Code-

declare

cursor c\_emps is select \* from employees\_copy for update;

v\_salary\_increase number:= 1.10;

v\_old\_salary number;

begin

for r\_emp in c\_emps loop

v\_old\_salary := r\_emp.salary;

r\_emp.salary := r\_emp.salary\*v\_salary\_increase + r\_emp.salary \* 0.8;

update employees\_copy set row = r\_emp where current of c\_emps;

dbms\_output.put\_line('The salary of : '|| r\_emp.employee\_id

|| ' is increased from '||v\_old\_salary||' to '||r\_emp.salary);

end loop;

end;

Output-

Statement processed.

The salary of : 1 is increased from 91877.79 to 174567.8

The salary of : 2 is increased from 74280.57 to 141133.08

The salary of : 3 is increased from 56492.73 to 107336.19

The salary of : 4 is increased from 84957.36 to 161418.98

The salary of : 5 is increased from 65705.99 to 124841.38

The salary of : 6 is increased from 54389.6 to 103340.24

The salary of : 7 is increased from 52664.9 to 100063.31

The salary of : 8 is increased from 86128.35 to 163643.87

The salary of : 9 is increased from 80011.26 to 152021.39

The salary of : 10 is increased from 79725.8 to 151479.02

Code-

create procedure increase\_salaries as

cursor c\_emps is select \* from employees\_copy for update;

v\_salary\_increase number := 1.10;

v\_old\_salary number;

begin

for r\_emp in c\_emps loop

v\_old\_salary := r\_emp.salary;

r\_emp.salary := r\_emp.salary \* v\_salary\_increase + r\_emp.salary \*0.4;

update employees\_copy set row = r\_emp where current of c\_emps;

dbms\_output.put\_line('The salary of : '|| r\_emp.employee\_id

|| ' is increased from '||v\_old\_salary||' to '||r\_emp.salary);

end loop;

end;

Output-

Procedure created.

Code-

begin

dbms\_output.put\_line('Increasing the salaries!...');

INCREASE\_SALARIES;

INCREASE\_SALARIES;

INCREASE\_SALARIES;

INCREASE\_SALARIES;

dbms\_output.put\_line('All the salaries are successfully increased!...');

end;

Output-

Statement processed.

Increasing the salaries!...

The salary of : 1 is increased from 174567.8 to 261851.7

The salary of : 2 is increased from 141133.08 to 211699.62

The salary of : 3 is increased from 107336.19 to 161004.29

The salary of : 4 is increased from 161418.98 to 242128.47

The salary of : 5 is increased from 124841.38 to 187262.07

The salary of : 6 is increased from 103340.24 to 155010.36

The salary of : 7 is increased from 100063.31 to 150094.97

The salary of : 8 is increased from 163643.87 to 245465.81

The salary of : 9 is increased from 152021.39 to 228032.09

The salary of : 10 is increased from 151479.02 to 227218.53

The salary of : 1 is increased from 261851.7 to 392777.55

The salary of : 2 is increased from 211699.62 to 317549.43

The salary of : 3 is increased from 161004.29 to 241506.44

The salary of : 4 is increased from 242128.47 to 363192.71

The salary of : 5 is increased from 187262.07 to 280893.11

The salary of : 6 is increased from 155010.36 to 232515.54

The salary of : 7 is increased from 150094.97 to 225142.46

The salary of : 8 is increased from 245465.81 to 368198.72

The salary of : 9 is increased from 228032.09 to 342048.14

The salary of : 10 is increased from 227218.53 to 340827.8

The salary of : 1 is increased from 392777.55 to 589166.33

The salary of : 2 is increased from 317549.43 to 476324.15

The salary of : 3 is increased from 241506.44 to 362259.66

The salary of : 4 is increased from 363192.71 to 544789.07

The salary of : 5 is increased from 280893.11 to 421339.67

The salary of : 6 is increased from 232515.54 to 348773.31

The salary of : 7 is increased from 225142.46 to 337713.69

The salary of : 8 is increased from 368198.72 to 552298.08

The salary of : 9 is increased from 342048.14 to 513072.21

The salary of : 10 is increased from 340827.8 to 511241.7

The salary of : 1 is increased from 589166.33 to 883749.5

The salary of : 2 is increased from 476324.15 to 714486.23

The salary of : 3 is increased from 362259.66 to 543389.49

The salary of : 4 is increased from 544789.07 to 817183.61

The salary of : 5 is increased from 421339.67 to 632009.51

The salary of : 6 is increased from 348773.31 to 523159.97

The salary of : 7 is increased from 337713.69 to 506570.54

The salary of : 8 is increased from 552298.08 to 828447.12

The salary of : 9 is increased from 513072.21 to 769608.32

The salary of : 10 is increased from 511241.7 to 766862.55

All the salaries are successfully increased!...

Code-

create or replace procedure increase\_salaries (v\_salary\_increase in number, v\_department\_id pls\_integer) as

cursor c\_emps is select \* from employees\_copy for update;

v\_old\_salary number;

begin

for r\_emp in c\_emps loop

v\_old\_salary := r\_emp.salary;

r\_emp.salary := r\_emp.salary \* v\_salary\_increase + r\_emp.salary \* 0.2;

update employees\_copy set row = r\_emp where current of c\_emps;

dbms\_output.put\_line('The salary of : '|| r\_emp.employee\_id

|| ' is increased from '||v\_old\_salary||' to '||r\_emp.salary);

end loop;

dbms\_output.put\_line('Procedure finished executing!');

end;

Output-

Procedure created.

Code-

create or replace procedure increase\_salaries

(v\_salary\_increase in out number, v\_department\_id pls\_integer, v\_affected\_employee\_count out number) as

cursor c\_emps is select \* from employees\_copy for update;

v\_old\_salary number;

v\_sal\_inc number := 0;

begin

v\_affected\_employee\_count := 0;

for r\_emp in c\_emps loop

v\_old\_salary := r\_emp.salary;

r\_emp.salary := r\_emp.salary \* v\_salary\_increase + r\_emp.salary \* 0.2;

update employees\_copy set row = r\_emp where current of c\_emps;

dbms\_output.put\_line('The salary of : '|| r\_emp.employee\_id

|| ' is increased from '||v\_old\_salary||' to '||r\_emp.salary);

v\_affected\_employee\_count := v\_affected\_employee\_count + 1;

v\_sal\_inc := v\_sal\_inc + v\_salary\_increase + 0.4;

end loop;

v\_salary\_increase := v\_sal\_inc / v\_affected\_employee\_count;

dbms\_output.put\_line('Procedure finished executing!');

end;

Output-Procedure created.

Code-

create or replace PROCEDURE PRINT(TEXT IN VARCHAR2 := 'This is the print function!.') IS

BEGIN

DBMS\_OUTPUT.PUT\_LINE(TEXT);

END;

Output-Procedure created.

Code-

exec print();

Output-

Statement processed.

This is the print function!.

Code-exec print(null);

Output-SQL Statement Output

Code-

cREATE OR REPLACE FUNCTION get\_avg\_sal (p\_dept\_id departments.department\_id%type) RETURN number AS

v\_avg\_sal number;

BEGIN

select avg(salary) into v\_avg\_sal from employees where department\_id = p\_dept\_id;

RETURN v\_avg\_sal;

END get\_avg\_sal;

----------------- using a function in begin-end block

declare

v\_avg\_salary number;

begin

v\_avg\_salary := get\_avg\_sal(50);

dbms\_output.put\_line(v\_avg\_salary);

end;

----------------- using functions in a select clause

select employee\_id,first\_name,salary,department\_id,get\_avg\_sal(department\_id) avg\_sal from employees;

----------------- using functions in group by, order by, where clauses

select get\_avg\_sal(department\_id) from employees

where salary > get\_avg\_sal(department\_id)

group by get\_avg\_sal(department\_id)

order by get\_avg\_sal(department\_id);

----------------- dropping a function

drop function get\_avg\_sal;

Output-

SQL Statement Output

Code-

create table emps\_high\_paid as select \* from employees where 1=2;

/

declare

procedure insert\_high\_paid\_emp(emp\_id employees.employee\_id%type) is

emp employees%rowtype;

begin

emp := get\_emp(emp\_id);

insert into emps\_high\_paid values emp;

end;

function get\_emp(emp\_num employees.employee\_id%type) return employees%rowtype is

emp employees%rowtype;

begin

select \* into emp from employees where employee\_id = emp\_num;

return emp;

end;

begin

for r\_emp in (select \* from employees) loop

if r\_emp.salary > 15000 then

insert\_high\_paid\_emp(r\_emp.employee\_id);

end if;

end loop;

end;

Output-

Table created.

Code-

declare

procedure insert\_high\_paid\_emp(emp\_id employees.employee\_id%type) is

emp employees%rowtype;

e\_id number;

function get\_emp(emp\_num employees.employee\_id%type) return employees%rowtype is

begin

select \* into emp from employees where employee\_id = emp\_num;

return emp;

end;

begin

emp := get\_emp(emp\_id);

insert into emps\_high\_paid values emp;

end;

begin

for r\_emp in (select \* from employees) loop

if r\_emp.salary > 15000 then

insert\_high\_paid\_emp(r\_emp.employee\_id);

end if;

end loop;

end;

Output-Statement processed.

Code-

declare

procedure insert\_high\_paid\_emp(p\_emp employees%rowtype) is

emp employees%rowtype;

e\_id number;

function get\_emp(emp\_num employees.employee\_id%type) return employees%rowtype is

begin

select \* into emp from employees where employee\_id = emp\_num;

return emp;

end;

function get\_emp(emp\_email employees.email%type) return employees%rowtype is

begin

select \* into emp from employees where email = emp\_email;

return emp;

end;

function get\_emp(f\_name varchar2, l\_name varchar2) return employees%rowtype is

begin

select \* into emp from employees where first\_name = f\_name and last\_name = l\_name;

return emp;

end;

begin

emp := get\_emp(p\_emp.employee\_id);

insert into emps\_high\_paid values emp;

emp := get\_emp(p\_emp.email);

insert into emps\_high\_paid values emp;

emp := get\_emp(p\_emp.first\_name,p\_emp.last\_name);

insert into emps\_high\_paid values emp;

end;

begin

for r\_emp in (select \* from employees) loop

if r\_emp.salary > 15000 then

insert\_high\_paid\_emp(r\_emp);

end if;

end loop;

end;

Output-

SQL Statement Output

Code-

create or replace function get\_emp(emp\_num employees.employee\_id%type) return employees%rowtype is

emp employees%rowtype;

begin

select \* into emp from employees where employee\_id = emp\_num;

return emp;

end;

Output-

Function created.

Code-

declare

v\_emp employees%rowtype;

begin

dbms\_output.put\_line('Fetching the employee data!..');

v\_emp := get\_emp(10);

dbms\_output.put\_line('Some information of the employee are : ');

dbms\_output.put\_line('The name of the employee is : '|| v\_emp.first\_name);

dbms\_output.put\_line('The email of the employee is : '|| v\_emp.email);

dbms\_output.put\_line('The salary of the employee is : '|| v\_emp.salary);

end;

Output-

Statement processed.

Fetching the employee data!..

Some information of the employee are :

The name of the employee is : First10

The email of the employee is : w4t97kq5@example.com

The salary of the employee is : 59922.12

Code-

create or replace function get\_emp(emp\_num employees.employee\_id%type) return employees%rowtype is

emp employees%rowtype;

begin

select \* into emp from employees where employee\_id = emp\_num;

return emp;

exception

when no\_data\_found then

dbms\_output.put\_line('There is no employee with the id '|| emp\_num);

end;

Output-Function created.

Code- create or replace function get\_emp(emp\_num employees.employee\_id%type) return employees%rowtype is

emp employees%rowtype;

begin

select \* into emp from employees where employee\_id = emp\_num;

return emp;

exception

when no\_data\_found then

dbms\_output.put\_line('There is no employee with the id '|| emp\_num);

raise no\_data\_found;

end;

Output-SQL Statement Output

Code-

create or replace function get\_emp(emp\_num employees.employee\_id%type) return employees%rowtype is

emp employees%rowtype;

begin

select \* into emp from employees where employee\_id = emp\_num;

return emp;

exception

when no\_data\_found then

dbms\_output.put\_line('There is no employee with the id '|| emp\_num);

raise no\_data\_found;

when others then

dbms\_output.put\_line('Something unexpected happened!.');

return null;

end;

Output-Function created.

Code-

Output-

Code-

Output-