UNIX

Lesson 19 : Database connectivity

Lesson Objectives

- At the end of the session you will be able to understand:
  - Connecting to a database
  - Executing stored procedure
  - Error handling

Profile Variable

- **Connecting to Database**

  A database is a collection of information that is organized so that it can easily be accessed, managed, and updated. In one view, databases can be classified according to types of content: bibliographic, full-text, numeric, and images.

- **Oracle Concepts - The SQL*Plus Command Line Interface**

  SQL*Plus also comes with a command line interface. This is the interface you use when working from the command line. To start SQL*Plus at the command line you must do the following:

  If connecting to a local database you must set the ORACLE_SID environment parameter to the name of your database. Here is an example of doing this in Windows, followed by an example of setting the ORACLE_SID in UNIX (as well as Oracle Home and path information so Unix can find the binaries

Profile Variable

Windows:

```
C:>Set ORACLE_SID=FLP_DB
```

UNIX/Linux:

```
export ORACLE_SID=FLP_DB
  export ORACLE_HOME=/oracle/product/10g/db_1
PATH=$PATH:$ORACLE_HOME/bin
```

2. Next, we start SQL*Plus with the sqlplus command. When starting SQL plus include the user name that you wish to connect to. Here are examples of the use of this command:

```
C:\>sqlplus scott
SQL*Plus: Release 10.1.0.2.0 - Production on Sun Feb 20 11:28:13 2005   Copyright (c)
1982, 2004, Oracle.  All rights reserved.
Enter password:   ***********
Connected to: Oracle Database 10g Enterprise Edition Release 10.1.0.2.0 - Production With
the Partitioning, OLAP and Data Mining options
SQL>
```

Sample Unix script

```
----ora_connect_eg.sh----
#!/bin/sh
user="scott"
pass="tiger"
var="$1"
sqlplus -S $user/$pass <<EOF
SELECT * FROM DEPT WHERE DEPTNO=$var;
exit;
EOF
```

Output

```
$ ksh  ora_connect_eg.sh 10

    DEPTNO DNAME          LOC
---------- -------------- -------------
        10 ACCOUNTING     NEW YORK
$
```

Calling oracle stored Procedure.

```
sqlplus -s scott@ftp_db/tiger<<END
execute test_proc(132,'raj');
commit;
exit;
END
```

Error handling using stored Procedure.

```
DECLARE
        Declaration section
BEGIN
        Exception section
EXCEPTION
 WHEN ex_name1 THEN
        -Error handling statements
 WHEN ex_name2 THEN
        -Error handling statements
 WHEN Others THEN
        -Error handling statements
END;
```

Example : error handling

```
---Procedure 1
PROCEDURE financial_review
IS
BEGIN
calc_profits (1996);
calc_expenses (1996);
DECLARE
v_str VARCHAR2(1);
BEGIN
v_str := 'abc';
END;
EXCEPTION
WHEN OTHERS
THEN
log_error;
END;

---Procedure 2
PROCEDURE calc_expenses
IS BEGIN
...
EXCEPTION
WHEN NO_DATA_FOUND
THEN
NULL;
END;
```

SUMMARY

- In this lesson, you have learnt:
  - Connecting data base using script
  - Calling stored procedures

Review Questions

- Question 1: How to form a connection to a database from script?