

```
-----  
--DBMS_OUTPUT  
-----
```

```
set serveroutput on;  
begin
```

```
    dbms_output.put_line('put_line');  
    dbms_output.put('put');  
    dbms_output.new_line;  
    dbms_output.put_line('put_line');
```

```
end;
```

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

```
    lines dbms_output.chararr;  
    num_lines number := 0;
```

```
begin
```

```
    -- enable the buffer with default size 20000  
    dbms_output.enable;
```

```
    dbms_output.put_line('Merhaba!');  
    dbms_output.put_line('İkinci satırı ekliyoruz!');  
    dbms_output.put_line('Üçüncü satırı ekliyoruz!');
```

```
    dbms_output.get_lines(lines, num_lines);
```

```
    for i in 1..num_lines loop
```

```
        dbms_output.put_line(i||'. satır: '||lines(i));
```

```
    end loop;
```

```
end;
```

```
-----  
--Oracle Dizini Oluşturma  
-----
```

```
$ docker exec -it db_oracle_registry bash
```

```
bash-4.2$ mkdir egitim_dir
```

```
bash-4.2$ pwd
```

```
CREATE OR REPLACE DIRECTORY EGITIM_DIR AS  
'C:\Oracle\product\12.2.0\Egitim_dir';
```

```
GRANT READ, WRITE ON DIRECTORY EGITIM_DIR TO EGITIM;
```

```
-----  
--UTL_FILE: Dosya Yazma  
-----
```

```
procedure dosya_yaz(p_dosya_ismi varchar2, p_dizin varchar2)  
is  
    filex utl_file.file_type;  
  
begin  
  
    filex := utl_file.fopen(p_dizin, p_dosya_ismi, 'W');  
    utl_file.put_line(filex, 'Merhaba');  
    utl_file.put_line(filex, 'Bu dosya PL/SQL eğitimi için oluşturuldu');  
    utl_file.put_line(filex, 'UTL_FILE paketi dosya okuma/yazma');  
    utl_file.fclose(filex);  
  
end;
```

```
-----  
--UTL_FILE: Dosya Okuma  
-----
```

```
procedure dosya_oku(p_dosya_ismi varchar2, p_dizin varchar2)  
is  
    filex utl_file.file_type;  
    satir varchar2(150);  
    satir_sayisi pls_integer := 1;  
begin  
    if not utl_file.is_open(filex) then  
        filex := utl_file.fopen(p_dizin, p_dosya_ismi, 'R');  
  
        begin  
            loop  
                utl_file.get_line(filex, satir);  
                dbms_output.put_line(satir_sayisi || '. satır: ' || satir);  
                satir_sayisi := satir_sayisi + 1;  
            end loop;  
            exception when no_data_found then  
                dbms_output.put_line('--- Dosya Sonu ---');  
        end;  
        dbms_output.put_line('Toplam: ' || satir_sayisi || ' satır okundu');  
        utl_file.fclose(filex);  
    end if;  
end;
```

```
-----  
--UTL_MAIL - Örnek-1  
-----
```

```
--connect as sysdba ve utl_mail paketi kurulur  
@ORACLE_HOME/rdbms/admin/utlmail.sql  
@ORACLE_HOME/rdbms/admin/prvtmail.sql  
  
--smtp parametresi ayarlanır  
ALTER SYSTEM SET SMTP_OUT_SERVER='smtp.server.com' SCOPE=SPFILE;  
  
--mail göndermek için örnek  
begin  
    utl_mail.send('tuncay@oracle.com', 'egitim@oracle.com'  
        message=> 'PL/SQL eğitim notlarını iyi alınız'  
        subject=> 'Eğitim notları');  
end;
```

```
-----  
--UTL_MAIL - Örnek-2  
-----
```

```
declare  
    l_attachment    raw(32767);  
  
begin  
  
    select rawimage  
        into l_attachment  
        from my_images  
    where id = 1;  
  
    utl_mail.send_attach_raw  
    (  
        sender      => 'me@domain.com',  
        recipients  => 'person1@domain.com,person2@domain.com',  
        subject     => 'UTL_MAIL Test',  
        message     => 'Mail testi yapılıyor, dikkate almayın!',  
        attachment  => l_attachment, --resim_getir('sql.gif', dizin)  
        att_filename => 'clouds.jpg'  
    );  
  
exception  
    when others then  
        raise_application_error(-20001,'Beklenmeyen bir hata oluştu: ' || sqlerrm);  
end;
```

```
-----  
--UTL_MAIL - RAW Dosya Oluşturma  
-----
```

```
create or replace function resim_getir  
(  
    p_dosya_ismi varchar2,  
    p_dizin      varchar2  
)  
return raw is  
  
    resim raw(32767);  
    dosya BFILE := BFILENAME(p_dizin, p_dosya_ismi);  
  
begin  
  
    DBMS_LOB.FILEOPEN(dosya, DBMS_LOB.FILE_READONLY);  
    resim := DBMS_LOB.SUBSTR(dosya);  
    DBMS_LOB.CLOSE(dosya);  
    return resim;  
  
end;
```

--UTL_MAIL - Text Dosya Oluşturma

```
create or replace function text_getir
(
    p_dosya_ismi varchar2,
    p_dizin varchar2
)
return varchar2 is

    icerik varchar2(32767);
    dosya BFILE := BFILENAME(p_dizin, p_dosya_ismi);

begin

    DBMS_LOB.FILEOPEN(dosya, DBMS_LOB.FILE_READONLY);
    icerik := UTL_RAW.CAST_TO_VARCHAR2(DBMS_LOB.SUBSTR(dosya));
    DBMS_LOB.CLOSE(dosya);
    return icerik;

end;
```

--ALIŞTIRMALARIN CEVAPLARI

--UTL_FILE: Dosya Yazma

```
create or replace procedure write_student_info(p_dir varchar2, p_file_name varchar2)
is
    filex utl_file.file_type;
begin
    filex := utl_file.fopen(p_dir, p_file_name, 'W');
    for std_row in (select * from student) loop
        utl_file.put_line(filex,
            'Name: ' || std_row.name ||
            ', Course: ' || std_row.course_name ||
            ', Class: ' || std_row.class_no);
    end loop;
    utl_file.fclose(filex);
end;
exec write_student_info('EGITIM_DIR', 'student_info.txt');
```

--UTL_FILE: Dosya Okuma

```
create or replace procedure read_student_info(p_dir varchar2, p_file_name varchar2)
is
    filex utl_file.file_type;
    row_text varchar2(300);
    row_count pls_integer := 0;
    sci_count pls_integer := 0;
begin
    if not utl_file.is_open(filex) then
        filex := utl_file.fopen(p_dir, p_file_name, 'R');

        begin
            loop
                utl_file.get_line(filex, row_text);

                if instr(lower(row_text), 'science') > 0 then
                    dbms_output.put_line(row_count || '. ' || row_text);
                    sci_count := sci_count + 1;
                end if;

                row_count := row_count + 1;

            end loop;

            exception
                when no_data_found then
                    dbms_output.put_line('--- End of file ---');

        end;
        dbms_output.put_line('Sum of rows: ' || row_count);
        dbms_output.put_line('Sum of science: ' || sci_count);
        utl_file.fclose(filex);

    end if;

    exception
        when utl_file.invalid_operation then
            dbms_output.put_line('File not found');
end;
```