
--Standart: Exceptions Declaring

```
create or replace package pck_excep is

    e_personel_yok exception;
    e_maas_yukse exception;
    e_sayi_degil exception;

    pragma exception_init(e_personel_yok, -20001);
    pragma exception_init(e_maas_yukse, -20002);
    pragma exception_init(e_sayi_degil, -20003);

end pck_excep;

create or replace procedure personel_bul (p_personel_id number)
is
    v_cnt number;

begin
    select count(*) into v_cnt from personel
    where personel_id = p_personel_id;

    if v_cnt = 0 then
        raise pck_excep.e_personel_yok;
    end if;
end;

begin
    personel_bul(6666);
exception
    when pck_excep.e_personel_yok then
        raise_application_error(SQLCODE,
            'Personel bulunamadı, lütfen kontrol ediniz');
end;
```

--Standart: Constants

```
create or replace package pck_const is

    c_min_maas          constant number := 3500;
    c_max_maas          constant number := 15000;
    c_zam_orani         constant number := 12;
    c_ilkyil_izin_gunu  constant integer := 14;
    c_pi               constant real := 3.14159;
    c_default_bolge     constant varchar2(10) := 'MARMARA';

end pck_const;
```

```
-----

declare
    v_cnt number;
Begin

    select count(*) into v_cnt from personel
    where maas < pck_const.c_min_maas;

    dbms_output.put_line(v_cnt);
    dbms_output.put_line(pck_const.c_min_maas);
end;
```

```
-----  
--Lokal Alt Programlar  
-----
```

```
declare
```

```
function kareal(sayi number)  
return number as  
begin  
    return sayi * sayi;  
end;
```

```
begin
```

```
    dbms_output.put_line(kareal(25));
```

```
end;
```

```
-----  
declare
```

```
cursor crs_personel is  
    select ad, soyad from personel  
    where unvan = 'GRUP MÜDÜRÜ';  
v_isim varchar2(50);
```

```
function isim_format(p_ad varchar2, p_soyad varchar2)  
return varchar2 as  
begin  
    return 'Adı: ' || p_ad || ', Soyadı: ' || p_soyad;  
end;
```

```
begin
```

```
    for rowx in crs_personel  
    loop  
        v_isim := isim_format(rowx.ad, rowx.soyad);  
        dbms_output.put_line(v_isim);  
    end loop;
```

```
end;
```

```
-----  
--Invoker Yetkileri: AUTHID CURRENT_USER  
-----
```

```
create or replace procedure dept_ekle
```

```
(  
    p_id number,  
    p_isim varchar2  
)
```

```
AUTHID CURRENT_USER
```

```
is
```

```
Begin
```

```
    insert into departman values(p_id, p_isim);
```

```
end;
```

```
--Connect as another user
```

```
Create departman table (with same name)
```

```
exec egitim.dept_ekle(444, 'Invoker Deneme');
```

--Autonomous Transactions (AT)

--log_iller tablosunu oluřturuyoruz

```
create table log_iller as
select il_kodu, il_adi, sysdate tarih from iller
where 1=0;
```

--Olulřturulan tabloya log yazan prosedürü yazıyoruz

```
create or replace procedure log_iller_yaz
(p_ilkodu number, p_iladi varchar2)
is
pragma autonomous_transaction;
begin
insert into log_iller values(p_ilkodu, p_iladi, sysdate);
commit;
end;

declare
v_ilkodu pls_integer := 100;
v_iladi varchar2(30) := 'Selçuklu';
v_bolgekodu pls_integer := 3;

begin
insert into iller values(v_ilkodu, v_iladi, v_bolgekodu);
log_iller_yaz(v_ilkodu, v_iladi);
rollback;
end;
```

--Performance: NOCOPY Hint

```
CREATE OR REPLACE procedure personel_bilgi2
( p_personel_id in number,
  p_ad out nocopy varchar2 )
is
begin

select ad into p_ad from personel
where personel_id = p_personel_id;

end;
```

--Performance: PARALLEL ENABLE

```
create or replace function per_isim_getir(p_id number)
return varchar2 parallel_enable
is
v_isim varchar2(40);
begin
select ad into v_isim
from personel
where personel_id = p_id;

return v_isim;
end;
```

--Performance: RESULT_CACHE

```
create or replace function per_isim_getir(p_id number)
return varchar2 result_cache
is
    v_isim varchar2(40);
begin
    select ad into v_isim
    from personel
    where personel_id = p_id;

    return v_isim;
end;
```

--Performance: DETERMINISTIC

```
create or replace function pass_number (i number)
return number
deterministic
is
begin
    dbms_output.put_line ('pass_number çalıştırıldı: '||i);
    return i;
end;
```

```
declare
    n number := 0;
begin
    for rec in (select pass_number (1) from dual
                connect by level < 5)
    loop
        n := n + 1;
    end loop;

    dbms_output.put_line (n);
end;
```

--Performance: RETURNING

```
create or replace procedure maas_guncelle (p_id number)
is
    v_sonuc varchar2(50);
begin
    update personel
    set maas = maas * 1.15
    where personel_id = p_id
    returning ad||' '||soyad || ' : '||maas into v_sonuc;

    dbms_output.put_line(v_sonuc);
end;
```

```

-----
declare
    v_sonuc varchar2(50);
begin
    insert into konum
    values(sq_temp.nextval, 'Erzurum Tortum', 25)
    returning konum_id||', '||konum_adi || ', '||il_kodu into v_sonuc;

    dbms_output.put_line(v_sonuc);
end;

```

```

-----
--Performance: Bulk Binding
-----

```

```

create table bulk_bind_table1 (sayi number, deger varchar2(20));
create table bulk_bind_table2 (sayi number, deger varchar2(20));

create or replace procedure bind_table_ornek is
    type type_sayi is table of number index by pls_integer;
    type type_deger is table of varchar2(20) index by pls_integer;
    v_sayi type_sayi;
    v_deger type_deger;

    sayac constant pls_integer := 100000;
    t1 integer;
    t2 integer;
    t3 integer;
begin
    for i in 1..sayac loop
        v_sayi(i) := i;
        v_deger(i) := 'Değer No: '||i;
    end loop;
    t1 := dbms_utility.get_time;
    for j in 1..sayac loop
        insert into bulk_bind_table1 values(v_sayi(j), v_deger(j));
    end loop;
    t2 := dbms_utility.get_time;
    forall j in 1..sayac
        insert into bulk_bind_table2 values(v_sayi(j), v_deger(j));
    t3 := dbms_utility.get_time;

    dbms_output.put_line('For Loop: ' || to_char((t2-t1)/100));
    dbms_output.put_line('ForAll : ' || to_char((t3-t2)/100));
    commit;
end;

```

--Performance: Sorgu İçinde Bulk Collect Into

```
create or replace procedure konum_getir
is
    type konum_type is table of konum%rowtype;

    v_konum konum_type;

begin

    select * bulk collect into v_konum from konum order by 1;

    for i in 1..v_konum.count loop
        dbms_output.put_line(v_konum(i).konum_id || ' ' ||
            v_konum(i).konum_adi);
    end loop;

end;
```

--Performance: Returning İle Bulk Collect Into

```
create or replace procedure maaslara_zam_yap(zam_orani number)
is
    type perid_type is table of number;
    type maas_type is table of personel.maas%type
        index by pls_integer;

    v_perid perid_type;
    v_yeni_maaslar maas_type;

begin

    select personel_id bulk collect into v_perid
    from personel order by personel_id;

    forall i in v_perid.first..v_perid.last
        update personel set maas = maas * zam_orani
        where personel_id = v_perid(i)
        returning maas bulk collect into v_yeni_maaslar;

    for i in 1..v_yeni_maaslar.count loop
        dbms_output.put_line(v_perid(i) || ' : ' || round(v_yeni_maaslar(i),2));
    end loop;

end;
```

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

```
-----  
--Lokal Alt Programlar  
-----
```

```
create or replace function create_emp_email (emp_id number)  
return varchar2  
is  
    v_first_name employees.first_name%type;  
    v_last_name   employees.last_name%type;  
    v_email       varchar2(27);  
    v_hosting     varchar2(12) := 'dbhunter.net';  
    v_return_msg  varchar2(200);  
  
    -- declare and define procedure  
  
    procedure create_email  
    (  
        name1    varchar2,  
        name2    varchar2,  
        company  varchar2  
    )  
    is  
        error_message varchar2(35) := '...email address is too long...';  
  
    begin  
  
        v_email := lower(name1) || '.' || lower(name2) || '@' || company;  
        v_return_msg := v_email;  
  
        exception  
        when value_error then  
            v_return_msg := error_message;  
  
    end create_email;  
  
begin  
  
    select first_name, last_name  
        into v_first_name, v_last_name  
    from employees  
    where employee_id = emp_id;  
  
    create_email(v_first_name, v_last_name, v_hosting);  
  
    return v_return_msg;  
  
end;  
/
```