Code-

create or replace trigger before\_statement\_emp\_cpy

before insert or update on employees\_copy

begin

dbms\_output.put\_line('Before Statement Trigger is Fired!.');

end;

Output-

Trigger created.

Code-

create or replace trigger after\_statement\_emp\_cpy

after insert or update on employees\_copy

begin

dbms\_output.put\_line('After Statement Trigger is Fired!.');

end;

Output-Trigger created.

Code-

create or replace trigger before\_row\_emp\_cpy

before insert or update on employees\_copy

for each row

begin

dbms\_output.put\_line('Before Row Trigger is Fired!.');

end;

Output-Trigger created.

Code-

create or replace trigger after\_row\_emp\_cpy

after insert or update on employees\_copy

for each row

begin

dbms\_output.put\_line('After Row Trigger is Fired!.');

end;

Output-SQL Statement Output

Code-

update employees\_copy set salary = salary + 100 where employee\_id = 10;

update employees\_copy set salary = salary + 100 where employee\_id = 9;

update employees\_copy set salary = salary + 100 where employee\_id = 1;

Output-

1 row(s) updated.

Before Statement Trigger is Fired!.

Before Row Trigger is Fired!.

After Row Trigger is Fired!.

After Statement Trigger is Fired!.

1 row(s) updated.

Before Statement Trigger is Fired!.

Before Row Trigger is Fired!.

After Row Trigger is Fired!.

After Statement Trigger is Fired!.

1 row(s) updated.

Before Statement Trigger is Fired!.

Before Row Trigger is Fired!.

After Row Trigger is Fired!.

After Statement Trigger is Fired!.

Code-

create or replace trigger before\_row\_emp\_cpy

before insert or update or delete on employees\_copy

referencing old as O new as N

for each row

begin

dbms\_output.put\_line('Before Row Trigger is Fired!.');

dbms\_output.put\_line('The Salary of Employee '||:o.employee\_id

||' -> Before:'|| :o.salary||' After:'||:n.salary);

end;

Output-

Trigger created.

Code-

create or replace trigger before\_row\_emp\_cpy

before insert or update or delete on employees\_copy

referencing old as O new as N

for each row

begin

dbms\_output.put\_line('Before Row Trigger is Fired!.');

dbms\_output.put\_line('The Salary of Employee '||:o.employee\_id

||' -> Before:'|| :o.salary||' After:'||:n.salary);

if inserting then

dbms\_output.put\_line('An INSERT occurred on employees\_copy table');

elsif deleting then

dbms\_output.put\_line('A DELETE occurred on employees\_copy table');

elsif updating ('salary') then

dbms\_output.put\_line('A DELETE occurred on the salary column');

elsif updating then

dbms\_output.put\_line('An UPDATE occurred on employees\_copy table');

end if;

end;

Output-Trigger created.

Code-

create or replace trigger trg\_comp\_emps

for insert or update or delete on employees\_copy

compound trigger

v\_dml\_type varchar2(10);

before statement is

begin

if inserting then

v\_dml\_type := 'INSERT';

elsif updating then

v\_dml\_type := 'UPDATE';

elsif deleting then

v\_dml\_type := 'DELETE';

end if;

dbms\_output.put\_line('Before statement section is executed with the '||v\_dml\_type ||' event!.');

end before statement;

before each row is

t number;

begin

dbms\_output.put\_line('Before row section is executed with the '||v\_dml\_type ||' event!.');

end before each row;

after each row is

begin

dbms\_output.put\_line('After row section is executed with the '||v\_dml\_type ||' event!.');

end after each row;

after statement is

begin

dbms\_output.put\_line('After statement section is executed with the '||v\_dml\_type ||' event!.');

end after statement;

end;

Output-Trigger created.