```
-------------------------------------------
--Paket Oluşturma
-------------------------------------------

create or replace package pck_genel as

end pck_genel;

---------

create or replace package body pck_genel as

end pck_genel;

-------------------------------------------
--Paket İçerisine Altprogram Ekleme
-------------------------------------------

--Body

create or replace package body pck_genel as

function date_diff(p_sure varchar2,p_d1 date, p_d2 date)
return number
as
    v_sonuc    number;
begin

select (p_d2 - p_d1) *
            decode( upper(p_sure),
                    'SS', 24*60*60, 'MI', 24*60, 'HH', 24, null )
    into v_sonuc from dual;

    return v_sonuc;

end;

end pck_genel;

--Spec

create or replace package pck_genel as

function date_diff(p_sure varchar2,p_d1 date, p_d2 date)
return number;

end pck_genel;
```

```
-------------------------------------------
--Paket Altprogramlarını Çağırma
-------------------------------------------

declare
    v_sure varchar2(20);
begin

    v_sure := pck_genel.date_diff('hh', sysdate-10, sysdate);
    dbms_output.put_line(v_sure);

end;

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

begin

    dbms_output.put_line(pck_genel.date_diff('hh', sysdate-10, sysdate));

end;

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

select pck_genel.date_diff('hh', sysdate-10, sysdate) sure from dual;

-------------------------------------------
--Paket İçerisine Altprogram Ekleme
-------------------------------------------

…
procedure out_yaz(p_deger varchar2)
is
begin
    dbms_output.put_line(p_deger);
end;

…

procedure maas_guncelle(p_id number, p_yeni_maas number)
is
begin

    update personel set maas = p_yeni_maas
    where personel_id = p_id;

    out_yaz('Güncellenen kayıt sayısı: ' || sql%rowcount);

end;
…
```

```
-------------------------------------------
--Body Bölümü Olmayan Paketler
-------------------------------------------

create or replace package pck_sabitler as

    c_mil2metre  constant number := 1609.3;
    c_fit2metre  constant number := 0.3048;
    c_metre2fit  constant number := 3.28;
    c_cm2inc     constant number := 0.39;

end pck_sabitler;

…

set serveroutput on;
begin

    dbms_output.put_line('50 Mil '|| 50 * pck_sabitler.c_mil2metre || ' metredir');

end;

-------------------------------------------
--Paketlerdeki Global Değişkenler
-------------------------------------------

create or replace package body pck_genel as

    v_global_number number := 0;

procedure set_global(p_deger number)
is
begin
    v_global_number := p_deger;
end;
…
…

set serveroutput on;
begin

    dbms_output.put_line(pck_genel.v_global_number);

    pck_genel.v_global_number := 100;

    dbms_output.put_line(pck_genel.v_global_number);

    pck_genel.set_global(20);

    dbms_output.put_line(pck_genel.v_global_number);

end;

-------------------------------------------
--Paketlerin Kodları Nerede Saklanıyor?
-------------------------------------------

select * from user_source
where name = 'TELNO_FORMATLA'
order by line;

select * from user_source
where name = 'PCK_GENEL'
      and type = 'PACKAGE BODY'
order by line;
```

```
-------------------------------------------
--Paketlerde Overloading
-------------------------------------------

…
procedure konum_ekle(p_konum_adi varchar2) is
    max_id integer;
begin

    select max(konum_id)+1 into max_id from konum;
    insert into konum values(max_id, p_konum_adi, 34);
    commit;

end;

procedure konum_ekle(p_konum_adi varchar2, p_il_kodu integer) is
    max_id integer;
begin

    select max(konum_id)+1 into max_id from konum;
    insert into konum values(max_id, p_konum_adi, p_il_kodu);
    commit;

end;
…

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

function date_diff(p_sure varchar2, p_d1 date, p_d2 date) return number as
    v_sonuc    number;
begin
    select (p_d2 - p_d1) *
            decode( upper(p_sure), 'SS', 24*60*60, 'MI', 24*60, 'HH', 24, null )
    into v_sonuc from dual;
    return v_sonuc;
end;

function date_diff(p_d1 date, p_d2 date) return number as
    v_sonuc    number;
    p_sure varchar2(2);
begin
    p_sure := 'ss';
    select (p_d2 - p_d1) *
            decode( upper(p_sure), 'SS', 24*60*60, 'MI', 24*60, 'HH', 24, null )
    into v_sonuc from dual;
    return v_sonuc;
end;

function date_diff(p_d1 date) return number as
    v_sonuc    number;
    p_sure varchar2(2) := 'ss';
    p_d2 date := sysdate;
begin
    select (p_d2 - p_d1) *
            decode( upper(p_sure), 'SS', 24*60*60, 'MI', 24*60, 'HH', 24, null )
    into v_sonuc from dual;
    return v_sonuc;
end;
```

```
-------------------------------------------
--Serially Reusable Paketler
-------------------------------------------

create or replace package not_serially_reusable_pkg as
  v_not_sr int := 0;
end;

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

create or replace package serially_reusable_pkg as
  pragma serially_reusable;
  v_sr int := 0;
end;

begin
    not_serially_reusable_pkg.v_not_sr := 100;
    serially_reusable_pkg.v_sr := 100;
end;

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

begin
    dbms_output.put_line ('not_serially: ' || not_serially_reusable_pkg.v_not_sr );
    dbms_output.put_line ('serially: ' || serially_reusable_pkg.v_sr );
end;

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

begin
    not_serially_reusable_pkg.v_not_sr := 100;
    serially_reusable_pkg.v_sr := 100;
    dbms_output.put_line ('not_serially: ' || not_serially_reusable_pkg.v_not_sr );
    dbms_output.put_line ('serially: ' || serially_reusable_pkg.v_sr );
end;
```

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

-----------------------------------------
--Paket İçerisine Altprogram Ekleme
-----------------------------------------

create or replace package pck_genel as

function yonetici_getir (p_personel_id number) return varchar2;

end pck_genel;

/

create or replace package body pck_genel as

    function yonetici_getir (p_personel_id number) return varchar2
    as
        v_yonetici_ismi varchar2(100);
    begin

        select
             pry.ad || ' ' ||pry.soyad
             into v_yonetici_ismi
        from personel pr, yonetici yn, personel pry
        where pr.yonetici_id = yn.yonetici_id
              and pry.personel_id = yn.personel_id
              and pr.personel_id = p_personel_id;

        return v_yonetici_ismi;

        exception
            when no_data_found then
                return 'Yönetici bulunamadı';

    end;

end pck_genel;
```

```
-----------------------------------------
--Paketlerdeki Global Değişkenler
-----------------------------------------

create or replace package pck_order_report as

procedure find_good_companys;

end pck_order_report;

/

create or replace package body pck_order_report as

    --global değişkenler tanımlanıyor
    vgb_category_id pls_integer;
    type tgb_order_id_type is varray(3) of integer;
    vgb_order_ids tgb_order_id_type := tgb_order_id_type(null, null, null);

procedure find_max_category
is
begin

    --stock ücreti en büyük miktara
    --sahip kategory tespit ediliyor
    select category_id into vgb_category_id from
    (
        select category_id,
            unit_price*units_in_stock as stock_price
        from products
        order by 2 desc
    )
    where rownum = 1;
    dbms_output.put_line('Category id: '||vgb_category_id);

end;

procedure find_orders
is
    --Yukarıda bulunan kategoideki en yüksek satış miktarlı
    --ilk 3 sipariş bulunuyor
    cursor crs_orders is
        select rownum, order_id from
        (
            select order_id, unit_price*quantity
            from order_details od
            where od.product_id in
                (
                    select product_id from products p
                    where category_id = vgb_category_id
                )
            order by 2 desc
        )
        where rownum < 4;
begin

    --Bulunan her bir sipariş id değer, global
    --bir varray dizisinin bir elemanına atanıyor
    for crs_row in crs_orders loop
        vgb_order_ids(crs_row.rownum) := crs_row.order_id;
        dbms_output.put_line('Order id: '||crs_row.order_id);
    end loop;

end;
```

```
procedure find_companys
is
    --Bulunan herbir siparişi veren şirketler bulunuyor
    --parmaetreli cursor kullanılıyor
    cursor crs_company(v_order_id integer) is
        select company_name from customers c
        where customer_id in
        (
            select customer_id from orders
            where order_id = v_order_id
        );
begin

    for i in 1..vgb_order_ids.count loop

        --Her bir sipariş id, cusror'e parametre olarak gönderiliyor
        for crs_row in crs_company(vgb_order_ids(i)) loop

            dbms_output.put_line('Company name: '|| crs_row.company_name);

        end loop;

    end loop;

end;

procedure find_good_companys
is
begin
    --Dışarıdan çağrılacak ana fonksiyon.
    --Diğer fonksiyonları sırayla çağırıyor
    find_max_category;
    find_orders;
    find_companys;
end;

end pck_order_report;
```