**PRACTICAL NO.1**

**AIM: Creating and working with insert/update/delete trigger using before/after clause.**

**1.1 Create a DML trigger that uses conditional predicted to determine which of its four possible triggering statement fired it.**

**Creating table.**

SQL> create table empl\_1(eidnumber,name varchar2(20),salnumber,dept varchar2(2

0),dob date);

Table created.

**Insert record.**

SQL> insert into empl\_1 values('&eid','&name','&sal','&dept','&dob');

Enter value for eid: 101

Enter value for name: Rutu

Enter value for sal: 20000

Enter value for dept: COMPUTER SCIENCE

Enter value for dob: 06 June 1999

old 1: insert into empl\_1 values('&eid','&name','&sal','&dept','&dob')

new 1: insert into empl\_1 values('101','Rutu','20000','COMPUTER SCIENCE','06 J

une 1999')

1 row created.

SQL> /

Enter value for eid: 102

Enter value for name: Shivani

Enter value for sal: 20000

Enter value for dept: COMPUTER SCIENCE

Enter value for dob: 05 June 1998

old 1: insert into empl\_1 values('&eid','&name','&sal','&dept','&dob')

new 1: insert into empl\_1 values('102','Shivani','20000','COMPUTER SCIENCE','0

5 June 1998')

1 row created.

SQL> /

Enter value for eid: 103

Enter value for name: Kanchan

Enter value for sal: 20000

Enter value for dept: COMPUTER SCIENCE

Enter value for dob: 08 Aug 1999

old 1: insert into empl\_1 values('&eid','&name','&sal','&dept','&dob')

new 1: insert into empl\_1 values('103','Kanchan','20000','COMPUTER SCIENCE','0

8 Aug 1999')

1 row created.]

**PROGRAM:**

SQL> create or replace trigger t

2 before

3 insert or

4 update of sal,dept or

5 delete

6 on empl\_1

7 begin

8 case

9 when inserting then

10 dbms\_output.put\_line('Inserting');

11 when updating('sal') then

12 dbms\_output.put\_line('Updating salary');

13 when updating ('dept') then

14 dbms\_output.put\_line('Updating Department');

15 when deleting then

16 dbms\_output.put\_line('Deleting');

17 end case;

18 end;

19 /

Trigger created.

**Inserting:**

SQL> insert into empl\_1 values('&eid','&name','&sal','&dept','&dob');

Enter value for eid: 106

Enter value for name: Sneha

Enter value for sal: 20000

Enter value for dept: COMPUTER SCIENCE

Enter value for dob: 23 Sep 1999

old 1: insert into empl\_1 values('&eid','&name','&sal','&dept','&dob')

new 1: insert into empl\_1 values('106','Sneha','20000','COMPUTER SCIENCE','23

Sep 1999')

**O/P:**

**Inserting**

1 row created.

**UPDATE Department:**

SQL> update empl\_1

2 setdept='IT'

3 whereeid=106;

**O/P:**

**Updating Department**

1 row updated.

**UPDATE Salary:**

SQL> update empl\_1

2 setsal=30000

3 whereeid=106;

**O/P:**

**Updating salary**

1 row updated.

**DELETE Column:**

SQL> delete from empl\_1

whereeid=106;

**O/P:**

**Deleting**

1 row deleted.

**Create Table**

SQL> Create table orders

2 (order\_id number(5),

3 quantity number(4),

4 cost\_per\_ithem number(6,2),

5 total\_cost number(8,2),

6 create\_date date,

7 created\_by varchar2(10)

8 );

**Table created.**

**CREATE TRIGGER statement to create AN BEFORE INSERT trigger**

SQL> Create or replace trigger orders\_before\_insert

2 Before insert

3 On orders

4 For each row

5 Declare

6 v\_username varchar(10);

7 begin

8 --find username of person performing insert into table

9 Select user into v\_username from dual;

10 --update create\_date field to current system date

11 :new.create\_date:=sysdate;

12 --update created\_by field to the username of the person performing the insert

13 :new.created\_by:=v\_username;

14 end;

15 /

**Trigger created.**

**Insert Data into table**

SQL> insert into orders values(&order\_id,&quantity,&cost\_per\_ithem,&total\_cost,'

&create\_date','&created\_by');

Enter value for order\_id: 1

Enter value for quantity: 2

Enter value for cost\_per\_ithem: 10

Enter value for total\_cost: 20

Enter value for create\_date: 2-jan-2017

Enter value for created\_by: Rutu

old 1: insert into orders values(&order\_id,&quantity,&cost\_per\_ithem,&total\_co

st,'&create\_date','&created\_by')

new 1: insert into orders values(1,2,10,20,'2-jan-2017','Rutu')

**1 row created.**

**Display data**

SQL> select \* from orders;

ORDER\_ID QUANTITY COST\_PER\_ITHEM TOTAL\_COST CREATE\_DA CREATED\_BY

---------- ---------- -------------- ---------- --------- ----------

1 2 10 20 29-SEP-17 SYSTEM

**After Insert**

SQL> CREATE TABLE orders1

2 (order\_id number(5),

3 quantity number(4),

4 cost\_per\_item number(6,2),

5 total\_cost number(8,2)

6 );

**Table created.**

SQL> CREATE TABLE orders\_audit

2 (order\_id number(5),

3 quantity number(4),

4 cost\_per\_item number(6,2),

5 total\_cost number(8,2),

6 username varchar2(30)

7 );

**Table created.**

SQL> Select \* from orders1;

**no rows selected**

SQL> Select \* from orders\_audit;

**no rows selected**

SQL> CREATE OR REPLACE TRIGGER orders1\_after\_insert

2 AFTER INSERT

3 ON orders1

4 FOR EACH ROW

5

6 DECLARE

7 v\_username varchar2(10);

8

9 BEGIN

10

11 -- Find username of performing the INSERT into the table

12 SELECT user INTO v\_username

13 FROM dual;

14

15 -- Insert record into audit table

16 INSERT INTO orders\_audit

17 (order\_id,

18 quantity,

19 cost\_per\_item,

20 total\_cost,

21 username)

22 VALUES

23 (:new.order\_id,

24 :new.quantity,

25 :new.cost\_per\_item,

26 :new.total\_cost,

27 v\_username);

28

29 END;

30 /

**Trigger created.**

SQL> Insert into orders1 values(&order\_id,&quantity,&cost\_per\_ithem,&total\_cost);

Enter value for order\_id: 1

Enter value for quantity: 2

Enter value for cost\_per\_ithem: 10

Enter value for total\_cost: 20

old 1: Insert into orders1 values(&order\_id,&quantity,&cost\_per\_ithem,&total\_c

ost)

new 1: Insert into orders1 values(1,2,10,20)

**1 row created.**

SQL> select \* from orders1;

ORDER\_ID QUANTITY COST\_PER\_ITEM TOTAL\_COST

---------- ---------- ------------- ----------

1 2 10 20

SQL> select \* from orders\_audit;

ORDER\_ID QUANTITY COST\_PER\_ITEM TOTAL\_COST USERNAME

---------- ---------- ------------- ---------- ------------------------------

1 2 10 20 SYSTEM

**BEFORE UPDATE Trigger**

SQL> Create table orders3

2 (order\_id number(5),

3 quantity number(4),

4 cost\_per\_item number(6,2),

5 total\_cost number(8,2),

6 updated\_date date,

7 updated\_by varchar2(10)

8 );

**Table created.**

SQL> CREATE OR REPLACE TRIGGER orders3\_before\_update

2 BEFORE INSERT

3 ON orders3

4 FOR EACH ROW

5

6 DECLARE

7 v\_username varchar2(10);

8

9 BEGIN

10

11 -- Find username of performing the UPDATE into the table

12 SELECT user INTO v\_username

13 FROM dual;

14

15 -- Update updated\_date field to current system date

16 :new.updated\_date:=sysdate;

17

18 -- Update updated\_by field to the username of the person performing the UPDATE

19 :new.updated\_by:=v\_username;

20 END;

21 /

**Trigger created.**

SQL> insert into orders3 values(&order\_id,&quantity,&cost\_per\_item,&total\_cost,'

&updated\_date','&updated\_by');

Enter value for order\_id: 1

Enter value for quantity: 2

Enter value for cost\_per\_item: 10

Enter value for total\_cost: 20

Enter value for updated\_date: 1-jan-2017

Enter value for updated\_by: Rutu

old 1: insert into orders3 values(&order\_id,&quantity,&cost\_per\_item,&total\_co

st,'&updated\_date','&updated\_by')

new 1: insert into orders3 values(1,2,10,20,'1-jan-2017','Rutu')

**1 row created.**

SQL> select \* from orders3;

ORDER\_ID QUANTITY COST\_PER\_ITEM TOTAL\_COST UPDATED\_D UPDATED\_BY

---------- ---------- ------------- ---------- --------- ----------

1 2 10 20 29-SEP-17 SYSTEM

SQL> update orders3 set quantity=3,total\_cost=30 where order\_id=1;

**1 row updated.**

SQL> select \* from orders3;

ORDER\_ID QUANTITY COST\_PER\_ITEM TOTAL\_COST UPDATED\_D UPDATED\_BY

---------- ---------- ------------- ---------- --------- ----------

1 3 10 30 29-SEP-17 SYSTEM

**AFTER UPDATE Trigger**

SQL> CREATE TABLE orders4

2 (order\_id number(5),

3 quantity number(4),

4 cost\_per\_item number(6,2),

5 total\_cost number(8,2)

6 );

**Table created.**

SQL> insert into orders4 values(&order\_id,&quantity,&cost\_per\_item,&total\_cost);

Enter value for order\_id: 1

Enter value for quantity: 2

Enter value for cost\_per\_item: 20

Enter value for total\_cost: 40

old 1: insert into orders4 values(&order\_id,&quantity,&cost\_per\_item,&total\_co

st)

new 1: insert into orders4 values(1,2,20,40)

**1 row created.**

SQL> select \* from orders4;

ORDER\_ID QUANTITY COST\_PER\_ITEM TOTAL\_COST

---------- ---------- ------------- ----------

1 2 20 40

SQL> CREATE TABLE orders04\_audit

2 (order\_id number(5),

3 quantity\_before number(4),

4 quantity\_after number(4),

5 username varchar2(30)

6 );

**Table created.**

SQL> Select \* from orders04\_audit;

**no rows selected**

SQL> CREATE OR REPLACE TRIGGER orders4\_after\_update

2 AFTER INSERT

3 ON orders4

4 FOR EACH ROW

5

6 DECLARE

7 v\_username varchar2(10);

8

9 BEGIN

10

11 -- Find username of performing the UPDATE into the table

12 SELECT user INTO v\_username

13 FROM dual;

14

15 -- Insert record into audit table

16 INSERT INTO orders04\_audit

17 (order\_id,

18 quantity\_before,

19 quantity\_after,

20 username)

21 VALUES

22 (:new.order\_id,

23 :old.quantity,

24 :new.quantity,

25 v\_username);

26

27 END;

28 /

**Trigger created.**

SQL> update orders4 set quantity=3,total\_cost=60 where order\_id=1;

**1 row updated.**

SQL> select \* from orders4;

ORDER\_ID QUANTITY COST\_PER\_ITEM TOTAL\_COST

---------- ---------- ------------- ----------

1 3 20 60

**BEFORE DELETE Trigger**

SQL> CREATE TABLE orders4

2 (order\_id number(5),

3 quantity number(4),

4 cost\_per\_item number(6,2),

5 total\_cost number(8,2)

6 );

**Table created.**

SQL> insert into orders4 values(&order\_id,&quantity,&cost\_per\_item,&total\_cost);

Enter value for order\_id: 2

Enter value for quantity: 2

Enter value for cost\_per\_item: 50

Enter value for total\_cost: 100

old 1: insert into orders4 values(&order\_id,&quantity,&cost\_per\_item,&total\_co

st)

new 1: insert into orders4 values(2,2,50,100)

**1 row created.**

SQL> /

Enter value for order\_id: 3

Enter value for quantity: 5

Enter value for cost\_per\_item: 5

Enter value for total\_cost: 25

old 1: insert into orders4 values(&order\_id,&quantity,&cost\_per\_item,&total\_co

st)

new 1: insert into orders4 values(3,5,5,25)

**1 row created.**

SQL> /

Enter value for order\_id: 4

Enter value for quantity: 1

Enter value for cost\_per\_item: 600

Enter value for total\_cost: 600

old 1: insert into orders4 values(&order\_id,&quantity,&cost\_per\_item,&total\_co

st)

new 1: insert into orders4 values(4,1,600,600)

**1 row created.**

SQL> /

Enter value for order\_id: 5

Enter value for quantity: 4

Enter value for cost\_per\_item: 10

Enter value for total\_cost: 40

old 1: insert into orders4 values(&order\_id,&quantity,&cost\_per\_item,&total\_co

st)

new 1: insert into orders4 values(5,4,10,40)

**1 row created.**

SQL> select \* from orders4;

ORDER\_ID QUANTITY COST\_PER\_ITEM TOTAL\_COST

---------- ---------- ------------- ----------

1 3 20 60

2 2 50 100

3 5 5 25

4 1 600 600

5 4 10 40

SQL> Create table orders\_audit\_delete

2 (order\_id number(5),

3 quantity number(4),

4 cost\_per\_item number(6,2),

5 total\_cost number(8,2),

6 delete\_date date,

7 delete\_by varchar2(10)

8 );

**Table created.**

SQL> select \* from orders\_1\_audit\_delete;

**no rows selected**

SQL> CREATE OR REPLACE TRIGGER orders\_before\_delete

2 BEFORE INSERT

3 ON orders4

4 FOR EACH ROW

5 DECLARE

6 v\_username varchar2(10);

7 BEGIN

8 --find username of performing the DELETE into the table

9 SELECT user INTO v\_username

10 FROM dual;

11 --Insert record into audit table

12 INSERT INTO orders\_audit\_delete

13 (order\_id,

14 quantity,

15 cost\_per\_item,

16 total\_cost,

17 delete\_date,

18 deleted\_by)

19 VALUES

20 (:new. order\_id,

21 :new. quantity,

22 :new. cost\_per\_item,

23 :new. total\_cost,

24 Sysdate,

25 v\_username);

26 END;

27 /

**Trigger created.**

SQL> Delete \* from orders4 where order\_id=1;

**1 row deleted.**

SQL>select \* from orders4;

ORDER\_ID QUANTITY COST\_PER\_ITEM TOTAL\_COST

---------- ---------- ------------- ----------

2 2 50 100

3 5 5 25

4 1 600 600

5 4 10 40

SQL>select \* from orders\_audit\_delete;

ORDER\_ID QUANTITY COST\_PER\_ITEM TOTAL\_COST DELETE\_DA

---------- ---------- ------------- ---------- ---------

DELETED\_BY

------------------------------

1 3 20 60 03-OCT-17

SYSTEM

**AFTER DELETE Trigger**

SQL> create table orders\_1

2 (order\_id number(5),

3 quantity number(4),

4 cost\_per\_item number(6,2),

5 total\_cost number(8,2)

6 );

**Table created.**

SQL> create table orders\_1\_audit\_delete

2 (order\_id number(5),

3 quantity number(4),

4 cost\_per\_item number(6,2),

5 total\_cost number(8,2),

6 delete\_date date,

7 deleted\_by varchar2(30)

8 );

**Table created.**

SQL> insert into orders\_1 values(&order\_id,&quantity,&cost\_per\_item,&total\_cost)

;

Enter value for order\_id: 1

Enter value for quantity: 3

Enter value for cost\_per\_item: 20

Enter value for total\_cost: 60

old 1: insert into orders\_1 values(&order\_id,&quantity,&cost\_per\_item,&total\_c

ost)

new 1: insert into orders\_1 values(1,3,20,60)

**1 row created.**

SQL> /

Enter value for order\_id: 2

Enter value for quantity: 2

Enter value for cost\_per\_item: 50

Enter value for total\_cost: 100

old 1: insert into orders\_1 values(&order\_id,&quantity,&cost\_per\_item,&total\_c

ost)

new 1: insert into orders\_1 values(2,2,50,100)

**1 row created.**

SQL> /

Enter value for order\_id: 3

Enter value for quantity: 5

Enter value for cost\_per\_item: 5

Enter value for total\_cost: 25

old 1: insert into orders\_1 values(&order\_id,&quantity,&cost\_per\_item,&total\_c

ost)

new 1: insert into orders\_1 values(3,5,5,25)

**1 row created.**

SQL> /

Enter value for order\_id: 4

Enter value for quantity: 1

Enter value for cost\_per\_item: 600

Enter value for total\_cost: 600

old 1: insert into orders\_1 values(&order\_id,&quantity,&cost\_per\_item,&total\_c

ost)

new 1: insert into orders\_1 values(4,1,600,600)

**1 row created.**

SQL> select \* from orders\_1\_audit\_delete;

**no rows selected**

SQL> CREATE OR REPLACE TRIGGER orders1\_after\_delete

2 AFTER DELETE

3 ON orders\_1

4 FOR EACH ROW

5

6 DECLARE

7 v\_username varchar2(10);

8

9 BEGIN

10

11 -- Find username of person performing the DELETE on the table

12 SELECT user INTO v\_username

13 FROM dual;

14

15 -- Insert record into audit table

16 INSERT INTO orders\_1\_audit\_delete

17 (order\_id,

18 quantity,

19 cost\_per\_item,

20 total\_cost,

21 delete\_date,

22 deleted\_by)

23 VALUES

24 (:old.order\_id,

25 :old.quantity,

26 :old.cost\_per\_item,

27 :old.total\_cost,

28 sysdate,

29 v\_username);

30

31 END;

32 /

**Trigger created.**

SQL> delete from orders\_1 where order\_id=1;

**1 row deleted.**

SQL> select \* from orders\_1;

ORDER\_ID QUANTITY COST\_PER\_ITEM TOTAL\_COST

---------- ---------- ------------- ----------

2 2 50 100

3 5 5 25

4 1 600 600

SQL> select \* from orders\_1\_audit\_delete;

ORDER\_ID QUANTITY COST\_PER\_ITEM TOTAL\_COST DELETE\_DA

---------- ---------- ------------- ---------- ---------

DELETED\_BY

------------------------------

1 3 20 60 03-OCT-17

SYSTEM