```
---------------------------------------------
--Trigger Oluşturma - DML
---------------------------------------------

create or replace trigger trg_dep_mesai
before insert
on departman
begin

    if to_char(sysdate, 'D') in (7,1) then
        raise_application_error(-20001,
    'Hafta sonu veri girişi yapılmamalıdır');
    end if;

end;

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

create or replace trigger trg_dep_mesai
before insert or update or delete
on departman
begin

    if to_char(sysdate, 'D') in (7,1) then
        if INSERTING then
            raise_application_error(-20001, 'Hafta sonu veri girişi yapılmamalıdır');
        elsif DELETING then
            raise_application_error(-20002, 'Hafta sonu veri silmesi yapılmamalıdır');
        elsif UPDATING ('DEPT_ID') then
            raise_application_error(-20003, 'Hafta sonu veri güncellemesi yapılmamalıdır');
        end if;
    end if;

end;
---------------------------------------------

insert into departman values(150, 'Yeni ofis');

update departman set dept_ismi = 'Sakarya Yeni Ofis'
where dept_id = 100;

delete departman where dept_id = 100;

---------------------------------------------
--Row Level Trigger
---------------------------------------------

CREATE OR REPLACE TRIGGER trg_per_maas_kontrol
before insert or update on personel
for each row
declare
    maas_ok boolean := true;
begin

    case :new.dept_id
         when 110 then if :new.maas > 10000 then maas_ok := false; end if;
         when 100 then if :new.maas > 12000 then maas_ok := false; end if;
    end case;

    if not maas_ok then
       raise_application_error(-20002, 'Maaş değerini yüksek girdiniz');
    end if;

end;
```

```
--------------------------------------------
--Trigger Oluşturma - NEW, OLD Yapısı
--------------------------------------------

create or replace trigger trg_per_maas
after update on personel
--REFERENCING NEW AS NEW OLD AS OLD
for each row
declare
    v_maas_farki number;
begin

    v_maas_farki := :new.maas - :old.maas;
    dbms_output.put_line('Eski Maaş:'  || :old.maas);
    dbms_output.put_line('Yeni Maaş:'  || :new.maas);
    dbms_output.put_line('Maaş Farkı:' || v_maas_farki);

end;

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

CREATE OR REPLACE TRIGGER TRG_KON_ILLER
BEFORE INSERT OR UPDATE ON KONUM
REFERENCING NEW AS yeni OLD AS eski
FOR EACH ROW
declare
    v_adet integer;
begin

    select count(*) into v_adet from iller
    where il_kodu = :yeni.il_kodu;

    if v_adet = 0 then
       raise_application_error(-20001, 'Geçersiz il kodu girdiniz');
    end if;

end;


--------------------------------------------
--Trigger When İfadesi
--------------------------------------------

create or replace trigger trg_per_maas_kontrol
before insert or update on personel
for each row
WHEN (new.unvan = 'MÜHENDİS')
declare
    maas_ok boolean := true;
begin

    case :new.dept_id
         when 110 then if :new.maas > 10000 then maas_ok := false; end if;
         when 100 then if :new.maas > 12000 then maas_ok := false; end if;
    end case;

    if not maas_ok then
       raise_application_error(-20002, 'Maaş değerini yüksek girdiniz');
    end if;

end;
```

```sql
--------------------------------------------

create or replace trigger trg_is_ilani
  before insert or update
  on is_ilanlari
  for each row
  when (instr(lower(new.baslik),'developer') > 0)
begin
    :new.platform := 'Linkedin';
end;

--------------------------------------------
--Trigger İle Veri Aktarma
--------------------------------------------

create table personel_silinen as select * from personel where 1=2;

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

create or replace trigger trg_per_silinen
before delete on personel
referencing new as new old as old
for each row
begin

    insert into personel_silinen
    values(:OLD.PERSONEL_ID, :OLD.AD, :OLD.SOYAD, :OLD.MAAS, :OLD.GIRIS_TARIHI,
           :OLD.CIKIS_TARIHI, :OLD.SEMT, :OLD.PRIM, :OLD.UNVAN, :OLD.IZIN_GUNU,
           :OLD.YONETICI_ID, :OLD.KONUM_ID, :OLD.DEPT_ID);

end;

--------------------------------------------
--Trigger İle Log Tutma
--------------------------------------------

create table departman_log
(
    islem_tipi       varchar2(20),
    islem_saati      date default sysdate,
    eski_dept_id     number,
    eski_dept_ismi   varchar2(100),
    yeni_dept_id     number,
    yeni_dept_ismi   varchar2(100),
    kullanici        varchar2(30)
);

create or replace trigger trg_dep_log
before insert or update or delete on departman
for each row
begin
  if INSERTING then
    insert into departman_log (islem_tipi,yeni_dept_id,yeni_dept_ismi,kullanici)
    values ('Insert', :new.dept_id, :new.dept_ismi, user);
  elsif DELETING then
    insert into departman_log (islem_tipi,eski_dept_id,eski_dept_ismi,kullanici)
    values ('Delete', :old.dept_id, :old.dept_ismi, user);
  elsif UPDATING then
    insert into departman_log
(islem_tipi,eski_dept_id,eski_dept_ismi,yeni_dept_id,yeni_dept_ismi,kullanici)
    values ('Update', :old.dept_id, :old.dept_ismi,
            :new.dept_id, :new.dept_ismi, user);
  end if;
end;
```

```
--------------------------------------------
--Trigger İçinde Prosedür Çağırma
--------------------------------------------

create or replace trigger trg_dep_log
before insert or update or delete on departman
referencing new as new old as old
for each row
declare
    v_islem varchar2(20);
begin

    if INSERTING then
        v_islem := 'Insert';
    elsif DELETING then
        v_islem := 'Delete';
    elsif UPDATING then
        v_islem := 'Update';
    end if;

    pck_genel.dep_log_yaz (v_islem, :old.dept_id, :old.dept_ismi,
    :new.dept_id, :new.dept_ismi);

end;

procedure dep_log_yaz(
    p_islem_tipi varchar2,
    p_eski_id integer,
    p_eski_isim varchar2,
    p_yeni_id integer,
    p_yeni_isim varchar2)
is
begin

    insert into departman_log
    values(p_islem_tipi, sysdate, p_eski_id, p_eski_isim,
           p_yeni_id, p_yeni_isim, user);

end;

--------------------------------------------
--Trigger İçinde Prosedür Çağırma - Alıştırma Script
--------------------------------------------

create table table_dml_log
(
    dml_type varchar2(1),
    dml_time date default sysdate,
    user_name varchar2(50) default user,
    table_name varchar2(50),
    column_name varchar2(50),
    column_old_value varchar2(250),
    column_new_value varchar2(250)
);
```

```
--------------------------------------------
--Trigger Instead OF
--------------------------------------------

create table personel_ozluk as
select personel_id, ad, soyad, giris_tarihi, cikis_tarihi,
       semt, izin_gunu, yonetici_id, konum_id, dept_id
from personel;

create table personel_ucret as
select personel_id, maas, nvl(prim,0) prim, unvan,
    trunc(dbms_random.value(5,20)) zam_orani
from personel;

create or replace view vw_personel as
select * from personel;

create or replace trigger trg_per_dagit
instead of insert or update or delete on vw_personel
for each row
begin

    if INSERTING then
        insert into personel_ozluk
        values(:new.personel_id, :new.ad, :new.soyad,
               :new.giris_tarihi, :new.cikis_tarihi,
               :new.semt, :new.izin_gunu, :new.yonetici_id,
               :new.konum_id, :new.dept_id);
    elsif DELETING then
        delete from personel_ozluk where personel_id = :old.personel_id;
        delete from personel_ucret where personel_id = :old.personel_id;
    elsif UPDATING('MAAS') or UPDATING('PRIM') then
        update personel_ucret set maas = :new.maas where personel_id = :old.personel_id;
    end if;

end;

--------------------------------------------
--Trigger Çalışma Sırası
--------------------------------------------

create or replace trigger trg_iller_follow1
before insert on iller
for each row
begin
  dbms_output.put_line('trg_iller_follow1 - Çalıştırıldı');
end;

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

create or replace trigger trg_iller_follow2
before insert on iller
for each row
follows trg_iller_follow1
begin
  dbms_output.put_line('trg_iller_follow2 - Çalıştırıldı');
end;
```

```
-----------------------------------------------------
--ALIŞTIRMALARIN CEVAPLARI
-----------------------------------------------------

--------------------------------------------
--Trigger Oluşturma - DML
--------------------------------------------

create or replace trigger trg_product_segment
before insert
on product_segment
declare
    v_count pls_integer;
begin

    select count(*) into v_count from product_segment;

    if v_count = 3 then
        raise_application_error(-20001, 'There can be a maximum of 3 segments');
    end if;

end;

--------------------------------------------
--Trigger Oluşturma - NEW, OLD Yapısı
--------------------------------------------

create or replace trigger trg_cars
before insert or update on cars
referencing new as new old as old
for each row

begin

    if INSERTING or UPDATING then

        if :new.price not between 10000 and 2000000 then

            raise_application_error(-20001, 'Price is out of limits');

        end if;

        if :new.discount is null then

            raise_application_error(-20002, 'Discount can not be null');

        end if;

    end if;

end;
```

```
-------------------------------------------
--Trigger İçinde Prosedür Çağırma
-------------------------------------------

create or replace procedure write_dml_log
(
    p_dml_type varchar2,
    p_table_name varchar2,
    p_column_name varchar2,
    p_column_old_value varchar2,
    p_column_new_value varchar2
)
is

begin

    insert into table_dml_log(dml_type, table_name,
    column_name, column_old_value, column_new_value)
    values(p_dml_type, p_table_name, p_column_name,
    p_column_old_value, p_column_new_value);

end;

create or replace trigger trg_cars
before update on cars
referencing new as new old as old
for each row

begin

    if UPDATING then

        if :new.brand <> :old.brand then
            write_dml_log('U', 'CARS', 'BRAND', :old.brand, :new.brand);
        end if;
        if :new.price <> :old.price then
            write_dml_log('U', 'CARS', 'PRICE', :old.price, :new.price);
        end if;
        if :new.discount <> :old.discount then
            write_dml_log('U', 'CARS', 'DISCOUNT', :old.discount, :new.discount);
        end if;

    end if;

end;
```