Cursors and Exceptions

Cursors

Practice 1

- Write a PL/SQL block that takes a department number from a user and increases the salary of all the employees belonging to the department by 10%. The block should display on the screen how many records are updated.

Practice 2

- Write a PL/SQL block to display the employee name, salary and their corresponding grades (by making use of the "salgrade" table) for the first five employees recorded in the "emp" table by making use of :
  - Simple for loop with "EXIT WHEN" condition
  - The "WHILE" condition
  - Cursor FOR LOOP

Practice 3

- Write a PL/SQL block that displays names and salaries of all CLERK's recorded in the "emp" table by making use of a cursor.

Practice 4

- Write a PL/SQL block that accepts a value of a job from user and displays the names, department numbers and salaries of the employees from the "emp" table having that job. The block makes use of a parameterized cursor. The user input is passed on to the cursor as a parameter.

Practice 5

- Write a PL/SQL block that updates the salaries of the employees as per the following rules.
  - If the job is CLERK and deptno is 10 then increase the salary by 20%
  - If the job is MANAGER and deptno is 20 then increase the salary by 5%
  - For all the other cases increase the salary by 10%

The block makes use of a cursor for performing the updates and ensures that the appropriate locks are taken on the data retrieved by the cursor, as the data is to be updated.

Practice 6

- Write a PL/SQL block that retrieves information from the "dept" and "emp" table

  Department Number :10 Department Name : XXXX

  _____

  EMPLOYEE NUMBER EMPLOYEE NAME SALARY JOB_ID

  Department Number :20 Department Name : XXXX

  _____

  EMPLOYEE NUMBER EMPLOYEE NAME SALARY JOB_ID

  .....and so on for all the departments recorded in the "dept" table.
  (Hint : In a loop, use a cursor to retrieve the deptno and the dname from the DEPT table, pass the deptno to retrieve the required column values from the "EMP" table, for those employees who work in that department.)
  Same using parameterized cursors

Exceptions

Practice 7

- Create a table named MESSAGES (err_message VARCHAR2(250))
- Write a PL/SQL block that accepts a salary value from a user and displays name of the employee having the salary value, on the screen.
  • If the salary entered returns more than one row, handle the exception with an appropriate Exception handler and insert into the MESSAGES table the message
  " More than one employee with salary of <input salary>"
  • If the salary entered does not return any rows, handle the exception with an appropriate Exception handler and insert into the MESSAGES table the message
  " No employee with salary of < input salary>"

• If the salary entered returns only one row, insert into the MESSAGES table the employee's name
• Handle any other exception with an appropriate Exception handler and insert into the MESSAGES table the message " Some other error occurred"

Practice 8

- Write a PL/SQL block that accepts all the column values for dept table as user inputs and inserts a record in the dept table. The block should give a name DUPLICATE_DEPT to the error for duplicate value of the primary key, deptno. (Use pragma EXCEPTION_INIT) The block should write a handler for handling the exception fired when a duplicated value is entered for deptno. The handler should have code for displaying an appropriate message on the screen when DUPLICATE_DEPT is fired.

Practice 9

- Write a PL/SQL BLOCK to check for more than one President (Job column) in the "emp" table. Create a user defined exception named DUPLICATE_PRESIDENT that should be raised when more than one president is found in the "emp" table. The block should handle the exception by displaying a message "MORE THAN ONE PRESIDENT" on the screen.

Practice 10

- Write a PL/SQL block that accepts the employee numbers of two employees as two user inputs empno1 and empno2. If empno1 exists in the "emp" table, then the block increases the salary of the employee by 10%. If empno2 exists in the "emp" table then the block increases his salary by 20%. The block should raise and handle the errors if empno1 or empno2 or both do not exist, by displaying the appropriate messages. Note that if empno1 does not exist but empno2 exists then the salary of empno2 must be increased. Also when both empno1 and empno2 do not exist then the error must be handled by the block.