EX.NO:01

DATE: 16/07/2021

TABLE MANIPILATION

AIM:

To write a SQL program to manipulate the DDL & DML Commands.

CREATE THE TABLE:

create table Student(Roll_no int(10)NOT NULL primary key(Roll_no),Name varchar(20),Age int(3),Address varchar(30));

Table created successfully.

INSERT THE TABLE:

insert into student values(19SUCS124,'M.Nanthakumar',20, 'Theni');
1 row created.
insert into student values(19SUCS135,'B.Sivaprakash',20, 'Madurai');
1 row created.
insert into student values(19SUCS121,'N.S.Jayaram',20, 'Chennai');
1 row created.

SELECTE THE TABLE:

Selecte * from Student;

Roll_no	Name	Age	Address
19SUCS124	M.Nanthakumar	20	Theni
19SUCS135	B.Sivaprakash	20	Madurai
19SUCS121	N.S.Jayaram	20	Chennai

ALTER THE TABLE:

alter table Student add(Phone_no int(10));

Roll_no	Name	Age	Address	Phone_no
19SUCS124	M.Nanthakumar	20	Theni	
19SUCS135	B.Sivaprakash	20	Madurai	

19SUCS121	N.S.Jayaram	20	Chennai	

DROP THE TABLE:

drop table student;

student table is droped(deleted).

TRUNCATE THE TABLE:

Truncate table Student;

Roll_no	Name	Age	Address	Phone_no

DESCRIBE:

ATTIBUDE	DATA TYPE
Roll_no	Int
Name	Varchar
Age	Int
Address	Varchar
Phone_no	Int

RESULT:

SQL Program is executed success fully

EX.NO:02
DATE: 24/07/2021

AGGERGATE FUNCTIONS

AIM:

To write a SQL program to perform aggregate functions.

CREATE A TABLE:

create table student2(Roll_no int(10),Name varchar(20),m1 int(5),m2 int(5),m3 int(5));

Table created

INSERT VALUES:

insert into student2 values('1','Arun','78','67','98');
insert into student2 values('2','Dinesh','76','87','67');
insert into student2 values('3','Jegan','78','79','56');
insert into student2 values('4','Mani','78','79','85');
insert into student2 values('5','Neru','76','57','87');
insert into student2 values('6','Vishal','87','78','99');
insert into student2 values('7','Suresh','96','56','87');
7 rows inserted

SUM:

select sum(m1+m2+m3) from student2;

Total: 1639

AVG:

select avg(m1+m2+m3)/3 from student2;

Avg: 78.047619

MAXIMUM:

select max(m1) from student2;

Max: 96

MINIMUM:

select min(m2) from student2;

Min: 56

COUNT:

select count(roll_no) from student2;

Count: 7

RESULT:

SQL Program is executed success fully

EX.NO:03

DATE: 27/07/2021

DUAL FUNCTION

AIM:

To write a SQL program to perform DUAL operations.

QUREIES:

select round(113.45) from dual;

Round: 113

select sqrt(17846) from dual;

Sqrt: 133.588922

select power(5,2) from dual;

Power:25

select abs(-275.5) from dual;

Abs:275.5

select ceil(6.75) from dual;

Cile:7

select floor(-6.75) from dual;

Floor:6

select length(72145678) from dual;

Length:8
select sin(90) from dual;

Sin: .893996664

select cos(0) from dual;

Cos: 1

select tan(45) from dual;

Tan: 1.61977519

select mod(25,7) from dual;

Mod: 4

select ASCII('a') from dual;

Ascii: 97

select sysdate from dual;

System Date: 08-NOV-21

select concat('hi','how are you') from dual;

Concat: hihow are you

select substr('oracle',2,4) from dual;

Substr: Racl

RESULT:

SQL Program is executed success fully

EX.NO:04

DATE: 10/08/2021

DATE FUNCTION

AIM:

To write a SQL program to perform date functions.

QUERIES:

Select sysdate,last_day(sysdate) from dual;

Sysdate last_day

25-oct -21 31-oct-21

Select sysdate +5 from dual;

Sysdate +5

30-oct -21

Select months_between('31-mar-2021', '28-feb-2021');

months_between('31-mar-2021', '28-feb-2021')

1

Select sysdate from dual;

Sysdate

25-oct -21

Select extract(day from sysdate) from dual;

extract(day from sysdate)

25

Select extract(month from sysdate) from dual;

extract(month from sysdate) from dual;

10

Select sysdate -5 from dual;

sysdate +5

20-oct-21

Select add_months(sysdate,1) from dual;

add_months

25-nov-21

Select next_day(sysdate, 'Tuesday') from dual;

next_day

26-oct-21

Select extract(year from sysdate) from dual;

extract(month from sysdate)

2021

RESULT:

SQL Program is executed success fully

EX.N0:05
DATE: 14/08/2021

JOINS

Aim:

To write a SQL program to perform various joins and set operations.

QUERIES:

CREATE A TABLE:

create table student(Roll_no int(3),Name varchar(20), Address varchar(30), Phone_no number(10),Age int(3));

Table created

INSERT RECORD:

insert into student values(&Roll_no,'&Name','&Address',&phone_no,&Age);

select * from student;

ROLL_NO	NAME	ADDRESS	PHONE_NO	AGE
1	Athi	Mdu	7809876590	19
2	Dhinesh	Trichy	9087908767	19
3	Jack	Che	9090908767	20
4	Magi	Selam	9999000087	19
5	Naren	Tpk	9876543210	19
6	Pattu	Theni	7878789878	19
7	Siva	Vir	7890978909	20
8	Visu	Din	9987678987	20

CREATE TABLE FOR STUDENT COURSE:

create table studentcourse (ELECTIVE_ID int(2),ROLL_NO int(2));

select * from stuentcourse;

ELECTIVE_ID	ROLL_NO
1	1
2	2
2	3
3	4
1	5

4	9
5	10
4	11

INNER JOIN

select sc.nme_id, s.name, s.age from student s inner join studentcourse sc on s.roll_no=sc.roll_no;

ELECTIVE_ID	NAME	AGE
1	Athi	19
2	Dhinesh	19
2	Jack	20
3	Magi	19
1	Naren	19

LEFT OUTER JOIN:

select s.name, sc.nme_id from student s left outer join studentcourse sc on s.roll_no=sc.roll_no;

NAME	ELECTIVE_ID
Athi	1
Dhinesh	2
Jack	2
Magi	3
Naren	1
Pavi	
Siva	
Visu	

8 rows selected.

RIGHT OUTER JOIN:

select s.name, sc.nme_id from student s right outer join studentcourse sc on s.roll_no=sc.roll_no;

NAME	ELECTIVE_ID
Athi	1
Dhinesh	2
Jack	2
Magi	3
Naren	1
	5
	4
	4

FULL OUTER JOIN:

select s.name, sc.nme_id from student s full outer join studentcourse sc on s.roll_no=sc.roll_no;

NAME	ELECTIVE_ID
Athi	1
Dhinesh	2
Jack	2
Magi	3
Naren	1
Pavi	5
Siva	4
Visu	4

11 rows selected.

NATURAL JOIN:

select * from student natural join studentcourse;

ROLL_NO	NAME	ADDEESS	PHONE_NO	AGE	ELECTIVE_ID
1	Athi	Mdu	7809876590	19	1
2	Dhinesh	Trichy	9087908767	19	2
3	Jack	Che	9090908767	20	2
4	Magi	Selam	9999000087	19	3
5	Naren	Tpk	9876543210	19	1

RESULT:

SQL Program is executed success fully

EX.NO:06

DATE: 17/08/2021

TRANSACTION CONTROL COMMANDS

AIM:

To write a SQL program to Transaction Management control command

CREATE TABLE:

create table student(id int(3), name varchar(20), age int(3), address varchar(10), salary int(10));

INSERT VALUES:

insert into student values(1,'siva',32,'mdu',10000);

insert into student values(2,'Dhoni',25,'che',10000);

insert into student values(3,'Naren',47,'sal',20000);

insert into student values(4,'Rana',25,'che',30000);

insert into student values(5,'Raina',35,'mdu',40000);

SELECT VALUES:

select * from values;

ID	NAME	AGE	ADDRESS	SALARY
1	Siva	32	Mdu	10000
2	Dhoni	25	Che	10000
3	Naren	47	Sal	20000
4	Rana	25	Che	30000
5	Raina	35	Mdu	40000

COMMIT:

delete from student where age = 25;

select * from student;

ID	NAME	AGE	ADDRESS	SALARY
1	Siva	32	Mdu	10000
3	Naren	47	Sal	20000
5	Raina	35	Mdu	40000

ROLLBACK:

delete from student where age = 25;

select * from student;

ID	NAME	AGE	ADDRESS	SALARY
1	Siva	32	Mdu	10000
2	Dhoni	25	Che	10000
3	Naren	47	Sal	20000
4	Rana	25	Che	30000
5	Raina	35	Mdu	40000

SAVEPOINT:

SAVEPOINT siva1;

delete from student where id = 1;

SAVEPOINT ball;

delete from student where id = 2;

SAVEPOINT bat;

delete from student where id = 3;

ROLLBACK bat;

select * from student;

ID	NAME	AGE	ADDRESS	SALARY
3	Naren	47	Sal	20000
4	Rana	25	Che	30000
5	Raina	35	Mdu	40000

RESULT:

SQL Program is executed success fully

EX.NO:07

DATE:20/09/2021

SQUARE NUMBER

AIM:

Write a program to square numbers using procedure in PL/SQL

QUERIES:

```
set serveroutput on;
 DECLARE
    a number;
    procedure squarenum(x in out number)
    Is
 BEGIN
    x:=x*x;
 END;
 BEGIN
    dbms_output.put_line('Enter the value of a=');
    a:=&a;
    squarenum(a);
    dbms_output.put_line('square of(a):'||a);
 END;
 /
```

OUTPUT:

Enter the value of a= 8

square of(a): 64

RESULT:

PL/SQL Program is executed success fully.

EX.NO:08

DATE: 23/09/2021

GREATEST NUMBER

AIM:

Write a program to Greatest Numbers using if and elseif in PL/SQL

QUERIES:

```
set serveroutput on
DECLARE
    a number;
    b number;
    c number;
BEGIN
    dbms_output.put_line('enter the value of a');
    a:=&a;
    dbms_output.put_line('enter the value of b');
    b:=&b;
    dbms_output.put_line('enter the value of c');
    c:=&c;
    if a>b and a>c then
    dbms_output.put_line('a is greatest');
    elsif b>a and b>c then
    dbms_output.put_line('b is greatest');
    else
    dbms_output.put_line('c is greatest');
    end if;
```

```
END;
/
```

OUTPUT:

ENTER THE VALUE OF A=10

ENTER THE VALUE OF B=9

ENTER THE VALUE OF C=8

A is greatest number

RESULT:

PL/SQL Program is executed success fully.

EX.NO:09

DATE: 25/09/2021

MULTIPLICATION TABLE

AIM:

Write a program to Multiplication Table using for loop in PL/SQL

QUERIES:

```
Set serveroutput on
DECLARE
    n number;
    i number;
BEGIN
    dbms_output.put_line('Enter the value of n');
    n:=&n;
    for i in 1..10 loop
    dbms_output.put_line(n||'*'||'='||n*i);
    end loop;
END;
/
```

OUTPUT:

Enter the value of n= 5

5*1=5

5*2=10

5*3=15

5*4=20

5*5=25

5*6=30

5*7=35

5*8=40

5*9=45

5*10=50

RESULT:

PL/SQL Program is executed success fully.

EX.NO:10

DATE: 27/09/2021

STUDENT MARK LIST

AIM:

Write a program to Student Mark List using if and elseif in PL/SQL

QUERIES:

```
set serveroutput on
 DECLARE
      dbms number;
         cn  number;
         os  number;
        cns number;
       total number;
        per number;
 BEGIN
        dbms:=&dbms;
            cn:=&cn;
            os:=&os;
           cns:=&cns;
         total:=(dbms+cn+os+cns);
            per:=(total/400)*100;
    if dbms<40 or cn<40 or os<40 or cns<40 then
    dbms_output.put_line('fail');
    if per>75 then
    dbms_output.put_line('Grade A');
```

```
    elsif per>65 and per<75 then
    dbms_output.put_line('Grade B');
    elsif per>55 and per<65 then
    dbms_output.put_line('Grade C');
    else
    dbms_output.put_line('invalid output');
    end if;
    end if;
    dbms_output.put_line('percentage is'||per);
   dbms_output.put_line('total is'||total);
END;
/
```

OUTPUT:

```
Enter the value of dbms : 89
Enter the value of cn   : 90
Enter the value of os   : 99
Enter the value of cns  : 98
                    Total : 376
                  Average : 94
```

RESULT:

PL/SQL Program is executed success fully.

EX.NO:11

DATE: 30/09/2021

FACTORIAL

AIM:

Write a program to Factorial using for loop in PL/SQL

QUERIES:

```
set serveroutput on
 DECLARE
     n number;
     fac number:=1;
     i number;
 BEGIN
     dbms_output.put_line('enter the valu of n');
     n:=&n;
     for i in 1..n loop
     fac:=fac*i;
     end loop;
     dbms_output.put_line('the factorial='||fac);
 END;
 /
```

OUTPUT:

Enter the value of n: 4

The Factorial: 24

RESULT:

PL/SQL Program is executed success fully

EX.NO:12

DATE: 04/10/2021

SUB QUERY

AIM:

Write a program to Sub Query in PL/SQL

QUERIES:

CREATE TABLE:

create table customer(id int,name varchar(20),age int,address varchar(20),salary int);

INSERT VALUES:

insert into customer values(&id,'&name',&age,'&address',&salary);

select * from customer;

ID	NAME	AGE	ADDRESS	SALARY
1	Bheem	29	Theni	21000
2	Krishna	30	Madurai	22000
3	Raju	28	Chennai	19000

SUB QUREY:

select * from customer where id in (select id from customer where salary>20000);

ID	NAME	AGE	ADDRESS	SALARY
1	Bheem	29		21000
2	Krishna	30		22000

RESULT:

PL/SQL Program is executed success fully

EX.NO:13

DATE: 05/10/2021

TRIGGER FUNCTIONS

AIM:

Write a program to Trigger Functions List in PL/SQL

QUERIES:

CREATE TABLE:

```
create table sum(no int,name varchar(20),salary int);
```

Table Created.

INSERT VALUES:

```
insert into sum values(&no,'&name',&salary);
```

NO	NAME	SALARY
1	Berlin	50000
2	Rio	50000
3	Denver	48000

CREATE TRIGGER:

```
create or replace trigger trg  before insert on sum
   for each row
   begin
   :new.name:=upper(:new.name);
   end;
   /
```

INSERT VALUES(new):

```
insert into sum values(&no,'&name',&salary);
```

NO	NAME	SALARY
1	Berlin	50000
2	Rio	50000
3	Denver	48000
4	TOKYO	52000

RESULT:

PL/SQL Program is executed success fully

EX.NO:14

DATE: 08/10/2021

EXCEPTION HANDLING

AIM:

Write a program to Exception Handling List using Exception in PL/SQL

QUERIES:

```
set serveroutput on;
 DECLARE
       a int;
       b int;
       c int;
 BEGIN
       a:=&a;
       b:=&b;
       c:=a/b;
      dbms_output.put_line('result =  '||c);
 EXCEPTION
      when ZERO_DIVIDE then
      dbms_output.put_line('division by 0 is not possible');
 END;
/
```

OUTPUT:

Enter the value of a:10

Enter the value of b:5

Result: 2

If we give

Enter the value of a:0

Enter the value of b:5

Result: division by 0 is not possible

RESULT:

PL/SQL Program is executed success fully

EX.NO:15

DATE: 12/10/2021

ARMSTRONG NUMBER

AIM:

Write a program to Armstrong Number List using while loop in PL/SQL

QUERIES:

Set serveroutput on

```
DECLARE

      n number(3);
      s number(3):=0;
      t number(3);

BEGIN

      n:=&n;
       t:=n;
       while t>0 loop
       s:=s+power((t mod 10),3);
        t:=trunc(t/10);
end loop;

     if(s=n) then
      dbms_output.put_line('The Given Number ' || n || ' is an Armstrong Number');
    else
      dbms_output.put_line('The Given Number ' || n || ' is Not an Armstrong Number');
    end if;

END;
/
```

OUTPUT

Enter the value of n: 153

The Given Number is an Armstrong Number

If we give

Enter the value of n: 155

The Given Number is Not an Armstrong Number

RESULT:

PL/SQL Program is executed success fully

EX.NO:16

DATE: 18/10/2021

PALINDROME NUMBER

AIM:

Write a program to Palindrome Number List using while loop in PL/SQL

QUERIES:

```
Set serveroutput on
 DECLARE
       n number;
      m number;
     rev number:=0;
        r number;
 BEGIN
       dbms_output.put_line('enter the number');
      n:=&n;
      m:=n;
      while n>0
      loop
      r=mod(n,10);
      rev:=(rev*10)+r;
      n:=trun c(n/10);
      end loop;
      if m=rev then
       dbms_output.put_line('Given number is a Palindrome');
      else
```

```
    dbms_output.put_line('Given number is a Not Palindrome');
   end if;
END;
/
```

OUTPUT:

Enter the number: 12321

Given number is a Palindrome

If we give

Enter the number: 12331

Given number is a Not Palindrome'

RESULT:

PL/SQL Program is executed success fully

EX.NO:17

DATE: 21/10/2021

FIBONACCI SERIES

AIM:

Write a program to Fibonacci Series List using for loop in PL/SQL

QUERIES:

```
Set serveroutput on
 DECLARE:
     First number:=0;
  Second number:=1;
    temp number;
        n number:=5;
        i number;
 BEGIN
     dbms_output.put_line('Fibonacci series is');
     dbms_output.put_line('First');
     dbms_output.put_line('Second');
     for i in 2..n loop
     temp:= First+Second;
     First:=Second;
 Second:=temp;
     dbms_output.put_line('temp');
     end loop;
 END;
 /
```

OUTPUT:

0

1

1

2

3

5

8

RESULT:

PL/SQL Program is executed success fully