```
select user from dual
select *from tab
select *from instructor

//Trigger to monitor who (user) is working on a particular table
create or replace trigger idu_instructor
before insert or delete or update on instructor
for each row
declare
            v_user varchar2(30);
begin
            select user into v_user from dual;
            DBMS_OUTPUT.put_line('you are working on instructor Mr.'|| v_user);
end;


delete from instructor where id=1010

update instructor
set salary=salary*2;

insert into instructor
values(1012,'Abdul Ali','CSE',101000)

select *from instructor_BACKUP


/* Monitoring the user and the type of  DML operation */

create or replace trigger idu2_instructor
before insert or delete or update on instructor
for each row
declare
            v_user varchar2(30);
begin
            select user into v_user from dual;
if inserting then
             DBMS_OUTPUT.put_line('One row inserted on instructor by Mr.'|| v_user);
elsif deleting then
            DBMS_OUTPUT.put_line('One row deleted on instructor by Mr.'|| v_user);
elsif updating then
            DBMS_OUTPUT.put_line('One row updated on instructor by Mr.'|| v_user);
end if;
end;
```

```
create table audit_instructor
(new_salary number(10,2), old_salary number(10,2), user_name varchar2(20), entry_date varchar2(20),
operation varchar2(20))



select *from audit_instructor
/* Audit Trigger */

create or replace trigger audit_instructor
before insert or delete or update on instructor
for each row
declare
        v_user varchar2(30);
        v_date varchar2(30);
begin
        select user, TO_CHAR(sysdate,'DD/MON/YYYY HH24:MI:SS') into v_user, v_date from dual;
if inserting then
        insert into audit_instructor (new_salary, old_salary, user_name, entry_date, operation)
        values(:NEW.salary,null,v_user,v_date,'insert');
elsif deleting then
        insert into audit_instructor (new_salary, old_salary, user_name, entry_date, operation)
        values(null,:OLD.salary,v_user,v_date,'delete');
elsif updating then
        Insert into audit_instructor (new_salary, old_salary, user_name, entry_date, operation)
        values(:NEW.salary,:OLD.salary,v_user,v_date,'update');

end if;
end;

desc instructor_backup

/* Making Synchronized backup copy of a table using DML */
create table instructor_backup as select *from instructor


create or replace trigger backup_instructor
before insert or delete or update on instructor
for each row

begin
```

```
if inserting then
             insert into instructor_backup (id, name, dept, salary)
            values(:NEW.id,:NEW.name,:NEW.dept,:NEW.salary);
elsif deleting then
            delete from instructor_backup where id=:OLD.id;
elsif updating then
            update instructor_backup
            set name=:NEW.name, dept=:new.dept, salary=:new.salary
            where id=:old.id;

end if;
end;

delete from instructor where id=1003

update instructor
set salary=salary/2
where id=1003;

insert into instructor
values(1003,'Mikrani 2','CSE',101000)

select *from instructor_BACKUP

select *from instructor_backup
DELETE FROM INSTRUCTOR_BACKUP



/* DDL Trigger with Schema Auditing */

create or replace trigger hr_audit_tr
after DDL on schema

begin
   insert into schema_audit
   values(sysdate, sys_context('USERENV','CURRENT_USER'), ora_dict_obj_type, ora_dict_obj_name,
ora_sysevent);
end;



create table schema_audit3
```

```
(ddl_date date, ddl_user varchar2(15), object_created varchar2(15), object_name varchar2(15),
ddl_operation varchar2(15))

select *from schema_audit
describe schema_audit


/* Transaction to OFF on Certain Day/Days*/

create or replace trigger instructor_off
before insert or delete or update on instructor
for each row
declare

begin

if (TO_CHAR(sysdate,'day')='thursday ') then
      raise_application_error(-20000,'Can not perform DML Operation');
  end if;
end;


/* Transaction to OFF on Certain Hours*/

create or replace trigger instructor_off
before insert or delete or update on instructor
for each row
declare

begin

if (TO_CHAR(sysdate,'HH24') between '20' and '23') then
      raise_application_error(-20000,'Can not perform DML Operation');
  end if;
end;
```

```
update instructor
set salary=salary-1000000

insert into instructor
values(1021,'Abir','MSE',200)

/* Extraction of information from sysdate */

SELECT TO_CHAR
   (SYSDATE, 'MM-DD-YYYY HH24:MI:SS') "NOW"
    FROM DUAL;

NOW
-------------------
04-13-2001 09:45:51


select * from all_triggers
where table_name = 'INSTRUCTOR'

select * from USER_TRIGGERS
where table_name = 'INSTRUCTOR'

drop trigger tri1
```