Triggers in Oracle

Triggers in Oracle

A trigger is a course of action to be taken when any DDL, DML event occurs.

It is a named PL/SQL block like a procedure.

But it is executed implicitly by Oracle Server.

Type of Triggers:

DML Triggers (Table Level)
DDL Triggers (Database Level)

Triggers are created to
Control DML
Implement complex business rule
Implement complex validations
Generate values for primary key
For auditing
Maintain replicas

General Syntax for Triggers:

```
create [or replace ] trigger trigger_name
[before | after | instead of ]
[insert [or] | update [or] | delete] on table_name
[referencing old as 0 new as n]
[for each row] [when (condition)]
[declare <declare section>]
begin
[exception]
end;
```

Before Trigger

Before trigger is executed before executing a DML command.

Sequence:

Trigger execution

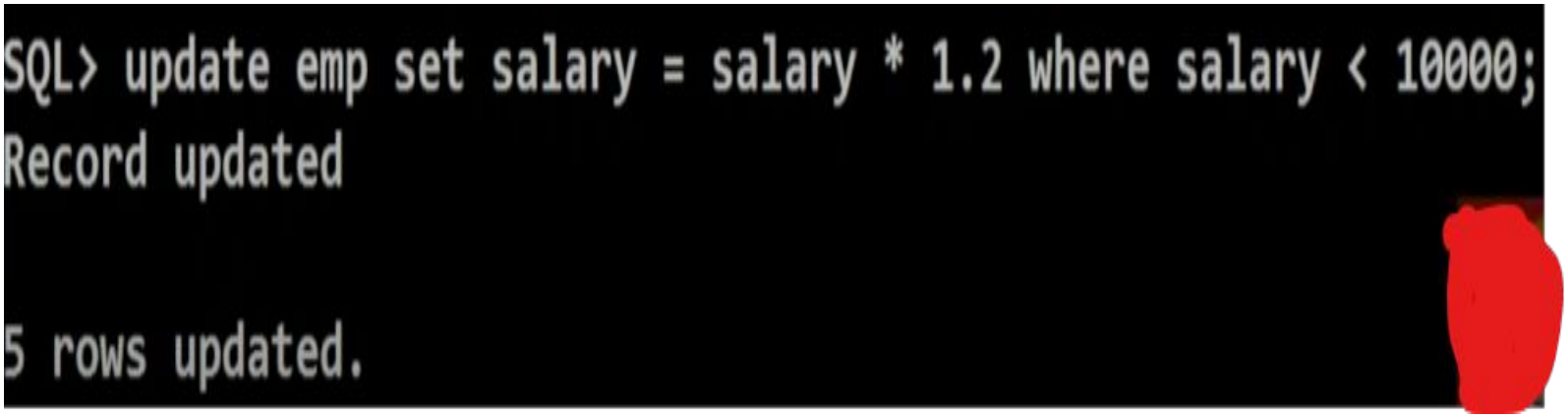
DML Execution

DML Triggers

These triggers are executed automatically whenever user performs DML operation on the table.

```
SQL> create or replace trigger emptrg2 before update on emp
  2  begin
  3  dbms_output.put_line('Record updated');
  4  end;
  5  /

Trigger created.

SQL>
```

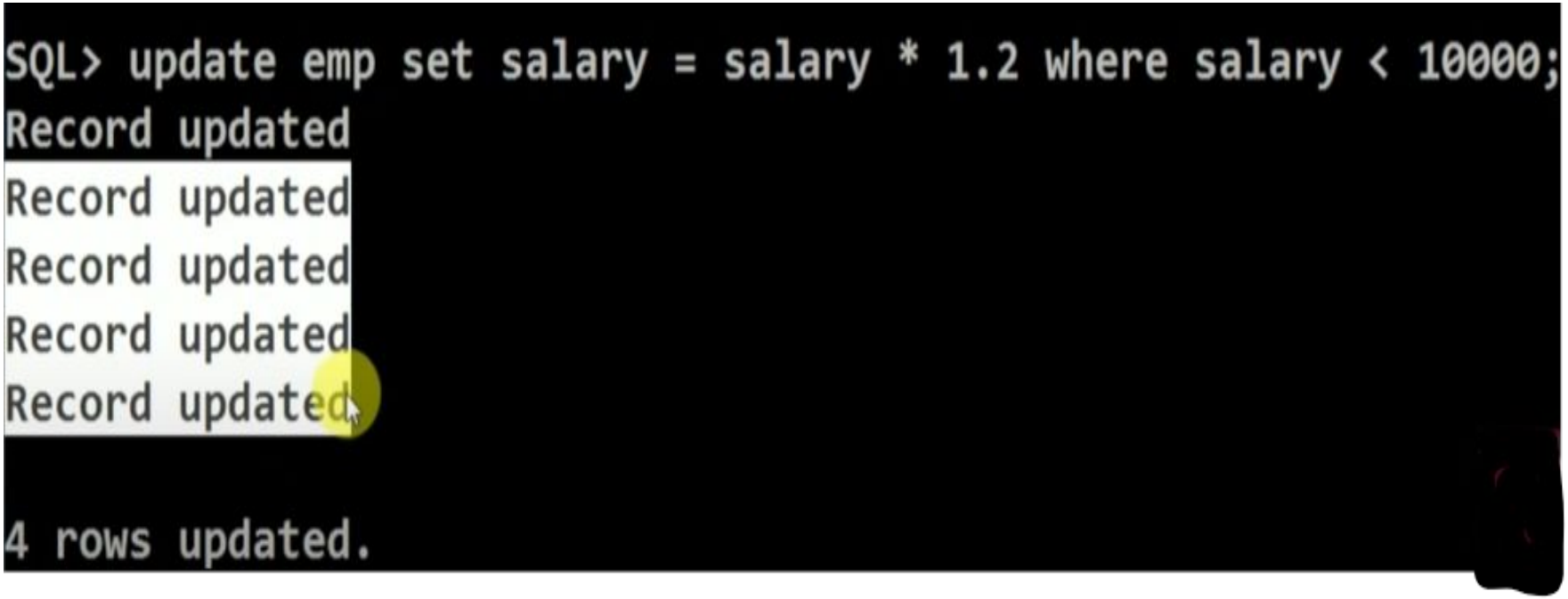
```
SQL> select * from emp;

       EID NAME                          SALARY
---------- ---------------- ----------
       100 Steven                         24000
       101 Neena                          17000
       102 Lex                            17000
       103 Alexander                       9000
       104 Bruce                           6000
       105 David                           4800
       106 Valli                           4800
       107 Diana                           4200

8 rows selected.
```


SQL> update emp set salary = salary * 1.2 where salary < 10000;
Record updated
5 rows updated.

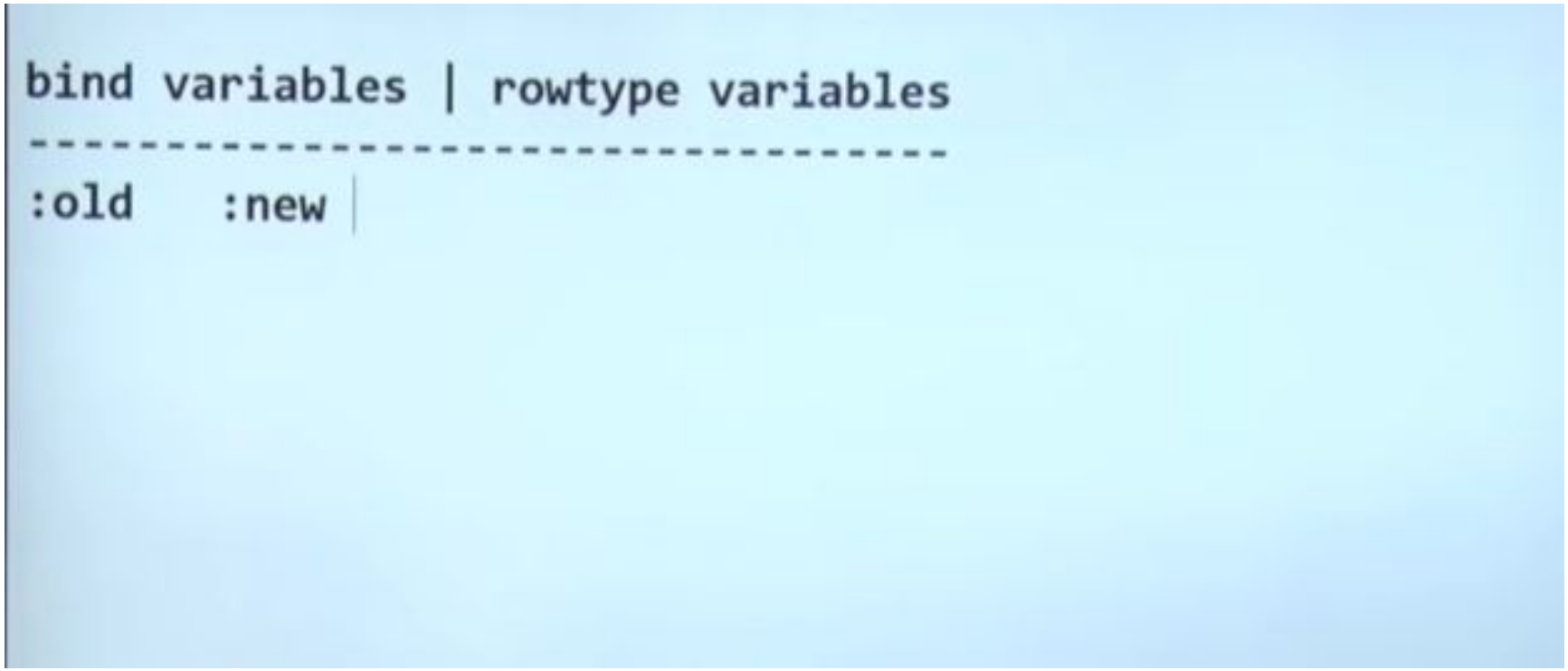
```
SQL> update emp set salary = salary * 1.2 where eid = 107;
Record updated

1 row updated.
```

```
SQL> select * from emp;

       EID NAME                      SALARY
---------- --------------- ----------
       100 Steven                     24000
       101 Neena                      17000
       102 Lex                        17000
       103 Alexander                  10800
       104 Bruce                       7200
       105 David                       5760
       106 Valli                       5760
       107 Diana                       6048

8 rows selected.
```

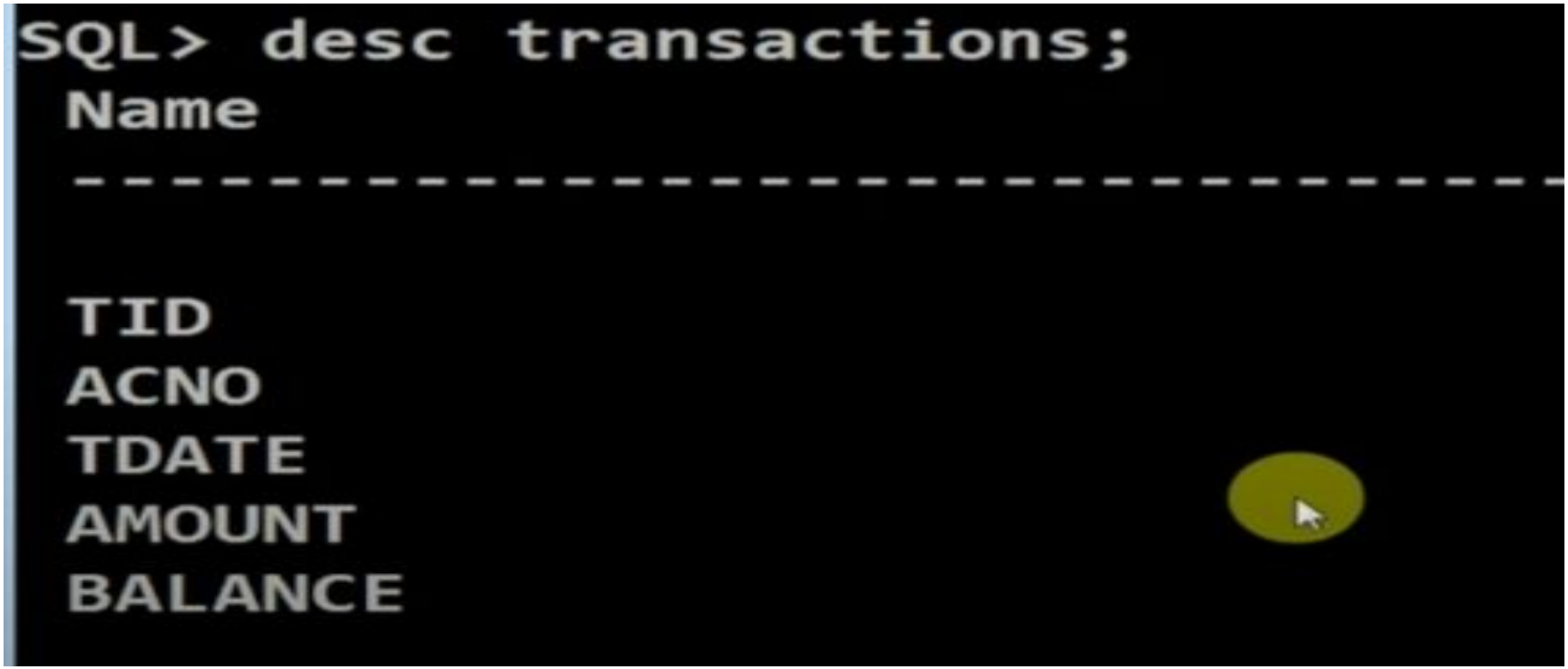
```
SQL> create or replace trigger emptrg1
  2  before update on emp for each row
  3  begin
  4  dbms_output.put_line('Record updated');
  5  end;
  6  /

Trigger created.
```


SQL> update emp set salary = salary * 1.2 where salary < 10000;
Record updated
Record updated
Record updated
Record updated
Record updated
4 rows updated.

```
SQL>  create or replace trigger hrtr1
  2  before drop on hr.schema
  3  begin
  4  dbms_output.put_line('Dropped');
  5  end;
  6  /

Trigger created.

SQL> drop table emp;
Dropped

Table dropped.
```


bind variables | rowtype variables
:old :new

```
SQL> select * from accounts;

      ACNO NAME                 BALANCE
---------- ---------- ----------
        10 A                    50000
        20 B                    75000
        30 C                    10000
        40 D                    92000
        50 E                    48000

SQL> select * from transactions;

no rows selected

SQL>
```


SQL> desc transactions;
Name
TID
ACNO
TDATE
AMOUNT
BALANCE


SQL> create sequence ac_seq;

```
create or replace trigger actr2
before update on accounts for each row
begin
insert into transactions
(tid, acno, tdate, amount, balance) values
(ac_seq.nextval, :new.acno, sysdate,
abs(:new.balance - :old.balance), :new.balance);
end;
/
```

```
      ACNO NAME            BALANCE
---------- ---------- ----------
        10 A                50000
        20 B                75000
        30 C                60000
        40 D                92000
        50 E                48000

SQL> select * from transactions;

       TID       ACNO TDATE          AMOUNT     BALANCE
---------- ---------- --------- ---------- ----------
         6         30 09-AUG-18      50000      60000

SQL> update accounts set balance = balance - 50000
  2  where acno = 40
```

THANK YOU