

DBMS Lab Exercise-10 Triggers in SQL

Sandip Datta

21BCE1163

Triggers in SQL

1. Consider the table Employee Table

Column Name	Data Type	Size	Description
<u>Empno</u>	NUMBER	4	Employee's Identification Number
Ename	VARCHAR2	30	Employee's Name
Job	VARCHAR2	15	Employee's Designation
Sal	NUMBER	8,2	Employee's Salary
DeptNo	NUMBER	2	Employee's Department id
Commission	NUMBER	7,2	Employee's Commission

- i. Create a trigger message that "a new row is inserted" after the record is added successfully.

```
create table emp_21bce1163(empno number(4),ename varchar2(30), job varchar2(15),sal
number(8,2),deptno number(2),com number(7,2));
```

```
create or replace trigger new_21bce1163
```

```
after insert on emp_21bce1163
```

```
for each row
```

```
begin
```

```
dbms_output.put_line('new row inserted');
```

```
end;
```

```
insert into emp_21bce1163 values(10,'sandip','employee',53652,10,400);
```

The screenshot displays the Oracle APEX SQL Workshop interface. At the top, the navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. The 'SQL Commands' tab is active, showing a schema of 'WKSP_111111111'. The language is set to 'PL/SQL' and the number of rows to display is '10'. The SQL command area contains the following code:

```
1 create table emp_21bce1163(empno number(4),ename varchar2(30), job varchar2(15),sal number(8,2),deptno number(2),com number(7,2));
2
3 create or replace trigger new_21bce1163
4 after insert on emp_21bce1163
5 for each row
6 begin
7 dbms_output.put_line('new row inserted');
8 end;
9
10 insert into emp_21bce1163 values(10,'sandip','employee',53652,10,400);
11
12
13
14
```

The 'Results' tab is selected, showing the output: 'new row inserted' and '1 row(s) inserted.' The footer includes the user 'sandip.datta2021@vitstudent.ac.in', the copyright notice 'Copyright © 1999, 2022, Oracle and/or its affiliates.', and the version 'Oracle APEX 22.2.4'.

- ii. Write a before delete trigger on Employee table. (If the entire record of an Employee is attempted to delete)

```
create table emp_21bce1163(empno number(4),ename varchar2(30), job varchar2(15),sal number(8,2),deptno number(2),com number(7,2));
```

```
create or replace trigger del_21bce1163
```

```
before delete on emp_21bce1163
```

for each row

begin

dbms_output.put_line('deleting one row');

end;

insert into emp_21bce1163 values(10,'sandip','employee',53652,10,400);

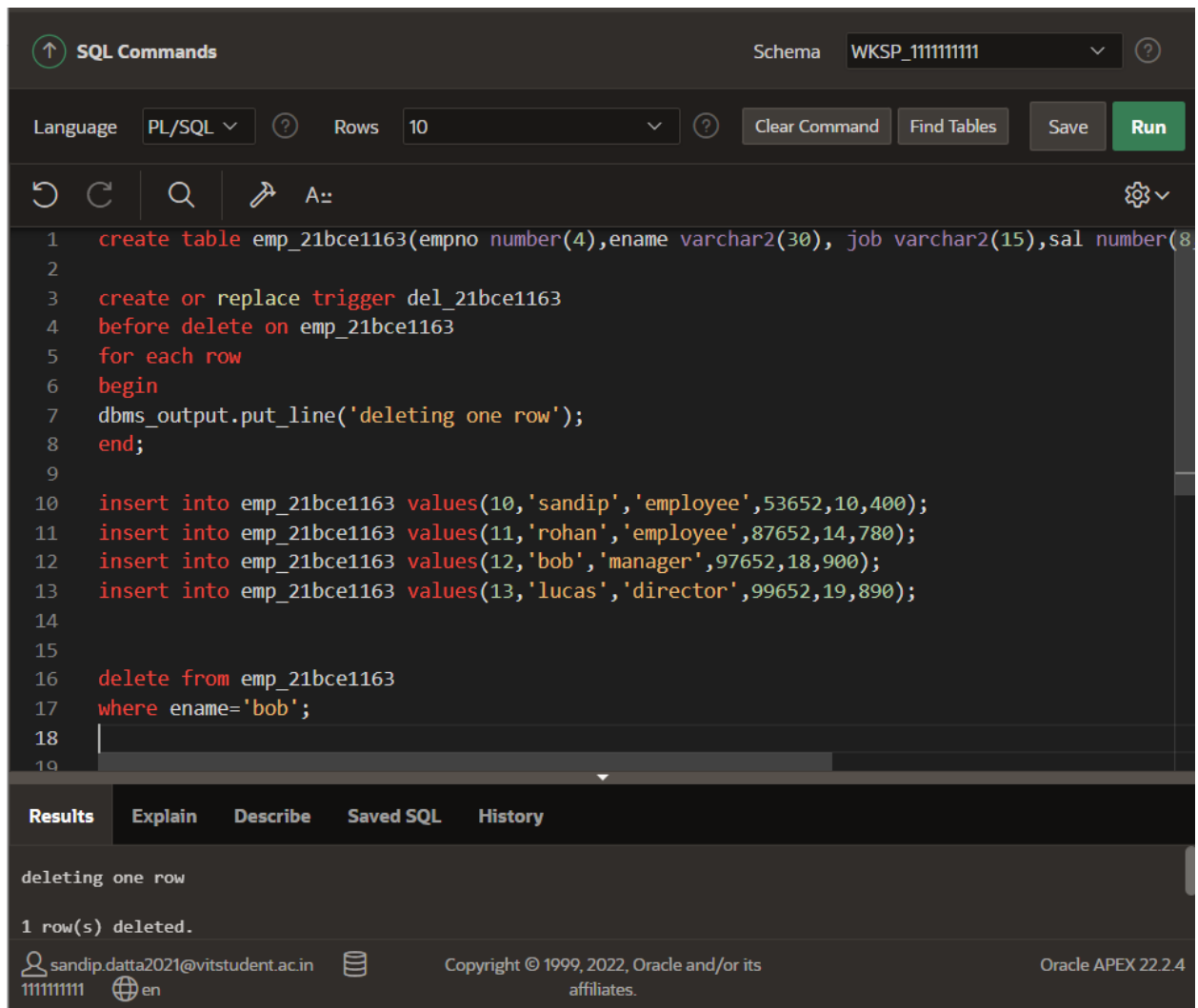
insert into emp_21bce1163 values(11,'rohan','employee',87652,14,780);

insert into emp_21bce1163 values(12,'bob','manager',97652,18,900);

insert into emp_21bce1163 values(13,'lucas','director',99652,19,890);

delete from emp_21bce1163

where ename='bob';



The screenshot shows the Oracle APEX SQL Commands interface. The schema is set to WKSP_111111111. The language is PL/SQL and the number of rows to display is 10. The script being executed is as follows:

```
1 create table emp_21bce1163(empno number(4),ename varchar2(30), job varchar2(15),sal number(8,2),deptno number(2),com number(7,2));
2
3 create or replace trigger del_21bce1163
4 before delete on emp_21bce1163
5 for each row
6 begin
7 dbms_output.put_line('deleting one row');
8 end;
9
10 insert into emp_21bce1163 values(10,'sandip','employee',53652,10,400);
11 insert into emp_21bce1163 values(11,'rohan','employee',87652,14,780);
12 insert into emp_21bce1163 values(12,'bob','manager',97652,18,900);
13 insert into emp_21bce1163 values(13,'lucas','director',99652,19,890);
14
15
16 delete from emp_21bce1163
17 where ename='bob';
18
19
```

The results pane shows the output of the execution:

```
deleting one row
1 row(s) deleted.
```

The footer of the interface shows the user sandip.datta2021@vitstudent.ac.in, the copyright notice Copyright © 1999, 2022, Oracle and/or its affiliates, and the Oracle APEX 22.2.4 version.

- iii. Write a trigger, for every any update on the salary column, the updated row is inserted into a new table.

```
create table emp_21bce1163(empno number(4),ename varchar2(30), job varchar2(15),sal number(8,2),deptno number(2),com number(7,2));
```

```
create table newemp_21bce1163(empno number(4),ename varchar2(30), job varchar2(15),sal number(8,2),deptno number(2),com number(7,2));
```

```
insert into emp_21bce1163 values(10,'sandip','employee',53652,10,400);
```

```
insert into emp_21bce1163 values(11,'rohan','employee',87652,14,780);
```

```
insert into emp_21bce1163 values(12,'bob','manager',97652,18,900);
```

```
insert into emp_21bce1163 values(13,'lucas','director',99652,19,890);
```

create or replace trigger update_21bce1163

after update of sal on emp_21bce1163

for each row

begin

insert into newemp_21bce1163(empno ,ename , job ,sal ,deptno ,com)

values(:new.empno ,:new.ename , :new.job ,:new.sal ,:new.deptno ,:new.com);

end;

update emp_21bce1163

set sal=10000

where ename='lucas';

select * from newemp_21bce1163;

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. A search bar and 'Send Data' button are on the right. The 'SQL Commands' tab is active, displaying a series of SQL statements. Below the editor, the 'Results' tab shows a table with one row of data.

```
4 insert into emp_21bce1163 values(11, 'roman', 'employee', 87652, 18, '880');
5 insert into emp_21bce1163 values(12, 'bob', 'manager', 97652, 18, 900);
6 insert into emp_21bce1163 values(13, 'lucas', 'director', 99652, 19, 890);
7
8 create or replace trigger update_21bce1163
9 after update of sal on emp_21bce1163
10 for each row
11 begin
12 insert into newemp_21bce1163(empno ,ename , job ,sal ,deptno ,com )
13 values(:new.empno ,:new.ename , :new.job ,:new.sal ,:new.deptno ,:new.com );
14 end;
15
16 update emp_21bce1163
17 set sal=10000
18 where ename='lucas';
19 select * from newemp_21bce1163;
```

EMPNO	ENAME	JOB	SAL	DEPTNO	COM
13	lucas	director	10000	19	890

1 rows returned in 0.03 seconds [Download](#)

Copyright © 1999, 2022, Oracle and/or its affiliates. Oracle APEX 22.2.4

iv. Store the count of employees having salary > 4000 in new table Audit

create table emp_21bce1163(empno number(4),ename varchar2(30), job varchar2(15),sal number(8,2),deptno number(2),com number(7,2));

create table newemp_21bce1163(empno number(4),ename varchar2(30), job varchar2(15),sal number(8,2),deptno number(2),com number(7,2));

```
insert into emp_21bce1163 values(10,'sandip','employee',53652,10,400);
insert into emp_21bce1163 values(11,'rohan','employee',87652,14,780);
insert into emp_21bce1163 values(12,'bob','manager',97652,18,900);
insert into emp_21bce1163 values(13,'lucas','director',99652,19,890);
```

```
create table audit_21bce1163 (id number generated by default as identity,
emcount number);
```

```
create or replace trigger salupaudit_21bce1163
after insert or update or delete on emp_21bce1163
```

```
begin
declare
emcount number;
begin
dbms_output.put_line('audit');
select count(*) into emcount from emp_21bce1163 where sal>4000;
insert into audit_21bce1163(emcount) values (emcount);
end;
end;
```

```
update emp_21bce1163
set sal=10000
where ename='rohan';
```

select * from audit_21bce1163;

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. A search bar and user profile 'Sandip Data' are on the right. The 'SQL Commands' panel is active, showing a SQL script with line numbers 10 to 30. The script includes a trigger definition, a count query, an insert statement, and an update statement. Below the script, the 'Results' tab is selected, displaying a table with two columns: 'ID' and 'EMCOUNT'. The table contains one row with ID 1 and EMCOUNT 3. The status bar at the bottom indicates '1 rows returned in 0.01 seconds' and includes a 'Download' link. The footer shows the user 'sandip.data2021@vitstudent.ac.in', the schema 'WSP_11111111', and the Oracle APEX version '22.2.4'.

```
10  emcount number);
11
12  create or replace trigger salupaudit_21bce1163
13  after insert or update or delete on emp_21bce1163
14
15  begin
16  declare
17  emcount number;
18  begin
19  dbms_output.put_line('audit');
20  select count(*) into emcount from emp_21bce1163 where sal>4000;
21  insert into audit_21bce1163(emcount) values (emcount);
22  end;
23  end;
24
25  update emp_21bce1163
26  set sal=10000
27  where ename='rohan';
28
29
30  select * from audit_21bce1163;
```

ID	EMCOUNT
1	3

1 rows returned in 0.01 seconds [Download](#)

v. Give 10% raise to all employees whose salary > 8000.

```
create table emp_21bce1163(empno number(4),ename varchar2(30), job varchar2(15),sal
number(8,2),deptno number(2),com number(7,2));
```

```
create table newemp_21bce1163(empno number(4),ename varchar2(30), job varchar2(15),sal
number(8,2),deptno number(2),com number(7,2));
```

```
insert into emp_21bce1163 values(10,'sandip','employee',53652,10,400);
```

```
insert into emp_21bce1163 values(11,'rohan','employee',87652,14,780);
```

```
insert into emp_21bce1163 values(12,'bob','manager',97652,18,900);
```

```
insert into emp_21bce1163 values(13,'lucas','director',99652,19,890);
```

```
create table audit_21bce1163 (id number generated by default as identity,
emcount number);
```

```
create or replace trigger salupaudit_21bce1163
```

after insert or update or delete on emp_21bce1163

begin

declare

emcount number;

begin

dbms_output.put_line('audit');

select count(*) into emcount from emp_21bce1163 where sal>4000;

insert into audit_21bce1163(emcount) values (emcount);

end;

end;

update emp_21bce1163

set sal=10000

where ename='rohan';

select * from audit_21bce1163;

update emp_21bce1163

set sal=sal*1.1

where sal>8000;


```

14
15 begin
16 declare
17   emcount number;
18 begin
19   dbms_output.put_line('audit');
20   select count(*) into emcount from emp_21bce1163 where sal>4000;
21   insert into audit_21bce1163(emcount) values (emcount);
22 end;
23 end;
24
25
26 update emp_21bce1163
27 set sal=10000
28 where ename='rohan';
29
30 select * from audit_21bce1163;
31
32 update emp_21bce1163
33 set sal=sal*1.1
34 where sal>8000;

```

Results Explain Describe Saved SQL History

audit

3 row(s) updated.

0.09 seconds

Results Explain Describe Saved SQL History

ID	EMCOUNT
21	3
1	3

2 rows returned in 0.01 seconds [Download](#)

- vi. For given table, create “Before Update” trigger preventing a change on Primary Key column

```
create table emp_21bce1163(empno number(4),ename varchar2(30), job varchar2(15),sal
number(8,2),deptno number(2),com number(7,2));
```

```
create table newemp_21bce1163(empno number(4),ename varchar2(30), job varchar2(15),sal
number(8,2),deptno number(2),com number(7,2));
```

```
insert into emp_21bce1163 values(10,'sandip','employee',53652,10,400);
```

```
insert into emp_21bce1163 values(11,'rohan','employee',87652,14,780);
```

```
insert into emp_21bce1163 values(12,'bob','manager',97652,18,900);
```

```
insert into emp_21bce1163 values(13,'lucas','director',99652,19,890);
```

```
create table audit_21bce1163 (id number generated by default as identity,  
emcount number);
```

```
create or replace trigger salupaudit_21bce1163  
after insert or update or delete on emp_21bce1163
```

```
begin
```

```
declare
```

```
emcount number;
```

```
begin
```

```
dbms_output.put_line('audit');
```

```
select count(*) into emcount from emp_21bce1163 where sal>4000;
```

```
insert into audit_21bce1163(emcount) values (emcount);
```

```
end;
```

```
end;
```

```
update emp_21bce1163
```

```
set sal=10000
```

```
where ename='rohan';
```

```
select * from audit_21bce1163;
```

update emp_21bce1163

set sal=sal*1.1

where sal>8000;

create or replace trigger upprevention

before update of empno on emp_21bce1163

for each row

begin

raise_application_error(-20000,'you can not update the primary key');

end;

update emp_21bce1163

set empno=100

where ename='rohan';

The screenshot displays the Oracle APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. A search bar and user profile 'Sandip Data' are on the right. The 'SQL Commands' section shows a trigger definition and an update statement. The 'Results' tab at the bottom displays an error message.

```
33 set sal=sal*1.1
34 where sal>8000;
35
36 create or replace trigger upprevention
37 before update of empno on emp_21bce1163
38 for each row
39
40 begin
41 raise_application_error(-20000,'you can not update the primary key');
42 end;
43
44 update emp_21bce1163
45 set empno=100
46 where ename='rohan';
```

Results Explain Describe Saved SQL History

```
ORA-20000: you can not update the primary key
ORA-06512: at "WKSP_1111111111.UPPREVENTION", line 2
ORA-04088: error during execution of trigger "WKSP_1111111111.UPPREVENTION"
ORA-06512: at "SYS.DBMS_SQL", line 1721

1. update emp_21bce1163
2. set empno=100
3. where ename='rohan';
```

Copyright © 1999, 2022, Oracle and/or its affiliates. Oracle APEX 22.2.4

