

PART - B

- ① Create two Tables P and Q as  
P(lid number, name, string); and  
Q(lid number, name, string);

- i) ~~Insert~~ The Tables with The above attributes.  
ii) insert rows into Table P and Q  
iii) display all records of both tables P and Q  
iv) display only records that exist in both Tables P and Q.  
v) display all records that are in Table P but not in Table Q.

Query 1:

Create Table P (lid number primary key, name varchar2 (20));

Create Table Q (lid number primary key, name varchar2 (20));

Query 2:-

Insert into P values (101, 'Ashu');  
Insert into P values (102, 'Smitha');  
Insert into P values (104, 'Javange');  
Insert into P values (107, 'Veena');

dec P:

Table	column	datatype	length
P	id	number	-
	name	varchar2	20

dec Q:

Table	column	datatype	length
P	id	number	-
	name	varchar2	20

Select \* from P;

ID	name
101	Ashu
102	Smitha
104	Javange

107	Veena
110	divya
112	sauranya
Select * from q;	
ID	name
103	Smitha
104	lavanya
106	bhaskar
110	divya
Query 3:	
ID	name
101	Asha
103	Smitha
104	lavanya
106	bhaskar
107	Veena
110	divya
112	sauranya
Query 4:	
ID	name
103	Smitha
104	lavanya
110	divya

Expt. No. ....	Date .....
Page No. 33	
insert into p values (110, 'divya'); insert into p values (112, 'sauranya'); insert into q values (103, 'Smitha'); insert into q values (104, 'lavanya'); insert into q values (106, 'bhaskar'); insert into q values (110, 'divya');	
Query 3: select * from p union select * from q;	
Query 4: select * from p intersect select * from q;	
Achiever	



Query 5!

ID	name
101	asha
107	vera
112	svanya

Expt. No. ....

Date .....

Page No. 34

Query 5!

select \* from P minus select \* from q

## ① Create The following Tables.

ITEM (Item code integer, Itemname Varchar(30), price decimal (10,2))

PURCHASE (Item code integer, quantity integer)

ii) atleast Insert 5 rows into each Table.

iii) demonstrate inner join as &amp; as join operation on these Tables

## i) Query 1:

Create Table Item (Item code integer, Item name Varchar(30), price decimal (10,2));

Create Table purchase (Item code integer quantity integer);

## Query 2:

insert into Item values ('4001', 'notebook', 30);

insert into Item values ('4002', 'pen', 40);

insert into Item values ('4003', 'writing pad', 85);

insert into Item values ('4004', 'Joke pad', 80);

insert into Item values ('4005', 'gum bottle', 95);

desc item;

Table	column	datatype	length
Item	Itemcode	number	-
	Itemname	Varchar 2	80
	price	number	-

desc purchase;

Table	column	datatype	length
Purchase	Itemcode	number	-
	Quantity	number	-

Query 2:

select \* from Item

Itemcode	Itemname	price
4001	note book	30
4002	pen	40
4003	writing pad	85
4004	Joke pad	80
4005	gum bottle	95



select \* from purchase;

Item code	Quantity
4001	10
4002	15
4003	5
4004	2
4005	1

Query 3:-

Item code	Item name	Price	Quantity
4001	note book	30	10
4002	pen	25	15
4003	writing pad	40	5
4004	int pad	20	2
4005	gum bottle	25	1

Expt No. ....

Date .....  
Page No. 36

Insert into Purchase Values

(4001, 10)  
(4002, 15)  
(4003, 5)  
(4004, 2)  
(4005, 1)

Query 3:-

select i.itemcode, i.itemname, i.price, p.quantity  
from item i inner join purchase p on i.itemcode =  
p.itemcode;

Q3) Create The following two tables person and job  
 person (pid varchar(5), name varchar(20));  
 job (jobid varchar(3));

- (i) Create the above table -
- ii) Insert rows into the two tables
- iii) demonstrate the set operation (cartesian product + (boys join) on these two tables.

\* Query 1:

create table person (pid varchar(5), name varchar(20));

Create table job (jobid varchar(3));

Query 2:

insert into person values ('101', 'Asha');  
 insert into person values ('102', 'Amitha');  
 insert into person values ('103', 'Anusha');  
 insert into person values ('104', 'Prakash');  
 insert into person values ('105', 'manju');  
 insert into person values ('106', 'pratharth');

desc person1;

Table	column	data type	length
person1	pid	varchar 2	50
	name	varchar 2	20

desc job1;

Table	column	data type	length
job 1	jobid	varchar 2	3

select \* from person1;

pid	name
101	asha
102	amitha
103	anusha
104	prakash
105	manju
106	pratharth



select \* from job ;

job id	
j1	
j2	

pid	name	job id
101	asha	j1
102	Amitha	j1
103	anusha	j1
104	Prakash	j1
105	manju	j1
106	prabanth	j1
101	asha	j2
102	Amitha	j2
103	anusha	j2
104	Prakash	j2

Expt No.

Page No. 32

Date

insert into job values ('j1');  
insert into job values ('j2');

query 3;

select P.id, P.name, J.jobid from person P cross  
join job J;

- Q1) Create the table with the following attributes
- Branch, branch (cname string, branch string, loan amt number (5))
  - customer loan (cname string, loan no string);
  - insert at least 3 rows in each table
  - demonstrate various types of arithmetical operations on the tables.

Query 1:

Create table branch\_loan (cname varchar(20), branch varchar(10), loan amt decimal(5));

Create table customer\_loan (cname varchar(20), loan no varchar(10));

Query 2:

insert into branch\_loan values ('a', 'L-170', 3000);

insert into branch\_loan values ('b', 'L-210', 5000);

insert into branch\_loan values ('c', 'L-340', 8000);

insert into customer\_loan values ('a', 'L-170', 3000);

insert into customer\_loan values ('b', 'L-210', 5000);

insert into customer\_loan values ('c', 'L-340', 8000);

select \* from customer\_loan;

due branch\_loan;

Table	column	datatype	length
branch	branch	varchar(20)	20
loan	loan no	varchar(20)	10
	loan amt	number	-

due customer\_loan;

Table	column	datatype	length
customer	customer	varchar(20)	20
loan	loan	varchar(20)	10

select \* from branch\_loan;

Branch	loan no	loan amt
X	L-170	3000
Y	L-210	5000
Z	L-340	8000

due name\_loan no

name	loan no
a	L-170
b	L-210
c	L-340



left outer join

name	loan no	loan amt	branch	loan no
X	L-170	3000	a	L-170
Y	L-340	8000	c	L-340
Z	L-210	5000	-	-

right outer join

name	loan no	loan amt	branch	loan no
X	L-170	3000	a	L-170
Z	L-340	8000	c	L-340
-	-	-	b	L-250

full outer join

name	loan no	loan amt	branch	loan no
X	L-170	3000	a	L-170
Z	L-340	8000	c	L-340
Y	L-210	5000	-	-
-	-	-	b	L-250

Expt No. ....

Page No. 44

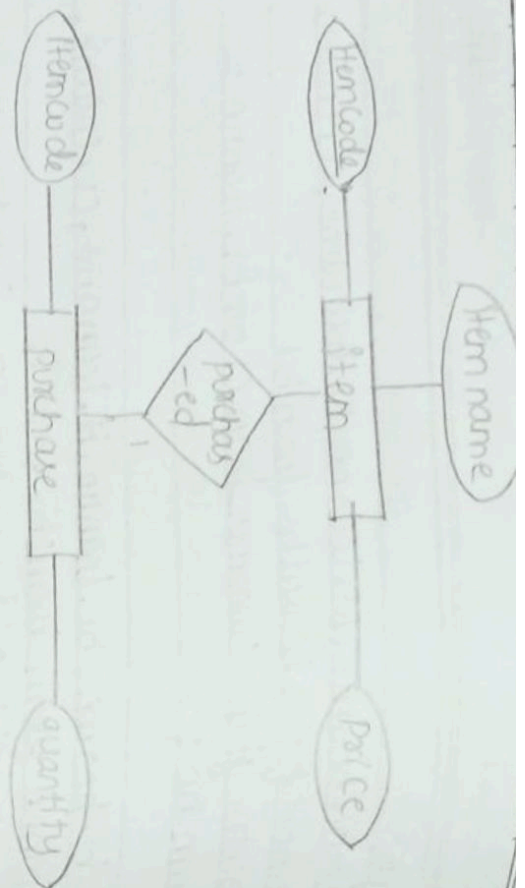
Ques 3:-

select b1.branch, b1.loan no, b1.loan amt, c1.branch,  
c1.loan no from branch=branch b1  
left outer join customer=loan c1 on b1.loan no =  
c1.loan no;

select b1.branch, b1.loan no, b1.loan amt, c1.branch,  
c1.loan no from branch=branch b1  
right outer join customer=loan c1 on b1.loan no =  
c1.loan no;

select b1.branch, b1.loan no, b1.loan amt, c1.branch,  
c1.loan no from branch=branch b1  
full outer join customer=loan c1 on b1.loan no =  
c1.loan no;





\* Query 1

disc items;

Table	Column	datatype	length
Items	Itemcode	number	-
	Itemname	varchar2	80
	Price	number	-

disc purchases;

Table	Column	datatype	length
Purchases	Itemcode	number	-
	Quantity	number	-

select from items;

Itemcode	Itemname	Price
1001	pen	7.5
1002	Pencil	3.5
1003	calculator	300
1004	addition book	80
1005	stapler	30

Expt No. ....

Page No. 41

Date .....

5) An inventory database has the following tables

Item (Item code: number, Item name: text, price: number (10,2));

Purchase (Item code: number, quantity: number)

1) Create the tables with the above attributes

ii) Enter atleast tuples into the tables

iii) List the items purchased

iv) display the total items purchased. (Listing must have the columns: Item code, Item name, Total quantity)

Query 1:-

Create Table Items (Item code int primary key, Item name varchar2 (80), price decimal (10,2));

Create Table purchases (Item code int, quantity int);

Query 2:-

Insert Into Items values (1001, 'pen', 7.50);

insert into Items values (1002, 'pencil', 3.50);

insert into Items values (1003, 'calculator', 300);

insert into Items values (1004, 'addition book', 80);

insert into Items values (1005, 'stapler', 30);



# selects from purchases

Itemcode	Quantity
1001	7
1002	20
1003	2
1004	5
1005	2

3) Item name  
 Pen  
 Pen  
 Calculator  
 Address book  
 Stapler

Item code	Quantity
1001	7
1002	20
1003	2
1004	5
1005	2

Query 5:

Total - Items - purchases  
 5

Expt No. ....

Page No. 112

Date

Insert into purchases values (1001, 7);  
 Insert into purchases values (1002, 20);  
 Insert into purchases values (1003, 2);  
 Insert into purchases values (1004, 5);  
 Insert into purchases values (1005, 2);

Query 3:

Select item name, Item purchases, Items where Item  
 -s. Item code = purchases. Item code;

Query 4:

Select items, Item code, purchases, quantity from  
 items, purchases where items. Item code = purchase  
 1. Item code;

Query 5:

Select count (Item code) total - items purchases  
 -es from purchases;

\* output

A 10  
B 7  
C 50

C is largest  
Statement processed.

Submit

Expt No. ....

Page No. 43

Date .....

6) Write a P/L Sol Program to find the largest of three numbers

```
declare
a number;
b number;
c number;
begin
a := 10;
b := 7;
c := 50;
if a > b and a > c then
  dbms_output.put_line('a is largest' || a);
else if b > a and b > c then
  dbms_output.put_line('b is largest' || b);
else
  dbms_output.put_line('c is largest' || c);
end if;
end;
```



\* output

: N 456

The reverse number is  
654  
statement processed.

Submit

Expt No. ....

Date .....

Page No. 44

7) Write a PLI SQL Program to generate reverse for  
Given number.

```
declare
n number(4) := :n;
s number(4) := 0;
r number(4);
begin
while n > 0
loop
r := mod(n, 10);
s := (s * 10) + r;
n := trunc(n / 10);
end loop;
dbms_output.put_line('The reverse number is');
dbms_output.put_line(s);
end;
```

\* output

$i$  N 6

Factorial of a number is 720.

Submit

Expt. No. ....

Page No. 45

8) write a PL/SQL program to find the factorial of a given number.

```
declare
  number (4) := 1;
  n number (4) := 6;
  f number (4) := 1;
begin
  for i in 1 .. n
  loop
    f := f * i;
  end loop;
  dbms_output.put_line ('factorial of a number
  is ' || f);
end;
```



Submit

```
* output
: Num 6
6 not prime
Statement processed
: num 7
7 is prime
Statement processed
```

Expt No. ....

Date .....  
Page No. 46

Q) Write a PL/SQL program to check whether given number is prime or not.

```
declare
num number;
i number := 1;
c number := 0;
begin
num := :num;
for i in 1..num
loop
if (c mod (num, i)) = 0)
Then
c := c + 1;
end if;
end loop;
if (c > 2)
Then
dbms_output.put_line (num || 'not prime');
else
dbms_output.put_line (num || 'is prime');
end if;
end;
```

\* output

N: 20

The Fibonacci series is

1  
1  
2  
3  
5  
8  
13

statement processed.

Submit

Expt No. ....

Page No. 47

Date .....

10) Write a PL/SQL Program to generate fibonacci series up to N.

declare

a number(3) := 1;

b number(3) := 1;

c number(3);

n number(3) := n;

begin

dbms\_output.put\_line('The fibonacci series

is:');

while a <= n loop

dbms\_output.put\_line(a);

c := a + b;

a := b;

b := c;

end loop;

end;

/

Achiever



\* Output

procedure created  
0.37 seconds

The sum is 30  
statement processed  
0.00 seconds

Exp No. \_\_\_\_\_

Date \_\_\_\_\_  
Page No. 142

ii) Write a PL/SQL program for creating a procedure for calculating sum of two numbers.

Creating the procedure:

```
create or replace procedure sum (n1 in number,  
n2 in number) is  
total number(6);  
begin  
total := n1 + n2;  
dbms_output.put_line ('The sum is ' || total);  
end;  
  
To call a sum the procedure:  
begin  
sum (10, 20);  
end
```

\* output :  
procedure created.  
0.01 seconds.

begin leap\_year (2020); end;  
The given year is leap\_year  
statement processed  
0.00 seconds.

begin leap\_year (2022); end;  
The given year is not leap\_year  
statement processed  
0.01 seconds.

Expt. No. ....

Page No. 49

Date .....

12) write a procedure to check The given year is leap year or not

Create an replace procedure leap\_year  
(y in number) is

begin

if  $y \bmod 4 = 0$  and  $y \bmod 100 < > 0$  or  $y$   
 $\bmod 400 = 0$  then

dbms\_output.put\_line ('The given year is  
leap year');

else

dbms\_output.put\_line ('The given year is not  
leap year');

end if;

end;

/

Procedure created