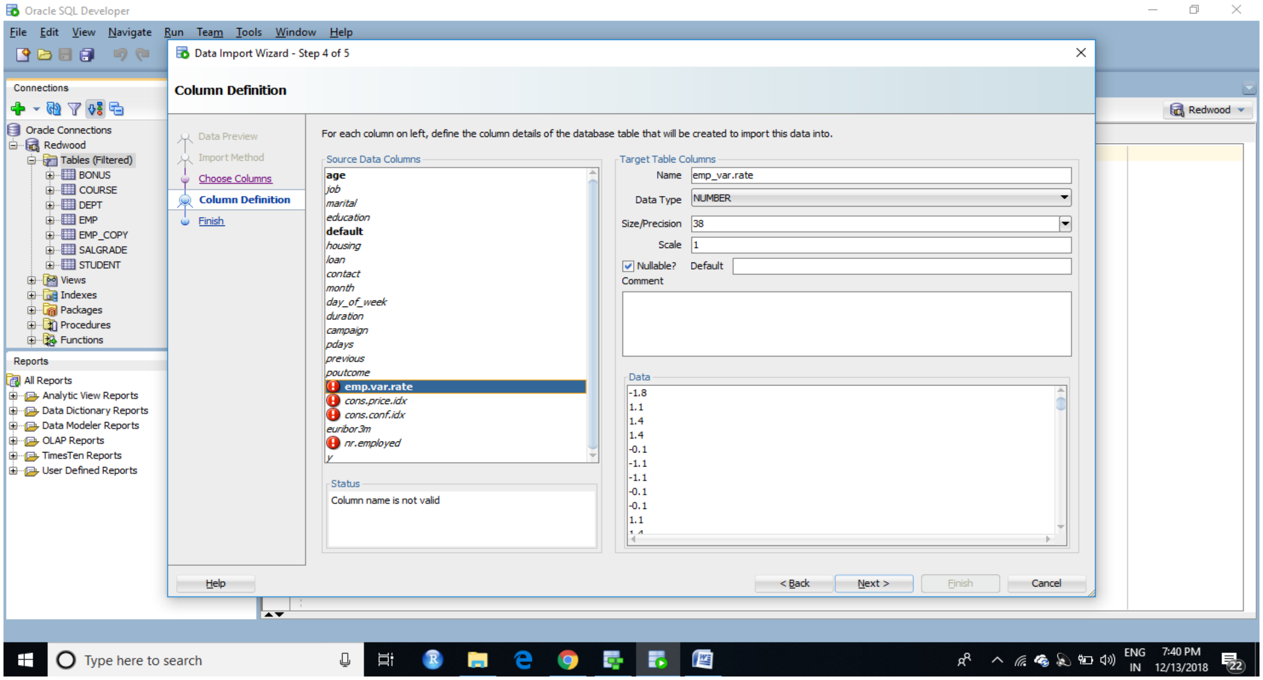
Available Columns

Selected Columns
age
job
marital
education
default
housing
loan
contact
month
day_of_week
duration
campaign
pdays
previous
poutcome
emp.var.rate
cons.price.idx
cons.conf.idx
euribor3m
nr.employed
y

File Contents

| age | job | marital | education | default | housing | loan | contact | month | day_of_week | duration |
|---|---|---|---|---|---|---|---|---|---|---|
| 59 | services | married | basic.4y | unknown | no | no | cellular | aug | thu | 93 |
| 57 | retired | married | university.d... | no | yes | no | cellular | apr | thu | 374 |
| 38 | entrepreneur | divorced | university.d... | no | yes | no | cellular | nov | thu | 252 |

5. *Change the variables name with '_' and* ***'default'****. And* ***next.***



Data Import Wizard - Step 4 of 5

Column Definition

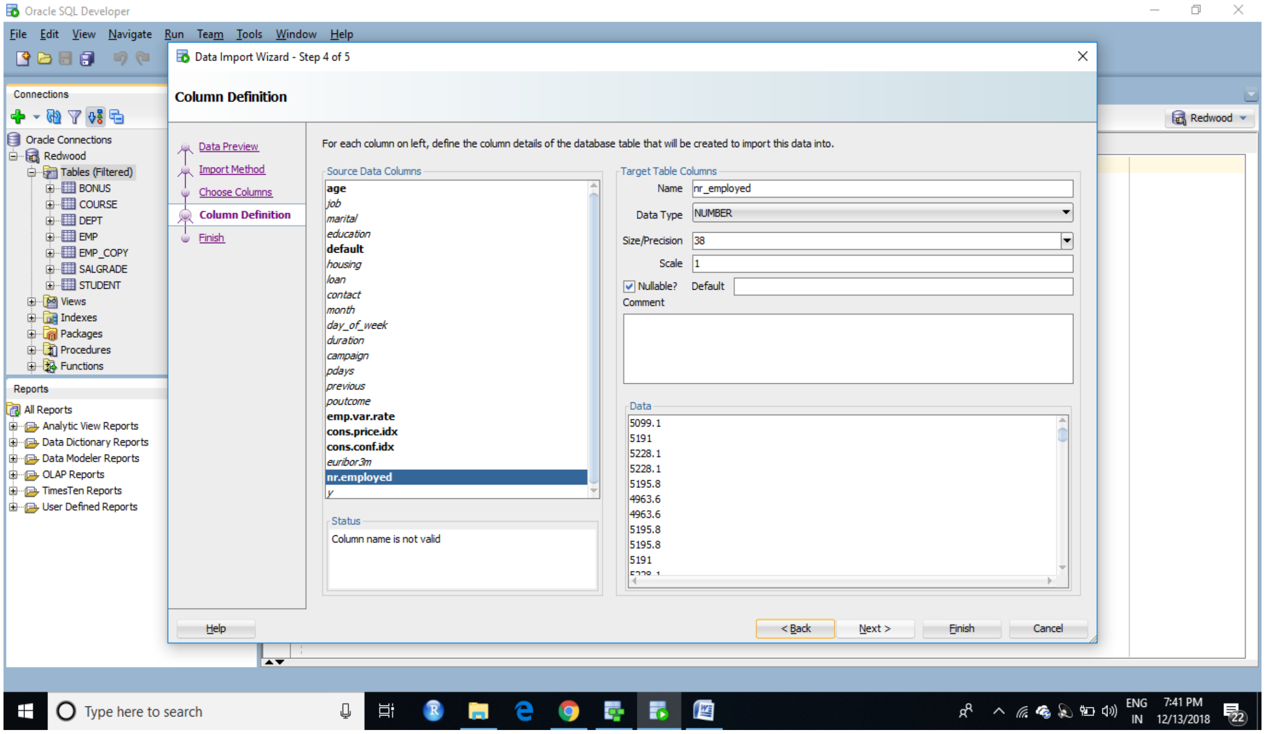
For each column on left, define the column details of the database table that will be created to import this data into.

Source Data Columns: age, job, marital, education, default, housing, loan, contact, month, day_of_week, duration, campaign, pdays, previous, poutcome, emp.var.rate, cons.price.idx, cons.conf.idx, euribor3m, nr.employed, y

Target Table Columns
Name: emp_var.rate
Data Type: NUMBER
Size/Precision: 38
Scale: 1
☑ Nullable? Default:
Comment:

Data
```
-1.8
1.1
1.4
1.4
-0.1
-1.1
-1.1
-0.1
-0.1
1.1
```

Status: Column name is not valid



Data Import Wizard - Step 4 of 5

Column Definition

For each column on left, define the column details of the database table that will be created to import this data into.

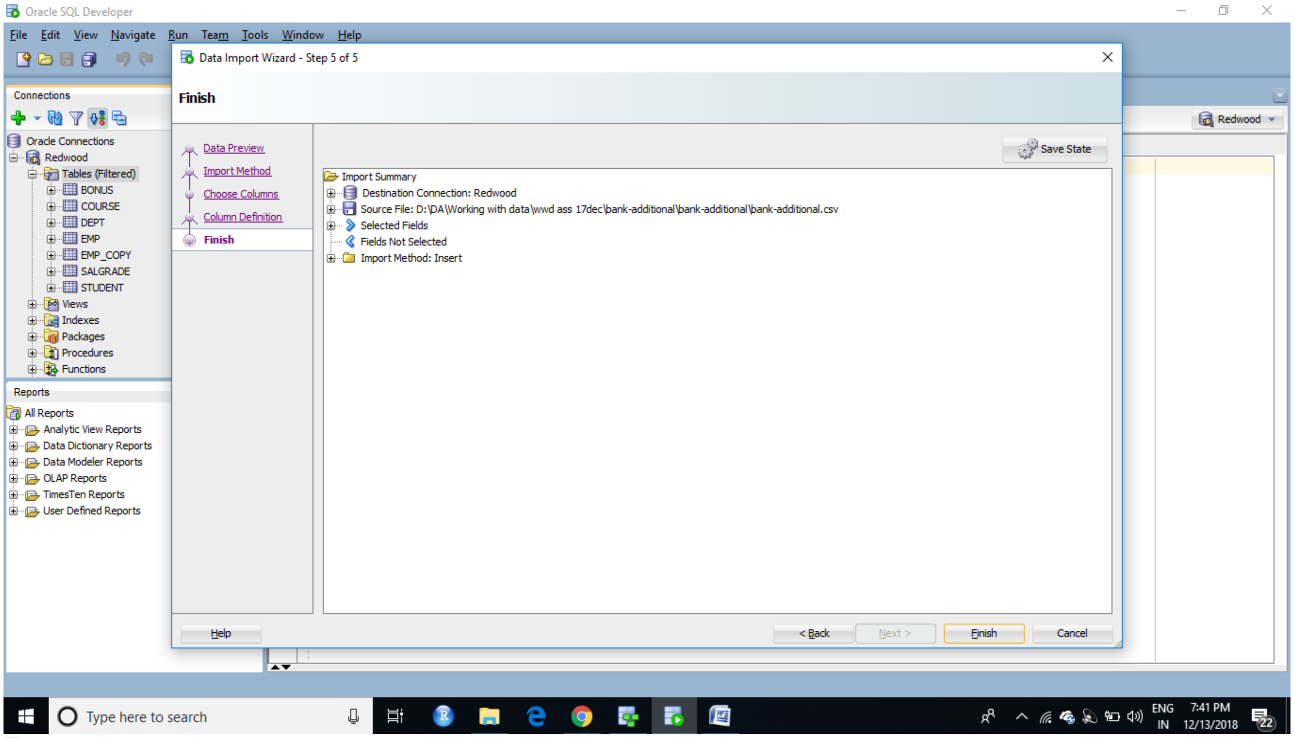
Source Data Columns: age, job, marital, education, default, housing, loan, contact, month, day_of_week, duration, campaign, pdays, previous, poutcome, emp.var.rate, cons.price.idx, cons.conf.idx, euribor3m, nr.employed, y

Target Table Columns
Name: nr_employed
Data Type: NUMBER
Size/Precision: 38
Scale: 1
☑ Nullable? Default:
Comment:

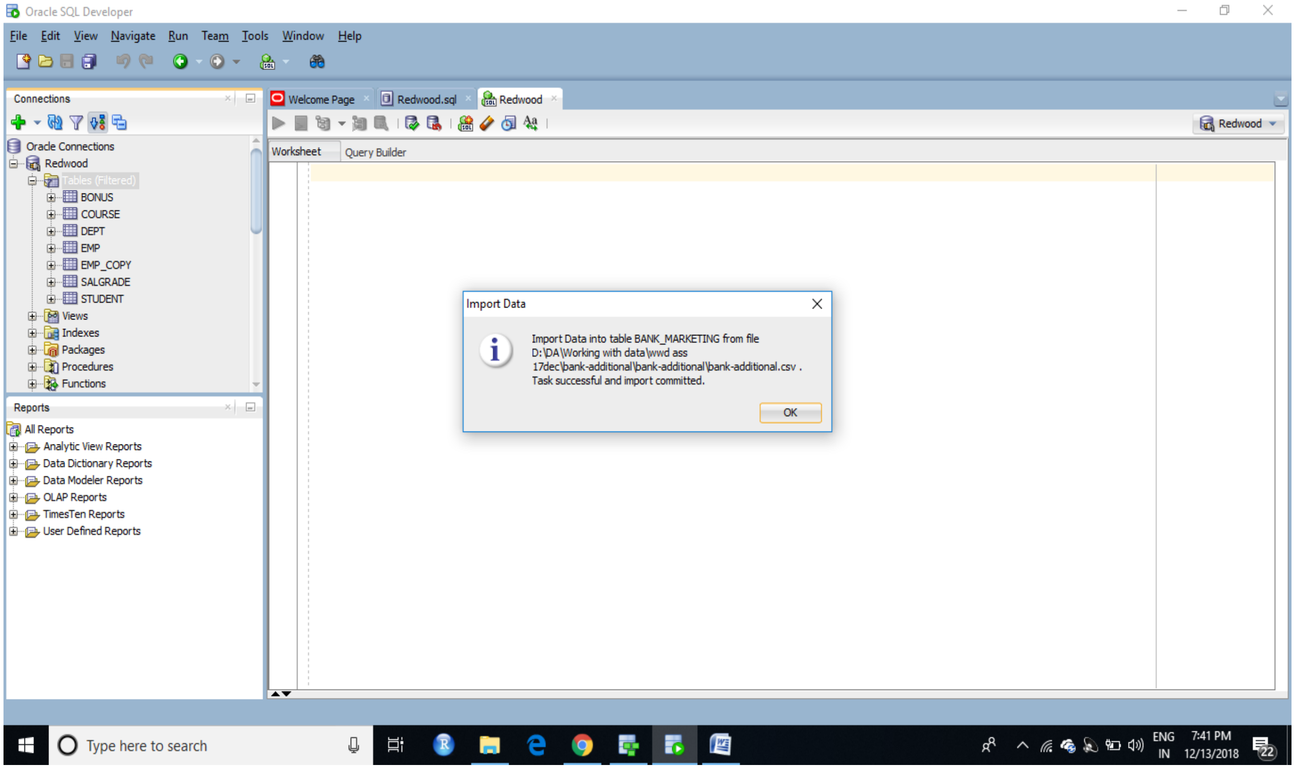
Data
```
5099.1
5191
5228.1
5228.1
5195.8
4963.6
4963.6
5195.8
5195.8
5191
```

Status: Column name is not valid

6. *Click **Finish.** Data imported to table B2.*



Import Summary
- Destination Connection: Redwood
- Source File: D:\DA\Working with data\wwd ass 17dec\bank-additional\bank-additional\bank-additional.csv
- Selected Fields
- Fields Not Selected
- Import Method: Insert

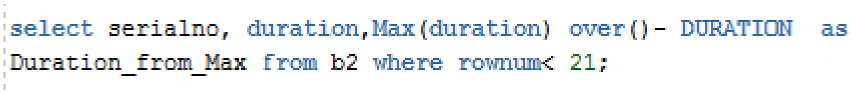


Import Data into table BANK_MARKETING from file D:\DA\Working with data\wwd ass 17dec\bank-additional\bank-additional\bank-additional.csv .
Task successful and import committed.

## 2.1 MAX

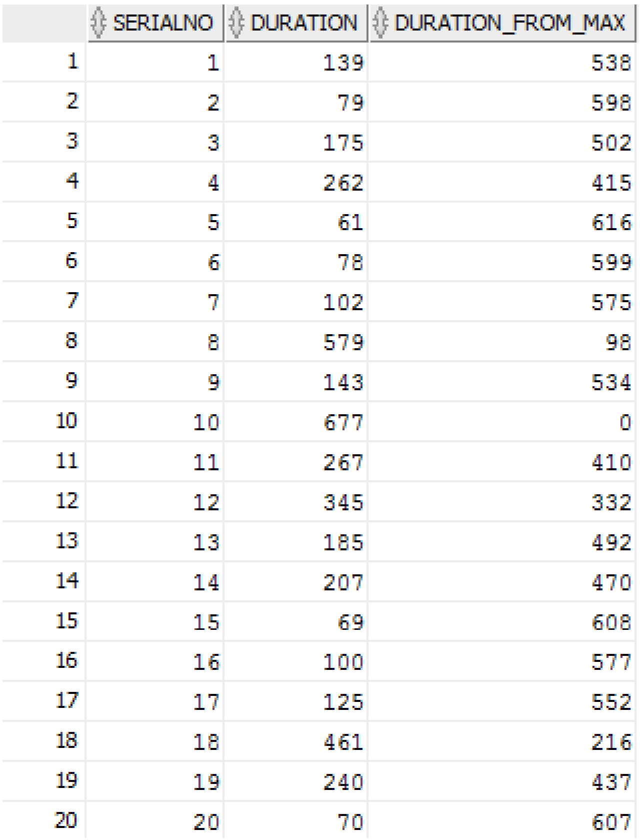
MAX is an analytical function and statistical function; it is used to find the maximum value in the attribute under observation. In the following query, duration from the maximum value in duration is found for the first 20 records. The result observed gives the difference of duration of calls from maximum duration of call with the clients.

CODE:



```
select serialno, duration,Max(duration) over()- DURATION  as
Duration_from_Max from b2 where rownum< 21;
```

OUTPUT:

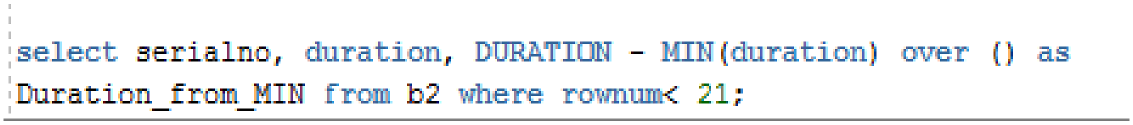


| | SERIALNO | DURATION | DURATION_FROM_MAX |
|---|---|---|---|
| 1 | 1 | 139 | 538 |
| 2 | 2 | 79 | 598 |
| 3 | 3 | 175 | 502 |
| 4 | 4 | 262 | 415 |
| 5 | 5 | 61 | 616 |
| 6 | 6 | 78 | 599 |
| 7 | 7 | 102 | 575 |
| 8 | 8 | 579 | 98 |
| 9 | 9 | 143 | 534 |
| 10 | 10 | 677 | 0 |
| 11 | 11 | 267 | 410 |
| 12 | 12 | 345 | 332 |
| 13 | 13 | 185 | 492 |
| 14 | 14 | 207 | 470 |
| 15 | 15 | 69 | 608 |
| 16 | 16 | 100 | 577 |
| 17 | 17 | 125 | 552 |
| 18 | 18 | 461 | 216 |
| 19 | 19 | 240 | 437 |
| 20 | 20 | 70 | 607 |

## 2.2 MIN

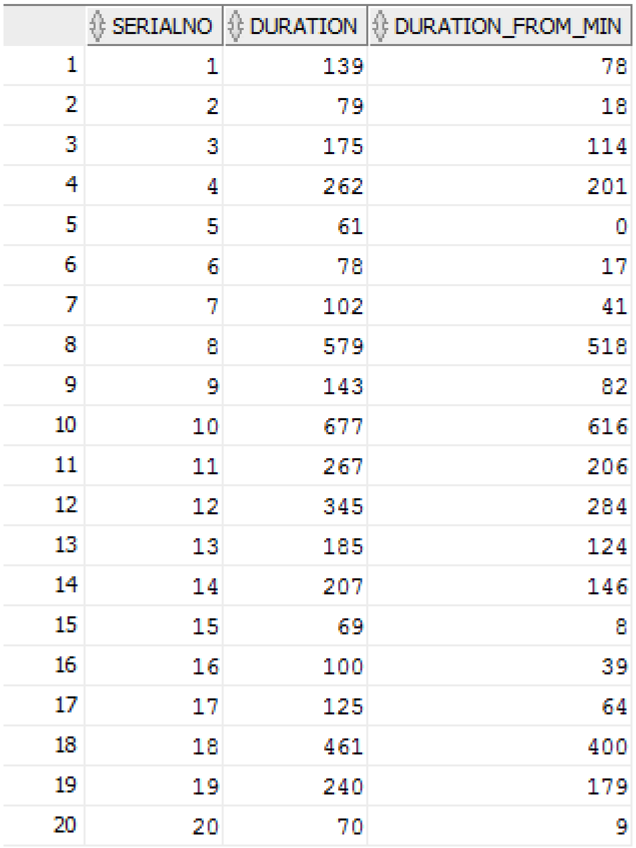
MIN is an analytical function and statistical function; it is used to find the minimum value in the attribute under observation. In the following query, duration from the minimum value in duration is found for the first 20 records. The result observed gives the difference from the minimum duration. The result observed gives the difference of maximum duration of call from duration of calls with the the clients.

*CODE:*



```
select serialno, duration, DURATION - MIN(duration) over () as
Duration_from_MIN from b2 where rownum< 21;
```

*OUTPUT:*

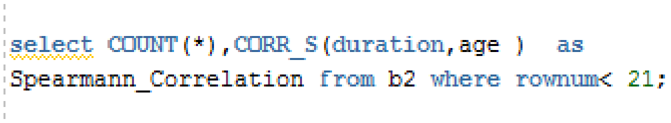


| | SERIALNO | DURATION | DURATION_FROM_MIN |
|---|---|---|---|
| 1 | 1 | 139 | 78 |
| 2 | 2 | 79 | 18 |
| 3 | 3 | 175 | 114 |
| 4 | 4 | 262 | 201 |
| 5 | 5 | 61 | 0 |
| 6 | 6 | 78 | 17 |
| 7 | 7 | 102 | 41 |
| 8 | 8 | 579 | 518 |
| 9 | 9 | 143 | 82 |
| 10 | 10 | 677 | 616 |
| 11 | 11 | 267 | 206 |
| 12 | 12 | 345 | 284 |
| 13 | 13 | 185 | 124 |
| 14 | 14 | 207 | 146 |
| 15 | 15 | 69 | 8 |
| 16 | 16 | 100 | 39 |
| 17 | 17 | 125 | 64 |
| 18 | 18 | 461 | 400 |
| 19 | 19 | 240 | 179 |
| 20 | 20 | 70 | 9 |

## 2.3 SPEARMAN CORRELATION

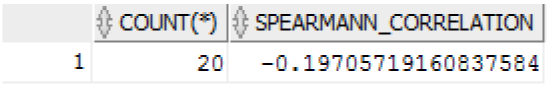
CORR_S calculates the spearman’s correlation coefficient. It’s a statistical function used for calculating correlation between two continuous variable. In the following query correlation between and duration and age has been found which tells a negative correlation is found between them of about 19%.

*CODE:*



```
select COUNT(*),CORR_S(duration,age )  as
Spearmann_Correlation from b2 where rownum< 21;
```

*OUTPUT:*

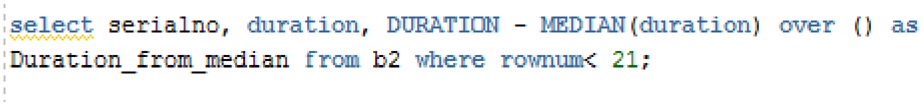


| | COUNT(*) | SPEARMANN_CORRELATION |
|---|---|---|
| 1 | 20 | -0.19705719160837584 |

## 2.4 MEDIAN

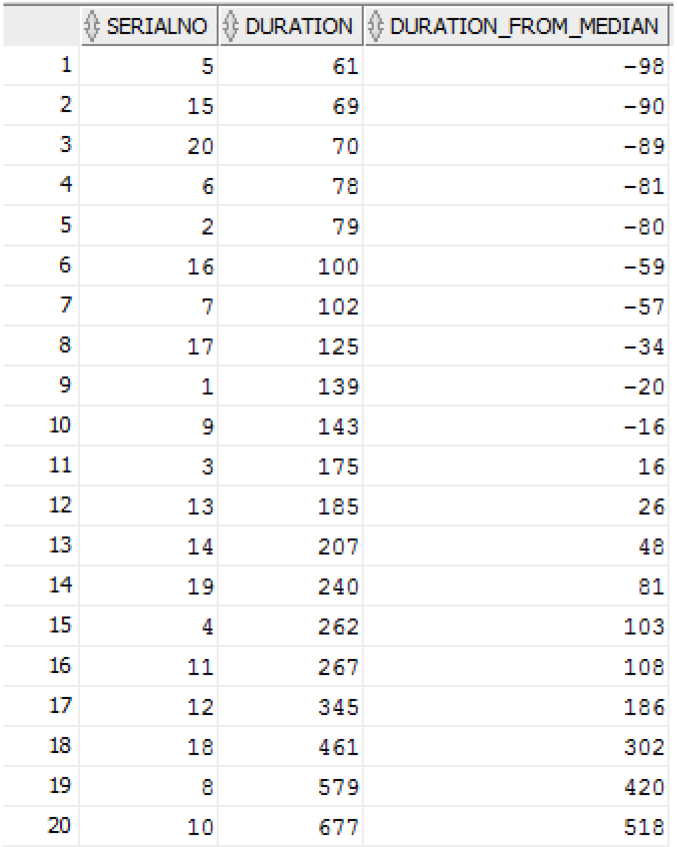
Median is an analytical function whch is used to find the median value in a ordered set of column. In the following query Median of duration is used to find the duration from median, which gives the user a critical analysis of duration using median function.

*CODE:*



```
select serialno, duration, DURATION - MEDIAN(duration) over () as
Duration_from_median from b2 where rownum< 21;
```

*OUTPUT:*

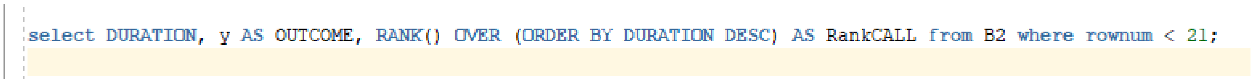


| | SERIALNO | DURATION | DURATION_FROM_MEDIAN |
|---|---|---|---|
| 1 | 5 | 61 | -98 |
| 2 | 15 | 69 | -90 |
| 3 | 20 | 70 | -89 |
| 4 | 6 | 78 | -81 |
| 5 | 2 | 79 | -80 |
| 6 | 16 | 100 | -59 |
| 7 | 7 | 102 | -57 |
| 8 | 17 | 125 | -34 |
| 9 | 1 | 139 | -20 |
| 10 | 9 | 143 | -16 |
| 11 | 3 | 175 | 16 |
| 12 | 13 | 185 | 26 |
| 13 | 14 | 207 | 48 |
| 14 | 19 | 240 | 81 |
| 15 | 4 | 262 | 103 |
| 16 | 11 | 267 | 108 |
| 17 | 12 | 345 | 186 |
| 18 | 18 | 461 | 302 |
| 19 | 8 | 579 | 420 |
| 20 | 10 | 677 | 518 |

## 2.5 RANK

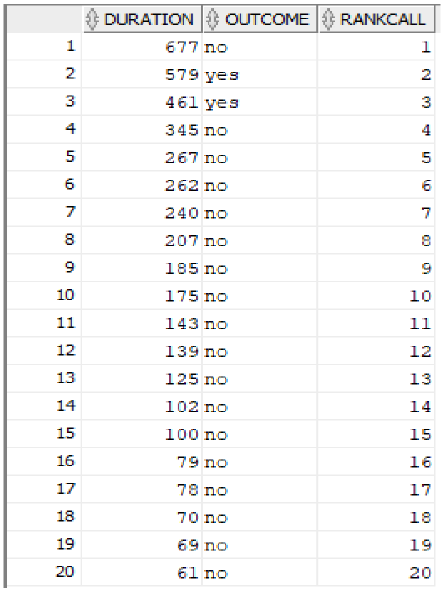
Rank is an analytical function, it computes rank of each row with given ordered specification to process. In the following query, Rank is provided in order of duration of calls to the clients or customers.

*CODE:*



```
select DURATION, y AS OUTCOME, RANK() OVER (ORDER BY DURATION DESC) AS RankCALL from B2 where rownum < 21;
```

*OUTPUT:*

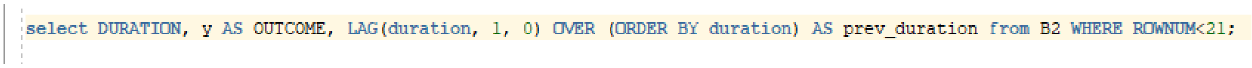


| | DURATION | OUTCOME | RANKCALL |
|---|---|---|---|
| 1 | 677 | no | 1 |
| 2 | 579 | yes | 2 |
| 3 | 461 | yes | 3 |
| 4 | 345 | no | 4 |
| 5 | 267 | no | 5 |
| 6 | 262 | no | 6 |
| 7 | 240 | no | 7 |
| 8 | 207 | no | 8 |
| 9 | 185 | no | 9 |
| 10 | 175 | no | 10 |
| 11 | 143 | no | 11 |
| 12 | 139 | no | 12 |
| 13 | 125 | no | 13 |
| 14 | 102 | no | 14 |
| 15 | 100 | no | 15 |
| 16 | 79 | no | 16 |
| 17 | 78 | no | 17 |
| 18 | 70 | no | 18 |
| 19 | 69 | no | 19 |
| 20 | 61 | no | 20 |

## *2.6 LAG*

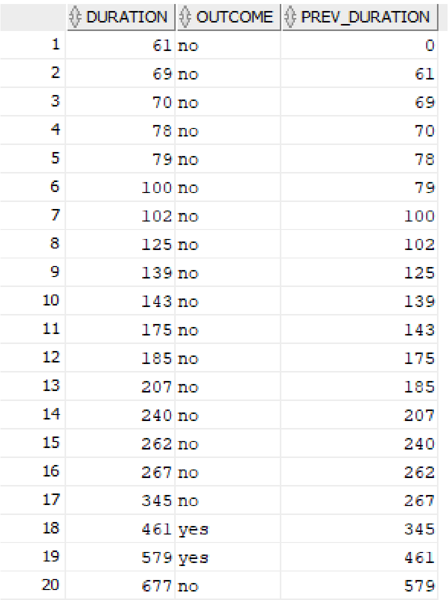
LAG is an analytical function which which provides the previous value in the ordered dataset. In the following query the previous duration of call to client has been provided for subscribing the long term deposit scheme.

*CODE:*



```
select DURATION, y AS OUTCOME, LAG(duration, 1, 0) OVER (ORDER BY duration) AS prev_duration from B2 WHERE ROWNUM<21;
```

*OUTPUT:*

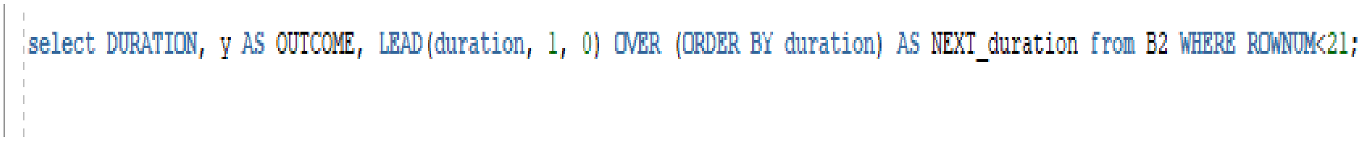


| | DURATION | OUTCOME | PREV_DURATION |
|---|---|---|---|
| 1 | 61 | no | 0 |
| 2 | 69 | no | 61 |
| 3 | 70 | no | 69 |
| 4 | 78 | no | 70 |
| 5 | 79 | no | 78 |
| 6 | 100 | no | 79 |
| 7 | 102 | no | 100 |
| 8 | 125 | no | 102 |
| 9 | 139 | no | 125 |
| 10 | 143 | no | 139 |
| 11 | 175 | no | 143 |
| 12 | 185 | no | 175 |
| 13 | 207 | no | 185 |
| 14 | 240 | no | 207 |
| 15 | 262 | no | 240 |
| 16 | 267 | no | 262 |
| 17 | 345 | no | 267 |
| 18 | 461 | yes | 345 |
| 19 | 579 | yes | 461 |
| 20 | 677 | no | 579 |

## 2.7LEAD

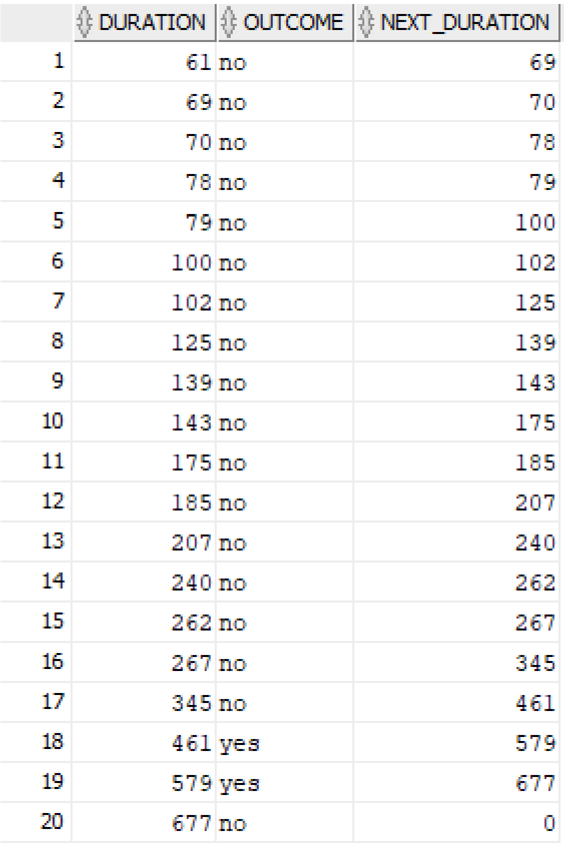
Lead is an analytical function which returns a query for the the next element in the order data where which provides a access to more than one row of a table at the same time without a self join. In the following query or example, the lead variable column has been created for duration which gives the duration of time the client maybe talking to the customer agent in order to subscribe the long term deposit plan.

*CODE:*



```
select DURATION, y AS OUTCOME, LEAD(duration, 1, 0) OVER (ORDER BY duration) AS NEXT_duration from B2 WHERE ROWNUM<21;
```

*OUTPUT:*

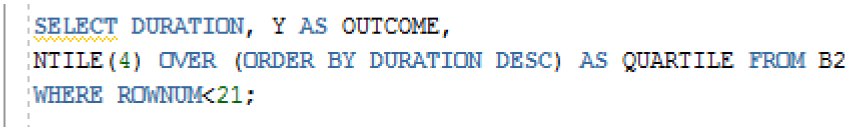


| | DURATION | OUTCOME | NEXT_DURATION |
|---|---|---|---|
| 1 | 61 | no | 69 |
| 2 | 69 | no | 70 |
| 3 | 70 | no | 78 |
| 4 | 78 | no | 79 |
| 5 | 79 | no | 100 |
| 6 | 100 | no | 102 |
| 7 | 102 | no | 125 |
| 8 | 125 | no | 139 |
| 9 | 139 | no | 143 |
| 10 | 143 | no | 175 |
| 11 | 175 | no | 185 |
| 12 | 185 | no | 207 |
| 13 | 207 | no | 240 |
| 14 | 240 | no | 262 |
| 15 | 262 | no | 267 |
| 16 | 267 | no | 345 |
| 17 | 345 | no | 461 |
| 18 | 461 | yes | 579 |
| 19 | 579 | yes | 677 |
| 20 | 677 | no | 0 |

## 2.8 NTILE

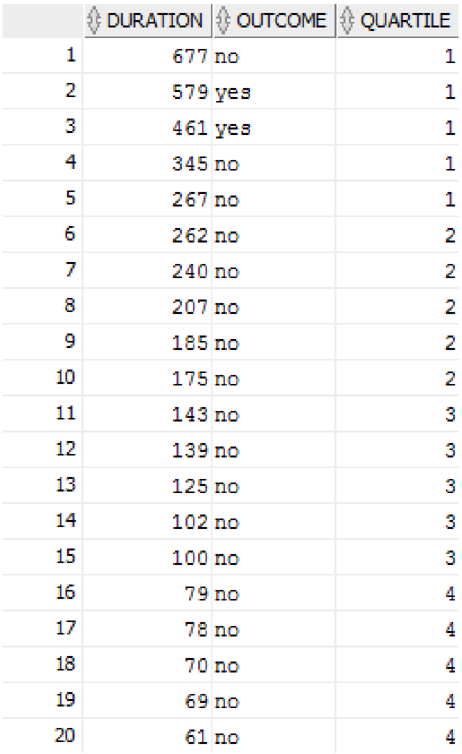
In SQL, NTILE is an analytical function, which divides an ordered dataset into number of buckets. From this example, we have divided the time duration range in calls taken by the client from the dataset, to get an idea how long the calls has been to client to subscribe the long term deposit plan.

*CODE:*



```
SELECT DURATION, Y AS OUTCOME,
NTILE(4) OVER (ORDER BY DURATION DESC) AS QUARTILE FROM B2
WHERE ROWNUM<21;
```

*OUTPUT:*



| | DURATION | OUTCOME | QUARTILE |
|---|---|---|---|
| 1 | 677 | no | 1 |
| 2 | 579 | yes | 1 |
| 3 | 461 | yes | 1 |
| 4 | 345 | no | 1 |
| 5 | 267 | no | 1 |
| 6 | 262 | no | 2 |
| 7 | 240 | no | 2 |
| 8 | 207 | no | 2 |
| 9 | 185 | no | 2 |
| 10 | 175 | no | 2 |
| 11 | 143 | no | 3 |
| 12 | 139 | no | 3 |
| 13 | 125 | no | 3 |
| 14 | 102 | no | 3 |
| 15 | 100 | no | 3 |
| 16 | 79 | no | 4 |
| 17 | 78 | no | 4 |
| 18 | 70 | no | 4 |
| 19 | 69 | no | 4 |
| 20 | 61 | no | 4 |

# 3. Part C (Machine Learning using SQL)

## *MODEL 1 (DECISION TREE)*

### *CREATING TRAIN DATASET*

*NOTE: SAMPLING DONE ON TRAINING DATASET WITH 50-50 YES AND NO OUTCOME VARAIBLE TO REMOVE PREDICTION BIASNESS TO ONLY NO VARAIBLE; AS NO IN OUTCOME VARAIBLES IS HIGHER IN THE DATASET.*

```
create table mining_data_train
as select * from B2
where y='yes';


create table m1
as select * from B2
where y='no';


select COUNT(*) FROM MINING_DATA_TRAIN;


insert into mining_data_train select * from m1
where rownum <= 4640 ;
```

### *CREATING TEST DATASET*

```
create table mining_data_test
```

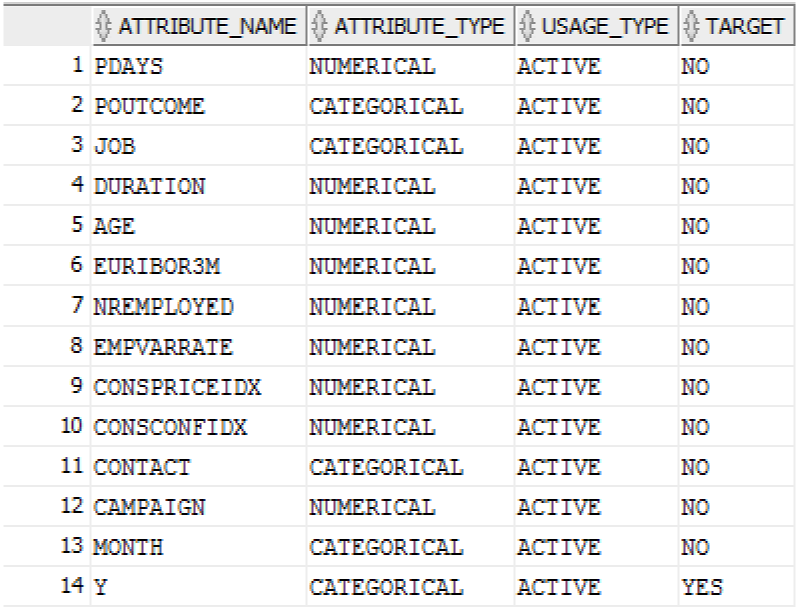
```
as select * FROM B2 where ORA_HASH(SERIALNO, 16, 5)= 0;
```

## Create the settings table

```
CREATE TABLE decision_tree_model_settings (

setting_name VARCHAR2(30),

setting_value VARCHAR2(30));


BEGIN

  INSERT INTO decision_tree_model_settings (setting_name, setting_value)

  VALUES (dbms_data_mining.algo_name,dbms_data_mining.algo_decision_tree);


  INSERT INTO decision_tree_model_settings (setting_name, setting_value)

  VALUES (dbms_data_mining.prep_auto,dbms_data_mining.prep_auto_on);

  COMMIT;

END;


BEGIN

DBMS_DATA_MINING.CREATE_MODEL(

  model_name => 'Decision_Tree_Model1',

  mining_function => dbms_data_mining.classification,

  data_table_name => 'mining_data_train',
  case_id_column_name => 'serialno',

  target_column_name => 'y',

  settings_table_name => 'decision_tree_model_settings');

END;
```

## VARAIBLES IN USAGE FOR DECISION TREE

```
SELECT attribute_name,

   attribute_type,

   usage_type,

   target

from all_mining_model_attributes

where model_name = 'DECISION_TREE_MODEL';
```



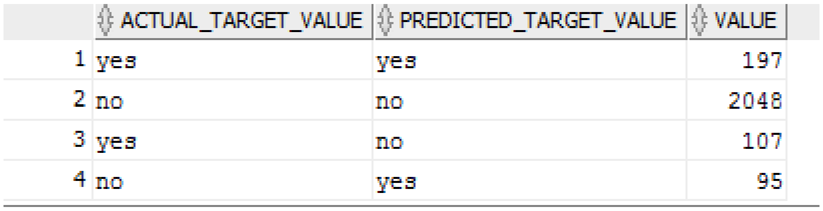
| | ATTRIBUTE_NAME | ATTRIBUTE_TYPE | USAGE_TYPE | TARGET |
|---|---|---|---|---|
| 1 | PDAYS | NUMERICAL | ACTIVE | NO |
| 2 | POUTCOME | CATEGORICAL | ACTIVE | NO |
| 3 | JOB | CATEGORICAL | ACTIVE | NO |
| 4 | DURATION | NUMERICAL | ACTIVE | NO |
| 5 | AGE | NUMERICAL | ACTIVE | NO |
| 6 | EURIBOR3M | NUMERICAL | ACTIVE | NO |
| 7 | NREMPLOYED | NUMERICAL | ACTIVE | NO |
| 8 | EMPVARRATE | NUMERICAL | ACTIVE | NO |
| 9 | CONSPRICEIDX | NUMERICAL | ACTIVE | NO |
| 10 | CONSCONFIDX | NUMERICAL | ACTIVE | NO |
| 11 | CONTACT | CATEGORICAL | ACTIVE | NO |
| 12 | CAMPAIGN | NUMERICAL | ACTIVE | NO |
| 13 | MONTH | CATEGORICAL | ACTIVE | NO |
| 14 | Y | CATEGORICAL | ACTIVE | YES |

*Figure 1: ATTRIBUTES IN DECISION TREE*

## CREATING CONFUSION MATRIX FOR DECISION TREE

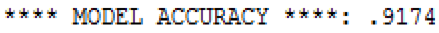
```
DECLARE
   v_accuracy NUMBER;
BEGIN
DBMS_DATA_MINING.COMPUTE_CONFUSION_MATRIX (
   accuracy => v_accuracy,
   apply_result_table_name => 'demo_class_dt_test_results',
   target_table_name => 'mining_data_test',
   case_id_column_name => 'serialno',
   target_column_name => 'y',
   confusion_matrix_table_name => 'demo_class_dt_confusion_matrix',
   score_column_name => 'PREDICTED_VALUE',
   score_criterion_column_name => 'PROBABILITY',
   cost_matrix_table_name => null,
   apply_result_schema_name => null,
   target_schema_name => null,
   cost_matrix_schema_name => null,
   score_criterion_type => 'PROBABILITY');
   DBMS_OUTPUT.PUT_LINE('**** MODEL ACCURACY ****: ' || ROUND(v_accuracy,4));
END;
```

```
SELECT *

FROM demo_class_dt_confusion_matrix;
```



| | ACTUAL_TARGET_VALUE | PREDICTED_TARGET_VALUE | VALUE |
|---|---|---|---|
| 1 | yes | yes | 197 |
| 2 | no | no | 2048 |
| 3 | yes | no | 107 |
| 4 | no | yes | 95 |

*Figure 2 : CONFUSION MATRIX DECISION TREE*



```
**** MODEL ACCURACY ****: .9174
```

## Model  2 – SUPPORT VECTOR MACHINE

### Create the settings table

```
CREATE TABLE SVM_settings (

setting_name VARCHAR2(30),

setting_value VARCHAR2(30));



BEGIN

   INSERT INTO SVM_settings (setting_name, setting_value)

   VALUES
(dbms_data_mining.algo_name,dbms_data_mining.ALGO_SUPPORT_VECTOR
_MACHINES);
```

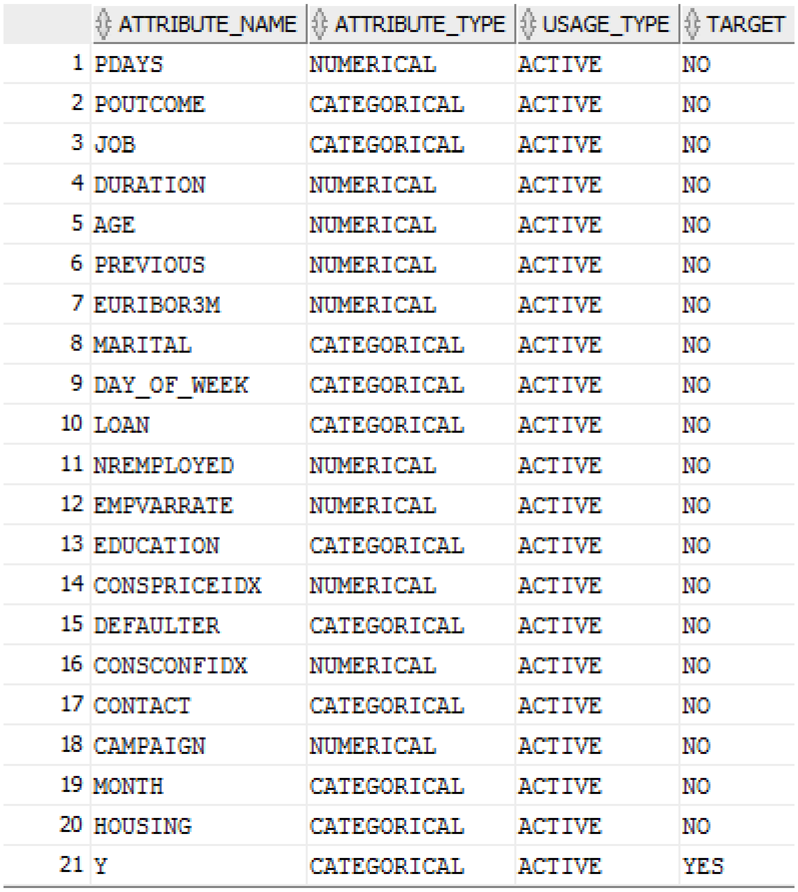
```
  INSERT INTO SVM_settings (setting_name, setting_value)
  VALUES (dbms_data_mining.prep_auto,dbms_data_mining.prep_auto_on);
  COMMIT;
END;
```

## CREATING SVM MODEL

```
BEGIN
DBMS_DATA_MINING.CREATE_MODEL(
  model_name => 'SVM1',
  mining_function => dbms_data_mining.classification,
  data_table_name => 'mining_data_train',
  case_id_column_name => 'serialno',
  target_column_name => 'y',
  settings_table_name => 'SVM_settings');
END;


-- describe the model settings tables
describe user_mining_model_settings
```

## ATTRIBUTE USED FOR SVM

```
SELECT attribute_name,
  attribute_type,
  usage_type,
  target
from all_mining_model_attributes
where model_name = 'SVM1';
```



| | ATTRIBUTE_NAME | ATTRIBUTE_TYPE | USAGE_TYPE | TARGET |
|---|---|---|---|---|
| 1 | PDAYS | NUMERICAL | ACTIVE | NO |
| 2 | POUTCOME | CATEGORICAL | ACTIVE | NO |
| 3 | JOB | CATEGORICAL | ACTIVE | NO |
| 4 | DURATION | NUMERICAL | ACTIVE | NO |
| 5 | AGE | NUMERICAL | ACTIVE | NO |
| 6 | PREVIOUS | NUMERICAL | ACTIVE | NO |
| 7 | EURIBOR3M | NUMERICAL | ACTIVE | NO |
| 8 | MARITAL | CATEGORICAL | ACTIVE | NO |
| 9 | DAY_OF_WEEK | CATEGORICAL | ACTIVE | NO |
| 10 | LOAN | CATEGORICAL | ACTIVE | NO |
| 11 | NREMPLOYED | NUMERICAL | ACTIVE | NO |
| 12 | EMPVARRATE | NUMERICAL | ACTIVE | NO |
| 13 | EDUCATION | CATEGORICAL | ACTIVE | NO |
| 14 | CONSPRICEIDX | NUMERICAL | ACTIVE | NO |
| 15 | DEFAULTER | CATEGORICAL | ACTIVE | NO |
| 16 | CONSCONFIDX | NUMERICAL | ACTIVE | NO |
| 17 | CONTACT | CATEGORICAL | ACTIVE | NO |
| 18 | CAMPAIGN | NUMERICAL | ACTIVE | NO |
| 19 | MONTH | CATEGORICAL | ACTIVE | NO |
| 20 | HOUSING | CATEGORICAL | ACTIVE | NO |
| 21 | Y | CATEGORICAL | ACTIVE | YES |

*Figure 3: VARAIBLE USED FOR SVM*

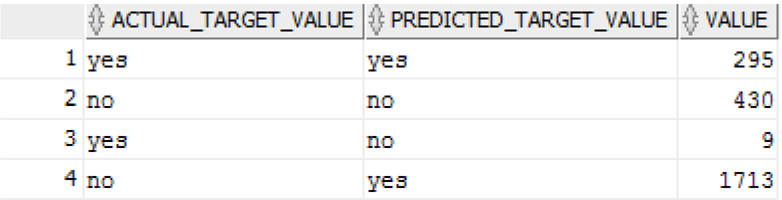
```
CREATE OR REPLACE VIEW demo_class_dt_test_results
AS
SELECT serialno,
  prediction(SVM1 USING *) predicted_value,
  prediction_probability(SVM1 USING *) probability
FROM mining_data_test;
```

```
SELECT * FROM demo_class_dt_test_results;
```

## CONFUSION MATRIX FOR SVM

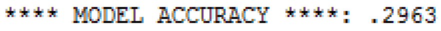
```
DECLARE
  v_accuracy NUMBER;
BEGIN
DBMS_DATA_MINING.COMPUTE_CONFUSION_MATRIX (
  accuracy => v_accuracy,
  apply_result_table_name => 'demo_class_dt_test_results',
  target_table_name => 'mining_data_test',
  case_id_column_name => 'serialno',
  target_column_name => 'y',
  confusion_matrix_table_name => 'SVM_confusion_matrix',
  score_column_name => 'PREDICTED_VALUE',
  score_criterion_column_name => 'PROBABILITY',
  cost_matrix_table_name => null,
  apply_result_schema_name => null,
  target_schema_name => null,
  cost_matrix_schema_name => null,
  score_criterion_type => 'PROBABILITY');
  DBMS_OUTPUT.PUT_LINE('**** MODEL ACCURACY ****: ' || ROUND(v_accuracy,4));
END;
```

```
SELECT * FROM SVM_confusion_matrix;
```



|  | ACTUAL_TARGET_VALUE | PREDICTED_TARGET_VALUE | VALUE |
|---|---|---|---|
| 1 | yes | yes | 295 |
| 2 | no | no | 430 |
| 3 | yes | no | 9 |
| 4 | no | yes | 1713 |

*Figure 4: SVM CONFUSION MATRIX*



```
**** MODEL ACCURACY ****: .2963
```

## Model 3 LINEAR REGRESSION

### Create the settings table

```
CREATE TABLE LINEARMODEL_settings (

setting_name VARCHAR2(30),

setting_value VARCHAR2(30));




BEGIN

  INSERT INTO LINEARMODEL_settings (setting_name, setting_value)

  VALUES
(dbms_data_mining.algo_name,dbms_data_mining.ALGO_GENERALIZED_LINEAR_MODEL);


  INSERT INTO LINEARMODEL_settings (setting_name, setting_value)

  VALUES (dbms_data_mining.prep_auto,dbms_data_mining.prep_auto_on);
  COMMIT;

END;
```

## BUILDING LINEAR REGRESSION MODEL

```
BEGIN

DBMS_DATA_MINING.CREATE_MODEL(

  model_name => 'LINEARMODEL',

  mining_function => dbms_data_mining.classification,

  data_table_name => 'mining_data_train',

  case_id_column_name => 'serialno',

  target_column_name => 'y',

  settings_table_name => 'LINEARMODEL_settings');

END;


-- describe the model settings tables

describe user_mining_model_settings
```

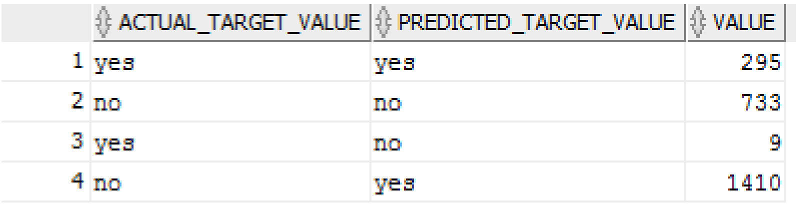
## ATTRIBUT USAGE IN LINEAR REGRESSION MODEL

```
SELECT attribute_name,

  attribute_type,

  usage_type,

  target

from all_mining_model_attributes
```

```
where model_name = 'LINEARMODEL';


CREATE OR REPLACE VIEW demo_class_dt_test_results

AS

SELECT serialno,

  prediction(LINEARMODEL USING *) predicted_value,

  prediction_probability(LINEARMODEL USING *) probability

FROM mining_data_test;




SELECT * FROM demo_class_dt_test_results;
```

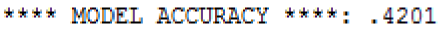
## CONFUSION MATRIX FOR LINEAR REGRESSION

```
DECLARE

  v_accuracy NUMBER;

BEGIN

DBMS_DATA_MINING.COMPUTE_CONFUSION_MATRIX (

  accuracy => v_accuracy,

  apply_result_table_name => 'demo_class_dt_test_results',

  target_table_name => 'mining_data_test',

  case_id_column_name => 'serialno',

  target_column_name => 'y',
  confusion_matrix_table_name => 'LINEARMODEL_confusion_matrix',

  score_column_name => 'PREDICTED_VALUE',

  score_criterion_column_name => 'PROBABILITY',

  cost_matrix_table_name => null,

  apply_result_schema_name => null,

  target_schema_name => null,

  cost_matrix_schema_name => null,

  score_criterion_type => 'PROBABILITY');

  DBMS_OUTPUT.PUT_LINE('**** MODEL ACCURACY ****: ' ||
ROUND(v_accuracy,4));

END;


SELECT * FROM LINEARMODEL_confusion_matrix;
```



| | ACTUAL_TARGET_VALUE | PREDICTED_TARGET_VALUE | VALUE |
|---|---|---|---|
| 1 | yes | yes | 295 |
| 2 | no | no | 733 |
| 3 | yes | no | 9 |
| 4 | no | yes | 1410 |

*Figure 5 CONFUSION MATRIX LINEAR REGRESSION*

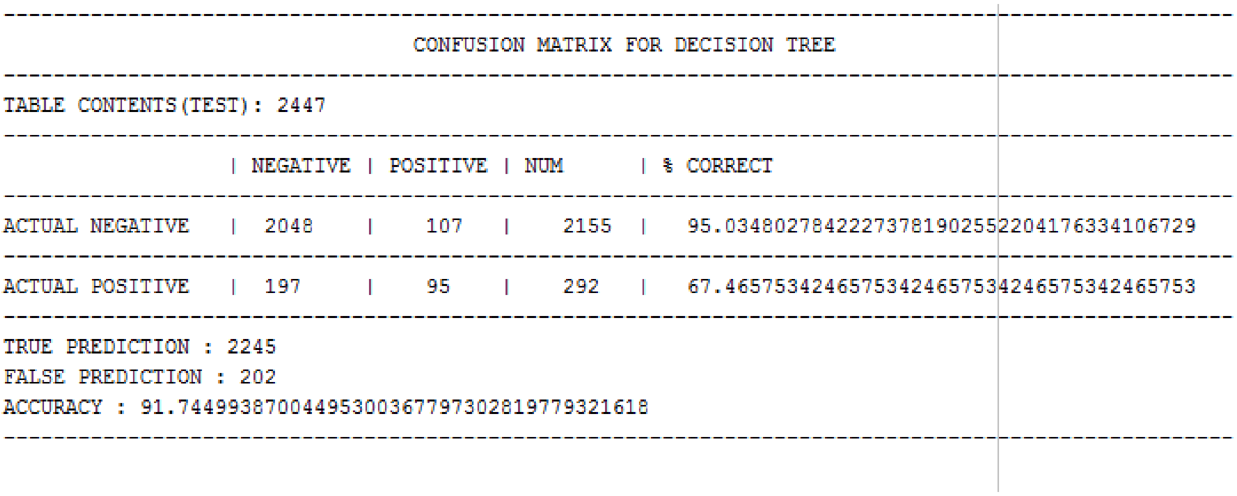


```
**** MODEL ACCURACY ****: .4201
```

## RESULTS AND OBSERVATION

The decision tree performed the best over all other algorithms, it is having an accuracy of 91.7%. Linear regression performed with an accuracy of 42.01%. SVM has performed with an accuracy of 29.63%.

## 4. Part D ( PL/SQL Code – CONFUSION MATRIX)



```
----------------------------------------------------------------------------------------------------
                                 CONFUSION MATRIX FOR DECISION TREE
----------------------------------------------------------------------------------------------------
TABLE CONTENTS(TEST): 2447
----------------------------------------------------------------------------------------------------
                  | NEGATIVE | POSITIVE | NUM      | % CORRECT
----------------------------------------------------------------------------------------------------
ACTUAL NEGATIVE   |  2048    |    107   |    2155  |   95.03480278422273781902552204176334106729
----------------------------------------------------------------------------------------------------
ACTUAL POSITIVE   |  197     |    95    |    292   |   67.46575342465753424657534246575342465753
----------------------------------------------------------------------------------------------------
TRUE PREDICTION : 2245
FALSE PREDICTION : 202
ACCURACY : 91.74499387004495300367797302819779321618
----------------------------------------------------------------------------------------------------
```

*Figure 6 Confusion matrix for decision tree*

We have calculated the confusion matrix of the decision tree model. The result generated is similar to the result generated in part C of assignment.

## APPENDIX FOR CONFUSION MATRIX

```
SELECT * FROM DEMO_CLASS_DT_TEST_RESULTS;


CREATE TABLE CONF_TRAIN AS SELECT
Y,PREDICTED_VALUE,D.SERIALNO FROM demo_class_dt_test_results D

INNER JOIN  MINING_DATA_TEST M ON M.SERIALNO = d.SERIALNO;
```

```
SELECT COUNT(*) FROM MINING_DATA_TEST;


SELECT COUNT(*) FROM CONF_TRAIN;


SELECT COUNT(*) FROM DEMO_CLASS_DT_TEST_RESULTS;


DECLARE
TN NUMBER;
TP NUMBER;
FN NUMBER;
FP NUMBER;
ACCURACY NUMBER;
N NUMBER;
N1 NUMBER;
N2 NUMBER;
CP1 NUMBER;
CP2 NUMBER;
TPP NUMBER;
FPP NUMBER;
BEGIN
select count(*) into   TP from   CONF_TRAIN WHERE PREDICTED_VALUE = 'yes' AND Y = 'yes';
select count(*) into   TN from   CONF_TRAIN WHERE PREDICTED_VALUE =
'no' AND Y = 'no';

select count(*) into   FP from   CONF_TRAIN WHERE PREDICTED_VALUE =
'yes' AND Y = 'no';

select count(*) into   FN from   CONF_TRAIN WHERE PREDICTED_VALUE =
'no' AND Y = 'yes';

SELECT COUNT(*) INTO N FROM MINING_DATA_TEST;

N1:=TN+FN;

N2:=TP+FP;

CP1:=TN/N1*100;

CP2:=TP/N2*100;

ACCURACY := (TP+TN)/(TP+TN+FP+FN)*100;

TPP:= TN+TP;

FPP:= FN+FP;

dbms_output.put_line('----------------------------------------------------------------------------
-----------------------------');

dbms_output.put_line('                              CONFUSION MATRIX FOR
DECISION TREE');

dbms_output.put_line('----------------------------------------------------------------------------
-----------------------------');

dbms_output.put_line('TABLE CONTENTS(TEST): '||N);

dbms_output.put_line('----------------------------------------------------------------------------
-----------------------------');
```

```
dbms_output.put_line('                  | NEGATIVE | POSITIVE | NUM      | % CORRECT');

dbms_output.put_line('-----------------------------------------------------------------------------------------------');

dbms_output.put_line('ACTUAL NEGATIVE  |  '||TN||'   |    '||FN||'   |    '||N1||'  | '||CP1 );

dbms_output.put_line('-----------------------------------------------------------------------------------------------');

dbms_output.put_line('ACTUAL POSITIVE  |  '||TP||'    |    '||FP||'    |    '||N2||'   | '||CP2 );

dbms_output.put_line('-----------------------------------------------------------------------------------------------');



dbms_output.put_line('TRUE PREDICTION : '||TPP);



dbms_output.put_line('FALSE PREDICTION : '||FPP);



dbms_output.put_line('ACCURACY : '||ACCURACY);

dbms_output.put_line('-----------------------------------------------------------------------------------------------');



END;
```