IT 440

FINAL PORTFOLIO

* BY PULKIT AGARWAL

CONTENTS

1. INTRODUCTION
2. SALES DATABASE QUERIES
3. IN CLASS QUERIES ON OWN DATABASE
4. PL/SQL QUERIES ON OWN DATABASE
5. PROCEDURES ON OWN DATABASE

Introduction

This portfolio has been compiled by me using the in-class assignments and homework. Most of the work has been done on my own database with the exception of one assignment where I used the sales database. My portfolio folder also contains the following:

* Excel Worksheet of my database
* SQL Files of all the queries, procedures and functions
* Data model of the database
* Word file of portfolio
* PDF of portfolio

The questions or the concerned queries have been commented.

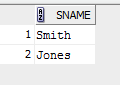
ASSIGNMENT 1 : 30 QUERIES ON SALES DATABASE WITH RESULTS

SELECT SNAME //Supplier name supplying maximum quantity of parts

FROM S

WHERE S# IN (SELECT S# FROM SP

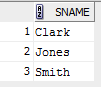
WHERE QTY =(SELECT MAX(QTY) FROM SP));



SELECT DISTINCT SNAME // supplier whose supplies red parts and whose

FROM S,SP,P // weight is greater than 10

WHERE S.S#=SP.S# AND SP.P#=P.P# AND P.COLOR='Red' and weight>10;

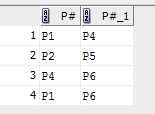


SELECT P1.P#, P2.P# // parts from the same city with no duplicates

FROM P P1, P P2

WHERE P1.P# < P2.P#

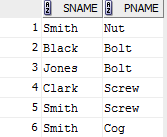
AND P1.CITY=P2.CITY;



SELECT SNAME,PNAME // supplier and part from the same city

FROM S,SP,P

WHERE S.S#=SP.S# AND SP.P#=P.P# AND P.CITY = S.CITY;



SELECT PNAME // part name that has min total quantity shipped

FROM P

WHERE P# IN (SELECT P# FROM SP

WHERE QTY =(SELECT MIN(QTY) FROM SP));

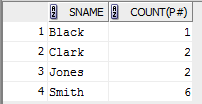


SELECT SNAME,COUNT(P#) // parts shipped by each supplier

FROM S,SP

WHERE S.S#=SP.S#

GROUP BY SNAME;



SELECT PNAME,SUM(QTY) // total quantity of all blue parts

FROM P,SP

WHERE P.P#=SP.P# AND P.COLOR='Blue'

group by PNAME

ORDER BY PNAME;



SELECT S.CITY, S.SNAME // suppliers not from the city from which smith is.

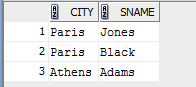
FROM S

WHERE S.CITY NOT IN

(SELECT CITY

FROM S

WHERE SNAME LIKE 'Smith');



SELECT \* FROM P // parts that are not from London and whose color is not blue

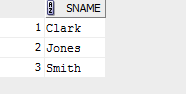
WHERE CITY <>'London' AND COLOR <>'Blue';



SELECT DISTINCT SNAME // name of the suppliers that supply a red part.

FROM S,SP,P

WHERE S.S#=SP.S# AND SP.P#=P.P# AND P.COLOR='Red';



SELECT P.PNAME // name of parts from supplier from Paris with qty>100 and

FROM P,SP,S // part is from Paris

WHERE S.S#=SP.S# AND SP.P#=P.P# AND S.CITY='Paris'

AND SP.qty > 100 AND P.CITY='Paris';



SELECT SUM(WEIGHT) // total weight of all the red parts

FROM P

WHERE COLOR='Red';



SELECT DISTINCT S1.SNAME,S2.SNAME

FROM S S1,S S2 // names of the suppliers that are from the same city

WHERE S1.S# < S2.S#

AND S1.CITY=S2.CITY;



SELECT SNAME

FROM S // supplier name that supplies the minimum quantity of parts

WHERE S# IN (SELECT S# FROM SP

WHERE QTY =(SELECT MIN(QTY) FROM SP));

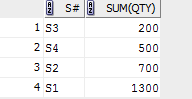


SELECT S#,SUM(QTY) // total quantity shipped for each part

FROM P,SP

WHERE SP.P#=P.P#

GROUP BY S#;

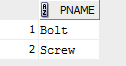


SELECT PNAME // part name that has the maximum quantity

FROM P

WHERE P# IN (SELECT P# FROM SP

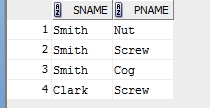
WHERE QTY =(SELECT MAX(QTY) FROM SP));



SELECT SNAME,PNAME // suppliers and parts that come from the London

FROM S,P,SP

WHERE S.S#=SP.S# AND SP.P#=P.P# AND P.CITY='London' AND S.CITY='London';

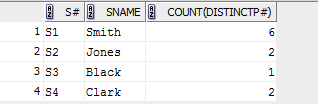


SELECT S.S#,SNAME,COUNT(distinct P#)

FROM S,SP // max number of unique parts supplied by every supplier

WHERE S.S#=SP.S#

GROUP BY S.S#,SNAME;



// name of the supplier that supplies both red part and weight is 14.

SELECT SNAME

FROM S,P,SP

WHERE S.S#=SP.S# AND SP.P#=P.P# AND WEIGHT=14 AND P.COLOR='Red';



// Top 2 and bottom 2 parts based on their quantity

SELECT 'TOP LIST',P#,QTY, RANKS

FROM(SELECT P#, SUM( QTY) AS QTY, ROW\_NUMBER () OVER ( ORDER BY SUM(QTY) DESC ) AS RANKS

FROM SP

GROUP BY P#)

WHERE RANKS < 3

UNION

SELECT 'BOTTOM',P#,QTY, RANKS

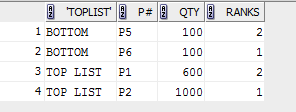
FROM(

SELECT P#, SUM( QTY) AS QTY, ROW\_NUMBER () OVER ( ORDER BY SUM(QTY) ASC ) AS RANKS

FROM SP

GROUP BY P#)

WHERE RANKS < 3;



// Names of the suppliers who supplies the most parts and least parts

SELECT 'Top\_supplier', S#

FROM(

SELECT S#, COUNT( P#) , ROW\_NUMBER() OVER ( ORDER BY COUNT( P#) DESC) AS RANKS

FROM SP

GROUP BY S# )

WHERE RANKS < 2

UNION

SELECT 'Least\_Supplier',S#

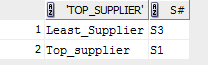
FROM(

SELECT S#, COUNT( P#) , ROW\_NUMBER() OVER ( ORDER BY COUNT( P#) ASC) AS RANKS

FROM SP

GROUP BY S# )

WHERE RANKS < 2;

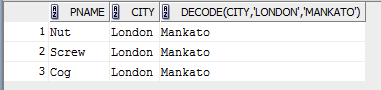


// Names of the suppliers who supplies the most parts and least parts

SELECT PNAME,city,DECODE(CITY,'London','Mankato')

FROM P

where city ='London';



// Top and bottom suppliers based on the quantity they supply.

SELECT 'Top\_supplier', qty

FROM(

SELECT S#, SUM(QTY) AS qty, ROW\_NUMBER() OVER ( ORDER BY S# ASC) AS RANKS

FROM SP

GROUP BY S# )

WHERE RANKS < 2

UNION

SELECT 'Least\_supplier',qty

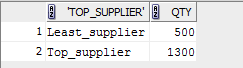
FROM(

SELECT S#, SUM(QTY) AS qty, ROW\_NUMBER() OVER ( ORDER BY S# DESC) AS RANKS

FROM SP

GROUP BY S# )

WHERE RANKS < 2



// Give the top 3 parts that weigh the most.

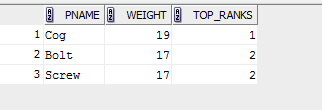
SELECT PNAME,WEIGHT,Top\_Ranks

FROM (SELECT PNAME,WEIGHT,

RANK() OVER (ORDER BY WEIGHT DESC) AS Top\_Ranks

FROM P)

where Top\_Ranks <=3;

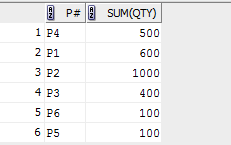


// Give the total qty for all the parts

SELECT P#,SUM(QTY)

FROM SP

GROUP BY P#;

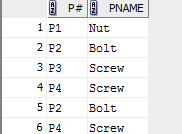


//Give the name of parts from supplier from London with qty>100 and part is from London

SELECT P.P#,PNAME

FROM SP,S,P

WHERE S.S#=SP.S# AND P.P#=SP.P# AND S.CITY='London' and qty>100;



// Rank suppliers on the total quantity and number of parts

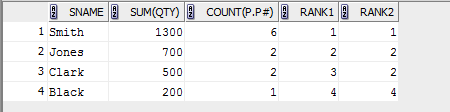
SELECT SNAME,SUM(QTY),COUNT(P.P#),RANK() OVER ( ORDER BY sum(QTY) DESC) AS RANK1,

RANK() OVER ( ORDER BY COUNT(P.P#) DESC) AS RANK2

FROM S,P,SP

WHERE S.S#=SP.S# AND P.P#=SP.P#

GROUP BY SNAME;



// Give the subtotals of the total part weights of the all the parts supplied by a supplier and give the final total weight.

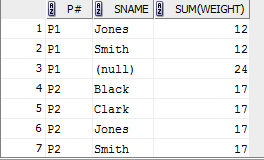
SELECT P.P#,SNAME,SUM(WEIGHT)

FROM P,SP,S

WHERE S.S#=SP.S# AND P.P#=SP.P#

GROUP BY ROLLUP(P.P#,SNAME)

ORDER BY P.P#;



ASSIGNMENT 2: IN CLASS QUERIES ADJUSTED ON OWN DATABASE

// Count employees in the same month

SELECT COUNT(\*),MONT

from (select ENO, EMP\_FNAME, EMP\_LNAME, EMP\_DOB

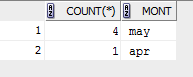
,to\_char(EMP\_DOB, 'DD-MON-YY') as EMP\_DOB\_CHAR

,to\_char(EMP\_DOB, '"Birth year is "YY') as Birth\_Year

,to\_char(EMP\_DOB, ' mon') as MONT

from employee )

group by MONT;



// Second method for the same

SELECT COUNT(\*),MONTH

from (select ENO, EMP\_FNAME, EMP\_LNAME, EMP\_DOB

,to\_char(EMP\_DOB, 'DD-MON-YY') as EMP\_DOB\_CHAR

,to\_char(EMP\_DOB, '"Birth year is "YY') as Birth\_Year

,SUBSTR(EMP\_DOB,4,3) MONTH

from employee )

group by MONTH

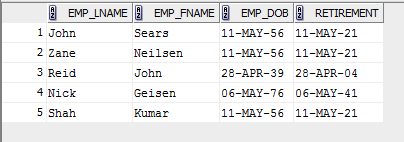


// Find retirement date of an employee

select EMP\_LNAME, EMP\_FNAME, EMP\_DOB

,add\_months(EMP\_DOB, 12\*65) as RETIREMENT

FROM EMPLOYEE;

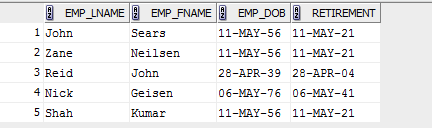


// Second method for the same

select EMP\_LNAME, EMP\_FNAME, EMP\_DOB

,add\_months(EMP\_DOB, 12\*65) as RETIREMENT

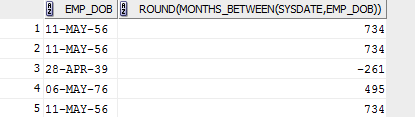
FROM EMPLOYEE;



// current age of employee in months

select EMP\_DOB ,ROUND(MONTHS\_BETWEEN(SYSDATE,EMP\_DOB))

from EMPLOYEE;



// Order employee based on their day of the week of the their birth day

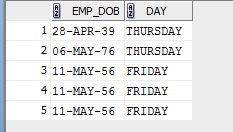
SELECT EMP\_DOB,TO\_CHAR(EMP\_DOB,'DAY') AS DAY

FROM EMPLOYEE

ORDER BY DECODE(RTRIM(TO\_CHAR(EMP\_DOB,'DAY'),' '),'SUNDAY',1, 'MONDAY',2,

'TUESDAY',3,'WEDNSDAY',4,

'THURSDAY',5,'FRIDAY' ,6,'SATURDAY',7);



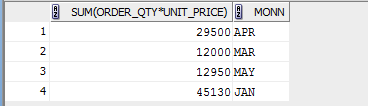
// total sales by month

select sum(ORDER\_QTY\*UNIT\_PRICE),substr(to\_char(CUST\_order\_date),4,3) as monn

from orders o ,PRODUCT l

where o.PNO=l.PNO

group by substr(to\_char(CUST\_order\_date),4,3);



// Total sales for employee

USING RLL UP

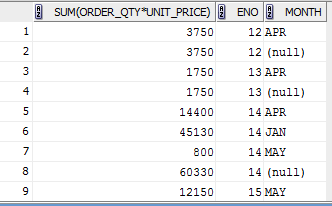
select sum(ORDER\_QTY\*UNIT\_PRICE),ENO,substr(to\_char(CUST\_ORDER\_DATE),4,3) MONTH

from orders o ,PRODUCT l

where o.PNO=l.PNO

group by ROLLUP(ENO,substr(to\_char(CUST\_ORDER\_DATE),4,3))

ORDER BY ENO,substr(to\_char(CUST\_ORDER\_DATE),4,3);



// Total sales for employee using cube method

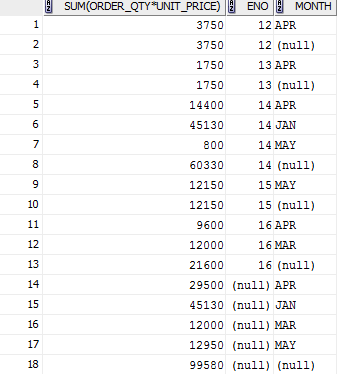
select sum(ORDER\_QTY\*UNIT\_PRICE),ENO,substr(to\_char(CUST\_ORDER\_DATE),4,3) MONTH

from orders o ,PRODUCT l

where o.PNO=l.PNO

group by CUBE(ENO,substr(to\_char(CUST\_ORDER\_DATE),4,3))

ORDER BY ENO,substr(to\_char(CUST\_ORDER\_DATE),4,3);



// Total sales for employee using group by method

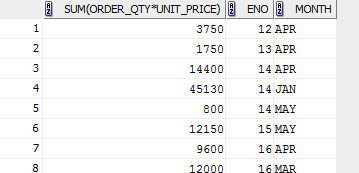
select sum(ORDER\_QTY\*UNIT\_PRICE),ENO,substr(to\_char(CUST\_ORDER\_DATE),4,3) MONTH

from orders o ,PRODUCT l

where o.PNO=l.PNO

group by ENO,substr(to\_char(CUST\_ORDER\_DATE),4,3)

ORDER BY ENO,substr(to\_char(CUST\_ORDER\_DATE),4,3);



// find employee ranked 2

EMPLOYE RANK 2

SELECT RANK ,EMP\_FNAME FROM (SELECT EMP\_FNAME, EMP\_LNAME , SALARY,

RANK() OVER ( ORDER BY SALARY Desc) AS RANK

FROM EMPLOYEE)

WHERE RANK =2;



// Using union to find both bottom and top at a time

SELECT EMP\_FNAME, EMP\_LNAME ,SALARY, Bottom\_Ranks

FROM (SELECT EMP\_FNAME, EMP\_LNAME ,SALARY,

RANK() OVER ( ORDER BY SALARY ASC) AS Bottom\_Ranks

FROM EMPLOYEE)

where Bottom\_Ranks <=3;

UNION

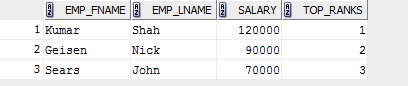
SELECT EMP\_FNAME, EMP\_LNAME, SALARY, Top\_Ranks

FROM (SELECT EMP\_FNAME, EMP\_LNAME ,SALARY,

DENSE\_RANK() OVER (ORDER BY SALARY DESC) AS Top\_Ranks

FROM EMPLOYEE)

where Top\_Ranks <=3;

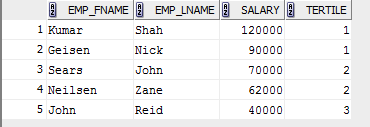


// finding ntile 90

SELECT EMP\_FNAME, EMP\_LNAME , SALARY AS SALARY,

NTILE(3) OVER ( ORDER BY SALARY DESC NULLS LAST) Tertile

FROM EMPLOYEE S;



// TOTAL SALES FOR EVERY MONTH USING PERCENTAGE TO RATTIO

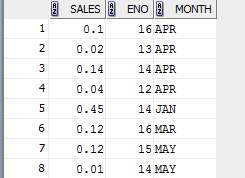
select ROUND(RATIO\_TO\_REPORT(sum(ORDER\_QTY\*UNIT\_PRICE)) OVER(),2) AS SALES,ENO,substr(to\_char(CUST\_ORDER\_DATE),4,3) MONTH

from orders o ,PRODUCT l

where o.PNO=l.PNO

group by ENO,substr(to\_char(CUST\_ORDER\_DATE),4,3)

ORDER BY 3,substr(to\_char(CUST\_ORDER\_DATE),4,3);



ASSIGNMENT 3: PL/SQL QUERIES ON OWN DATABASE

// PL/SQL hello world program

set serveroutput on;

DECLARE

BEGIN

DBMS\_OUTPUT.PUT\_LINE(' \*\*\*\*\* Hello World \*\*\*\*\*');

DBMS\_OUTPUT.PUT\_LINE(' SUMMER CLASS IN ONE WEEK IS WONDERFUL AND UNBLIEBLE');

END;



// simple employee count program

SET SERVEROUTPUT ON;

declare

var1 varchar(24);

begin

select 'Number of Employees: ' || count(\*)

into var1

from employee;

dbms\_output.put\_line (var1);

exception

when others then dbms\_output.put\_line('ERROR OCCURRED');

end;



// Employee count program with error handling

SET SERVEROUTPUT ON;

declare

var1 varchar(24);

begin

select 'Number of Employees: ' || count(\*)

into var1

from employee;

dbms\_output.put\_line (var1);

exception

when others then dbms\_output.put\_line('ERROR OCCURRED');

end;

/



// Finding retirement date using PL/SQL

SET SERVEROUTPUT ON;

DECLARE

retirement\_date date;

emp\_var employee%rowtype;

BEGIN

SELECT min(EMP\_DOB)

INTO emp\_var.EMP\_DOB

FROM employee;

retirement\_date := add\_months(emp\_var.EMP\_DOB, 12\*65);

dbms\_output.put\_line ('Retirement date is '|| to\_char(retirement\_date,'dd/mm/yyyy'));

END;

/



// Current month program

set serveroutput on;

declare

current\_month char(3);

begin

select to\_char(sysdate, 'MON') into current\_month from dual;

if current\_month = 'JAN' then

dbms\_output.put\_line ('My daughter Jane was born in Jan');

elsif current\_month = 'FEB' then

dbms\_output.put\_line ('My good friend Ron was born in Feb');

elsif current\_month = 'MAR' then

dbms\_output.put\_line ('My father was born in March');

elsif current\_month = 'APR' then

dbms\_output.put\_line ('I was born in April');

elsif current\_month = 'MAY' then

dbms\_output.put\_line ('My son Matt was born in May');

elsif current\_month = 'OCT' then

dbms\_output.put\_line ('My wife was born in October');

else

dbms\_output.put\_line ('I do not have any relatives

born in '||current\_month);

end if;

end;



// finding oldest employee

set serveroutput on;

DECLARE

oldest\_birth\_date date;

EMP\_LNAME employee.EMP\_LNAME%type;

EMP\_FNAME employee.EMP\_FNAME%type;

cursor find\_old\_b\_day is select min(EMP\_DOB) from employee;

cursor id\_emp is select EMP\_LNAME, EMP\_FNAME from employee where EMP\_DOB = oldest\_birth\_date;

BEGIN

open find\_old\_b\_day;

fetch find\_old\_b\_day into oldest\_birth\_date;

close find\_old\_b\_day;

open id\_emp;

fetch id\_emp into EMP\_LNAME, EMP\_FNAME;

close id\_emp;

dbms\_output.put\_line ('The Oldest employee Is ' ||EMP\_FNAME||' '||EMP\_LNAME);

END;

/



// finding oldest employee program 2

set serveroutput on;

DECLARE

cursor find\_old\_b\_day is

select min(EMP\_DOB) day from employee;

old\_date find\_old\_b\_day%rowtype;

cursor id\_emp is select EMP\_LNAME, EMP\_FNAME from employee where EMP\_DOB = old\_date.day;

id id\_emp%rowtype;

BEGIN

open find\_old\_b\_day;

fetch find\_old\_b\_day into old\_date;

close find\_old\_b\_day;

open id\_emp;

fetch id\_emp into id;

close id\_emp;

dbms\_output.put\_line ('The Oldest Employee Is ' ||id.EMP\_FNAME||' '||id.EMP\_LNAME);

END;

/



// Printing first three employees

set serveroutput on;

declare

counter\_variable number := 1;

cursor a is select EMP\_LNAME, EMP\_FNAME from employee;

cur\_var a%rowtype;

begin

open a;

loop

exit when counter\_variable = 4;

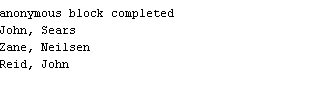
fetch a into cur\_var;

dbms\_output.put\_line (cur\_var.EMP\_LNAME || ', '||cur\_var.EMP\_FNAME);

counter\_variable := counter\_variable +1;

end loop;

end;



// printing all employees

set serveroutput on;

declare

cursor a is select EMP\_LNAME, EMP\_FNAME from employee;

cur\_var a%rowtype;

begin

open a;

fetch a into cur\_var;

while a%found

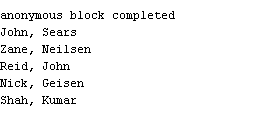
loop

dbms\_output.put\_line (cur\_var.EMP\_LNAME|| ', '||cur\_var.EMP\_FNAME);

fetch a into cur\_var;

end loop;

end;



// Printing all employees born in Mankato

set serveroutput on ;

declare

cursor a is select employee\_no, EMP\_LNAME, EMP\_FNAME, EMP\_DOB from employee where lower(city) = 'mankato';

cur\_var a%rowtype;

begin

dbms\_output.put\_line('EMP Last First Date');

dbms\_output.put\_line('NO Name Name of Birth');

dbms\_output.put\_line('---- --------------- ---------- ---------');

open a;

fetch a into cur\_var;

while a%found

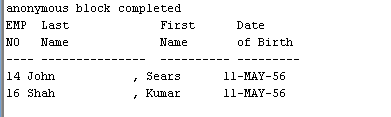
loop

dbms\_output.put\_line (cur\_var.employee\_no|| ' '||rpad(cur\_var.EMP\_LNAME,15)|| ', '||rpad(cur\_var.EMP\_FNAME,10)|| ' '||cur\_var.EMP\_DOB);

fetch a into cur\_var;

end loop;

end;



// Prinitng all products with their names and category

set serveroutput on

declare

cursor a is select PNO, prod\_name, prod\_category from product;

prod\_var a%rowtype;

begin

open a;

for cnt\_var in 1..5

loop

fetch a into prod\_var;

dbms\_output.put\_line(to\_char(cnt\_var)

||' '||prod\_var.PNO

||' '||prod\_var.prod\_name

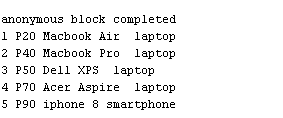
||' '||prod\_var.prod\_category);

end loop;

close a;

end;

/



// Printing product numbers along with their unit prices

set serveroutput on ;

BEGIN

dbms\_output.put\_line('PNO '||' '|| 'UNIT\_PRICE');

for cnt\_var in (SELECT PNO, UNIT\_PRICE from product)

LOOP

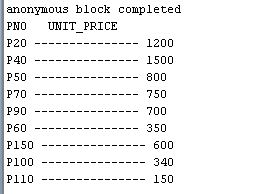
dbms\_output.put\_line(cnt\_var.PNO

||' --------------- '||cnt\_var.UNIT\_PRICE);

END LOOP;

END;

/



ASSIGNMENT 4: PROCEDURS ON OWN DATABASE

// Hello world procedure with name

Set serveroutput on;

create or replace procedure firstproc is

begin

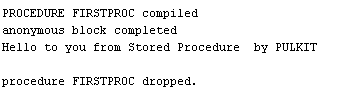
dbms\_output.put\_line ('Hello to you from Stored Procedure by PULKIT ');

end;

/

execute firstproc ;

drop procedure firstproc ;



// Printing employees from Mankato and their date of births

Set serveroutput on;

create or replace procedure list\_emp (city2 in varchar2) is

begin

for a in (select e.EMP\_LNAME, e.EMP\_FNAME, e.EMP\_DOB

from employee e

where upper(e.city1) = upper(city2) )

loop

dbms\_output.put\_line('Name='||a.EMP\_LNAME||' '||a.EMP\_FNAME);

dbms\_output.put\_line('EMP\_DOB='||a.EMP\_DOB);

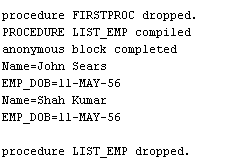
end loop;

end;

/

exec list\_emp('Mankato');

drop procedure list\_emp;



Set serveroutput on;

declare

procedure Hello is

begin

dbms\_output.put\_line ('Hello to you from Stored Procedure');

end;

procedure list\_position (position1 in varchar2) is

begin

for a in (select \* from employee where city1=position1)

loop

dbms\_output.put\_line (a.city1);

end loop;

end;

begin

dbms\_output.put\_line ('\*\*\* Results of the first procedure');

hello;

dbms\_output.put\_line ('\*');

dbms\_output.put\_line ('\*\*\* Results of the second procedure');

dbms\_output.put\_line ('position ');

dbms\_output.put\_line ('---------');

list\_position('Manager');

end;

/

