DEPARTMENT OF COMPUTER SCIENCE

Aim:

Create student(htno, name, percentage_of_marks, DOJ, d_id) and Department(d_id, d_name, hod, contactno, e-mail) tables with relevant primary key, foreign key and other constraints. Perform the following.

a). Insert three students details in two departments.

b). Display all students order by dept_no.

c). Display all students in each department who has highest percentage.

```
SQL> insert into student2 values('201k0002,'srinivas',75,'23-oct-
1998',9224517887,20);

ERROR:
ORA-01756: quoted string not properly terminated


SQL> insert into student2 values('201k0002','srinivas',75,'23-oct-
1998',9224517887,20);

1 row created.

SQL> insert into student2 values('201k0003','krishna',85,'23-oct-
1997',922451789,20);

1 row created.

SQL> select * from student2;

HTNO      SNAME     PERCENTAGE DOJ       CONTACT_NO      D_ID
-----------------------
201f0001  ramu            57 22-OCT-01 9224589765        10
201f0002  ragu            65 22-JUN-01 9224512345        10
201f0003  roja           75 12-JAN-00 9224512377         10
201k0001  sujatha          45 11-JAN-03 9224518887        20
201k0002  srinivas        75 23-OCT-98 9224517887        20
201k0003  krishna         85 23-OCT-97  922451789        20

6 rows selected.

SQL> insert into department values(10,'mca
  2  ','ramu','xyz@gmail.com','201f0001');

1 row created.

SQL> insert into department values(20,'mba
  2  ','sujat','abc@gmail.com','201k0001');

1 row created.

SQL> select * from department;

    D_ID D_NA HOD   E_MAIL              HTNO
-------- ---- ----- ------------------- ----------
      10 mca  ramu  xyz@gmail.com       201f0001
      20 mba  sujat abc@gmail.com       201k0001

SQL>
```

```
SQL> create table student2(htno varchar2(10) primary key,sname
varchar2(10),percentage number(6,2),
  2  doj date,contact_no number(12),d_id number(3));

Table created.

SQL> create table department(d_id number(3),d_name varchar2(4),hod
varchar2(5),e_mail varchar2(20) ,
  2  htno varchar2(10),foreign key(htno)references student2(htno));

Table created.
```

a) Insert three student details in two departments.

```
SQL> insert into student2 values('201f0001','ramu',57,'22-oct-
2001',9224589765,10);

1 row created.

SQL> insert into student2 values('201f0002','ragu',65,'22-jun-
2001',9224512345,10);

1 row created.

SQL> insert into student2 values('201f0003','roja',75,'12-jan-
2000',9224512377,10);

1 row created.

SQL> insert into student2 values('201k0001','sujatha',45,'11-jan-
2003',9224518887,20);

1 row created.
```

A) Insert 3 student details in 1 departments.

```
SQL> insert into student2 values('201f0001','ramu',57,'22-oct-
2001',9224589765,10);

1 row created.

SQL> insert into student2 values('201f0002','ragu',65,'22-jun-
2001',9224512345,10);

1 row created.

SQL> insert into student2 values('201f0003','roja',75,'12-jan-
2000',9224512377,null);

1 row created.
```

b) Insert data into department table and use commit,rollback and savepoint.

```
SQL> insert into department
values(10,'mca','ramu','abc@gmail.com','201f0001');

1 row created.

SQL> commit;

Commit complete.

SQL> select * from department;

      D_ID D_NA HOD   E_MAIL              HTNO
---------- ---- ----- -------------------- ----------
        10 mca  ramu  abc@gmail.com       201f0001
```

DEPARTMENT OF COMPUTER SCIENCE

Aim:

Create student(htno, name, percentage_of_marks, DOJ, did) and
Department(d_id, d_name, hod, contactno, e-mail) tables with relevant primary key, foreign key and other Constraints.

Perform the following :

a). Insert 3 student details in 1 department

b). Insert data into student table and use Commit, rollback and savepoint.

c). Alter any one of the field.

```
SQL> insert into department
values(10,'mca1','ramu','abc@gmail.com','201f0001');

1 row created.

SQL> rollback;

Rollback complete.

SQL> select * from department;

    D_ID D_NA HOD   E_MAIL              HTNO
---------- ---- ----- -------------------- ----------
      10 mca  ramu  abc@gmail.com       201f0001

SQL> savepoint a;

Savepoint created.

SQL> insert into department
values(10,'mca1','ramu','abc@gmail.com','201f0001');

1 row created.

SQL> savepoint b;

Savepoint created.

SQL> insert into department
values(10,'mca2','ramu','abc@gmail.com','201f0001');

1 row created.

SQL> select * from department;

    D_ID D_NA HOD   E_MAIL              HTNO
---------- ---- ----- -------------------- ----------
      10 mca1 ramu  abc@gmail.com      201f0001
      10 mca2 ramu  abc@gmail.com      201f0001
      10 mca ramu  abc@gmail.com  201f0001
SQL> rollback to b;
Rollback complete.
SQL> select * from department;

    D_ID D_NA HOD   E_MAIL              HTNO
---------- ---- ----- -------------------- ----------
      10 mca1 ramu  abc@gmail.com      201f0001
      10 mca ramu  abc@gmail.com  201f0001
```

SQL> select * from student2;

HTNO	SNAME	PERCENTAGE	DOJ	CONTACT_NO	D_ID
201f0001	ramu	57	22-OCT-01	9224589765	10
201f0002	ragu	65	22-JUN-01	9224512345	10
201f0003	roja	75	12-JAN-00	9224512377	

a) Find Average percentages of all students.

SQL> select avg(percentage) from student2;

AVG(PERCENTAGE)
65.6666667

b) Count the number of Students.

SQL> select count(*) from student2;

COUNT(*)
3

c) Find the Maxminum percentage .

SQL> select max(percentage) from student2;

MAX(PERCENTAGE)
75

d) Find Minimum Percentage

SQL> select max(percentage) from student2;

MAX(PERCENTAGE)
75

f) Calculate sum of percentages of all the students.

b) Display all students order by dept_no.

SQL> select * from student2 order by d_id desc;

HTNO	SNAME	PERCENTAGE	DOB	CONTACT_NO	D_ID
201k0002	srinivas	75	23-OCT-98	9224517887	20
201k0001	sujatha	45	11-JAN-03	9224518887	20
201k0003	krishna	85	23-OCT-97	922451789	20
201f0002	ragu	65	22-JUN-01	9224512345	10
201f0001	ramu	57	22-OCT-01	9224589765	10
201f0003	roja	75	12-JAN-00	9224512377	10

6 rows selected.

c) Display all students in each departent who has highest percentage.

SQL> select d_id,max(percentage) from student2 group by d_id;

D_ID	MAX(PERCENTAGE)
20	85
10	75

Suzy.

DEPARTMENT OF COMPUTER SCIENCE

Aim:

Queries Using Aggregate Function's :

(Count, SUM, AVG, MAX, and MIN).

```
SUBS
----
Inda

SQL> select ltrim(' ----- ramu','-')from dual;

LTRI
----
ramu

SQL> select rtrim('ramu----- ','-')from dual;

RTRI
---
ramu

SQL> select lpad('ramu',7,'*')from dual;

LPAD('R
-------
***ramu

SQL> select rpad('ramu',7,'*')from dual;

RPAD('R
-------
ramu***
```

B) Date Functions:-

```
SQL> select sysdate from dual;

SYSDATE
------
28-APR-22

SQL> select months_between(sysdate,'01-JAN-2021')from dual;

MONTHS_BETWEEN(SYSDATE,'01-JAN-2021')
-------------------------------------
                           15.8959943

SQL> select add_months(sysdate,10) from dual;

ADD_MONTH
---------
28-FEB-23
```

A) **String Functions**

```
SQL> select lower('RAMU') from dual;

LOWE
----
ramu

SQL> select upper('ramu')from dual;

UPPE
----
RAMU

SQL> select length('ramu') from dual;

LENGTH('RAMU')
--------------
             4

SQL> select initcap('sukumar atmaukuru') from dual;

INITCAP('SUKUMARA
-----------------
Sukumar Atmaukuru

SQL> select concat('govinda','7hills')from dual;

CONCAT('GOVIN
-------------
govinda7hills

SQL> select substr('govinda',1,4) from dual;

SUBS
----
govi

SQL> select substr('govinda',-4)from dual;
```

SQL> rollback to a;
Rollback complete.

SQL> select * from department;

D_ID	D_NA	HOD	E_MAIL	HTNO
10	mca	ramu	abc@gmail.com	201f0001

SQL>

c) alter any one of the field.

SQL> desc student;

Name	Null?	Type
ROLL_NO	NOT NULL	NUMBER(2)
SNAME		VARCHAR2(10)
DOB		DATE
AGE		NUMBER(2)
M_NO		NUMBER(11)

SQL> alter table student modify m_no number(14);

Table altered.

SQL> desc student;

Name	Null?	Type
ROLL_NO	NOT NULL	NUMBER(2)
SNAME		VARCHAR2(10)
DOB		DATE
AGE		NUMBER(2)
M_NO		NUMBER(14)

SQL> select sum(percentage) from student2;

SUM(PERCENTAGE)

197

Sum

SQL> select next_day(sysdate,'mon')from dual;

NEXT_DAY(

02-MAY-22

C) Conversion Functions

SQL> select to_char(sysdate,'d')from dual;

T

-

5

SQL> select to_char(sysdate,'dd')from dual;

TO

--

28

SQL> select to_char(sysdate,'ddd')from dual;

TO_

118

SQL> select to_char(sysdate,'yy')from dual;

TO

--

22

SQL> select to_char(sysdate,'mm')from dual;

TO

--

04

SQL> select to_char(sysdate,'DY')from dual;

TO_

THU

SQL> select to_char(sysdate,'MON')from dual;

TO_

APR

DEPARTMENT OF COMPUTER SCIENCE

Aim:

Queries Using String Function's:

(concatenation, lpad, rpad, ltrim, rtrim, lower, Upper, initCap, length, Substr, Instr)

Date Function's:

(sysdate, next_day, add_moths, last_day, months_between)

Conversion function's (to_char, to_date).

PL/SQL PROGRAM

```
SQL> select * from student1;

       SID SNAME          SUB1       SUB2       SUB3
---------- ---------- ---------- ---------- ----------
         1 srinu              75         64         85
         2 gopi               35         44         55
         3 ramu               55         84         95
         4 somu               24         54         36

SQL> declare
  2 type t1 is varray(10) of student1%rowtype;
  3 a t1:=t1();
  4 c number(2):=0;
  5 d number(4):=0;
  6 e exception;
  7 begin
  8 a.extend(9);
  9 select * bulk collect into a from student1;
 10 for i in a.first .. a.last
 11 loop
 12 d:=a(i).sub1+a(i).sub2+a(i).sub3;
 13 if d > 200 then
 14 dbms_output.put_line(a(i).sid||''||a(i).sname||''||a(i).sub1||'
'||a(i).sub2||''||a(i).sub3)
;
 15 c:=c+1;
 16 end if;
 17 end loop;
 18 if c=0 then
 19 raise e;
 20 end if;
 21 exception
 22 when e then
 23 dbms_output.put_line('No students with first grade');
 24 end;
 25 /
1 srinu 75 64 85
3 ramu 55 84 95
PL/SQL procedure successfully completed.
```

01/04/22

DEPARTMENT OF COMPUTER SCIENCE

Aim:

Creation of simple pl/sql program which includes declaration section, executable section, and exception handling section (Ex:- student marks can be selected from table and printed for those who secured first class and exception can be raised if the no records were found).

Aim:- a). Develop the program that includes the Feature NESTED IF.

PL/SQL PROGRAM

```
SQL> declare
  2  a number(3);
  3  b number(3);
  4  c number(3);
  5  begin
  6  a:=&a;
  7  b:=&b;
  8  c:=&c;
  9  if a>b and a>c then
 10  Dbms_output.put_line('A is big');
 11  elsif b>a and b>c then
 12  dbms_output.put_line('B id big');
 13  elsif c>a and c>b then
 14  dbms_output.put_line('C is big');
 15 else
 16  dbms_output.put_line('Three values are equal');
 17 end if;
 18  end;
 19  /
Enter value for a: 10
old  6: a:=&a;
new  6:  a:=10;
Enter value for b: 20
old  7: b:=&b;
new  7: b:=20;
Enter value for c: 30
old  8: c:=&c;
new 8: c:=30;
C is big

PL/SQL procedure successfully completed.

SQL> /
Enter value for a: 30
old  6: a:=&a;
new  6:  a:=30;
Enter value for b: 20
```

Aim:- Write pl/sql procedure for an Application Using Exception Handling?

PL/SQL PROGRAM

```
SQL> create table employee(eid number(7),ename varchar2(10),salary
number(7));

Table created.

SQL> insert into employee values(1,'suku',7000);

1 row created.

SQL> insert into employee values(2,'suresh',9000);

1 row created.

SQL> insert into employee values(3,'vasavi',10000);

1 row created.

SQL> select * from employee;

       EID ENAME      SALARY
---------- ---------- ----------
         1 suku         7000
         2 suresh       9000
         3 vasavi       10000

SQL> create or replace procedure exp7(a number)
  2  is
  3  b number(3);
  4  c varchar2(10);
  5  d number(5);
  6  begin
  7  select eid,ename,salary into b,c,d from employee where eid=a;
  8  dbms_output.put_line('Employee id:' ||'-'||b);
  9  dbms_output.put_line('Employee salary:'||'-'||d);
 10  dbms_output.put_line('Employee Name:'||'-'||c);
 11  exception
```

SQL> select to_char(sysdate,'MONTH')from dual;

TO_CHAR(S

APRIL

SQL> select to_date('28-apr-2022')from dual;

TO_DATE('

28-APR-2

Suny

```
4  choice number(2);


5  result number(2);
 6  begin
 7  dbms_output.put_line('Enter the two values:');
 8  first:=&first;
 9  second:=&second;
10  choice:=&choice;
11  case
12  when choice=1 then
13  choice:=first+second;
14 dbms_output.put_line('Sum is:'||' '||choice);
15 when choice=2 then
16  choice:=first-second;
17  dbms_output.put_line('Difference is:'||' '||choice);
18 when choice=3 then
19 dbms_output.put_line('Product is:'||' '||choice);
20 else
21 dbms_output.put_line('You have given rong choice');
22 end case;
23  end;
24  /

Enter value for first: 10
old  8: first:=&first;
new  8: first:=10;
Enter value for second: 20
old  9: second:=&second;
new  9: second:=20;
Enter value for choice: 1
old  10: choice:=&choice;
new 10: choice:=1;
Enter the two values:
Sum is: 30

PL/SQL procedure successfully completed.
SQL> /

Enter value for first: 20
old  8: first:=&first;
new  8: first:=20;
Enter value for second: 10
old  9: second:=&second;
new  9: second:=10;
Enter value for choice: 2
old  10: choice:=&choice;
new 10: choice:=2;
Enter the two values:
Difference is: 10
```

```
PL/SQL procedure successfully completed.
SQL> /

Enter value for first: 10
old  8: first:=&first;
new  8: first:=10;
Enter value for second: 5
old  9: second:=&second;
new  9:  second:=5;
Enter value for choice: 3
old  10: choice:=&choice;
new 10: choice:=3;
Enter the two values:
Product is: 3

PL/SQL procedure successfully completed.
SQL> /

Enter value for first: 10
old  8: first:=&first;
new  8: first:=10;
Enter value for second: 10
old  9: second:=&second;
new  9: second:=10;
Enter value for choice: 4
old  10: choice:=&choice;
new 10: choice:=4;
Enter the two values:
You have given wrong choice
PL/SQL procedure successfully completed
```

```
old  7: b:=&b;
new  7: b:=20;


Enter value for c: 10
old  8: c:=&c;


new 8: c:=10;
A is big


PL/SQL procedure successfully completed.
SQL> /

Enter value for a: 20
old  6: a:=&a;
new   6:  a:=20;
Enter value for b: 30
old  7: b:=&b;
new   7: b:=30;
Enter value for c: 10
old  8: c:=&c;
new  8: c:=10;
B id big

PL/SQL procedure successfully completed.

SQL> /
Enter value for a: 10
old  6: a:=&a;
new   6:  a:=10;
Enter value for b: 10
old  7: b:=&b;
new   7: b:=10;
Enter value for c: 10
old  8: c:=&c;
new  8: c:=10;
Three values are equal

PL/SQL procedure successfully completed.

SQL>
```

B) Develop a Program that includes the features CASE statement.

```
SQL> declare
 2  first number(2);
 3  second number(2);
```

Aim:- Write a PL/SQL block for transaction operations of typical application using triggers.

PL/SQL PROGRAM

```
SQL> create or replace trigger exp9
  2  after update on employee
  3  begin
  4  dbms_output.put_line(
  5  'Trigger Raised');
  6  end;
  7  /

Trigger created.

SQL> select * from employee;

       EID ENAME          SALARY
---------- ---------- ----------
         1 suku             7000
         2 suresh           9000
         3 vasavi          10000
         2 suresh           7000

SQL> set serveroutput on;

SQL> begin
  2  update employee set ename='venkat' where eid=1;
  3  end;
  4  /

Trigger Raised

PL/SQL procedure successfully completed.
```

The package is database object which encapsulates procedures,functions,cursors,global variables into single unit. In oracle , every package having 2 parts. These are

1) package specification.
2) package body.

Syntax to package specification:

Create or replace package packagename

Aim:- Write PL/SQL Procedure for an Application Using Cursors.

PL/SQL PROGRAM

```
SQL> create or replace procedure exp8
 2 is
 3 cursor c1 is select * from employee;
 4 r employee%rowtype;
 5 begin
 6 open c1;
 7 dbms_output.put_line('eid       ename    salary');
 8 dbms_output.put_line('-------------------------- ');
 9 loop
10  fetch c1 into r;
11  exit when c1%notfound;
12  dbms_output.put_line(r.eid || r.ename || r.salary);
13 end loop;
14 close c1;
15 end;
16 /

Procedure created.

SQL> set serveroutput on

SQL> call exp8();

eid     ename    salary
-------------------------
1       suku  7000
2       suresh  9000
3       vasavi  10000
2       suresh  7000

Call completed.
```

```
12  when no_data_found then
13  dbms_output.put_line('Employee does not exist');
14 when too_many_rows then
15  dbms_output.put_line('Many employees are existing with given id');


16  end;
17  /
Procedure created.
SQL> set serveroutput on



SQL> call exp7(1);

Employee id:-1
Employee salary:-7000
Employee Name:-suku
Call completed.
SQL> call exp7(8);

Employee does not exist
Call completed.

SQL> insert into employee values(2,'suresh',7000);

1 row created.

SQL> call exp7(2);

Many employees are existing with given id
Call completed.
```

Suku

Is/as
Global variables declarations;
Cursor declarations;
Procedure declarations;

Function declarations;
End;

Syntax to Package body:

Create or replace package body packagename
Is
Procedure implementation;
Function implementation;
End;

Suny

Aim:- Demonstrate various Joins by create two tables and entries at least 2 records in each table.

SQL> select * from sailors;

SID	SNAME	RATING	AGE
22	dustin	7	45
31	lubber	7	45
58	Rusty	10	35

SQL> select * from reserves;

SID	BID	DAY
22	10	10-OCT-96
58	13	11-DEC-96

1) EQUI JOIN:-

```
SQL> select s.sid,s.sname,s.rating,s.age,r.bid,r.day
 2  from sailors s ,reserves r
 3  where s.sid=r.sid;
```

SID	SNAME	RATING	AGE	BID	DAY
22	dustin	7	45	10	10-OCT-96
58	Rusty	10	35	13	11-DEC-96

(or)

SQL> select * from sailors join reserves using(sid);

SID	SNAME	RATING	AGE	BID	DAY
22	dustin	7	45	10	10-OCT-96
58	Rusty	10	35	13	11-DEC-96

2) Natural Joins:-

SQL> select * from sailors natural join reserves;

```
22  dbms_output.put_line('Current Balance:'||'-'||c_bal);
23  exception
24  when no_data_found then
25  dbms_output.put_line('Account does not exist');
26 end with_draw;
27  end;
28  /

Package body created.

SQL> begin
 2  dbms_output.put_line('Deposit');
 3  bank_operations.deposit(&account_number,&ammount);
 4  dbms_output.put_line('WithDraw');
 5  bank_operations.with_draw(&account_number,&ammount);
 6  end;
 7  /

Enter value for account_number: 10
Enter value for ammount: 200
old  3: bank_operations.deposit(&account_number,&ammount);
new  3: bank_operations.deposit(10,200);
Enter value for account_number: 20
Enter value for ammount: 400
old  5: bank_operations.with_draw(&account_number,&ammount);
new  5: bank_operations.with_draw(20,400);
Deposit
Current Balance:-7200
WithDraw
Current Balance:-8400

PL/SQL procedure successfully completed.

SQL> select * from account;

    ACNO CNAME                BAL
-------- -------------------------
      10 suresh               7200
      20 venkat               8400
```

Suresh

Aim:- Write a PL/SQL block for transaction Operation's of typical application Using package.

PL/SQL PROGRAM

```
SQL> select * from account;

    ACNO CNAME          BAL
-------- ---------- ----------
      10 suresh          7000
      20 venkat          9000

SQL> create or replace package bank_operations
  2 is
  3 c_bal number(5):=0;
  4 procedure deposit(a_no number,amt number);
  5 procedure with_draw(a_no number,amt number);
  6 end;
  7 /

Package created.

SQL> create or replace package body bank_operations
  2 is
  3 procedure deposit(a_no number,amt number)
  4 is
  5 begin
  6 select bal into c_bal from account where acno=a_no;
  7 c_bal:=c_bal+amt;
  8 update account set bal=c_bal where acno=a_no;
  9 commit;
 10 dbms_output.put_line('Current Balance:'||'-'||c_bal);
 11 exception
 12 when no_data_found then
 13 dbms_output.put_line('Account does not exist');
 14 end deposit;
 15 procedure with_draw(a_no number,amt number)
 16 is
 17 begin
 18 select bal into c_bal from account where acno=a_no;
 19 c_bal:=c_bal-amt;
 20 update account set bal=c_bal where acno=a_no;
 21 commit;
```

SID	SNAME	RATING	AGE	BID	DAY
22	dustin	7	45	10	10-OCT-96
58	Rusty	10	35	13	11-DEC-96

3.left Outer Join:-

SQL> select * from sailors left outer join reserves on sailors.sid=reserves.sid;

SID	SNAME	RATING	AGE	SID	BID	DAY
22	dustin	7	45	22	10	10-OCT-96
58	Rusty	10	35	58	13	11-DEC-96
31	lubber	7	45			

4)Full-outer join:-

SQL> select * from sailors full outer join reserves on sailors.sid=reserves.sid;

SID	SNAME	RATING	AGE	SID	BID	DAY
22	dustin	7	45	22	10	10-OCT-96
31	lubber	7	45			
58	Rusty	10	35	58	13	11-DEC-96