```
-----------------------------------------
--Prosedürler - Örnek
-----------------------------------------

create procedure karesini_al
is
begin

    for i in reverse 1..10 loop

        dbms_output.put_line(lpad(i,2,' ') || ' ==> karesi : ' || i**2);

    end loop;

end;

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

create or replace procedure uzman_yazdir is

    cursor c_personel is
        select ad, soyad, maas from personel
        where unvan = 'UZMAN';

begin

    for row_per in c_personel loop

        dbms_output.put_line(row_per.ad ||' '||
                             row_per.soyad || ': '||
                             row_per.maas);
    end loop;

end;

-----------------------------------------
--Prosedürler - Parametreli
-----------------------------------------

create or replace procedure personel_yazdir (p_unvan varchar2)
is

    cursor c_personel is
        select ad, soyad, maas from personel
        where unvan = p_unvan;

begin

    for row_per in c_personel loop

        dbms_output.put_line(row_per.ad    || ' ' ||
                             row_per.soyad || ': '||
                             row_per.maas  || ', '||
                             row_per.unvan);
    end loop;

end;
```

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

create or replace procedure konum_ekle
    (
        p_konum_id konum.konum_id%type,
        p_konum_adi varchar2,
        p_il_kodu number
    )
is

begin

    insert into konum
    values(p_konum_id, p_konum_adi, p_il_kodu);

    commit;

end;

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

create or replace procedure personel_bilgi
    (
        p_personel_id number
    )
is
    v_ad varchar2(50);
    v_unvan varchar2(30);
    v_maas number;
begin

    select ad, unvan, maas into v_ad, v_unvan, v_maas
    from personel
    where personel_id = p_personel_id;

    dbms_output.put_line(v_ad || ', ' || v_unvan || ': ' || v_maas);

end;

-----------------------------------------
--Prosedürler – Parametreli (OUT)
-----------------------------------------

create or replace procedure personel_bilgi
    (
        p_personel_id   in  number,
        p_ad            out varchar2,
        p_maas          out number
    )
is
begin
    select ad, maas into p_ad, p_maas
    from personel
    where personel_id = p_personel_id;
end;

declare
    v_ad varchar2(50);
    v_maas number;

begin
    personel_bilgi(5020, v_ad, v_maas);
    dbms_output.put_line(v_ad || ': ' || v_maas);
end;
```

```
-------------------------------------------
--Prosedürler - Parametreli (IN OUT)
-------------------------------------------

create or replace procedure telno_formatla
    (
      p_telno  IN OUT  varchar2
    )
is
begin

    p_telno := '(' ||
               substr(p_telno,1,3) || ') '||
               substr(p_telno,4,3) || ' ' ||
               substr(p_telno,7,2) || ' ' ||
               substr(p_telno,9,2);

end;

declare
    v_telefon_no varchar2(20) := '5859638541';
begin

    telno_formatla(v_telefon_no);
    dbms_output.put_line(v_telefon_no);

end;

-------------------------------------------
--Prosedürler - Parametreleri Dinamik Verme
-------------------------------------------

create table faaliyetler
(
     faaliyet_id   number,
     faaliyet      varchar2(100),
     faaliyet_gunu date
);

create procedure faaliyet_ekle
    (
        p_id  number := -1,
        p_adi varchar2 default 'Doğum günü',
        p_gunu date default sysdate
    )
is
begin

    insert into faaliyetler values(p_f_id, p_f_adi, p_f_gunu);
    commit;

end;

exec faaliyet_ekle;
exec faaliyet_ekle(7);
exec faaliyet_ekle(10, 'Haftalık raporlar');
exec faaliyet_ekle(1, 'Yılbaşı partisi', to_date('31.12.2020', 'dd.mm.yyyy'));
exec faaliyet_ekle(p_id => 2, p_adi => 'Dünya yazılımcılar günü', p_gunu => sysdate+10);
exec faaliyet_ekle(p_gunu => sysdate-20, p_id => 3, p_adi => 'Ramazan bayramı');
exec faaliyet_ekle(4, 'Eşimin doğum günü partisi', p_gunu => add_months(sysdate, 2));
exec faaliyet_ekle(p_id=>5, 'Hoşgeldin bahar pikniği', p_gunu => sysdate-22); --!!
exec faaliyet_ekle(p_id=>6, p_adi => 'Proje kapanış etkinliği', sysdate);  --!!
exec faaliyet_ekle(p_adi ='Günlük faaliyetler');
exec faaliyet_ekle(p_gunu => to_date('01.01.2021','dd.mm.yyyy'));
```

```
-----------------------------------------
--Fonksiyonlar - Örnek
-----------------------------------------

create or replace function f_faktoryel (p_sayi number)
return number
is
    v_sonuc number := 1;

begin

    for i in reverse 1..p_sayi loop

        v_sonuc := v_sonuc * i;

    end loop;

    return v_sonuc;
end;

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

create or replace function f_ucret_duzeyi(p_id number)
return varchar2
is
    v_ucret_duzey varchar2(30);

begin

    select uc.aciklama into v_ucret_duzey
    from personel pr, ucret_duzey uc
    where personel_id = p_id
        and pr.maas between uc.maas_alt_limit and uc.maas_ust_limit;

    return v_ucret_duzey;

end;

-----------------------------------------
--Fonksiyonları SQL İçinde Kullanma
-----------------------------------------

select ad, soyad, maas,
       f_ucret_duzeyi(personel_id) ucret_duzeyi
from personel

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

select ucret_duzeyi, count(*) adet from
(
    select f_ucret_duzeyi(personel_id) ucret_duzeyi
    from personel
)
group by ucret_duzeyi
```

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

create or replace function f_kesinti
(p_id personel.personel_id%type)
return number
is
    v_kesinti number;

begin

    select decode(unvan,
        'UZMAN', 0.05,
        'MÜDÜR', 0.08,
        'GRUP MÜDÜRÜ', 0.20,
        0) * maas into v_kesinti
    from personel
    where personel_id = p_id;

    return v_kesinti;

end;

select ad, soyad, unvan,
       f_kesinti(personel_id) kesinti
from personel
order by 4 desc;

select dept_id,
   max(f_kesinti(personel_id)) maks_kesinti
from personel
group by dept_id

-----------------------------------------
--Fonksiyonlar - Örnek
-----------------------------------------

create or replace function f_date_diff
(
    p_sure_tipi in varchar2,
    p_d1        in date,
    p_d2        in date
)
return number
as
    v_sonuc    number;

--  p_sure_tipi değeri=> ss : Saniye, mi : Dakika, hh : Saat

begin

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

    return v_sonuc;
end;
```

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

create or replace function f_zam_orani_hesapla(p_id number) return number
is
    v_zam_orani number;
    v_unvan varchar2(20);

begin

    select unvan into v_unvan from personel
    where personel_id = p_id;

    case v_unvan
        when 'UZMAN' then v_zam_orani := 1.05;
        when 'MÜDÜR' then v_zam_orani := 1.10;
        when 'TEKNİKER' then v_zam_orani := 1.07;
        else
            raise_application_error(-20001, 'Bu unvana ait zam oranı bulunamadı');
    end case;

    return v_zam_orani;
    exception
        when no_data_found then
            raise_application_error(-20002, p_id || ' numaralı personel bulunamadı!!');
            return null;

end;

-----------------------------------------
--Fonksiyonlar – Result Cache
-----------------------------------------

create or replace function f_bilgi_rc(p_id number)
return varchar2
result_cache
is
    v_ad varchar2(40);
begin

    select ad into v_ad
    from personel
    where personel_id = p_id;

    dbms_output.put_line(p_id|| ': '||v_ad);
    return v_ad;

end;

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

declare
    type t_sicil is table of number;
    v_sicil t_sicil;
    v_cikti varchar2(50);

begin
    v_sicil := t_sicil(5010, 5020, 5030, 5010, 5050);
    for i in 1..v_sicil.count loop

        v_cikti := f_bilgi_rc(v_sicil(i));
        dbms_output.put_line(v_cikti);

    end loop;
end;
```

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

-----------------------------------------
--Prosedürler
-----------------------------------------

create or replace procedure print_grand_lux_products
is

     cursor c_product is
        select name, price, price * discount as discount,
            price - price * discount as net_price
        from product p, product_segment ps
        where p.segment_id = ps.id
            and ps.segment = 'Grand Luxury';

begin

    for row_product in c_product loop

        dbms_output.put_line(row_product.name ||' Price: '||
                             row_product.price || ' Discount: '||
                             row_product.discount|| ' Net Price: '||
                             row_product.net_price);
    end loop;

end;

-----------------------------------------
--Prosedürler - Parametreli
-----------------------------------------

create or replace procedure top_ten_orders(p_ship_via number)
is
    cursor c_order_info is
        select * from
        (
            select first_name, last_name, freight
            from orders o, employees e
            where o.ship_via = p_ship_via
                and o.employee_id = e.employee_id
            order by freight desc
        )
        where rownum < 11;

begin

    for v_order_info in c_order_info loop

        dbms_output.put_line(v_order_info.first_name || ' ' ||
            v_order_info.last_name || ': ' ||
            v_order_info.freight);

    end loop;

end;
```

```
-------------------------------------------
--Fonksiyonları SQL İçinde Kullanma
-------------------------------------------

create or replace function get_manager (p_emp_id number)
return varchar2
result_cache
is
    v_manager_name varchar2(100);

begin

    select first_name || ' ' || last_name
        into v_manager_name
    from employees
    where employee_id = p_emp_id;

    return v_manager_name;

end;

-------------------------------------------
--Fonksiyonlar
-------------------------------------------

create or replace function get_total_orders(
    p_year pls_integer
)
return number
is
    v_total_orders number := 0;

begin
    -- get total sales
    select sum(unit_price * quantity)
        into v_total_orders
    from order_details
        inner join orders using (order_id)
    where shipped_date is not null
    group by extract(year from order_date)
    having extract(year from order_date) = p_year;

    return v_total_orders;
end;
```