Lab – 7 : Dipesh Singh – 190905520

Question 1 : The HRD manager has decided to raise the salary of all the Instructors in a given department number by 5%. Whenever, any such raise is given to the instructor, a record for the same is maintained in the salary_raise table. It includes the Instuctor Id, the date when the raise was given and the actual raise amount. Write a PL/SQL block to update the salary of each Instructor and insert a record in the salary_raise table.

salary_raise(Instructor_Id, Raise_date, Raise_amt)

```
create table salaryraise(
    id number(8),
    raise_date date,
    raise_amt number(8)
);
declare dt constant varchar(20) := '09/06/2021';
cursor c is
select *
from instructor;
begin for ins in c loop
insert into salaryraise
values(
        ins.id,
        to_date(dt, 'dd/mm/yyyy'),
        ins.salary * 0.05
    );
end loop;
update instructor
set salary = salary * 1.05;
end;
```

```
SQL> declare dt constant varchar(20) := '09/06/2021';
  2  cursor c is
  3  select *
  4  from instructor;
  5  begin for ins in c loop
  6  insert into salaryraise
  7  values(
  8             ins.id,
  9             to_date(dt, 'dd/mm/yyyy'),
 10             ins.salary * 0.05
 11     );
 12  end loop;
 13  update instructor
 14  set salary = salary * 1.05;
 15  end;
 16  /

PL/SQL procedure successfully completed.

SQL> select * from salaryraise;

        ID RAISE_DAT  RAISE_AMT
---------- --------- ----------
     63395 09-JUN-21       4717
     78699 09-JUN-21       2965
     96895 09-JUN-21       5996
      4233 09-JUN-21       4440
      4034 09-JUN-21       3069
     50885 09-JUN-21       1629
     79653 09-JUN-21       4490
     50330 09-JUN-21       5401
     80759 09-JUN-21       2277
```

Question 2 : Write a PL/SQL block that will display the ID, name, dept_name and tot_cred of the first 10 students with lowest total credit.

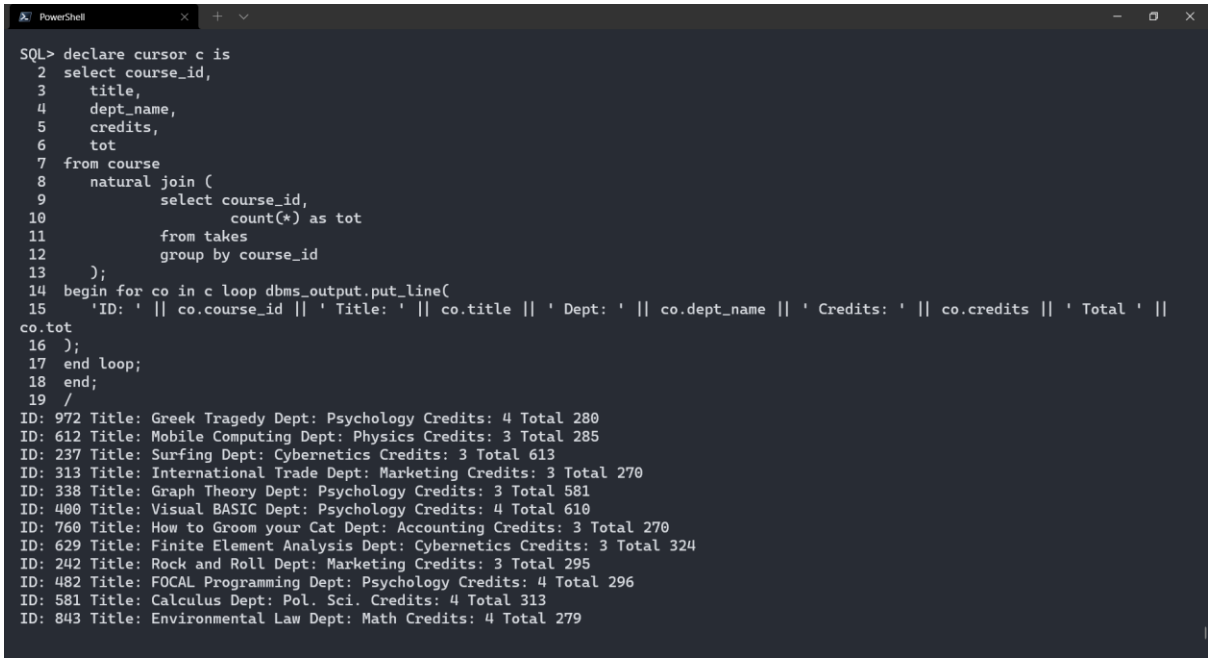
```
declare cursor c is
select *
from student
order by tot_cred asc;
stu student %rowtype;
cnt number(5);
begin cnt := 0;
open c;
loop fetch c into stu;
dbms_output.put_line(
    'ID:' || stu.id || ' Name:' || stu.name || ' Dept:' || stu.dept_name
|| ' Credits:' || stu.tot_cred
);
cnt := cnt + 1;
exit
when cnt >= 10;
end loop;
close c;
end;
```

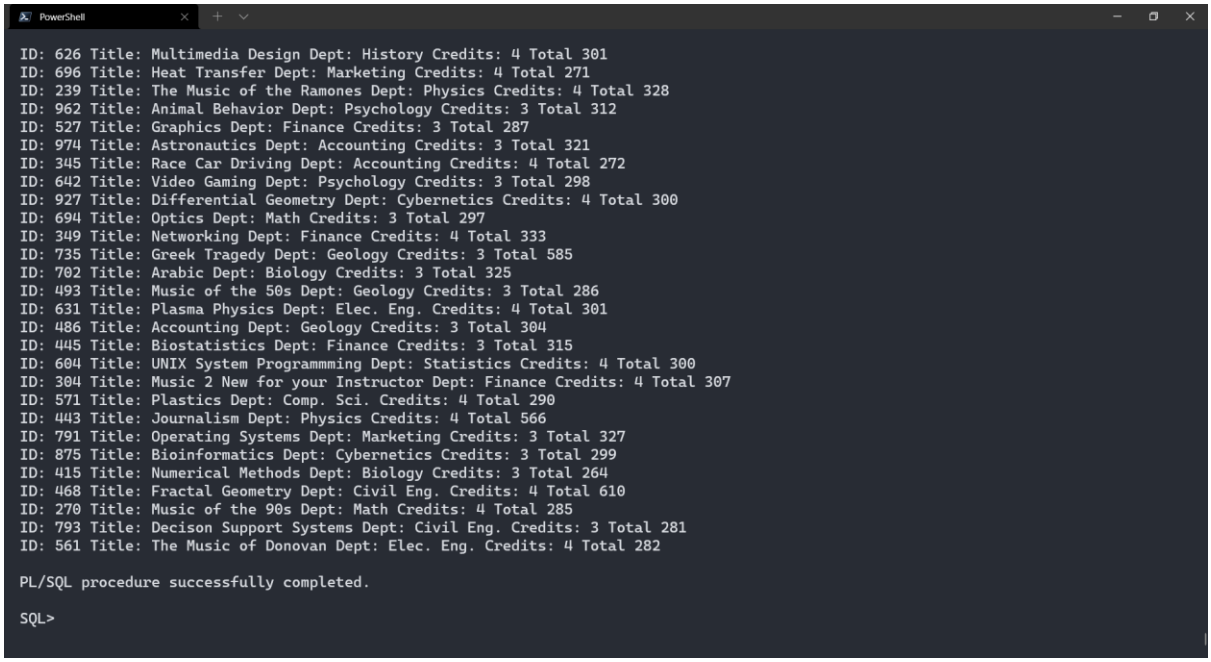
```
SQL> declare cursor c is
  2  select *
  3  from student
  4  order by tot_cred asc;
  5  stu student %rowtype;
  6  cnt number(5);
  7  begin cnt := 0;
  8  open c;
  9  loop fetch c into stu;
 10  dbms_output.put_line(
 11     'ID:' || stu.id || ' Name:' || stu.name || ' Dept:' || stu.dept_name || ' Credits:' || stu.tot_cred
 12  );
 13  cnt := cnt + 1;
 14  exit
 15  when cnt ≥ 10;
 16  end loop;
 17  close c;
 18  end;
 19  /
ID:4582 Name:Zaniolo Dept:Math Credits:0
ID:93571 Name:Kato Dept:Psychology Credits:0
ID:59908 Name:Cox Dept:Civil Eng. Credits:0
ID:48053 Name:Macias Dept:Comp. Sci. Credits:0
ID:82646 Name:Nirenbu Dept:Biology Credits:0
ID:39157 Name:Loull Dept:Accounting Credits:0
ID:14032 Name:Belhadji Dept:Elec. Eng. Credits:0
ID:81610 Name:Ching Dept:Languages Credits:0
ID:81175 Name:Zelek Dept:Biology Credits:0
ID:11201 Name:Bianchi Dept:Statistics Credits:0

PL/SQL procedure successfully completed.
```

Question 3 : Print the Course details and the total number of students registered for each course along with the course details -(Course-id, title, dept-name, credits, tot_student_no)

```
declare cursor c is
select course_id,
    title,
    dept_name,
    credits,
    tot
from course
    natural join (
        select course_id,
            count(*) as tot
        from takes
        group by course_id
    );
begin for co in c loop dbms_output.put_line(
    'ID: ' || co.course_id || ' Title: ' || co.title || ' Dept: ' || co.d
ept_name || ' Credits: ' || co.credits || ' Total ' || co.tot
);
end loop;
end;
```





Question 4 : Find all students who take the course with Course-id: 747 and if he/ she has less than 30 total credit (tot-cred), deregister the student from that course. (Delete the entry in Takes table)

```
declare cursor c is
select *
from takes
where course_id = '747';
cre student.tot_cred %type;
cnt number(8);
begin cnt := 0;
for s in c loop
select tot_cred into cre
from student
where id = s.id;
if cre < 30 then
delete from takes
where course_id = '747'
    and id = s.id;
dbms_output.put_line('deleted : ' || s.id || ' credits : ' || cre);
cnt := cnt + 1;
end if;
end loop;
dbms_output.put_line(
    cnt || ' students de-enrolled from the course 747'
);
end;
```

```
SQL> declare cursor c is
  2  select *
  3  from takes
  4  where course_id = '747';
  5  cre student.tot_cred %type;
  6  cnt number(8);
  7  begin cnt := 0;
  8  for s in c loop
  9  select tot_cred into cre
 10  from student
 11  where id = s.id;
 12  if cre < 30 then
 13  delete from takes
 14  where course_id = '747'
 15     and id = s.id;
 16  dbms_output.put_line('deleted : ' || s.id || ' credits : ' || cre);
 17  cnt := cnt + 1;
 18  end if;
 19  end loop;
 20  dbms_output.put_line(
 21     cnt || ' students de-enrolled from the course 747'
 22  );
 23  end;
 24  /
deleted : 74016 credits : 15
deleted : 1110 credits : 23
deleted : 90779 credits : 24
deleted : 73542 credits : 8
deleted : 28518 credits : 20
deleted : 15883 credits : 24
deleted : 14065 credits : 7
deleted : 70021 credits : 16
deleted : 16405 credits : 5
deleted : 89393 credits : 21
deleted : 94173 credits : 16
```

```
deleted : 2629 credits : 4
deleted : 24796 credits : 18
deleted : 40303 credits : 25
deleted : 26494 credits : 28
deleted : 90181 credits : 23
deleted : 83022 credits : 10
deleted : 42843 credits : 18
deleted : 69752 credits : 24
deleted : 19245 credits : 4
deleted : 77729 credits : 26
deleted : 44258 credits : 28
deleted : 85746 credits : 5
deleted : 41683 credits : 20
deleted : 66008 credits : 25
deleted : 6195 credits : 20
deleted : 13757 credits : 3
deleted : 64914 credits : 17
deleted : 65979 credits : 22
deleted : 79210 credits : 14
deleted : 32217 credits : 14
deleted : 98830 credits : 13
deleted : 84167 credits : 20
deleted : 18809 credits : 20
deleted : 70452 credits : 18
deleted : 68330 credits : 13
deleted : 78481 credits : 23
deleted : 7514 credits : 11
deleted : 85226 credits : 18
deleted : 30845 credits : 19
deleted : 68150 credits : 3
65 students de-enrolled from the course 747

PL/SQL procedure successfully completed.

SQL>
```

Question 5 : Alter StudentTable(refer Lab No. 8 Exercise)by resetting column LetterGrade to F. Then write a PL/SQL block to update the table by mapping GPA to the corresponding letter grade foreach student.

```
declare cursor c is
select *
from studenttable for
update;
begin for stu in c loop if stu.gpa > 4
and stu.gpa <= 5 then
update studenttable
set grade = 'e'
where current of c;
elsif stu.gpa > 5
and stu.gpa <= 6 then
update studenttable
set grade = 'd'
where current of c;
elsif stu.gpa > 6
and stu.gpa <= 7 then
update studenttable
set grade = 'c'
where current of c;
elsif stu.gpa > 7
and stu.gpa <= 8 then
update studenttable
set grade = 'b'
where current of c;
elsif stu.gpa > 8
and stu.gpa <= 9 then
update studenttable
set grade = 'a'
where current of c;
elsif stu.gpa > 9
and stu.gpa <= 10 then
update studenttable
set grade = 'a+'
where current of c;
end if;
end loop;
end;
```

```
SQL> declare cursor c is
  2  select *
  3  from studenttable for
  4  update;
  5  begin for stu in c loop if stu.gpa > 4
  6  and stu.gpa ≤ 5 then
  7  update studenttable
  8  set grade = 'e'
  9  where current of c;
 10  elsif stu.gpa > 5
 11  and stu.gpa ≤ 6 then
 12  update studenttable
 13  set grade = 'd'
 14  where current of c;
 15  elsif stu.gpa > 6
 16  and stu.gpa ≤ 7 then
 17  update studenttable
 18  set grade = 'c'
 19  where current of c;
 20  elsif stu.gpa > 7
 21  and stu.gpa ≤ 8 then
 22  update studenttable
 23  set grade = 'b'
 24  where current of c;
 25  elsif stu.gpa > 8
 26  and stu.gpa ≤ 9 then
 27  update studenttable
 28  set grade = 'a'
 29  where current of c;
 30  elsif stu.gpa > 9
 31  and stu.gpa ≤ 10 then
 32  update studenttable
 33  set grade = 'a+'
 34  where current of c;
 35  end if;
 36  end loop;
 37  end;
 38  /

PL/SQL procedure successfully completed.
```

```
PL/SQL procedure successfully completed.

SQL> select * from studenttable;

    ROLLNO        GPA GR
---------- ---------- --
         1        5.8 d
         2        6.5 c
         3        3.4 f
         4        7.8 b
         5        9.5 a+
         6        -.2 f

6 rows selected.

SQL>
```

Question 6 : Write a PL/SQL block to print the list of Instructors teaching a specified course.

```
declare cursor c(courseid teaches.course_id %type) is
select id
from teaches
where course_id = courseid;
i instructor %rowtype;
begin for ins in c(& courseid) loop
select * into i
from instructor
where id = ins.id;
dbms_output.put_line(
    'ID: ' || i.id || ' Name: ' || i.name || ' Dept: ' || i.dept_name ||
' Salary: ' || i.salary
);
end loop;
end;
/
```

```
SQL> declare cursor c(courseid teaches.course_id %type) is
  2  select id
  3  from teaches
  4  where course_id = courseid;
  5  i instructor %rowtype;
  6  begin for ins in c(& courseid) loop
  7  select * into i
  8  from instructor
  9  where id = ins.id;
 10  dbms_output.put_line(
 11     'ID: ' || i.id || ' Name: ' || i.name || ' Dept: ' || i.dept_name || ' Salary: ' || i.salary
 12  );
 13  end loop;
 14  end;
 15  /
Enter value for courseid: '274'
old   6: begin for ins in c(& courseid) loop
new   6: begin for ins in c('274') loop
ID: 34175 Name: Bondi Dept: Comp. Sci. Salary: 121242.57

PL/SQL procedure successfully completed.

SQL>
```

Question 7 : Write a PL/SQL block to list the students who have registered for a course taught by his/her advisor.

```
declare cursor a is
select unique t.id as s,
    s.id as i
from takes t,
    teaches s
where t.course_id = s.course_id;
cursor b(s student.id %type, i instructor.id %type) is
select unique s_id
from advisor
where s_id = s
    and i_id = i;
st student %rowtype;
cnt number(8);
begin cnt := 0;
for tuple in a loop for stu in b(tuple.s, tuple.i) loop
select * into st
from student
where id = stu.s_id;
dbms_output.put_line(st.name || ' ' || st.id || ' ' || st.dept_name);
cnt := cnt + 1;
end loop;
end loop;
dbms_output.put_line(cnt || ' rows selected');
end;
/
```

```
SQL> declare cursor a is
  2  select unique t.id as s,
  3     s.id as i
  4  from takes t,
  5     teaches s
  6  where t.course_id = s.course_id;
  7  cursor b(s student.id %type, i instructor.id %type) is
  8  select unique s_id
  9  from advisor
 10  where s_id = s
 11     and i_id = i;
 12  st student %rowtype;
 13  cnt number(8);
 14  begin cnt := 0;
 15  for tuple in a loop for stu in b(tuple.s, tuple.i) loop
 16  select * into st
 17  from student
 18  where id = stu.s_id;
 19  dbms_output.put_line(st.name || ' ' || st.id || ' ' || st.dept_name);
 20  cnt := cnt + 1;
 21  end loop;
 22  end loop;
 23  dbms_output.put_line(cnt || ' rows selected');
 24  end;
 25  /
Rote 82688 Cybernetics
Boulah 39115 Civil Eng.
Okaf 10663 Geology
Dhav 108 Biology
Lin 42843 Mech. Eng.
Zuyev 38121 English
Jordan 84239 Languages
```

```
Qvi 66229 Civil Eng.
Seyfert 25718 Athletics
McDonald 87044 Accounting
Cacciari 89297 Astronomy
Tsantis 42556 Languages
Mathur 28538 Statistics
Andert 85809 Geology
Fournier 20002 Accounting
Franchet 86736 Finance
Peip 82970 Mech. Eng.
Yap 77898 Marketing
Bersk 50386 Elec. Eng.
Otsuki 72165 Psychology
Sakamoto 99760 Athletics
Cox 21766 Astronomy
Dooley 70965 Languages
Kashima 858 Psychology
Teng 45826 Mech. Eng.
Savelieva 28004 Finance
Marlet 78767 Geology
Curutchet 32954 Languages
Roses 65681 English
Moskow 435 Languages
Kawakami 33645 Comp. Sci.
Bravo 99780 English
Arndt 34569 Accounting
Youseffi 30021 History
447 rows selected

PL/SQL procedure successfully completed.

SQL>
```

Question 8 : Write a PL/SQL block that updates the salary of ‘Biology’ department instructors by 20%. Subsequently, check the whether the department budget can support the raise. If not, undo the raise given to the instructors.

```
declare cursor c is
select *
from instructor
where dept_name = 'Biology' for update;
cnt number(20);
temp number(20);
begin savepoint a;
cnt := 0;
for ins in c loop cnt := cnt + ins.salary * 1.2;
update instructor
set salary = salary * 1.2
where current of c;
end loop;
select budget into temp
from department
where dept_name = 'Biology';
if temp < cnt then rollback to savepoint a;
else commit;
end if;
end;
/
```

```
SQL> declare cursor c is
  2  select *
  3  from instructor
  4  where dept_name = 'Biology' for update;
  5  cnt number(20);
  6  temp number(20);
  7  begin savepoint a;
  8  cnt := 0;
  9  for ins in c loop cnt := cnt + ins.salary * 1.2;
 10  update instructor
 11  set salary = salary * 1.2
 12  where current of c;
 13  end loop;
 14  select budget into temp
 15  from department
 16  where dept_name = 'Biology';
 17  if temp < cnt then rollback to savepoint a;
 18  else commit;
 19  end if;
 20  end;
 21  /

PL/SQL procedure successfully completed.

SQL> select * from instructor where dept_name='Biology';

ID    NAME                 DEPT_NAME               SALARY
----- -------------------- -------------------- ----------
80759 Queiroz              Biology                57378.29
81991 Valtchev             Biology                97065.59

SQL> select * from department where dept_name='Biology';

DEPT_NAME            BUILDING            BUDGET
-------------------- --------------- ----------
Biology              Candlestick      647610.55

SQL>
```