

Lab 5. PL/SQL Cursors, Exceptions

Point Distribution:

Option A	
Question Number	Points
7a, 8a	6 pts x 2
6c, 8c, 8e	6 pts x 3
Other questions	3.5 pts x 20
Total	100 pts

Option B	
Question Number	Points
1, 2	50 pts x 2
Total	100 pts

Submission:

- You may choose to work with Option A or Option B for credit. Please indicate which part you wish to submit. If you submit both, only Option A will be graded. If you choose to submit Option B, please make sure that you know the answers to Option A.

Objectives:

- Practice implementing implicit PL/SQL Cursors
- Practice implementing explicit PL/SQL Cursors
- Practice implementing PL/SQL Cursor FOR Loops
- Practice handling exceptions
- Familiarization with standard Oracle exceptions

Option A

Explicit Cursors

- 1a** Using the code provided below, create the **print_employee_roster** procedure. The procedure uses an explicit cursor.

Worksheet	Query Builder
1	<code>CREATE OR REPLACE PROCEDURE print_employee_roster</code>
2	<code>AS</code>
3	<code>current_employeeid employee.employeeid%TYPE;</code>
4	<code>current_lastname employee.lastname%TYPE;</code>
5	<code>current_firstname employee.firstname%TYPE;</code>
6	
7	<code>CURSOR all_employees</code>
8	<code>IS</code>
9	<code>SELECT employeeid, lastname, firstname FROM employee;</code>
10	
11	<code>BEGIN</code>
12	<code>OPEN all_employees;</code>
13	
14	<code>FETCH all_employees INTO current_employeeid, current_lastname, current_firstname;</code>
15	
16	<code>WHILE all_employees%FOUND LOOP</code>
17	
18	<code>DBMS_OUTPUT.PUT (RPAD(current_employeeid, 15, ' '));</code>
19	<code>DBMS_OUTPUT.PUT (RPAD(current_lastname, 30, ' '));</code>
20	<code>DBMS_OUTPUT.PUT_LINE (current_firstname);</code>
21	<code>FETCH all_employees INTO current_employeeid, current_lastname, current_firstname;</code>
22	<code>END LOOP;</code>
23	
24	<code>CLOSE all_employees;</code>
25	<code>END print_employee_roster;</code>

Note: the procedure uses 3 local variables of %TYPE to hold the “fetched” values.

- 1b** Execute the **print_employee_roster** procedure. Include the results in your submission.

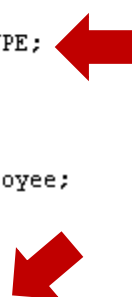
- 2a** Improve the **print_employee_roster** procedure by using a single local variable of %ROWTYPE (instead of the 3 local variables it currently uses) as shown below.

Worksheet Query Builder

```

1  CREATE OR REPLACE PROCEDURE print_employee_roster
2  AS
3
4  CURSOR all_employees
5  IS
6      SELECT employeeid, lastname, firstname FROM employee;
7
8      current_employee all_employees%ROWTYPE;
9  BEGIN
10     OPEN all_employees;
11
12     FETCH all_employees INTO current_employee;
13
14     WHILE all_employees%FOUND LOOP
15
16         DBMS_OUTPUT.PUT (RPAD(current_employee.employeeid, 15, ' '));
17         DBMS_OUTPUT.PUT (RPAD(current_employee.lastname, 30, ' '));
18         DBMS_OUTPUT.PUT_LINE (current_employee.firstname);
19         FETCH all_employees INTO current_employee;
20     END LOOP;
21
22     CLOSE all_employees;
23 END print_employee_roster;
24

```




- 2b** Execute the **print_employee_roster** procedure. Include the results in your submission.
- 2c** What is the database object that the **current_employee** variable is based upon?

- 3a** Modify the cursor definition to include concatenation, formatting, and an alias as shown below.

Worksheet Query Builder

```
1 CREATE OR REPLACE PROCEDURE print_employee_roster
2 AS
3
4 CURSOR all_employees
5 IS
6     SELECT employeeid,
7            lastname || ', ' || firstname as name
8     FROM employee;
9
10 current_employee all_employees%ROWTYPE;
11 BEGIN
12     OPEN all_employees;
13
14     FETCH all_employees INTO current_employee;
15
16     WHILE all_employees%FOUND LOOP
17
18         DBMS_OUTPUT.PUT (RPAD(current_employee.employeeid, 15, ' '));
19         DBMS_OUTPUT.PUT_LINE (current_employee.name);
20         FETCH all_employees INTO current_employee;
21     END LOOP;
22
23     CLOSE all_employees;
24 END print_employee_roster;
```



- 3b** Execute the **print_employee_roster** procedure. Include the results in your submission.

Cursor FOR Loops


4a Modify the **print_employee_roster** procedure to use a cursor FOR loop as shown below.

Worksheet Query Builder

```

1 CREATE OR REPLACE PROCEDURE print_employee_roster
2 AS
3
4     CURSOR all_employees
5     IS
6         SELECT employeeid,
7                lastname || ', ' || firstname as name
8         FROM employee;
9
10    BEGIN
11
12        FOR current_employee IN all_employees LOOP
13
14            DBMS_OUTPUT.PUT (RPAD(current_employee.employeeid, 15, ' '));
15            DBMS_OUTPUT.PUT_LINE (current_employee.name);
16        END LOOP;
17
18    END print_employee_roster;
19

```



Be certain to remove any unnecessary variables from your procedure.

4b Execute the **print_employee_roster** procedure. Include the results in your submission.

4c Which of the processing steps associated with cursor usage are implicitly handled by the PL/SQL engine?

Parameterized Cursors

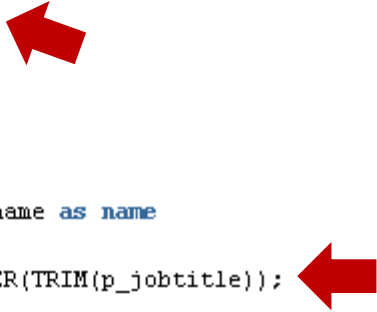
- 5a** Modify the **print_employee_roster** procedure to accept a parameter that is subsequently used in the WHERE clause of the cursor to restrict the data retrieved by the query.

Worksheet Query Builder

```

1  CREATE OR REPLACE PROCEDURE print_employee_roster
2  (p_jobtitle IN employee.jobtitle%type)
3  AS
4
5      CURSOR all_employees
6      IS
7          SELECT  employeeid,
8                  lastname || ', ' || firstname as name
9          FROM    employee
10         WHERE   UPPER(TRIM(JOBTITLE)) = UPPER(TRIM(p_jobtitle));
11
12 BEGIN
13
14     FOR current_employee IN all_employees LOOP
15
16         DBMS_OUTPUT.PUT (RPAD(current_employee.employeeid, 15, ' '));
17         DBMS_OUTPUT.PUT_LINE (current_employee.name);
18     END LOOP;
19
20 END print_employee_roster;

```



- 5b** What is the code necessary to display all 'sales' personnel at Eagle Electronics using the **print_employee_roster** procedure? Include the results in your submission.


- 5c** What is the code necessary to display all 'assembly' personnel at Eagle Electronics using the **print_employee_roster** procedure? Include the results in your submission.

- 5d** What happens if you provide the procedure a non-existent job title? For example: 'student'. Does the procedure crash? Or display nothing? Or what?

Exception Handling

- 6a** Modify the **print_employee_roster** procedure to gracefully handle exceptions in a “generic” manner (e.g., treat all exceptions the same).

Worksheet	Query Builder
1	CREATE OR REPLACE PROCEDURE print_employee_roster
2	(p_jobtitle IN employee.jobtitle% type)
3	AS
4	
5	CURSOR all_employees
6	IS
7	SELECT employeeid,
8	lastname ', ' firstname as name
9	FROM employee
10	WHERE UPPER(TRIM(JOBTITLE)) = UPPER(TRIM(p_jobtitle));
11	
12	BEGIN
13	
14	FOR current_employee IN all_employees LOOP
15	
16	DBMS_OUTPUT.PUT (RPAD(current_employee.employeeid, 15, ' '));
17	DBMS_OUTPUT.PUT_LINE (current_employee.name);
18	END LOOP ;
19	
20	EXCEPTION
21	
22	WHEN OTHERS THEN
23	DBMS_OUTPUT.PUT (SQLCODE);
24	DBMS_OUTPUT.PUT (': ');
25	DBMS_OUTPUT.PUT_LINE (SUBSTR(SQLERRM, 1, 100));
26	END print_employee_roster;
27	



- 6b** What code is now necessary to display all ‘assembly’ personnel at Eagle Electronics using the **print_employee_roster** procedure? Include the results in your submission.
- 6c** If an exception was to occur, what would be displayed to the user? You will need to understand what SQLCODE and SQLERRM functions do

Write Your Own Code	
7a	<p>Create a procedure named customer_roster that accepts 1 parameter: state. The procedure should display the <u>company name</u>, <u>city</u>, <u>state</u>, and <u>contact name (in Last, First format)</u> for all customers that are located in the state specified.</p> <p>The procedure should:</p> <ul style="list-style-type: none"> • Use an explicit cursor and a WHILE Loop. • Format the output so it is easy to read (e.g., make certain the columns line up and formatting is standard). <p>Handle any exceptions that might occur gracefully, displaying the error number and message to the user.</p>
7b	<p>If we wanted to determine the customers that Eagle Electronics has in Georgia (GA) using the customer_roster procedure, what is the code necessary to execute the procedure? Verify your solution works by running it. Include the results in your submission.</p>
8a	<p>Create a procedure named customer_search that accepts 1 parameter: name. The procedure should display the <u>company name</u>, <u>contact first name</u>, <u>contact last name</u>, and <u>contact title</u> for any customer contact whose <u>last name</u> contains the characters string specified by the user (e.g., use LIKE functionality). The search should be case insensitive and ignore leading and trailing spaces.</p> <p>The procedure should:</p> <ul style="list-style-type: none"> • Use an explicit cursor. • Format the output so it is easy to read (e.g., make certain the columns line up and formatting is standard). <p>Handle any exceptions that might occur gracefully, displaying the error number and message to the user.</p>
8b	<p>If we wanted to use the customer_search procedure to identify all customer contacts having a last name containing the character string 'NA', what code is necessary to execute the procedure? Verify your solution works by running it. Include the results in your submission.</p>
8c	<p>Modify the search functionality of the customer_search procedure created in 8a. The search should now display the information for customer contacts <u>whose first name or last name</u> contains the specified character string.</p>
8d	<p>If we wanted to use the customer_search procedure to identify all customer contacts having a first or last name containing the character string 'na', what code is necessary to execute the procedure? Verify your solution works by running it. Include the results in your submission.</p>

8e	Modify the search functionality of the customer_search procedure in 8c to using an IMPLICIT cursor with a FOR loop.
8f	Repeat 8d with the updated procedure.

Option B

Optional Questions: EAGLE database	
1	<p>Create a procedure that would: given an employee first name, last name, and date of birth, find employees that they supervise. Order the supervised employees by last name. The procedure should:</p> <ul style="list-style-type: none"> • Use first name, last name, date of birth as input to your procedure • Use exception if any part of input is missing • Use exception if such employee doesn't exist (it should not be case sensitive) • Use exception if an employee doesn't supervise anyone • Use exception if date is invalid • Use exceptions for anything else that you can think of. • When you list the supervised employees, use a cursor and a FOR loop (implicit OPEN, FETCH) • Prior to listing the supervised employees, print a message to the screen indicating how many employees are there that match given criteria.
2	<p>Write a function that would: given an employee name, return the number of people they supervise. The function should:</p> <ul style="list-style-type: none"> • Use employee id name as input (using the format <firstname><space><lastname>) • Use exception if any part of input is missing • Use exception if invalid last name is entered, however, allow for variations in first name, such as initial. • If a user enters a name that corresponds to more than one employee, return the overall sum for all of them, not individual numbers. • Use exceptions for anything else that you can think of. • Use a parameterized cursor and a WHILE loop to display all supervised employees that you can find for a given name. • If an employee with such name doesn't exist, suggest a similarly spelled name that does exist.