

Oracle PL/SQL

Advanced Level Uygulamalar

(1)

Mehmet ÇELEBİOĞLU

Oracle DBA, Forms&Reports, ISO 27001 (MSA / LA), PL/SQL Developer

mcAkademi9@gmail.com

www.udemy.com'da Yayında Olan Önceki Dersler

Oracle Database Kurulum ve Konfigurasyon

Oracle SQL, Sıfırdan İleri Düzeye

Oracle PL/SQL, Veritabanı Programlama

Öğrencilerime diledikleri zaman canlı destek verdiğim

<https://www.facebook.com/groups/CELEBIHOCA/> grubuma üye olmayı unutmayınız.

ORACLE PL/SQL.....	1
TEXT DOSYALARLA ÇALIŞMAK.....	1
TEXT DOSYADAN VERİ OKUMAK	2
TEXT DOSYAYA VERİ YAZMAK 1	3
TEXT DOSYAYA VERİ YAZMAK 2	4
TEXT DOSYAYA VERİ YAZMAK 3	5
XML TABLOLARI İLE ÇALIŞMAK	6
SYS_XMLAGG	7
XMLELEMENT.....	8
XMLFOREST.....	9
SORGU SONUCUNU XML FORMATINA ÇEVİRMEK (TABLO KOLONLARI XML TAG OLARAK KULLANMAK)	10
ÖRNEK UYGULAMA (A): BİR TABLODAKİ BİLGİLERİ XML FORMATINA ÇEVİRELİM	11
ÖRNEK UYGULAMA (B): XML FORMATINDAKİ BİR VERİYİ STANDART VERİ FORMATINA ÇEVİRELİM	14
XML DOSYA OLUŞTURMA 2. YÖNTEM (XML TABLODAN OKUYUP DOSYA.XML OLUŞTURMAK).....	16
XML DOSYALARI OKUMAK İÇİN FONKSİYONLAR	17
EXTRACT	18
EXTRACTVALUE	19
XMLSEQUENCE.....	20
WEB SERVİS UYGULAMASI (TCMB GÜNLÜK KUR)	21
GÖREV ZAMANLAMA TANIMLAMALARI (JOB).....	23
EXCEL TABLOLARI İLE ÇALIŞMAK.....	24
DIRECTORY CREATE ETMEK VE KULLANICI YETKİLENDİRMEK	25
VERİLERİ EXCEL'E ATACAK PROCEDURE YAZALIM (1. YÖNTEM).....	26
VERİLERİ EXCEL'E ATALIM (2. YÖNTEM)	28
EXCEL'DEN VERİ OKUMAK –SQL DEVELOPER VE TOAD	30
ERİŞİM KONTROL LİSTESİ (ACL) (ORA-24247 HATASI).....	31

TEXT DOSYALARLA ÇALIŞMAK

Bu bölümde TEXT dosya oluşturmak, TEXT dosyadan veri okumak, TEXT dosyaya veri yazmak konularında örnekler yapacağız.

TEXT Dosyadan Veri Okumak

```
SQL> START ...EMP_DEPT.sql;

CREATE or replace DIRECTORY USER_DIR AS 'd:\PLSQL_ADVANCED\TEXTS';

GRANT READ, WRITE ON DIRECTORY USER_DIR TO MEHMET;

DECLARE
Paragraf VARCHAR2(1200); --32767
txtdosya1 UTL_FILE.FILE_TYPE;
BEGIN
    Txtdosya1 := UTL_FILE.FOPEN('TEXT_DIR','dosya.txt','R');
    Loop
        BEGIN
            UTL_FILE.GET_LINE(txtdosya1,Paragraf);
            dbms_output.put_line(Paragraf);
            EXCEPTION WHEN No_Data_Found THEN EXIT;
        END;
    end loop;
IF UTL_FILE.IS_OPEN(txtdosya1) THEN
    dbms_output.put_line('File1 is Open');
    UTL_FILE.FCLOSE(txtdosya1);
END IF;
END;
```

Text Dosyaya Veri Yazmak 1

Bu dersimizde de bir Text dosyadan veri okuyup başka bir Text dosyaya veri yazacağız.

```
DECLARE
Paragraf VARCHAR2(1200); --32767
txtdosya1 UTL_FILE.FILE_TYPE;
txtdosya2 UTL_FILE.FILE_TYPE;
I NUMBER :=0;
BEGIN
    Txtdosya1 := UTL_FILE.FOPEN('TEXT_DIR','dosya1.txt','R');
    Txtdosya2 := UTL_FILE.FOPEN('TEXT_DIR','dosya2.txt','W');
    Loop
        I := I + 1;
        BEGIN
            UTL_FILE.GET_LINE(txtdosya1,Paragraf);
            dbms_output.put_line(Paragraf);
            UTL_FILE.PUT_LINE(txtdosya2,I || ' . Satır' || Paragraf);
            EXCEPTION WHEN No_Data_Found THEN EXIT;
        END;
    end loop;
    IF UTL_FILE.IS_OPEN(txtdosya1) THEN
        dbms_output.put_line('File1 is Open');
        UTL_FILE.FCLOSE(txtdosya1);
    END IF;
    IF UTL_FILE.IS_OPEN(txtdosya2) THEN
        dbms_output.put_line('File2 is Open');
        UTL_FILE.FCLOSE(txtdosya2);
    END IF;
END;
```

Text Dosyaya Veri Yazmak 2

Bu dersimizde de veritabanındaki bir tablodan veri okuyup Text dosyaya yazacağız.

```
DECLARE
Paragraf VARCHAR2(1200); --32767
txtdosya1 UTL_FILE.FILE_TYPE;
CURSOR EMP_OKU IS SELECT EMPNO, ENAME, JOB, SAL FROM EMP ORDER BY EMPNO;
EMP_REC EMP_OKU%ROWTYPE;

BEGIN
    Txtdosya1 := UTL_FILE.FOPEN('TEXT_DIR','dosya.txt','W');
    Paragraf := 'Sicil No' || ' ' || 'İsmi' || ' ' || 'İş' || ' ' || 'Maaşı'; -- Başlıkları attık
    UTL_FILE.PUT_LINE(txtdosya1,Paragraf);

    OPEN EMP_OKU;
    Loop
        FETCH EMP_OKU INTO EMP_REC;
        EXIT WHEN EMP_OKU%NOTFOUND;
        Paragraf := Emp_rec.Empno || emp_rec.ename || emp_rec.job || emp_rec.sal;
        UTL_FILE.PUT_LINE(txtdosya1,Paragraf);
    end loop;
    IF UTL_FILE.IS_OPEN(txtdosya1) THEN
        dbms_output.put_line('File1 is Open');
        UTL_FILE.FCLOSE(txtdosya1);
    END IF;
END;
```

Text Dosyaya Veri Yazmak 3

Bu dersimizde de veritabanındaki iki tablodan veri okuyup Text dosyaya yazacağız.

```
DECLARE
Paragraf VARCHAR2(1200); --32767
txtdosya1 UTL_FILE.FILE_TYPE;

CURSOR DEPT_OKU IS SELECT DEPTNO, DNAME FROM DEPT ORDER BY DEPTNO;
DEPT_REC DEPT_OKU%ROWTYPE;

CURSOR EMP_OKU(p_deptno emp.deptno%type) IS SELECT EMPNO, ENAME, JOB, SAL FROM
EMP ORDER BY EMPNO;
EMP_REC EMP_OKU%ROWTYPE;

BEGIN
    Txtdosya1 := UTL_FILE.FOPEN('TEXT_DIR','dosya.txt','W');

    OPEN DEPT_OKU;
    LOOP
        FETCH DEPT_OKU INTO DEPT_REC;
        EXIT WHEN DEPT_OKU%NOTFOUND;
        UTL_FILE.PUT_LINE(txtdosya1, ' '); -- Boş Satır Geçtik
        Paragraf := 'Bölüm No / Bölüm Adı : ' || dept_rec.deptno || ' / ' || dept_rec.dname;
        UTL_FILE.PUT_LINE(txtdosya1, paragraf);
        Paragraf := 'Sicil No' || ' İsmi ' || ' İş ' || ' Maaşı '; -- Başlıkları attık
        UTL_FILE.PUT_LINE(txtdosya1, Paragraf);
        OPEN EMP_OKU(dept_rec.deptno);
        Loop
            FETCH EMP_OKU INTO EMP_REC;
            EXIT WHEN EMP_OKU%NOTFOUND;
            Paragraf := Emp_rec.Empno || emp_rec.ename || emp_rec.job || emp_rec.sal;
            UTL_FILE.PUT_LINE(txtdosya1, Paragraf);
        end loop;
        CLOSE EMP_OKU;
    END LOOP;
    CLOSE DEPT_OKU;

    IF UTL_FILE.IS_OPEN(txtdosya1) THEN
        UTL_FILE.FCLOSE(txtdosya1);
    END IF;
END;
```

XML TABLOLARI İLE ÇALIŞMAK

SYS_XMLAGG, XMLELEMENT, XMLFOREST

SYS_XMLAGG

Oracle PL / SQL'de, SYS_XMLAGG, altındaki giriş XML örneklerinin değerlerini birleştirerek bir XML belgesinin bağımsız bir düğüm ögesini üreten bir fonksiyondur. ROWSET, varsayılan düğüm ögesi adıdır. Sırasıyla iki XMLtype Örnek tipi ve XMLformat örnek türü parametresini kabul eder.

Sözdizimi

SYS_XMLAGG(ifade, Format)

Basit Örnek

```
SQL> SELECT SYS_XMLAGG(sys_xmlgen(ename)) from EMP;
```

XMLELEMENT

İlişkisel verilerden XML öğeleri oluşturmak için SQL / XML standart XMLElement işlevini kullanır. Argümanlar bir öğe adı, öğe için isteğe bağlı öznitelik koleksiyonu ve öğe içeriğini oluşturan sıfır veya daha fazla ek argüman olarak alır. Bir XMLType örneği döndürür.

Basit Örnek

```
SELECT XMLElement("Tarih", HIREDATE) FROM EMP;

SELECT XMLElement("Emp",
                  XMLElement("isim", ENAME),
                  XMLElement("BASTAR", hiredate)) AS "RESULT"

FROM emp e;
```

XMLFOREST

Bir XML öğeleri kümesi oluşturmak için SQL / XML standart işlev XMLForest'i kullanırsınız. Argümanları, isteğe bağlı diğer adlarla birlikte değerlendirilecek ifadelerdir

Örnek (Kullanıcı Tanımlı Data Tipinde Çıktı Elde Etmek)

Önce birkaç tane OBJECT ve TYPE oluşturalım.

```
CREATE OR REPLACE TYPE emp_t AS OBJECT ("@EMPNO" NUMBER(4),  
                                         ENAME VARCHAR2(10));  
  
CREATE OR REPLACE TYPE emplist_t AS TABLE OF emp_t;  
  
CREATE OR REPLACE TYPE dept_t AS OBJECT ("@DEPTNO" NUMBER(2),  
                                         DNAME VARCHAR2(14),  
                                         EMP_LIST emplist_t);
```

```
SELECT XMLForest(  
    dept_t(DEPTNO,  
           DNAME,  
           cast(MULTISET  
               (SELECT EMPNO, ENAME  
                 FROM EMP e WHERE e.DEPTNO = d.DEPTNO)  
               AS emplist_t  
           )  
    ) AS "Department"  
) AS deptxml  
FROM DEPT d  
WHERE DEPTNO=10;
```

Sorgu Sonucunu XML Formatına Çevirmek (Tablo Kolarları XML Tag Olarak Kullanmak)

```
SELECT sys_xmlagg(  
    xmlelement(  
        "EMP",  
        xmlforest(e.empno, e.ename, e.job, e.mgr, e.hiredate, e.sal, e.comm)  
    )  
    )  
FROM emp e;
```

The screenshot shows the Oracle SQL Developer interface. The left pane displays the database schema with tables like EMP, EMP2, EMP4, EMP5, ILCLER, ILILLER, KURLAR, SALES, SALGRADE, SEHRLER, SPARIS, and TAX. The main window shows a query in the Worksheet tab:

```
SELECT sys_xmlagg(  
    xmlelement(  
        "EMP",  
        xmlforest(e.empno, e.ename, e.job, e.mgr, e.hiredate, e.sal, e.comm)  
    )  
    )  
FROM emp e;
```

The Query Result tab shows the XML output of the query. The XML is a single row with a root element <EMP> containing employee data. The XML structure is as follows:

```
<?xml version="1.0"?><ROWSET><EMP><EMPNO>7639</EMPNO><ENAME>KING</ENAME><JOB>PRESIDENT</JOB><HIREDATE>1981-11-17</HIREDATE><SAL>60000</SAL></EMP></ROWSET>
```

The XML output is truncated in the screenshot, showing only the first few rows of the result set.

Örnek Uygulama (a): Bir Tablodaki Bilgileri XML formatına Çevirelim

Bu çalışmamızda EMP tablosundaki bilgileri XML formatına çevirecek, XMLTYPE türünde bir tabloya INSERT edeceğiz.

XMLTYPE türündeki Tabloları

CREATE TABLE <Table-Name> OF XMLTYPE sözdizimiyle create ederiz.

EMP_XML isimli bir tablo create edelim.

CREATE TABLE EMP_XML OF XMLTYPE;

Şimdi EMP tablosundaki bilgileri XML tipine çevirip EMP_XML tablosuna atalım.

DECLARE

D_xmltype XMLTYPE;

BEGIN

SELECT sys_xmlagg(

xmlelement(

"EMP",

xmlforest(e.empno, e.ename, e.job, e.mgr, e.hiredate, e.sal, e.comm)

)

)

INTO D_xmltype

FROM emp e;

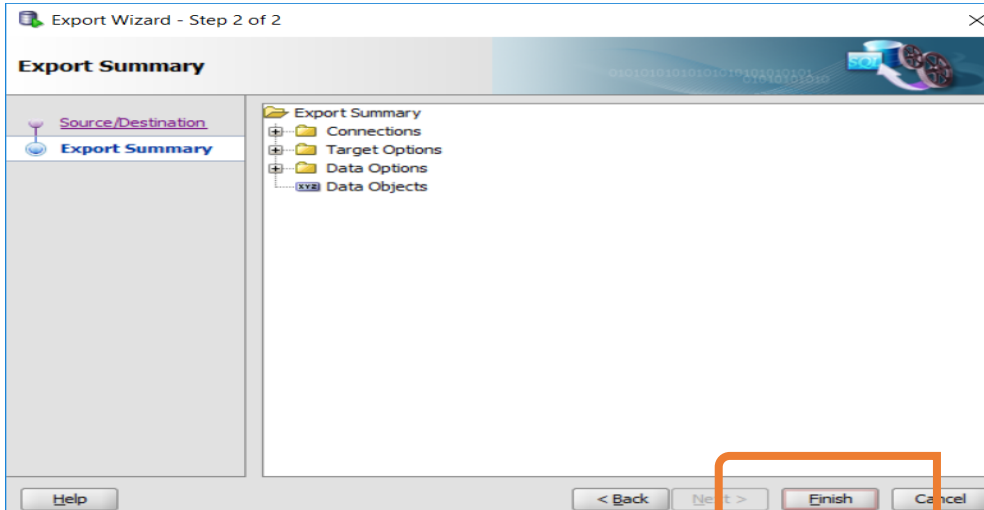
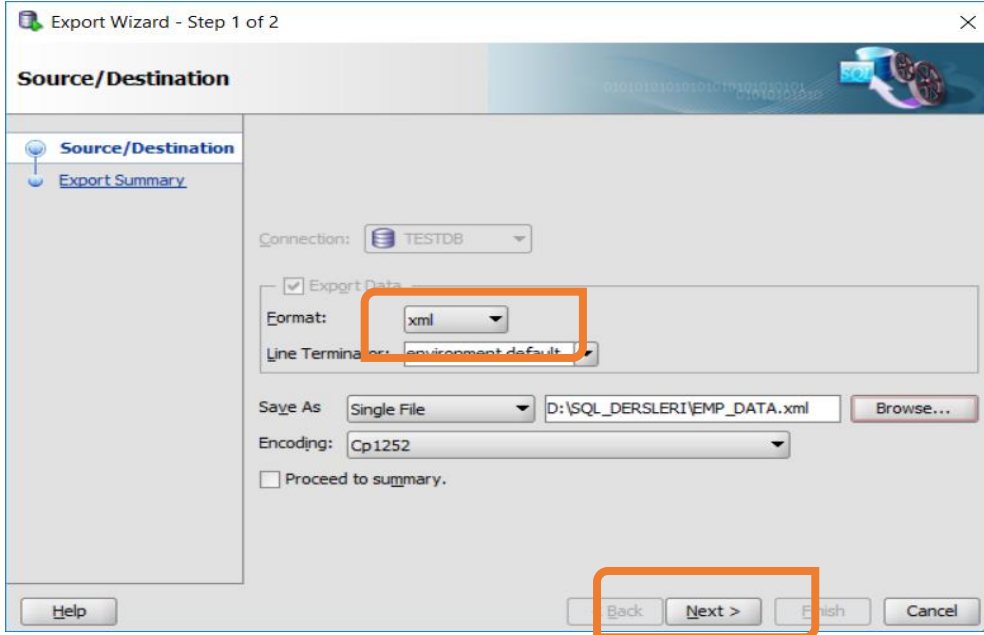
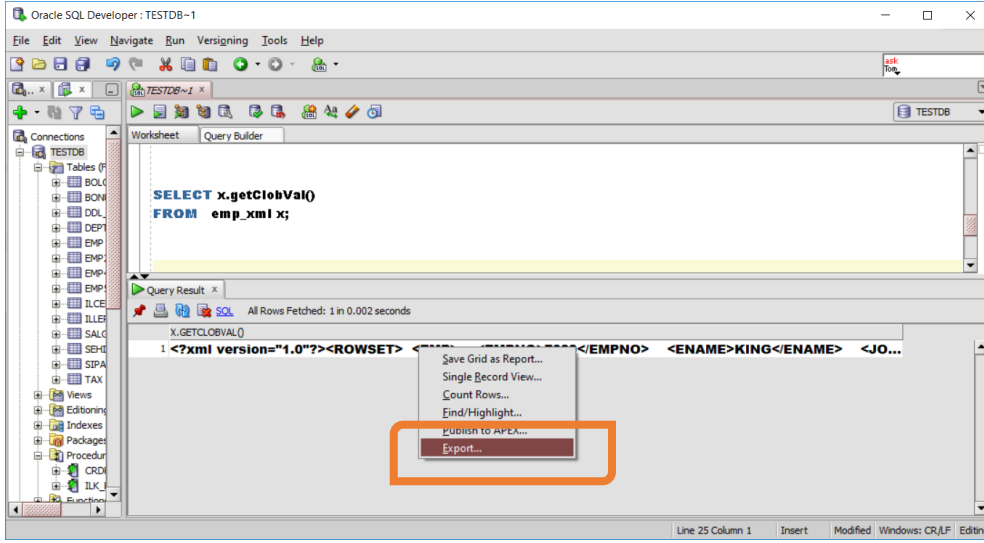
INSERT INTO EMP_XML VALUES (D_xmltype);

COMMIT;

END;

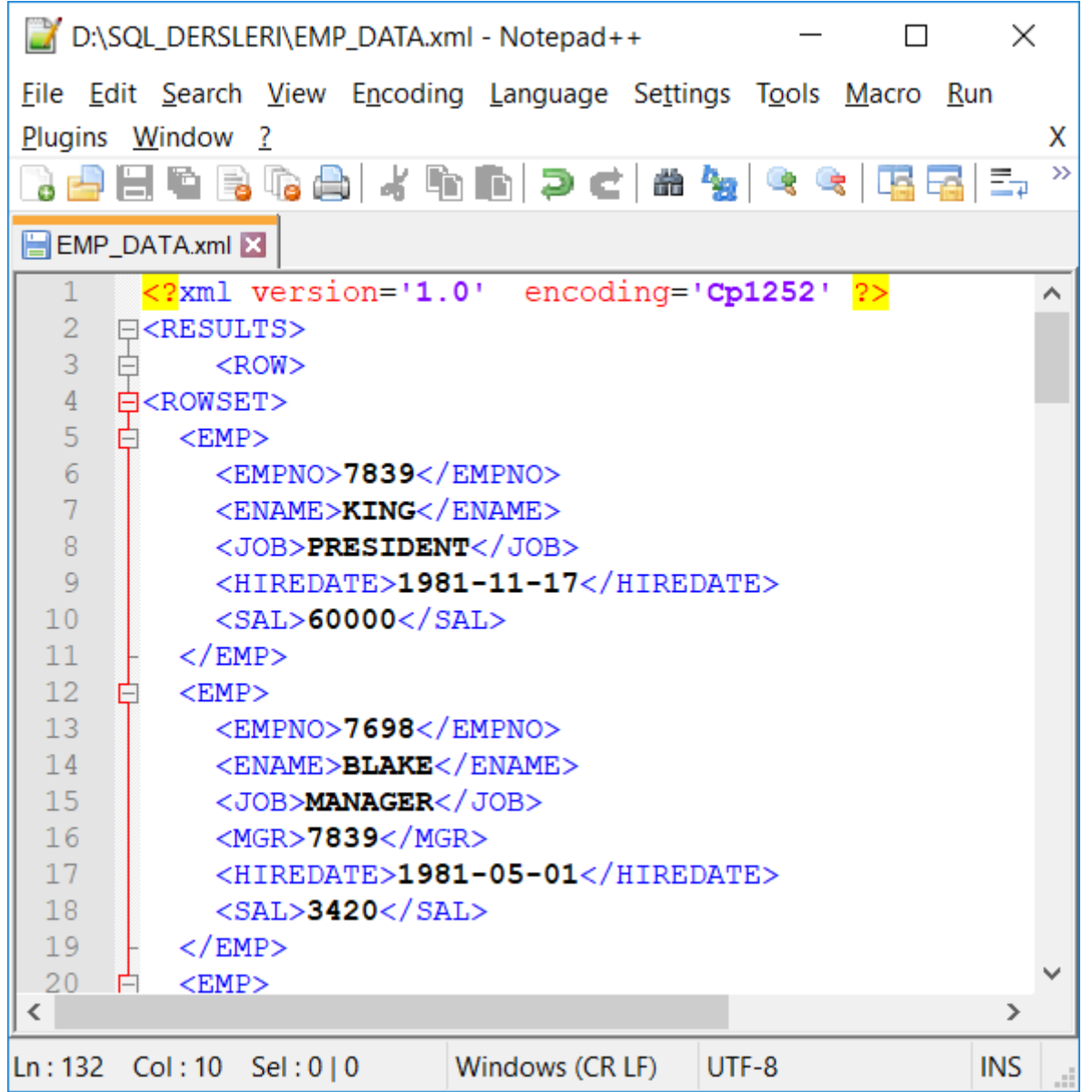
Şimdi Select Edelim ve çıktıyı File Sisteme xml formatında saklayalım.

```
SELECT x.getClobVal() FROM emp_xml x;
```



Oluşturduğumuz XML dosyayı Notepad++ ile açarsak şu şekilde görünecektir. Stil uygulamadığımız için Browserda açtığımızda düzgün görülmeyecektir.

Bizim burada amacımız zaten Tablo verilerinin kolayca XML formatına nasıl çevrildiğini görmek idi.



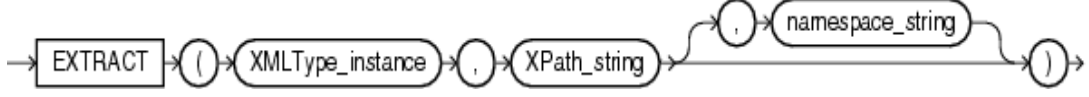
```
<?xml version='1.0' encoding='Cp1252' ?>
<RESULTS>
  <ROW>
    <ROWSET>
      <EMP>
        <EMPNO>7839</EMPNO>
        <ENAME>KING</ENAME>
        <JOB>PRESIDENT</JOB>
        <HIREDATE>1981-11-17</HIREDATE>
        <SAL>60000</SAL>
      </EMP>
      <EMP>
        <EMPNO>7698</EMPNO>
        <ENAME>BLAKE</ENAME>
        <JOB>MANAGER</JOB>
        <MGR>7839</MGR>
        <HIREDATE>1981-05-01</HIREDATE>
        <SAL>3420</SAL>
      </EMP>
      <EMP>
```

Ln : 132 Col : 10 Sel : 0 | 0 Windows (CR LF) UTF-8 INS

Örnek Uygulama (b): XML Formatındaki Bir Veriyi Standart Veri Formatına Çevirelim

EXTRACT

EXTRACT fonksiyonu, bir XMLType veri tipini ve bir XPath ifadesini argüman olarak alır, alt fonksiyonlarıyla ilgili düğümdeki (tag) değerini döndürür.



Şimdi de XML tablodan veriyi etiketleri kaldırarak okuyalım

```
SELECT extract(value(d), '//EMPNO/text()').getNumberVal() AS empno,
```

```
extract(value(d), '//ENAME/text()').getStringVal() AS ename,
```

```
extract(value(d), '//JOB/text()').getStringVal() AS job,
```

```
extract(value(d), '//MGR/text()').getNumberVal() AS mgr,
```

```
extract(value(d), '//HIREDATE/text()').getStringVal() AS hiredate,
```

```
extract(value(d), '//SAL/text()').getNumberVal() AS sal
```

```
FROM EMP_XML x,
```

```
table(xmlsequence(extract(value(x), '/ROWSET/EMP'))) d;
```

```
/* Pipeline */
```

Oracle SQL Developer: TESTDB~1

Worksheet: TESTDB~1

```
SELECT extract(value(d), '//EMPNO/text()').getNumberVal() AS empno,
extract(value(d), '//ENAME/text()').getStringVal() AS ename,
extract(value(d), '//JOB/text()').getStringVal() AS job,
extract(value(d), '//MGR/text()').getNumberVal() AS mgr,
extract(value(d), '//HIREDATE/text()').getStringVal() AS hiredate,
extract(value(d), '//SAL/text()').getNumberVal() AS sal
FROM EMP_XML x,
table(xmlsequence(extract(value(x), '/ROWSET/EMP'))) d;
```

Script Output: All Rows Fetched: 16 in 0.027 seconds

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL
1	7839	KING	PRESIDENT	(null)	1981-11-17	60000
2	7698	BLAKE	MANAGER	7839	1981-05-01	3420
3	7782	CLARK	MANAGER	7839	1981-06-09	2940
4	7566	JONES	MANAGER	7839	1981-04-02	3570
5	7788	SCOTT	ANALYST	7566	1987-03-20	3600
6	7902	FORD	ANALYST	7566	1981-12-03	3600
7	7369	SMITH	CLERK	7902	1980-12-17	960
8	7499	ALLEN	SALESMAN	7698	1981-02-20	1920
9	7521	WARD	SALESMAN	7698	1981-02-22	1500
10	7654	MARTIN	SALESMAN	7698	1981-09-28	1500
11	7844	TURNER	SALESMAN	7698	1981-09-08	1800

Table MEHMET.null@TESTDB | Line 1 Column 2 | Insert | Modified | Windows: CR/LF | Editing

Artık XML tablodan verileri düzgün formatta elde ettik. Bu şekilde elde ettiğimiz verileri dilediğimiz gibi kullanabiliriz.

Bir sonraki uygulamamızda bir XML dosyadan verileri okuyup Oracle Table formatta (standart Format) bir tabloya INSERT edeceğiz.

XML DOSYA OLUŞTURMA 2. YÖNTEM (xml Tablodan Okuyup dosya.xml oluşturmak)

```
DECLARE
  D_xmltype XMLTYPE;
  txtdosya2 UTL_FILE.FILE_TYPE;
  AA VARCHAR2(4000);
BEGIN
  Txtdosya2 := UTL_FILE.FOPEN('TEXT_DIR','dosya3.XML','W');
  FOR I IN (SELECT sys_xmllagg(
    xmlelement(
      "EMP",
      xmlforest(e.empno, e.ename, e.job, e.mgr, e.hiredate, e.sal, e.comm)
    )
  ) XX
  FROM emp e ) LOOP
  BEGIN
  INSERT INTO EMP_XML2 VALUES (I.XX);
  COMMIT;
  END;
  END LOOP;
  SELECT x.getClobVal() INTO AA FROM emp_xml2 x;
  UTL_FILE.PUT_LINE(txtdosya2,AA);
  UTL_FILE.FCLOSE(txtdosya2);
END;
```

XML Dosyaları Okumak İin Fonksiyonlar

Bu kısımda 3 fonksiyon hakkında kısa bilgi vereceėiz.

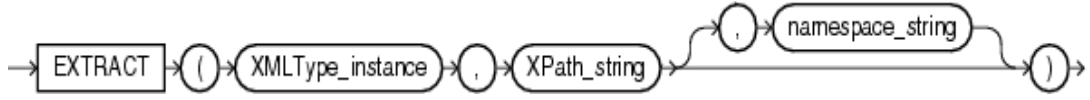
EXTRACT, EXTRACTVALUE, XMLSEQUENCE

Bu fonksiyonların her biri bařlı bařına bir konudur. Burada kısa bir bilgi vermemizin amacı az sonra yapacağımız Merkez Bankası gnlk kurlarını sistemimize alma uygulaması iindir.

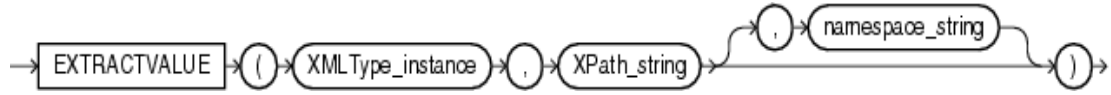
TCMB gnlk kurları XML formatında yayınlamaktadır.

EXTRACT

EXTRACT işlevi, bir XMLType veri tipini ve bir XPath ifadesini argüman olarak alır, alt fonksiyonlarıyla ilgili düğümdeki (tag) değerini döndürür.



EXTRACTVALUE



**** Grafikler Oracle.com adresinden alınmıştır.**

EXTRACTVALUE işlevi, bir XMLType veri tipini ve bir XPath ifadesini argüman olarak alır ve ilgili düğümdeki (tag) değeri döndürür.

XMLSEQUENCE

XMLSEQUENCE fonksiyonu, XMLTYPE sorgularından dönen değerleri birden çok satıra bölmek için kullanılır.

WEB SERVİS UYGULAMASI (TCMB Günlük Kur)

(XML Dosyadan Veri Okumak)

Bu Çalışmada Türkiye Cumhuriyet Merkez Bankasından Günlük Kurları çekip veritabanımızdaki bir tabloya saklayacağız.

```
CREATE TABLE GUNLUKKUR
(
  TARİH   DATE,
  KOD     VARCHAR2(4 BYTE),
  ALIS    NUMBER(12,4),
  SATIS   NUMBER(12,4),
  EFALIS  NUMBER(12,4),
  EFSATIS NUMBER(12,4)
);

CREATE OR REPLACE PROCEDURE WSL_GET_KURLAR
IS
  TCMB_url HTTPURITYPE;
  TCMB_xml XMLTYPE;
BEGIN
  TCMB_url := httpuritype.createuri('http://www.tcmb.gov.tr/kurlar/today.xml');
  TCMB_xml := TCMB_url.getxml ();
  FOR OKU
  IN (SELECT EXTRACTVALUE (VALUE (x), '/Tarih_Date/@Tarih') Tarih,
        EXTRACTVALUE (VALUE (p), '/Currency/@CurrencyCode') Kod,
        EXTRACTVALUE (VALUE (p), '/Currency/ForexBuying') Alis,
        EXTRACTVALUE (VALUE (p), '/Currency/ForexSelling') Satis,
        EXTRACTVALUE (VALUE (p), '/Currency/BanknoteBuying') Efalis,
        EXTRACTVALUE (VALUE (p), '/Currency/BanknoteSelling') Efsatis
      FROM TABLE (XMLSEQUENCE (EXTRACT (TCMB_xml, '/Tarih_Date')) x,
        TABLE (XMLSEQUENCE (EXTRACT (VALUE (x), '/Tarih_Date/Currency')) p
      WHERE EXTRACTVALUE (VALUE (p), '/Currency/@CurrencyCode') IN
        ('USD', 'EUR')
      )
  )
  LOOP
    INSERT INTO KURLAR (TARİH, KOD, ALIS, SATIS, EFALIS, EFSATIS)
    SELECT TRUNC(SYSDATE), TO_CHAR (oku.kod),
      NVL (REPLACE (oku.Alis, ',', '.'), 0), NVL (REPLACE (oku.Satis, ',', '.'), 0),
      NVL (REPLACE (oku.Efalis, ',', '.'), 0), NVL (REPLACE (oku.Efsatis, ',', '.'), 0)
    FROM DUAL;
    COMMIT;
  END LOOP;
END;
```

Yukarıdaki komutları çalıştırıp GUNLUKKUR Procedure'yi oluşturduk.

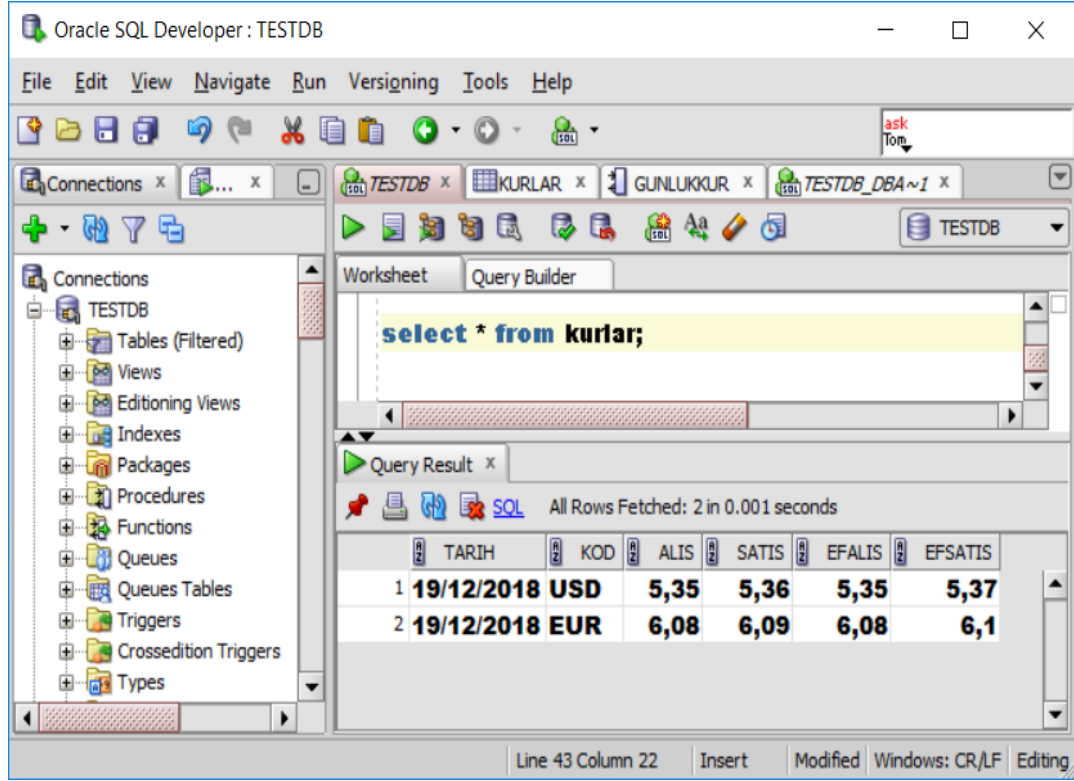
Şimdi GUNLUKKUR procedureyi çalıştırıp bugüne ait kurları sistemimize çekelim.

SQL> EXECUTE GUNLUKKUR;

(Çalıştırdığınızda Ora-24247 hatası alırsanız ACL ayarları yapılmamış demektir. Kitabın sonundaki Erişim Kontrol Listesi (ACL) (Ora-24247) başlığı altındaki işlemleri yapınız.

Günlük kuru sistemimize kaydeden GUNLUKKUR procedureyi çalıştırdık. Şimdi listeleyelim.

SQL> SELECT * FROM KURLAR;



Bu Procedure'yi Bir JOB'a bağlarsak günlük kurları her gün otomatik olarak çekip sistemimize yazabiliriz.

Görev Zamanlama Tanımlamaları (JOB)

Veritabanında belirli aralıklarla çalışmasını istediğimiz programlar olabilir. Bu tür işlemlere Zamanlanmış Görevler yani JOB'lar denir.

JOB konusu PL/SQL konusu dışında olduğu için burada detaylara girmeyeceğiz.

TCMB Kurlarını çeken Procedureyi her gün belirli bir saatte alsın istiyorsak şu şekilde bir tanımlama yapmalıyız.

Aşağıdaki JOB, GUNLUKKUR proceduresini her gün saat 11:59'da çalışmasını sağlar. (Günlük kurlar her gün saat 11'de açılır)...

```
BEGIN
  Sys.Dbms_Scheduler.Create_Job (
    job_name => 'TCMBKUR4',
    job_type => 'STORED_PROCEDURE',
    job_action => '"MEHMET"."GUNLUKKUR"',
    Number_Of_Arguments => 0,
    start_date  => TO_TIMESTAMP_TZ('2019-01-05 11:59:00 Europe/Istanbul','YYYY-MM-DD
HH24.MI.SS TZR'),
    repeat_interval => 'FREQ=DAILY;BYTIME=115900',
    End_Date => Null,
    job_class => '"SYS"."DEFAULT_JOB_CLASS"',
    enabled => FALSE,
    auto_drop => FALSE,
    comments => '',
    credential_name => NULL,
    destination_name => NULL);
END;
```

EXCEL TABLolarI İLE ÇALIŞMAK

Bu çalışmamızda PL/SQL ile Veritabanındaki verileri Excel dosyasına atmayı ve Excel dosyalarından verileri Veritabanına çekmeyi işleyeceğiz.

PL/SQL ile Excel'e veri atma – veri çekme işlemlerini iki farklı yolla yapacağız.

Birincisi basit bir PL/SQL Procedure yazarak, ikincisi ise daha kullanışlı bir metotla işlemleri gerçekleştireceğiz.

Daha önce de belirtildiği üzere Veritabanı ile Filesystem (İşletim sistemi) doğrudan haberleşemez. Veritabanı ve File system arasında dosya okuma – yazma işlemleri için öncelikle veritabanı içerisinde ilgili FileSystem'i işaret eden bir tanımlama yapmamız gerekmektedir.

DIRECTORY Create Etmek ve Kullanıcı Yetkilendirmek

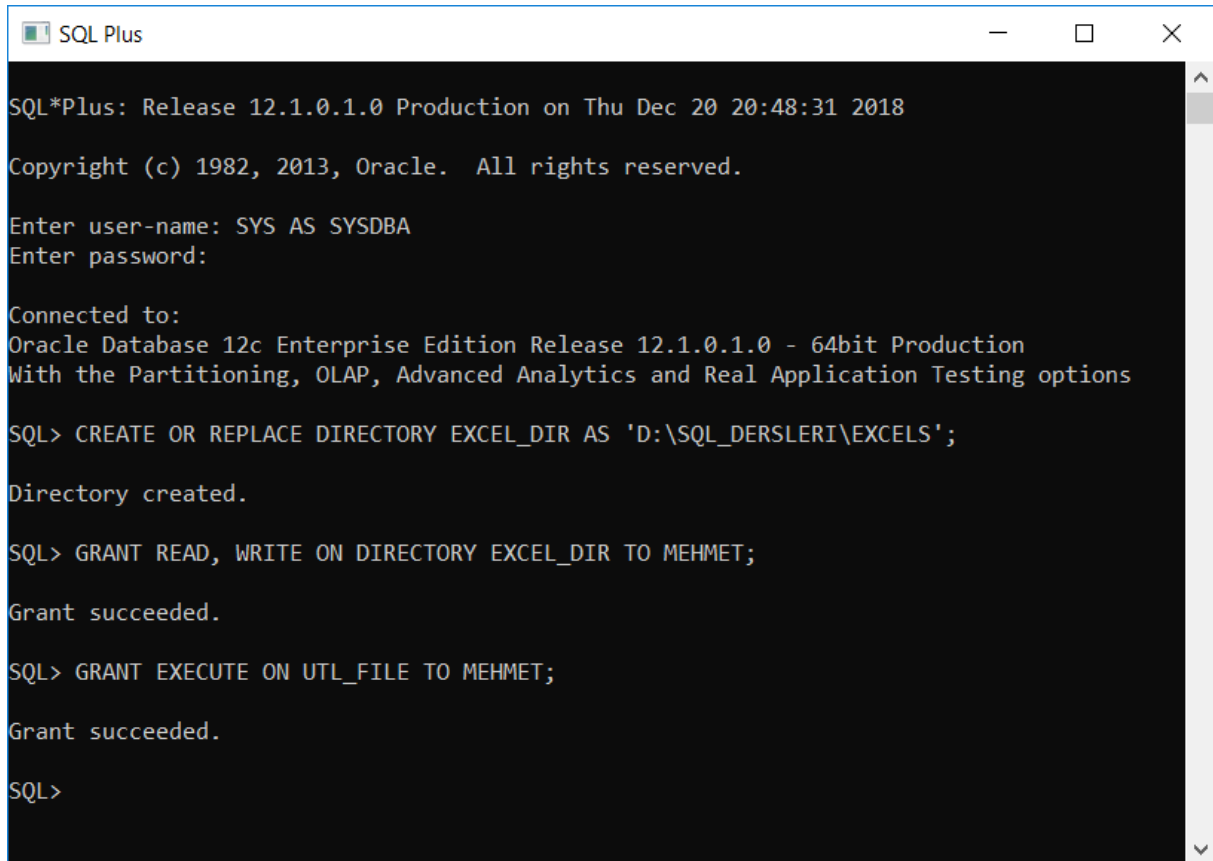
Veritabanına DBA yetkili bir kullanıcı ile (SYS veya SYSTEM) ile Connect olalım.

SQLPLUS, SQL Developer, TOAD veya başka bir araçla

SQL> CREATE OR REPLACE DIRECTORY EXCEL_DIR 'D:\SQL_DERSLERI\EXCELS';

SQL> GRANT READ, WRITE, ON DIRECTORY EXCEL_DIR TO MEHMET;

SQL> GRANT EXECUTE ON UTL_FILE TO MEHMET;

A screenshot of a SQL Plus command window. The window title is "SQL Plus". The text inside shows the SQL*Plus release information, copyright notice, and connection details for user SYS AS SYSDBA. The user has entered the password and is now connected to an Oracle Database 12c Enterprise Edition. The user then enters three SQL commands: 1. "CREATE OR REPLACE DIRECTORY EXCEL_DIR AS 'D:\SQL_DERSLERI\EXCELS';" which results in "Directory created.". 2. "GRANT READ, WRITE ON DIRECTORY EXCEL_DIR TO MEHMET;" which results in "Grant succeeded.". 3. "GRANT EXECUTE ON UTL_FILE TO MEHMET;" which results in "Grant succeeded.". The prompt "SQL>" is visible at the bottom.

```
SQL*Plus: Release 12.1.0.1.0 Production on Thu Dec 20 20:48:31 2018

Copyright (c) 1982, 2013, Oracle. All rights reserved.

Enter user-name: SYS AS SYSDBA
Enter password:

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application Testing options

SQL> CREATE OR REPLACE DIRECTORY EXCEL_DIR AS 'D:\SQL_DERSLERI\EXCELS';

Directory created.

SQL> GRANT READ, WRITE ON DIRECTORY EXCEL_DIR TO MEHMET;

Grant succeeded.

SQL> GRANT EXECUTE ON UTL_FILE TO MEHMET;

Grant succeeded.

SQL>
```

Bu komut veritabanına işletim sistemi tarafında bir folder'ı tanıtır fakat ilgili kolasörü create edemez. O yüzden File System tarafından ilgili klasörü kendimiz create etmeliyiz.

(File system – Windows tarafında belirtilen adreste EXCELS isimli bir klasör create etmeliyiz. Aksi halde oluşturacağımız Procedure çalışma esnasında hata verir)

Verileri Excel'e Atacak Procedure Yazalım (1. Yöntem)

SQL Developer veya SQLPLUS'da yazılabilir. Veya Notepad gibi bir editörde yazıp SQL Developer veya Sqlplus'da bu scripti çalıştırabiliriz. Biz SQL Developer kullanacağız.

SQL developer açıp (Bizim örneğimizde) MEHMET kullanıcısı ile veritabanına bağlanıyoruz.

```
CREATE OR REPLACE PROCEDURE EMP_TO_CSV (p_seperator IN VARCHAR2) AS
/* P_SEPERATOR Parametresi veriler arasına konulacak ayırteci deęerdir. noktalı virgöl (;), Slash (/),
yıldız .. gibi işaretleler olabilir. Bizim örneğimizde ; veya / ile sınırlandıracağız
*/
```

```
CURSOR EMP_data IS
SELECT E.empno, E.ename, E.job, E.mgr,
       TO_CHAR(E.hiredate,'DD-MON-YYYY') AS hiredate,
       E.sal, E.comm, E.deptno, d.dname
FROM   emp E, dept d
where e.deptno = d.deptno
ORDER BY e.ename;

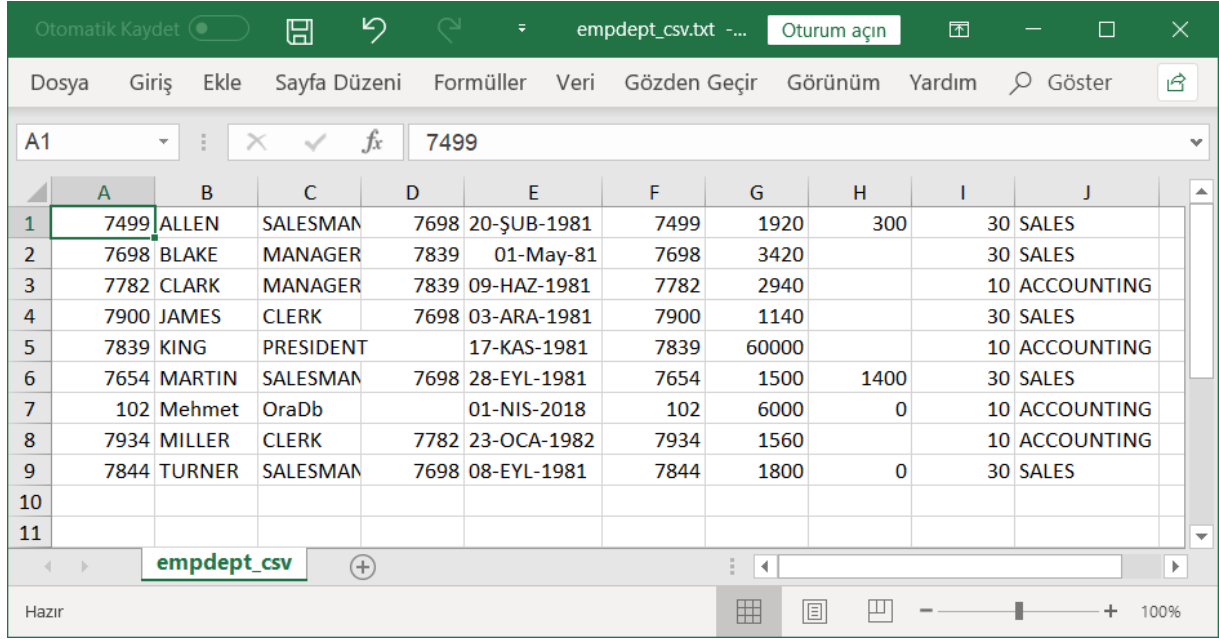
t_file UTL_FILE.FILE_TYPE;
hatali_seperator EXCEPTION;
BEGIN
IF P_SEPERATOR NOT IN (';', '/') THEN
  RAISE HATALI_SEPERATOR;
END IF;
t_file := UTL_FILE.FOPEN(location => 'EXCEL_DIR',
                        filename => 'empdept_csv.txt',
                        open_mode => 'w',
                        max_linesize => 32767);
FOR CURSOR_rec IN EMP_data LOOP
  UTL_FILE.PUT_LINE(t_file,
                    cursor_rec.empno || P_SEPERATOR ||
                    cursor_rec.ename || P_SEPERATOR ||
                    cursor_rec.job || P_SEPERATOR ||
                    cursor_rec.mgr || P_SEPERATOR ||
                    cursor_rec.hiredate || P_SEPERATOR ||
                    cursor_rec.empno || P_SEPERATOR ||
                    cursor_rec.sal || P_SEPERATOR ||
                    cursor_rec.comm || P_SEPERATOR ||
                    cursor_rec.deptno || P_SEPERATOR ||
                    cursor_rec.dname || P_SEPERATOR);
END LOOP;
CLOSE OKU;
UTL_FILE.FCLOSE(t_file);
EXCEPTION
WHEN HATALI_SEPERATOR THEN
  UTL_FILE.FCLOSE(t_file);
  DBMS_OUTPUT.PUT_LINE('Hatalı Seperator Girdiniz... Seperaor ; veya / Olabilir!');
WHEN OTHERS THEN
  UTL_FILE.FCLOSE(t_file);
  RAISE;
END;
```

Bu Procedureyi çalıştıralım.

SQL> EXECUTE EMP_TO_CSV('/');

Bu şekilde çalıştırdığımızda, bir önceki adımda tanımladığımız DIRECTORY (EXCEL_DIR, bizim örneğimizde **D:\SQL_DESLER\EXCELS**) altına empdept_csv.txt ismiyle bir text dosya oluşturur.

Bu dosyayı Excel ile açarsak ve ilgili seperatoru gösterirsek, sorgudan gelen bilgileri Excel'e atmış olduğunu görürüz.



	A	B	C	D	E	F	G	H	I	J
1	7499	ALLEN	SALESMAN	7698	20-ŞUB-1981	7499	1920	300	30	SALES
2	7698	BLAKE	MANAGER	7839	01-May-81	7698	3420		30	SALES
3	7782	CLARK	MANAGER	7839	09-HAZ-1981	7782	2940		10	ACCOUNTING
4	7900	JAMES	CLERK	7698	03-ARA-1981	7900	1140		30	SALES
5	7839	KING	PRESIDENT		17-KAS-1981	7839	60000		10	ACCOUNTING
6	7654	MARTIN	SALESMAN	7698	28-EYL-1981	7654	1500	1400	30	SALES
7	102	Mehmet	OraDb		01-NIS-2018	102	6000	0	10	ACCOUNTING
8	7934	MILLER	CLERK	7782	23-OCA-1982	7934	1560		10	ACCOUNTING
9	7844	TURNER	SALESMAN	7698	08-EYL-1981	7844	1800	0	30	SALES
10										
11										

Verileri Excel'e Atalım (2. Yöntem)

Bu çalışmamızda <https://technology.amis.nl/2011/02/19/create-an-excel-file-with-plsql/> adresinde açık kaynak (open source) olarak yayınlanan ve Anton Scheffer tarafından free olarak yayınlanan PL/SQL paketlerini kullanacağız. Paketin içerisinde de belirtildiği üzere kullanacağımız bu paketin kullanılması ve lisanslanması ile ilgili herhangi bir kısıtlama konulmamıştır. Bu PL/SQL paketini free olarak yayınlayan Mr. Anton Scheffer'e teşekkür ederim.

Paket üzerinde çalışmalarımıza uygun olarak pek çok değişiklik yaptım. Son şeklini dersime ekleyeceğim. Ayrıca paketin orijinalini yukarıda vermiş olduğum adresten indirebilirsiniz.

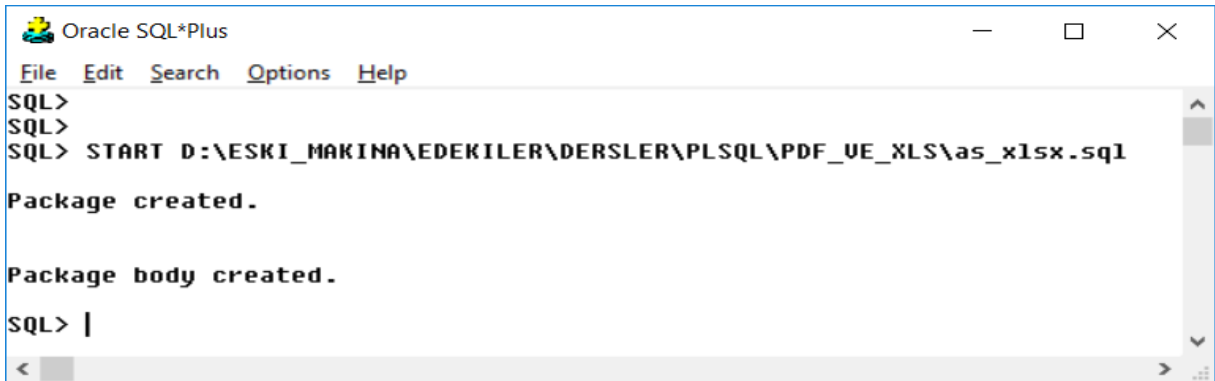
Çalışmaya başlamadan önce bu kitaptaki “[DIRECTORY Create Etmek ve Kullanıcı Yetkilendirmek](#)” başlığı altındaki işlemleri tekrar gözden geçiriniz.

**

İlk olarak yukarıdaki adresten ya da bu derse eklediğim AS_XLSX.sql dosyasını şemamızda çalıştırıyoruz. Bu script şemamızda AS_XLSX isimli bir paket oluşturur.

```
SQL> [FULL-PATH] \ as_xlsx.sql
```

(Siz paketi hangi klasöre indirdiyseniz FULL-PATH kısmına o yolu yazacaksınız...



```
Oracle SQL*Plus
File Edit Search Options Help
SQL>
SQL>
SQL> START D:\ESKI_MAKINA\EDEKILER\DERSLER\PLSQL\PDF_UE_XLS\as_xlsx.sql

Package created.

Package body created.

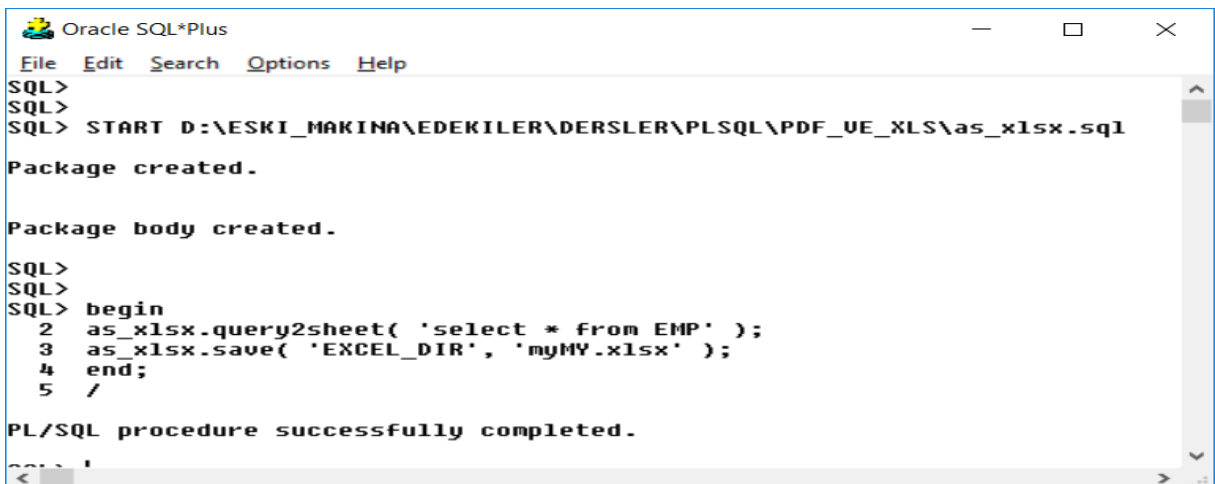
SQL> |
```

Ben paketi SQLPLUS'da çalıştırdım. Siz SQL Developer'da da çalıştırabilirsiniz.

```
SQL> [FULL-PATH] \ as_xlsx.sql
```

Paket hatasız şemamızda oluşturuldu.

Şimdi test edelim.



```
Oracle SQL*Plus
File Edit Search Options Help
SQL>
SQL>
SQL> START D:\ESKI_MAKINA\EDEKILER\DERSLER\PLSQL\PDF_UE_XLS\as_xlsx.sql

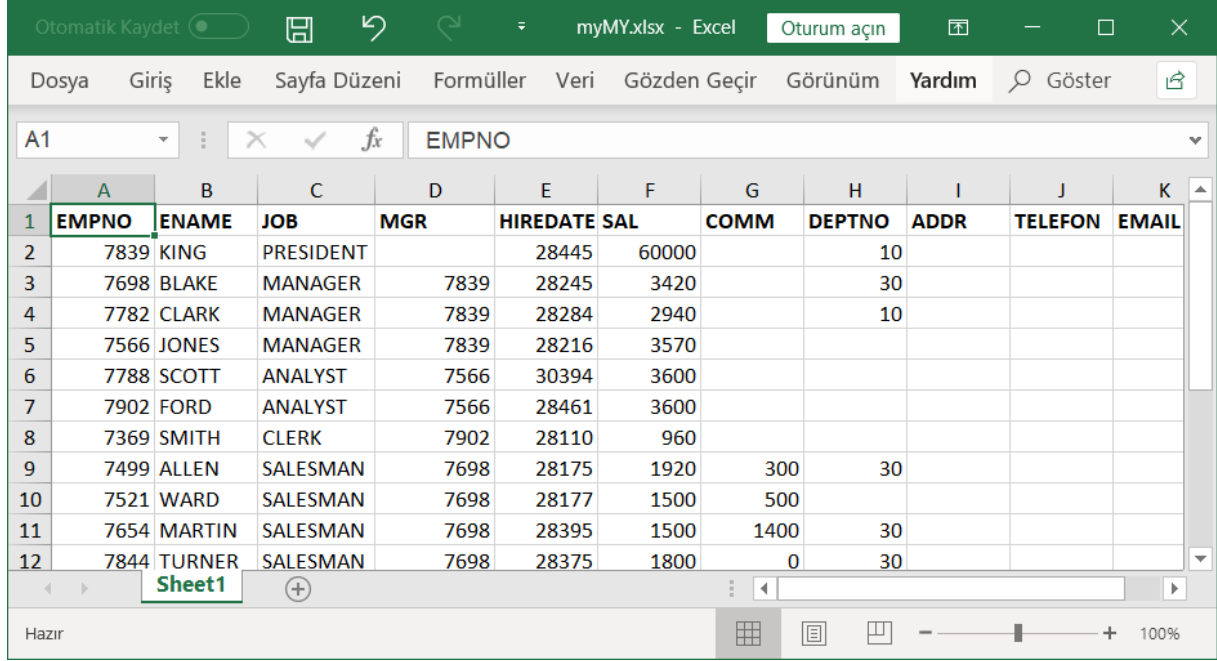
Package created.

Package body created.

SQL>
SQL>
SQL> begin
2  as_xlsx.query2sheet( 'select * from EMP' );
3  as_xlsx.save( 'EXCEL_DIR', 'myMY.xlsx' );
4  end;
5  /

PL/SQL procedure successfully completed.
```

EXCEL_DIR Directory olarak hangi [PATH'i tanımladıysanız](#) o klasörde **myMy.xlsx** çıktısı oluşmuştur. İlgili klasörde myMY.xlsx dosyasını açıyoruz.



EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO	ADDR	TELEFON	EMAIL
7839	KING	PRESIDENT		28445	60000		10			
7698	BLAKE	MANAGER	7839	28245	3420		30			
7782	CLARK	MANAGER	7839	28284	2940		10			
7566	JONES	MANAGER	7839	28216	3570					
7788	SCOTT	ANALYST	7566	30394	3600					
7902	FORD	ANALYST	7566	28461	3600					
7369	SMITH	CLERK	7902	28110	960					
7499	ALLEN	SALESMAN	7698	28175	1920	300	30			
7521	WARD	SALESMAN	7698	28177	1500	500				
7654	MARTIN	SALESMAN	7698	28395	1500	1400	30			
7844	TURNER	SALESMAN	7698	28375	1800	0	30			

Örnek2

```
begin
as_xlsx.new_sheet('tablespace1');
as_xlsx.query2sheet( 'select 1 r from dual', p_sheet => 1 );
as_xlsx.new_sheet('tablespace2');
as_xlsx.query2sheet( 'select 2 r from dual', p_sheet => 2 );
as_xlsx.new_sheet( 'tablespace3');
as_xlsx.query2sheet( 'select 3 r from dual', true, 'EXCEL_DIR', 'my5.xlsx', 3 );
end;
```

Örnek3

```
BEGIN
as_xlsx.new_sheet('EMP');
as_xlsx.query2sheet( 'select ENAME ENAME, EMPNO EMPNO from EMP', p_sheet =>1);
as_xlsx.new_sheet('DEPT');
as_xlsx.query2sheet( 'select DEPTNO BOLUMNO, DNAME BOLUMISMI from DEPT', p_sheet =>2);
as_xlsx.save('EXCEL_DIR','deneme.xlsx');
END;
```

as_XLSX paketini kullanarak verilerimizi Excel'e bu şekilde atabilmekteyiz.

Erişim Kontrol Listesi (ACL) (Ora-24247 Hatası)

Eğer önceden Erişim Kontrol Listesi (ACL Access Control List) oluşturulmadıysa, bir web hizmetini çağırmak veya bir smtp hizmetine bağlanmaya çalışıldığında aşağıdaki hatayı alırsanız:

ORA-24247: network access denied by access control list (ACL)

Hatanın Sebebi;

“Hedef sunucuya hiçbir erişim kontrol listesi (ACL) atanmamış veya hedef ana bilgisayara erişmek için gerekli ayrıcalık, erişim kontrol listesinde kullanıcıya verilmemiş.”

Bu hata, Oracle 11g'den itibaren yaygın bir şekilde alınır.

utl_tcp, utl_smtp, utl_mail, utl_http ve utl_inaddr gibi ağlar üzerinden ağ kaynaklarının kullanılması, veritabanına ciddi bir güvenlik tehdidine maruz bırakmıştır. Çünkü kullanıcıya bu paketlerin kullanma izni verilmişse bu paketlerle herhangi bir bilgisayara bağlanmak için başka bir sınırlama daha önceki versiyonlarda yoktu.

Oracle 11g ve sonraki versiyonlarda, erişim kontrol listelerini (ACL) kullanarak ağ hizmetlerine sınırlama getirdi.

Bu yeni özellik, Veritabanı yöneticisi (DBA, sys, system) hangi kullanıcının hangi bilgisayara bağlanabileceğini kontrol altına almasını sağladı.

CÖZÜM

- 1) Create an ACL (Eğer yoksa)
- 2) Add privileges (Network kaynaklarının kullanım haklarını ilgili kullanıcıya ata)
- 3) Assign the ACL(Bağlantı yapılacak adrese ACL'yi ata)

**

1) ACL Var mı?

```
SELECT * FROM dba_network_ACLS;
```

Eğer bağlanılmaya çalışılan adres HOST alanında yoksa, yeni bir ACL oluştur.

```
begin
dbms_network_acl_admin.create_acl (
acl => 'http_permissions.xml', -- veya Herhangi bir isim olabilir
description => 'HTTP Access',
principal => 'MEHMET', -- Networke bağlanacak Kullanıcı
is_grant => TRUE, privilege => 'connect',
start_date => null, end_date => null
);
end;

commit;
```

Bu işlem yeni bir ACL oluşturacak ve MEHMET kullanıcısına CONNECT yetkisini verecek.

2) ACL mevcutsa, kullanıcıya uygun ayrıcalık verildiğini doğrulamak için aşağıdaki sorguyu çalıştırın.

```
SELECT * FROM dba_network_acl_privileges
where principal='MEHMET';
```

UTL_TCP, UTL_HTTP, UTL_SMTP ve UTL_MAIL paketlerini kullanabilmesi için MEHMET kullanıcıasına CONNECT yetkisi verilecek.

```
begin
    DBMS_NETWORK_ACL_ADMIN.ADD_PRIVILEGE(acl => 'http_permissions.xml',
    principal => 'MEHMET',
    is_grant => true,
    privilege => 'connect');
end;
commit;
```

Bağlanılacak adresin IP'sinden HOST isminin çözümlenmesi gerekiyorsa RESOLVE yetkisinin de MEHMET kullanıcıasına verilmesi gerekir.

```
begin
    DBMS_NETWORK_ACL_ADMIN.ADD_PRIVILEGE(acl => 'http_permissions.xml',
    principal => 'MEHMET',
    is_grant => true,
    privilege => 'resolve');
end;
commit;
```

3) Son olarak hedef ACL hedef adrese atanması gerekir.

```
BEGIN
    dbms_network_acl_admin.assign_acl (
    acl => 'http_permissions.xml',
    host => 'NETWORK ADDRESS', /*Hedef Bilgisayar ismi veya IP adresi bizim örneğimizde -
    *.www.tcmb.gov.tr*/
    lower_port => 80,
    upper_port => 80
    );
END;
```

DİKKAT

Herhangi bir ana bilgisayara sadece bir ACL atanabileceğini unutmamak önemlidir. Bir hedefe yeni bir ACL atanırsa eski ACL atanmamış duruma gelir.

Fakat eski ACL silinmez. Bu karışıklığa neden olur.