```
----------------------------------------------------------------------------------
---------------------------------
---------------------Using the DBMS_OUTPUT Package (Code Samples)----------------
----------------------------------------------------------------------------------
-------------------------------
EXEC dbms_output.put_line('Test No:1');
/
SET SERVEROUTPUT ON;
EXEC dbms_output.put_line('Test No:2');
/
EXEC dbms_output.put('Test No:3');
/
EXEC dbms_output.put_line('Test No:4');
/
SET SERVEROUTPUT OFF
/
CREATE TABLE temp_table(ID NUMBER GENERATED ALWAYS AS IDENTITY, text
VARCHAR2(1000));
/
EXEC dbms_output.enable;
EXEC dbms_output.put_line('Hi');
/
DECLARE
    v_buffer VARCHAR2(1000);
    v_status INTEGER;
BEGIN
    dbms_output.put('...');
    dbms_output.put_line('Hello');
    dbms_output.put_line('How are you');
    FOR I IN 1..10 LOOP
        dbms_output.get_line(v_buffer,v_status);
        IF v_status = 0 THEN
            INSERT INTO temp_table(text) VALUES (v_buffer);
        END IF;
    END LOOP;
END;
/
SELECT * FROM temp_table;
/
SET SERVEROUTPUT ON;
DECLARE
    v_buffer VARCHAR2(1000);
    v_status INTEGER;
BEGIN
    dbms_output.put('...');
    dbms_output.put_line('Hello');
    dbms_output.put_line('How are you');
    dbms_output.get_line(v_buffer,v_status);
END;
/
SET SERVEROUTPUT OFF;
EXEC dbms_output.enable;
/
DECLARE
    v_buffer dbms_output.chararr;
    v_num_lines INTEGER:= 30;
BEGIN
    dbms_output.put('...');
    dbms_output.put_line('Hello');
    dbms_output.put_line('How are you');
    dbms_output.get_lines(v_buffer,v_num_lines);
    FOR i IN 1..v_num_lines LOOP
        INSERT INTO temp_table(text) VALUES (v_buffer(I));
    END LOOP;
END;
/
DROP TABLE temp_table;

-----------------------------------------------------------------------------------
----------------------------------
---------------------Using the UTL_FILE Package (Code
Samples)----------------------
-----------------------------------------------------------------------------------
--------------------------------

-------------------CREATE DIRECTORY------------------------------------
CREATE DIRECTORY test_dir AS 'C:\My Folder';
/
-------------------GET ALL THE EXISTING DIRECTORIES--------------------
SELECT * FROM all_directories;
/
-------------------READ FROM A FILE------------------------------------
SET SERVEROUTPUT ON;
DECLARE
    v_file UTL_FILE.FILE_TYPE;
    v_line VARCHAR2(32767);
BEGIN
    v_file := UTL_FILE.FOPEN('TEST_DIR', 'temp file.txt', 'R', 32767);
    LOOP
        UTL_FILE.GET_LINE(v_file, v_line);
        dbms_output.put_line (v_line);
    END LOOP;
    EXCEPTION
        WHEN no_data_found THEN
            dbms_output.put_line('The whole file is read!');
            UTL_FILE.FCLOSE(v_file);
END;
/
-------------------GRANT OR REVOKE READ-WRITE PRIVILEGES---------------
GRANT READ, WRITE ON DIRECTORY test_dir TO hr;
REVOKE READ, WRITE ON DIRECTORY test_dir FROM hr;
/
-------------------WRITE TO A FILE USING PUT_LINE PROCEDURE-------------
DECLARE
    v_file UTL_FILE.FILE_TYPE;
BEGIN
    v_file := UTL_FILE.FOPEN('TEST_DIR', 'temp file.txt', 'w', 32767);
    FOR r_emp IN (select * from employees) LOOP
        UTL_FILE.PUT_LINE(v_file, r_emp.first_name||' '||r_emp.last_name);
    END LOOP;
    UTL_FILE.FCLOSE(v_file);
END;
/
-------------------WRITE TO A FILE USING PUT AND NEW_LINE---------------
DECLARE
    v_file UTL_FILE.FILE_TYPE;
BEGIN
    v_file := UTL_FILE.FOPEN('TEST_DIR', 'temp file.txt', 'w', 32767);
    FOR r_emp IN (select * from employees) LOOP
        UTL_FILE.PUT(v_file, r_emp.first_name||' '||r_emp.last_name);
        UTL_FILE.NEW_LINE(v_file);
    END LOOP;
    UTL_FILE.FCLOSE(v_file);
END;
/
-------------------WRITE TO A FILE USING PUTF---------------------------
DECLARE
    v_file UTL_FILE.FILE_TYPE;
BEGIN
    v_file := UTL_FILE.FOPEN('TEST_DIR', 'temp file.txt', 'w', 32767);
    FOR r_emp IN (select * from employees) LOOP
        UTL_FILE.PUTF(v_file, '--> %s %s',r_emp.first_name,r_emp.last_name);
        --UTL_FILE.NEW_LINE(v_file);
        --UTL_FILE.PUTF(v_file, '--> %s %s\n',r_emp.first_name,r_emp.last_name);
    END LOOP;
    UTL_FILE.FCLOSE(v_file);
END;
/
-------------------USING FFLUSH TO WRITE IMMEDIATELY-------------------
DECLARE
    v_file UTL_FILE.FILE_TYPE;
BEGIN
    v_file := UTL_FILE.FOPEN('TEST_DIR', 'temp file.txt', 'w', 32767);
    FOR r_emp IN (select * from employees) LOOP
        UTL_FILE.PUT_LINE(v_file,r_emp.first_name||' '||r_emp.last_name);
        --UTL_FILE.FFLUSH(v_file);
        --UTL_FILE.PUT_LINE(v_file,r_emp.first_name||' '||r_emp.last_name,true);
        DBMS_SESSION.SLEEP(1);
    END LOOP;
    UTL_FILE.FCLOSE(v_file);
END;
/
-------------------CHECK FILE ATTRIBUTES-----------------------------
DECLARE
    v_fexists       BOOLEAN;
    v_file_length   NUMBER;
    v_block_size    BINARY_INTEGER;
BEGIN
    UTL_FILE.FGETATTR('TEST_DIR','temp
file.txt',v_fexists,v_file_length,v_block_size);
    IF v_fexists THEN
        DBMS_OUTPUT.PUT_LINE('The file exists');
        DBMS_OUTPUT.PUT_LINE('Its length is     :'||v_file_length);
        DBMS_OUTPUT.PUT_LINE('Its block size is :'||v_block_size);
    ELSE
        DBMS_OUTPUT.PUT_LINE('The file does not exist!');
    END IF;
END;
/
-------------------COPY THE FILE---------------------------------------
EXECUTE UTL_FILE.FCOPY('TEST_DIR','temp file.txt','TEST_DIR','temp file copy.txt');
/
-------------------COPY THE FILE EX2-----------------------------------
EXECUTE UTL_FILE.FCOPY('TEST_DIR','temp file.txt','TEST_DIR','temp file
copy2.txt',1,5);
/
-------------------RENAME THE FILE-------------------------------------
```

```
EXECUTE UTL_FILE.FRENAME('TEST_DIR','temp file copy2.txt','TEST_DIR','temp file
renamed.txt');
/
-------------------REMOVE THE FILE------------------------------------
EXECUTE UTL_FILE.FREMOVE('TEST_DIR','temp file renamed.txt');
EXECUTE UTL_FILE.FREMOVE('TEST_DIR','temp file copy.txt');
EXECUTE UTL_FILE.FREMOVE('TEST_DIR','temp file.txt');
/
-------------------DROP THE DIRECTORY---------------------------------
DROP DIRECTORY test_dir;

----------------------------------------------------------------------------------
----------------------------------
---------------------Using the UTL_MAIL Package (Code Samples)---------------------
----------------------------------------------------------------------------------
--------------------------------
--Sending an email with the least number of parameters
BEGIN
    UTL_MAIL.send(
                  sender     => 'somebody@somedomain.com',
                  recipients => 'oraclemaster@outlook.com',
                  subject    => 'Example 1: Test Email Subject',
                  message    => 'This is a test email from someone.'
                 );
END;
/
--Sending an email with specific names to the sender and recipients
BEGIN
    UTL_MAIL.send(
                  sender     => 'Some Person <somebody@somedomain.com>',
                  recipients => 'Oracle Masters <oraclemaster@outlook.com>',
                  subject    => 'Example 2: Test Email Subject',
                  message    => 'This is a test email from someone.'
                 );
END;
/
--Sending an email with using all of the parameters
BEGIN
    UTL_MAIL.send(
                  sender     => 'somebody@somedomain.com',
                  recipients => 'oraclemaster@outlook.com',
                  cc         =>
'somemanager@somedomain.something,someotherperson@somedomain.something',
                  bcc        => 'someothermanager@somedomain.com',
                  subject    => 'Example 3: Test Email Subject',
                  message    => 'This is a test email from someone.',
                  mime_type  => 'text/plain; charset=us-ascii',
                  priority   => 1,
                  replyto    => 'somereplyaddress@somedomain.com'
                 );
END;
/
--Sending an email by dynamically filling the message body
DECLARE
    cursor cur_top_earning_emps is
                    select employee_id, first_name, last_name, salary
                    from hr.employees
                    where salary > 10000
                    order by salary desc;
    v_message varchar2(32767);
BEGIN
    v_message := 'EMPLOYEE ID'||CHR(9)||'FIRST NAME'||CHR(9)||'LAST NAME'||
CHR(9)||'EMPLOYEE ID'||CHR(13);
    for r_top in cur_top_earning_emps loop
        v_message := v_message||r_top.employee_id||CHR(9)||r_top.first_name||
CHR(9)||r_top.last_name||CHR(9)||r_top.salary||CHR(13);
    end loop;

    UTL_MAIL.send(
                  sender     => 'topearnings@somedns.com',
                  recipients => 'oraclemaster@outlook.com',
                  subject    => 'Example 4: The Employees Earning More Than
$10000',
                  message    => v_message
                 );
END;
/
--Sending an HTTP mail
DECLARE
    cursor cur_top_earning_emps is
                    select employee_id, first_name, last_name, salary
                    from hr.employees
                    where salary > 10000
                    order by salary desc;
    v_message varchar2(32767);
BEGIN
    v_message := '<!DOCTYPE html>
                    <html>
                       <head>
                          <meta charset=''Cp1252''>
                          <title>Top Earning Employees</title>
                          <meta name="viewport" content="width=device-width,
initial-scale=1.0">
                          <style>
                             * {
                             margin: 0;
                             padding: 0;
                             }
                             body {
                             font: 14px/1.4 Georgia, Serif;
                             }
                             /*
                             Generic Styling, for Desktops/Laptops
                             */
                             table {
                             width: 100%;
                             border-collapse: collapse;
                             }
                             /* Zebra striping */
                             tr:nth-of-type(odd) {
                             background: #eee;
                             }
                             th {
                             background: #333;
                             color: white;
                             font-weight: bold;
                             }
                             td, th {
```

```
                             padding: 6px;
                             border: 1px solid #9B9B9B;
                             text-align: left;
                             }
                             @media
                             only screen and (max-width: 760px),
                             (min-device-width: 768px) and (max-device-width:
1024px)  {
                             table, thead, tbody, th, td, tr { display: block; }
                             thead tr { position: absolute;top: -9999px;left: -
9999px;}
                             tr { border: 1px solid #9B9B9B; }
                             td { border: none;border-bottom: 1px solid #9B9B9B;
position: relative;padding-left: 50%; }
                             td:before { position: absolute;top: 6px;left:
6px;width: 45%; padding-right: 10px; white-space: nowrap;}
                             /*
                             Label the data
                             */
                             td:nth-of-type(0):before { content: "EMPLOYEE_ID"; }
                             td:nth-of-type(1):before { content: "FIRST_NAME"; }
                             td:nth-of-type(2):before { content: "LAST_NAME"; }
                             td:nth-of-type(3):before { content: "SALARY"; }
                             }
                             }
                          </style>
                          <!--<![endif]-->
                       </head>
                       <body>
                          <h1 style = ''text-align :center;
color:green;''>Employees Earning more than $10.000 Per/month</h1>
                          <br>
                          <table>
                             <thead>
                                <tr>
                                   <th>EMPLOYEE_ID</th>
                                   <th>FIRST_NAME</th>
                                   <th>LAST_NAME</th>
                                   <th>SALARY</th>
                                </tr>
                             </thead>
                             <tbody id="data">';
    for r_top in cur_top_earning_emps loop
        v_message := v_message|| '<tr>'||
                                     '<td
align="right">'||r_top.employee_id||'</td>'||
                                     '<td>'||r_top.first_name||'</td>'||
                                     '<td>'||r_top.last_name||'</td>'||
                                     '<td align="right">'||r_top.salary||'</td>'||
                                 '</tr>';

    end loop;
    v_message := v_message||'           </tbody>
                                      </table>
                                   </body>
                                </html>';
    UTL_MAIL.send(
                  sender     => 'topearnings@somedns.com',
                  recipients => 'oraclemaster@outlook.com',
                  subject    => 'Example 5: The Employees Earning More Than $10000
(HTML Formatted)',
                  message    => v_message,
                  mime_type  => 'text/html'
                 );
END;
/
------------------SEND ATTACH RAW------------
--Create a temp table
CREATE TABLE temp_table(
  id        NUMBER,
  blob_data BLOB
);
/
--2) Create a directory object
CREATE OR REPLACE DIRECTORY BLOB_DIR AS 'C:\blob_directory\';
/
--3)Write a PL/SQL Block to load your external file into a BLOB/CLOB column.
DECLARE
  v_bfile       BFILE;
  v_blob        BLOB;
  v_dest_offset INTEGER := 1;
  v_src_offset  INTEGER := 1;
BEGIN
  INSERT INTO temp_table (id, blob_data)
      VALUES (222,  empty_blob())
          RETURNING blob_data INTO v_blob;

  v_bfile := BFILENAME('BLOB_DIR', 'test_file.jpeg');
  DBMS_LOB.fileopen(v_bfile, DBMS_LOB.file_readonly);
  DBMS_LOB.loadblobfromfile (
             dest_lob    => v_blob,              -- Destination lob
             src_bfile   => v_bfile,             -- Source file path and name in
the OS
             amount      => DBMS_LOB.lobmaxsize, -- Maximum LOB size.
             dest_offset => v_dest_offset,       -- Destination offset.
             src_offset  => v_src_offset);       -- Source offset.
  DBMS_LOB.fileclose(v_bfile);
  COMMIT;
END;
/
--4) Check the table to see if we could insert the blob file or not
SELECT * FROM temp_table;
/
--5) Send email with an attachment
DECLARE
    v_file BLOB;
    v_rawbuf RAW(32767);
BEGIN
    select blob_data into v_file from temp_table where rownum = 1;
    v_rawbuf := dbms_lob.substr(v_file);
    UTL_MAIL.send_attach_raw
    (
        sender => 'somebody@somedomain.com',
        recipients => 'oraclemaster@outlook.com',
        subject => 'Example 6: Attachment Test',
        message => 'This is a raw data',
        attachment => v_rawbuf,
        att_inline => TRUE,
        att_filename => 'testImage.jpeg'
    );
END;
/
DROP DIRECTORY blob_dir;
DROP TABLE temp_table;
/
--5) Send email with a text attachment
BEGIN
    UTL_MAIL.send_attach_varchar2
    (
        sender => 'somebody@somedomain.com',
        recipients => 'oraclemaster@outlook.com',
        subject => 'Example 7: Text Attachment Test',
        message => 'This is a text data',
        attachment => 'This is the text that will be written inside of the text
file.',
        att_inline => TRUE,
        att_filename => 'testTextFile.txt'
    );
END;






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






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






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