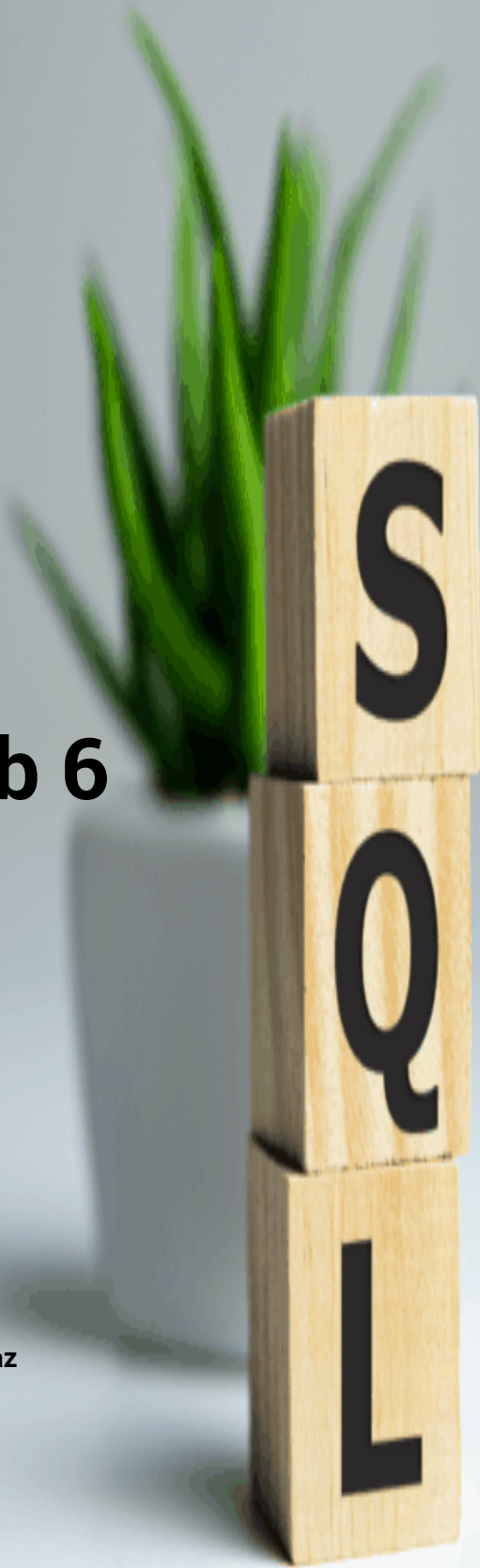


PLSQL Lab 6

Aya Sabry

Submitted to: Eng. Yahia Momtaz

Dec 15, 2022



Task

1.

1. Create plsql block and to check for all employees using cursor; and update their commission_pct based on the salary

SALARY < 7000 :	COMM = 0.1
7000 <= SALARY < 10000	COMM = 0.15
10000 <= SALARY < 15000	COMM = 0.2
15000 <= SALARY	COMM = 0.25

```
1  set serveroutput on
2  ----AYA SABRY LAB 6
3  declare
4      cursor emp_cursor is
5          select * from employees;
6  begin
7      for emp_record in emp_cursor loop
8          if emp_record.salary <7000 then
9              emp_record.COMMISSION_PCT :=0.1;
10             elsif emp_record.salary >=7000 and emp_record.salary <10000 then
11                 emp_record.COMMISSION_PCT :=0.15;
12             elsif emp_record.salary >=10000 and emp_record.salary <15000 then
13
14                 emp_record.COMMISSION_PCT :=.2;
15             else
16                 emp_record.COMMISSION_PCT:=.25;
17             end if;
18             dbms_output.put_line('Id = '||emp_record.employee_id|| ' , Name = '||emp_record.last_name||',Salary
19                 update employees
20                 set COMMISSION_PCT = emp_record.COMMISSION_PCT
21                 where employee_id= emp_record.employee_id;
22                 commit;
23             end loop;
24 end;
25 --select employee_id,COMMISSION_PCT from employees;
```

Id = 106 , Name = Pataballa ,Salary 4800 , His COMM.1
 Id = 108 , Name = Greenberg ,Salary 12008 , His COMM.2
 Id = 106 , Name = Pataballa ,Salary 4800 , His COMM.1
 Id = 108 , Name = Greenberg ,Salary 12008 , His COMM.2
 Id = 109 , Name = Faviert ,Salary 9000 , His COMM.15
 Id = 111 , Name = Sciarra ,Salary 7700 , His COMM.15
 Id = 113 , Name = Popp ,Salary 6900 , His COMM.1
 Id = 114 , Name = Raphaely ,Salary 11000 , His COMM.2
 Id = 116 , Name = Baida ,Salary 2900 , His COMM.1
 Id = 117 , Name = Tobias ,Salary 2800 , His COMM.1
 Id = 119 , Name = Colmenares ,Salary 2500 , His COMM.1
 Id = 121 , Name = Fripp ,Salary 8200 , His COMM.15
 Id = 122 , Name = Kaufling ,Salary 7900 , His COMM.15
 Id = 124 , Name = Mourgos ,Salary 5800 , His COMM.1
 Id = 126 , Name = Mikkilineni ,Salary 2700 , His COMM.1
 Id = 127 , Name = Landry ,Salary 2400 , His COMM.1
 Id = 129 , Name = Bissot ,Salary 3300 , His COMM.1
 Id = 130 , Name = Atkinson ,Salary 2800 , His COMM.1
 Id = 132 , Name = Olson ,Salary 2100 , His COMM.1
 Id = 134 , Name = Rogers ,Salary 2900 , His COMM.1
 Id = 135 , Name = Gee ,Salary 2400 , His COMM.1
 Id = 137 , Name = Ladwig ,Salary 3600 , His COMM.1
 Id = 138 , Name = Stiles ,Salary 3200 , His COMM.1
 Id = 140 , Name = Patel ,Salary 2500 , His COMM.1
 Id = 142 , Name = Davies ,Salary 3100 , His COMM.1

EMPLOYEE_ID	COMMISSION_PCT
111	0.15
113	0.1
114	0.2
116	0.1
117	0.1
119	0.1
121	0.15
122	0.15
124	0.1
126	0.1
127	0.1
129	0.1
130	0.1

2.

Alter table employees then add column retired_bonus

Create plsql block to calculate the retired salary for all employees using cursor and update retired_bonus column

Retired bonus = no of working months * 10 % of his current salary

Only for those employees have passed 18 years of their hired date

```
---AYA SABRY LAB 6 -----
declare
cursor c_emp2 is
SELECT* FROM employees where (trunc( months_between(sysdate,hire_date) / 12 )) >18 ;
begin
FOR emp_record IN c_emp2 LOOP
update employees
set retired_bonus=trunc( months_between(sysdate,hire_date)) *(0.1*salary)
where employee_id= emp_record.employee_id;
commit;
dbms_output.put_line('ID = '||emp_Record.employee_id||' , Salary : '||emp_Record.salary||' Working Years : '||(trunc( months_between(sysdate,emp_record.hire_date) / 12 ))||
' ,Retired Bonus: '||emp_Record.retired_bonus);
dbms_output.put_line('_____');
END LOOP;
END;
/*
ALTER TABLE employees
ADD retired_bonus number(8,2) default 0;
*/
```

ID = 108 , Salary : 12008 Working Years : 20 ,Retired Bonus: 291794.4
ID = 109 , Salary : 9000 Working Years : 20 ,Retired Bonus: 218700
ID = 114 , Salary : 11000 Working Years : 20 ,Retired Bonus: 264000
ID = 122 , Salary : 7900 Working Years : 19 ,Retired Bonus: 185650
ID = 137 , Salary : 3600 Working Years : 19 ,Retired Bonus: 83520
ID = 100 , Salary : 24000 Working Years : 19 ,Retired Bonus: 559200
ID = 102 , Salary : 17000 Working Years : 21 ,Retired Bonus: 445400
ID = 115 , Salary : 3100 Working Years : 19 ,Retired Bonus: 72540
ID = 141 , Salary : 3500 Working Years : 19 ,Retired Bonus: 80150
ID = 200 , Salary : 4400 Working Years : 19 ,Retired Bonus: 101200
ID = 203 , Salary : 6500 Working Years : 20 ,Retired Bonus: 159900
ID = 204 , Salary : 10000 Working Years : 20 ,Retired Bonus: 246000
ID = 205 , Salary : 12008 Working Years : 20 ,Retired Bonus: 295396.8

3.

Create plsql block using cursor to print last name, department name, city, country name for all employees employee (without using join | sub query)

```
1  set serveroutput on
2  ----AYA SABRY LAB 6-----
3  declare
4      cursor emp_cursore is
5          select * from employees
6          where department_id is not null;
7  v_LOCATION_ID LOCATIONS.LOCATION_ID%type;
8  v_department_id number(4);
9  v_COUNTRY_ID LOCATIONS.COUNTRY_ID%type;
10 v_city LOCATIONS.city%type;
11 v_COUNTRY_NAME COUNTRIES.COUNTRY_NAME%type;
12
13 v_department_name varchar2(100);
14 begin
15
16     for emp_record in emp_cursore loop
17         dbms_output.put_line('ID = '||emp_Record.employee_id);
18         dbms_output.put_line('Emp name is : '||emp_Record.last_name);
19         select department_id
20             into v_department_id
21             from employees
22             where employee_id = emp_Record.employee_id;
23         select department_name, LOCATION_ID into v_department_name, v_LOCATION_ID
24             from departments
25             where department_id = v_department_id;
26
27         -- emp_name --
28         where employee_id = emp_Record.employee_id;
29         select department_name, LOCATION_ID into v_department_name, v_LOCATION_ID
30             from departments
31             where department_id = v_department_id;
32         dbms_output.put_line('Department Name is '||v_department_name);
33         select city, COUNTRY_ID into v_city, v_COUNTRY_ID
34             from locations
35             where LOCATION_ID=v_LOCATION_ID;
36         dbms_output.put_line('City = '||v_city);
37         select COUNTRY_NAME into v_COUNTRY_NAME
38             from countries
39             where COUNTRY_ID=v_COUNTRY_ID;
40         dbms_output.put_line('Country Name= '||v_COUNTRY_NAME);
41         dbms_output.put_line('_____ ');
42     end loop;
43 end;
```

1	ID = 106
2	Emp name is : Pataballa
3	Department Name is IT
4	City = Southlake
5	Country Name= United States of America
6	
7	ID = 108
8	Emp name is : Greenberg
9	Department Name is Finance
10	City = Seattle
11	Country Name= United States of America
12	
13	ID = 109
14	Emp name is : Faviet
15	Department Name is Finance
16	City = Seattle
17	Country Name= United States of America
18	
19	ID = 111
20	Emp name is : Sciarra
21	Department Name is Finance
22	City = Seattle
23	Country Name= United States of America
24	
25	ID = 113
26	Emp name is : Deena

4.

Create plsql block that loop over employees table and

Increase only those working in 'IT' department by 10% of their salary.

```

set serveroutput on size 1000000
----AYA SABRY LAB 6----
declare

    cursor c_emp1 is
    SELECT* FROM employees WHERE job_id like '%IT%';
begin
    FOR emp_record IN c_emp1 LOOP
        update employees
        set salary=salary +(0.1*salary)
        where employee_id= emp_record.employee_id;
        commit;

        dbms_output.put_line('ID = '||emp_Record.employee_id);
        dbms_output.put_line('Emp Salary after 10% is : '||emp_Record.salary);
        dbms_output.put_line('HIS JOB : '||emp_Record.job_id);
        dbms_output.put_line('_____');
    END LOOP;
END;
```

```
ID = 106
Emp Salary is : 4800
HIS JOB : IT_PROG

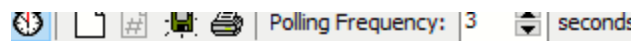
ID = 207
Emp Salary is : 18000
HIS JOB : IT_PROG

ID = 103
Emp Salary is : 9000
HIS JOB : IT_PROG

ID = 104
Emp Salary is : 17000
HIS JOB : IT_PROG

ID = 105
Emp Salary is : 4800
HIS JOB : IT_PROG

ID = 107
Emp Salary is : 4200
HIS JOB : IT_PROG
```

 Polling Frequency: 3 seconds

```
ID = 106
Emp Salary after 10% is : 5280
HIS JOB : IT_PROG

ID = 207
Emp Salary after 10% is : 19800
HIS JOB : IT_PROG

ID = 103
Emp Salary after 10% is : 9900
HIS JOB : IT_PROG

ID = 104
Emp Salary after 10% is : 18700
HIS JOB : IT_PROG

ID = 105
Emp Salary after 10% is : 5280
```

5.

Create empty table `employees_again2` and use Cursor loop to insert only employees whose `job_id = 'SA_REP'` to the new table with double salary [`salary * 2`]

```
--create table employees_again2 as select EMPLOYEE_ID, LAST_NAME, SALARY, JOB_ID from employees where 1=2;

----AYA SABRY LAB 6----
• set serveroutput on
• declare
  cursor c_emp1 is
  SELECT * FROM employees WHERE job_id = 'SA_REP';
  begin
  FOR emp_record IN c_emp1 LOOP
  INSERT INTO employees_again2(EMPLOYEE_ID, LAST_NAME, SALARY, JOB_ID) VALUES
  (emp_record.EMPLOYEE_ID, emp_record.LAST_NAME, (2*emp_record.SALARY), emp_record.JOB_ID);
  commit;

  dbms_output.put_line('ID = ' || emp_record.employee_id || ', Name : ' || emp_record.LAST_NAME || ', Salary : ' || emp_record.salary || ' JOB ID: ' || emp_record.job_id);
  dbms_output.put_line('-----');
  END LOOP;
END;

--select * from employees_again2;
```

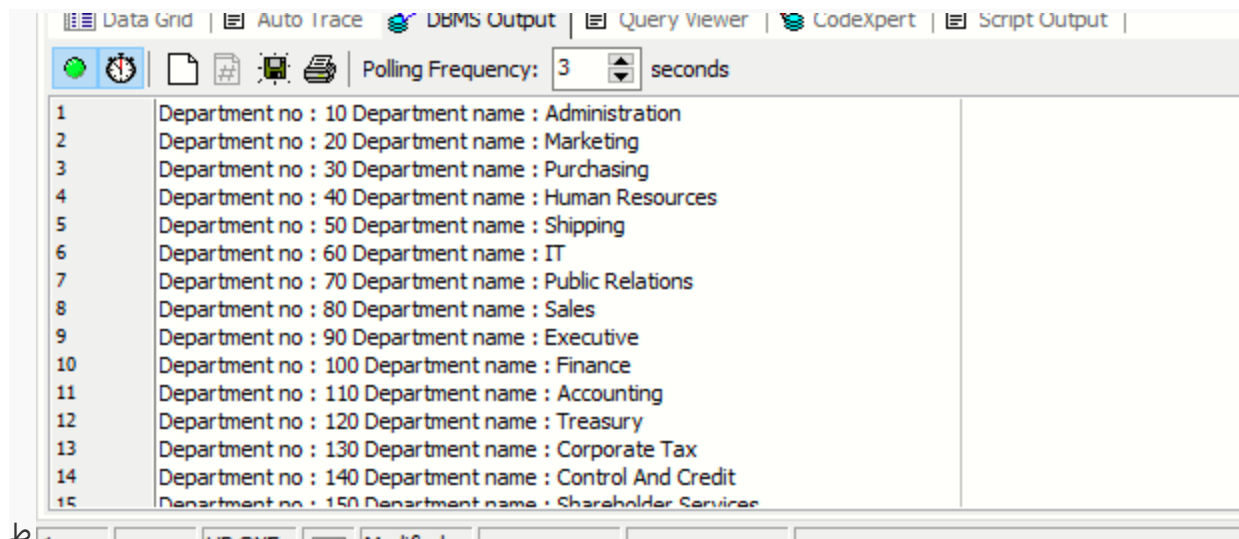
```
ID = 150 ,Name :Tucker , Salary : 10000 JOB ID: SA_REP
-----
ID = 152 ,Name :Hall , Salary : 9000 JOB ID: SA_REP
-----
ID = 153 ,Name :Olsen , Salary : 8000 JOB ID: SA_REP
-----
ID = 155 ,Name :Tuvault , Salary : 7000 JOB ID: SA_REP
-----
ID = 157 ,Name :Sully , Salary : 9500 JOB ID: SA_REP
-----
ID = 159 ,Name :Smith , Salary : 8000 JOB ID: SA_REP
-----
ID = 160 ,Name :Doran , Salary : 7500 JOB ID: SA_REP
```


EMPLOYEE_ID	LAST_NAME	SALARY	JOB_ID
150	Tucker	20000	SA_...
152	Hall	18000	SA_...
153	Olsen	16000	SA_...
155	Tuvault	14000	SA_...
157	Sully	19000	SA_...
159	Smith	16000	SA_...
160	Doran	15000	SA_...
162	Vishney	21000	SA_...
164	Marvins	14400	SA_...
165	Lee	13600	SA_...
167	Banda	12400	SA_...
169	Bloom	20000	SA_...
170	Fox	19200	SA_...
172	Bates	14600	SA_...
174	Abel	22000	SA_...

6.

Using cursor loop to loop over departments and print dept_id, dept_name

```
---AYA SABRY LAB 6---  
  
begin  
for DEP in (select department_id ,department_name from departments)  
loop  
    dbms_output.put_line('Department no : ' || DEP.department_id || ' Department name : ' || DEP.department_name);  
end loop;  
end;
```



The screenshot shows the SQL Developer interface with the DBMS Output window open. The window displays 15 rows of output, each representing a department. The output is formatted as 'Department no : [dept_id] Department name : [dept_name]'. The departments listed are: Administration (10), Marketing (20), Purchasing (30), Human Resources (40), Shipping (50), IT (60), Public Relations (70), Sales (80), Executive (90), Finance (100), Accounting (110), Treasury (120), Corporate Tax (130), Control And Credit (140), and Shareholder Services (150).

Line	Output
1	Department no : 10 Department name : Administration
2	Department no : 20 Department name : Marketing
3	Department no : 30 Department name : Purchasing
4	Department no : 40 Department name : Human Resources
5	Department no : 50 Department name : Shipping
6	Department no : 60 Department name : IT
7	Department no : 70 Department name : Public Relations
8	Department no : 80 Department name : Sales
9	Department no : 90 Department name : Executive
10	Department no : 100 Department name : Finance
11	Department no : 110 Department name : Accounting
12	Department no : 120 Department name : Treasury
13	Department no : 130 Department name : Corporate Tax
14	Department no : 140 Department name : Control And Credit
15	Department no : 150 Department name : Shareholder Services

7.

Create plsql block that insert new Department

With these data

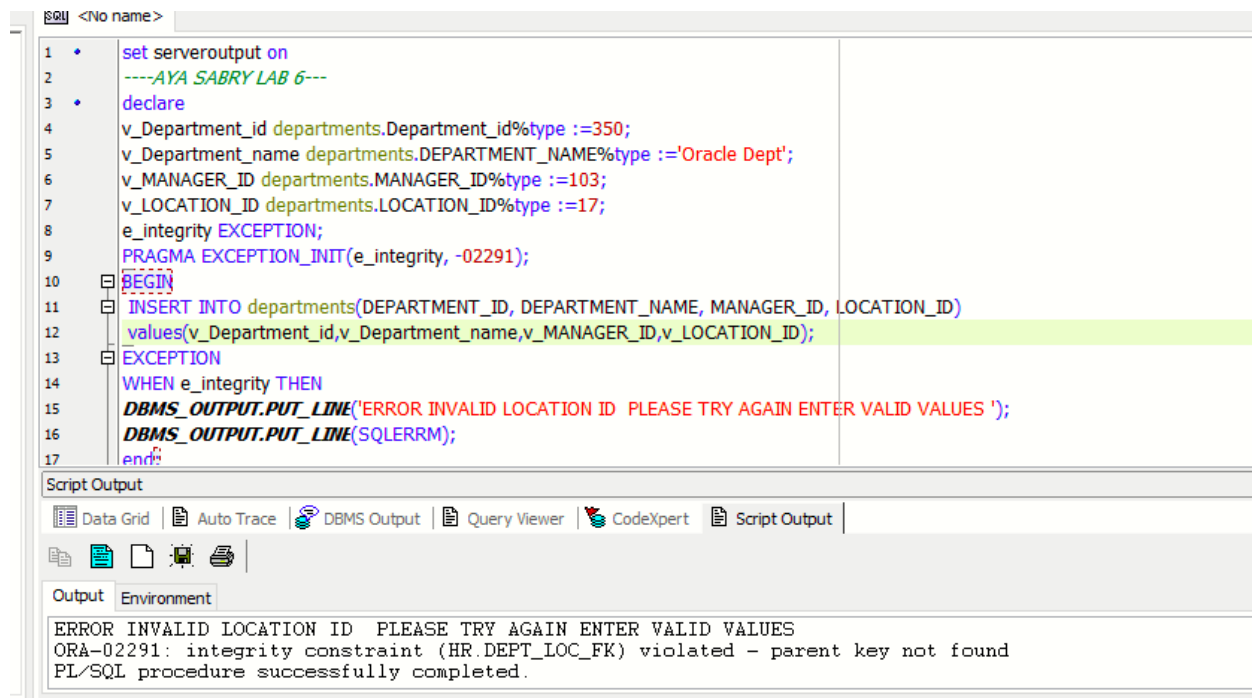
Department_id = 350

Department name = Oracle Dept

Manager id = 103

Location Id = 17

Handle exception as needed



```
1 • set serveroutput on
2 ----AYA SABRY LAB 6---
3 • declare
4 v_Deptment_id departments.Department_id%type :=350;
5 v_Deptment_name departments.DEPARTMENT_NAME%type :='Oracle Dept';
6 v_MANAGER_ID departments.MANAGER_ID%type :=103;
7 v_LOCATION_ID departments.LOCATION_ID%type :=17;
8 e_integrity EXCEPTION;
9 PRAGMA EXCEPTION_INIT(e_integrity, -02291);
10 BEGIN
11 INSERT INTO departments(DEPARTMENT_ID, DEPARTMENT_NAME, MANAGER_ID, LOCATION_ID)
12 values(v_Deptment_id,v_Deptment_name,v_MANAGER_ID,v_LOCATION_ID);
13 EXCEPTION
14 WHEN e_integrity THEN
15 DBMS_OUTPUT.PUT_LINE('ERROR INVALID LOCATION ID PLEASE TRY AGAIN ENTER VALID VALUES ');
16 DBMS_OUTPUT.PUT_LINE(SQLERRM);
17 end;
```

Script Output

Data Grid | Auto Trace | DBMS Output | Query Viewer | CodeXpert | Script Output

Output Environment

ERROR INVALID LOCATION ID PLEASE TRY AGAIN ENTER VALID VALUES
ORA-02291: integrity constraint (HR.DEPT_LOC_FK) violated - parent key not found
PL/SQL procedure successfully completed.