```sql
--------------------------------------------
--Compound Trigger Oluşturma
--------------------------------------------

CREATE OR REPLACE TRIGGER TRG_PER_COMP
FOR INSERT OR UPDATE ON PERSONEL
COMPOUND TRIGGER

    BEFORE STATEMENT IS
    BEGIN
        dbms_output.put_line('Before Statement Çalıştı');
    END BEFORE STATEMENT;

    BEFORE EACH ROW IS
    BEGIN
        dbms_output.put_line('Before Each Row Çalıştı');
    END BEFORE EACH ROW;

    AFTER EACH ROW IS
    BEGIN
        dbms_output.put_line('After Each Row Çalıştı');
    END AFTER EACH ROW;

    AFTER STATEMENT IS
    BEGIN
        dbms_output.put_line('After Statement Çalıştı');
    END AFTER STATEMENT;

END;

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

create or replace trigger trg_per_comp
for insert or update on personel
compound trigger

    type t_per_array is table of number
        index by binary_integer;
    v_pers t_per_array;
    v_islem varchar2(1);

    after each row is
    begin
        v_pers(v_pers.count+1) := :new.personel_id;
    end after each row;

    after statement is
    begin
        if INSERTING then v_islem := 'I';
        else v_islem := 'U'; end if;
        forall i in 1 .. v_pers.count
            insert into per_log
            values (systimestamp, v_islem, v_pers(i));
    end after statement;

end;

create table per_log
(
    zaman        timestamp not null,
    islem_tipi   varchar2(1) not null,
    personel_id  integer not null
);
```

```
insert into personel
(personel_id, ad, soyad, maas)
values(6002, 'İsmet', 'Hayran', 8000);

insert into personel
(personel_id, ad, soyad, maas)
select 1000+rownum,
       'PersonelAd'||rownum,
       'PersonelSoyad'||rownum,
       round(dbms_random.value(2000, 10000),2)
from dual
connect by level <= 5;

update personel set prim = 150
where prim = 100;

--------------------------------------------
--Mutating Tables
--------------------------------------------

create or replace trigger tg_pers_mut
before insert or update of maas on personel
for each row
when (new.dept_id = 100)
declare
    v_min_maas number;
    v_max_maas number;
begin

    select min(maas), max(maas)
    into v_min_maas, v_max_maas
    from personel where dept_id = 100;

    if not (:new.maas between v_min_maas and v_max_maas) then
       raise_application_error(-20005, 'Maaş değeri sınırların dışındadır');
    end if;

end;

--------------------------------------------
--Compound Trigger İle Mutating Sorununu Çözme
--------------------------------------------

create or replace trigger tg_pers_mut
for insert or update of maas on personel
when (new.dept_id = 100)
compound trigger
    v_min_maas number;
    v_max_maas number;

    before statement is
    begin
        select min(maas), max(maas)
        into v_min_maas, v_max_maas
        from personel where dept_id = 100;
    end before statement;

    after each row is
    begin
        if not (:new.maas between v_min_maas and v_max_maas) then
           raise_application_error(-20005, 'Maaş değeri sınırların dışındadır');
        end if;
    end after each row;
end;
```

```
--------------------------------------------
--Compound Trigger - Alıştırma Scripti
--------------------------------------------

create or replace trigger trg_total_discount
    after insert or update
    on product_segment
    for each row
declare
    v_total_discount pls_integer;

begin

    -- get the total discount
    select sum (discount) into v_total_discount
    from product_segment;

    -- check total discount
    if v_total_discount + :new.discount - :old.discount > 0.25 then

        --:new.discount := :old.discount;
        update product_segment set discount = :old.discount
        where id = :new.id;

    end if;

end;
/

--------------------------------------------
--DDL Trigger - Schema
--------------------------------------------

create or replace trigger trg_no_drop
before drop on egitim.schema
begin
    raise_application_error(-20005,
    'Bu şema üzerinde herhangi bir obje drop edemezsiniz');
end;

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

CREATE OR REPLACE trigger EGITIM.TRG_DDL_SCHEMA
before create or alter or drop on egitim.schema
--when (ora_dict_obj_type = 'table')
begin

    dbms_output.put_line('Event Type:'   || ora_sysevent);
    dbms_output.put_line('Object Owner:' || ora_dict_obj_owner);
    dbms_output.put_line('Object Type:'  || ora_dict_obj_type);
    dbms_output.put_line('Object Name:'  || ora_dict_obj_name);

end;
```

```
--------------------------------------------

create or replace trigger trg_ddl_schema2
before create or alter
on schema
declare
    v_sqltext  dbms_standard.ora_name_list_t;
    v_sqlcount pls_integer;

begin

    v_sqlcount := ora_sql_txt(v_sqltext);

    for i in 1..v_sqlcount loop

        dbms_output.put_line('sqltext(' || i || ')=' || v_sqltext(i));

    end loop;
end;

--------------------------------------------
--DDL Trigger - Schema - Alıştırma Scripti
--------------------------------------------

create table ddl_log
(
     operation_type varchar2(50),
     object_owner varchar2(50),
     object_name varchar2(50),
     user_name varchar2(50),
     ddl_time date,
     sql_text clob
);

--------------------------------------------
--DDL Trigger - Database
--------------------------------------------

create table log_triggering
(
    kullanci_id varchar2(30),
    log_zamani timestamp,
    islem varchar2(50)
)

create or replace trigger trg_logon
after logon
on database
begin

    insert into log_triggering
    values(USER, SYSDATE,
           'Kullanıcı Login');

end;

create or replace trigger trg_logoff
before logoff
on database
begin
    insert into log_triggering
    values(USER, SYSDATE,
           'Kullanıcı Logoff');
end;
```

```
--------------------------------------------

CREATE TABLE log_info_session
(
    username    VARCHAR2(30),
    logon_date  DATE,
    session_id  VARCHAR2(30),
    ip_addr     VARCHAR2(30),
    hostname    VARCHAR2(30),
    auth_type   VARCHAR2(30)
);

CREATE OR REPLACE TRIGGER TRG_LOGOFF
BEFORE LOGOFF ON DATABASE
DECLARE
 session_id VARCHAR2(30);
 ip_addr    VARCHAR2(30);
 hostname   VARCHAR2(30);
 auth_type  VARCHAR2(30);
BEGIN

    SELECT sys_context ('USERENV', 'SESSIONID'),
           sys_context ('USERENV', 'IP_ADDRESS'),
           sys_context ('USERENV', 'HOST'),
           sys_context ('USERENV', 'AUTHENTICATION_TYPE')
    INTO session_id, ip_addr, hostname, auth_type
    FROM dual;

    INSERT INTO log_info_session VALUES
    (user, sysdate, session_id, ip_addr, hostname, auth_type);

END;
```

```
--------------------------------------------

CREATE TABLE LOG_ALL_DDL_DB
(
  USERNAME      VARCHAR2(30),
  LOG_DATE      DATE,
  SESSION_ID    VARCHAR2(30),
  IP_ADDR       VARCHAR2(30),
  HOSTNAME      VARCHAR2(30),
  AUTH_TYPE     VARCHAR2(30),
  EVENT_TYPE    VARCHAR2(30),
  OBJECT_OWNER  VARCHAR2(30),
  OBJECT_TYPE   VARCHAR2(30),
  OBJECT_NAME   VARCHAR2(30),
  SQL_TEXT      CLOB
);

CREATE OR REPLACE TRIGGER TRG_ALL_DDL_DB
AFTER DDL ON DATABASE
declare
    session_id varchar2(30);
    ip_addr    varchar2(30);
    hostname   varchar2(30);
    auth_type  varchar2(30);
    v_sqltext  dbms_standard.ora_name_list_t;
    v_sqlcount pls_integer;
    v_ddl_text clob;
begin
    v_sqlcount := ora_sql_txt(v_sqltext);
    for i in 1..v_sqlcount loop
        v_ddl_text := v_ddl_text || v_sqltext(i);
    end loop;

    select sys_context ('USERENV', 'SESSIONID'),
           sys_context ('USERENV', 'IP_ADDRESS'),
           sys_context ('USERENV', 'HOST'),
           sys_context ('USERENV', 'AUTHENTICATION_TYPE')
    into session_id, ip_addr, hostname, auth_type
    from dual;

    insert into log_all_ddl_db values
    (user, sysdate, session_id, ip_addr, hostname, auth_type,
     ora_sysevent, ora_dict_obj_owner, ora_dict_obj_type,
     ora_dict_obj_name, v_ddl_text);
end;
```

```
--------------------------------------------

CREATE OR REPLACE TRIGGER TRG_DDL_TABLO_KONTROL
AFTER DDL ON DATABASE
declare
    tabloKontrol integer := 0;
begin

    if ORA_SYSEVENT in ('CREATE') and ORA_DICT_OBJ_TYPE = 'TABLE' then

        select count(*) into tabloKontrol
        from dba_tables
        where table_name = ORA_DICT_OBJ_NAME;

        if tabloKontrol > 1 then

            raise_application_error(-20001,
                '['||ORA_DICT_OBJ_NAME||
                '] isimli tablodan başka bir şemada da bulunmaktadır.' ||
                'Lütfen tablo ismini değiştiriniz.');

        end if;
    end if;
end;
```

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

--------------------------------------------
--Compound Trigger
--------------------------------------------

create or replace trigger trg_total_discount
    for insert or update on product_segment
    compound trigger

    v_total_discount number;

    before statement is
    begin

        -- get the total discount
        select sum (discount) into v_total_discount
        from product_segment;

    end before statement;

    before each row is
    begin

        -- check total discount
        if v_total_discount + :new.discount - :old.discount > 0.25 then
            :new.discount := :old.discount;
        end if;

    end before each row;

end;
/

--------------------------------------------
--DDL Trigger - Schema
--------------------------------------------

create or replace trigger trg_schema_ddl
    after ddl on schema
declare
    v_sqltext_col  dbms_standard.ora_name_list_t;
    v_sqlcount pls_integer;
    v_sqltext clob;

begin

    v_sqlcount := ora_sql_txt(v_sqltext_col);
    v_sqltext := '';

    for i in 1..v_sqlcount loop

        v_sqltext := v_sqltext || v_sqltext_col(i);

    end loop;

    insert into ddl_log
    select ora_sysevent, ora_dict_obj_owner,
        ora_dict_obj_name, user, sysdate, v_sqltext
    from dual;

end;
```

```
--------------------------------------------
--DDL Trigger - Database
--------------------------------------------

create or replace trigger trg_ddl_truncate_control
before ddl on database
begin

    if ora_sysevent in ('TRUNCATE') and USER = 'TEST' and ora_dict_obj_owner = 'EGITIM' then

        raise_application_error(-20001,
            'The TEST user cannot truncate the table in the EGITIM schema!');

    end if;

end;
```