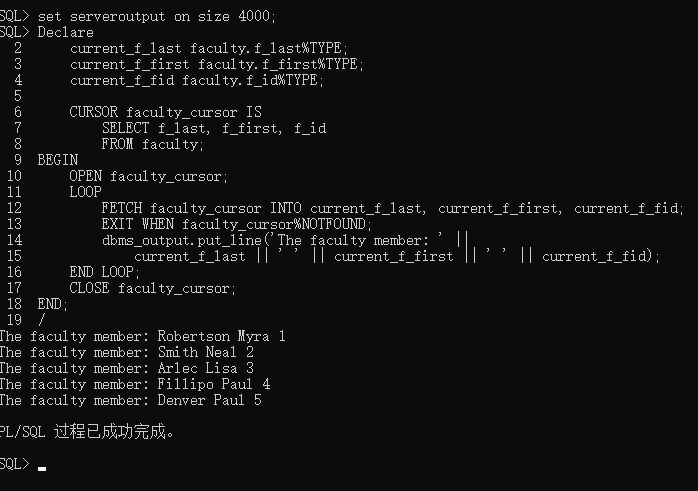
Lab 3: Database Processing using PL/SQL Programming

Submit all PL/SQL programs and their output to be saved into a text file Programs\_Lab3.sql with sql extension.

1. Explicit cursor using %TYPE anchored declaration.

1. **Declare**
2. current\_f\_last faculty.f\_last%TYPE;
3. current\_f\_first faculty.f\_first%TYPE;
4. current\_f\_fid faculty.f\_id%TYPE;
6. **CURSOR** faculty\_cursor **IS**
7. **SELECT** f\_last, f\_first, f\_id
8. **FROM** faculty;
9. **BEGIN**
10. **OPEN** faculty\_cursor;
11. LOOP
12. **FETCH** faculty\_cursor **INTO** current\_f\_last, current\_f\_first, current\_f\_fid;
13. EXIT **WHEN** faculty\_cursor%NOTFOUND;
14. dbms\_output.put\_line('The faculty member: ' ||
15. current\_f\_last || ' ' || current\_f\_first || ' ' || current\_f\_fid);
16. **END** LOOP;
17. **CLOSE** faculty\_cursor;
18. **END**;



2. Execute the script file registration.sql for creating tables of Registration System.

a) Using explicit cursor, and the reserved word %TYPE variables to display explicit cursor values, Edit PL/SQL program to display all level classes and the description of class using the nested IF/THEN/ELSE as shown in Figure 1.

1. **DECLARE**
2. current\_lc\_desc levelclass.lc\_desc%TYPE;
3. current\_lc\_min levelclass.lc\_min%TYPE;
4. current\_lc\_max levelclass.lc\_max%TYPE;
5. **CURSOR** lc\_cursor **IS**
6. **SELECT** lc\_desc, lc\_min, lc\_max
7. **FROM** levelclass;
8. **BEGIN**
9. **OPEN** lc\_cursor;
10. LOOP
11. **FETCH** lc\_cursor **INTO** current\_lc\_desc, current\_lc\_min, current\_lc\_max;
12. EXIT **WHEN** lc\_cursor%NOTFOUND;
13. IF current\_lc\_max <= 12 **THEN**
14. dbms\_output.put\_line('LC Type 1, Desc of Class: ' ||
15. 'Small Class, Need to upgrade this class building');
16. ELSIF current\_lc\_max <= 30 AND current\_lc\_min > 12 **THEN**
17. dbms\_output.put\_line('LC Type 2, Desc of Class: ' ||
18. 'Medium Class, Need to Renovate this class building');
19. ELSIF current\_lc\_max <= 90 AND current\_lc\_min > 30 **THEN**
20. dbms\_output.put\_line('LC Type 3, Desc of Class: ' ||
21. 'Large Class, No Renovation for this class building');
22. **ELSE**
23. dbms\_output.put\_line('LC Type 4, Desc of Class: ' ||
24. 'Extra Large Class, Need to equip this class building with new speakers');
25. **END** IF;
26. **END** LOOP;
27. **CLOSE** lc\_cursor;
28. **END**;



b) Using **Explicit Cursor**, Edit PL/SQL program to display all students’ names within **student** table along with the first name and last name of their respective supervisor to befetched from **faculty** table as shown in Figure 2. Use **%ROWTYPE** variable to displayexplicit cursor values.

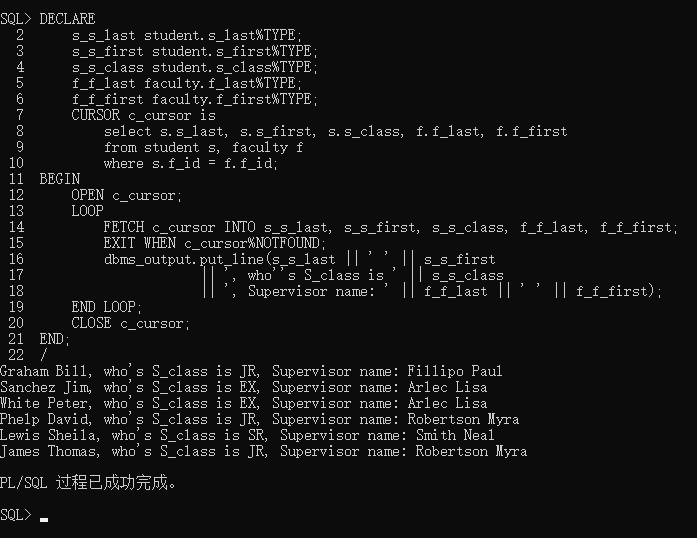
Solution I

1. **DECLARE**
2. current\_s\_last student.s\_last%TYPE;
3. current\_s\_first student.s\_first%TYPE;
4. current\_s\_class student.s\_class%TYPE;
5. current\_f\_id student.f\_id%TYPE;
7. **CURSOR** s\_cursor **IS**
8. **SELECT** s\_last, s\_first, s\_class, f\_id
9. **FROM** student;
10. **BEGIN**
11. **OPEN** s\_cursor;
12. LOOP
13. **FETCH** s\_cursor **INTO** current\_s\_last, current\_s\_first,
14. current\_s\_class, current\_f\_id;
15. EXIT **WHEN** s\_cursor%NOTFOUND;
16. **DECLARE**
17. current\_f\_last faculty.f\_last%TYPE;
18. current\_f\_first faculty.f\_first%TYPE;
20. **CURSOR** f\_cursor **IS**
21. **select** f\_last, f\_first **from** faculty
22. **WHERE** f\_id = current\_f\_id;
23. **BEGIN**
24. **OPEN** f\_cursor;
25. **FETCH** f\_cursor **INTO** current\_f\_last, current\_f\_first;
26. dbms\_output.put\_line(current\_s\_last || ' ' || current\_s\_first
27. || ', who''s S\_class is ' || current\_s\_class
28. || ', Supervisor name: ' || current\_f\_last || ' ' || current\_f\_first);
29. **CLOSE** f\_cursor;
30. **END**;
31. **END** LOOP;
32. **CLOSE** s\_cursor;
33. **END**;



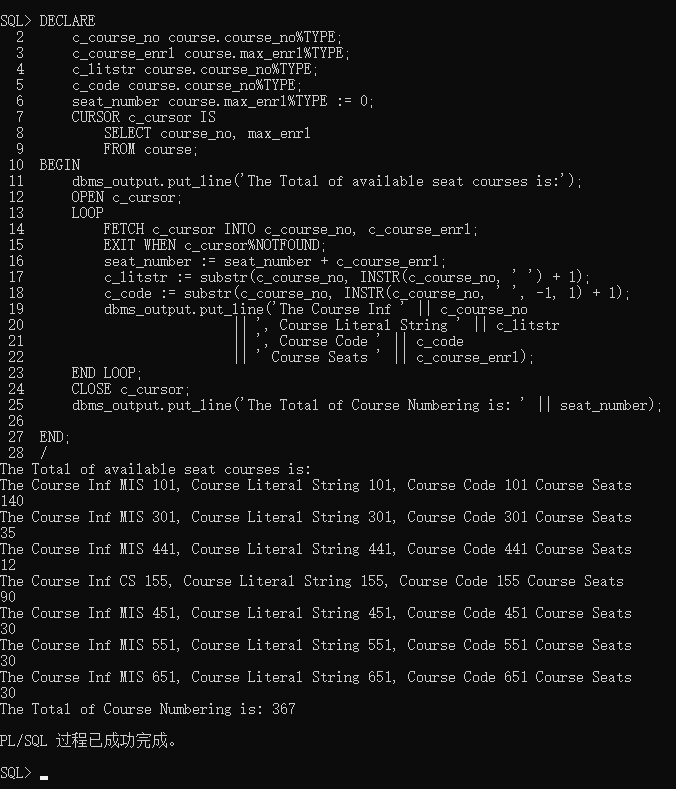
Solution II

1. **DECLARE**
2. s\_s\_last student.s\_last%TYPE;
3. s\_s\_first student.s\_first%TYPE;
4. s\_s\_class student.s\_class%TYPE;
5. f\_f\_last faculty.f\_last%TYPE;
6. f\_f\_first faculty.f\_first%TYPE;
7. **CURSOR** c\_cursor **is**
8. **select** s.s\_last, s.s\_first, s.s\_class, f.f\_last, f.f\_first
9. **from** student s, faculty f
10. **where** s.f\_id = f.f\_id;
11. **BEGIN**
12. **OPEN** c\_cursor;
13. LOOP
14. **FETCH** c\_cursor **INTO** s\_s\_last, s\_s\_first, s\_s\_class, f\_f\_last, f\_f\_first;
15. EXIT **WHEN** c\_cursor%NOTFOUND;
16. dbms\_output.put\_line(s\_s\_last || ' ' || s\_s\_first
17. || ', who''s S\_class is ' || s\_s\_class
18. || ', Supervisor name: ' || f\_f\_last || ' ' || f\_f\_first);
19. **END** LOOP;
20. **CLOSE** c\_cursor;
21. **END**;



c) Using **explicit cursor**, and the **anchored declaration %TYPE** to display explicit cursorvalues. Edit PL/SQL program to display the total of all available seats (**max\_enrl**) of allcourses. Display the output as shown in Figure 3.

1. **DECLARE**
2. c\_course\_no course.course\_no%TYPE;
3. c\_course\_enrl course.max\_enrl%TYPE;
4. c\_litstr course.course\_no%TYPE;
5. c\_code course.course\_no%TYPE;
6. seat\_number course.max\_enrl%TYPE := 0;
7. **CURSOR** c\_cursor **IS**
8. **SELECT** course\_no, max\_enrl
9. **FROM** course;
10. **BEGIN**
11. dbms\_output.put\_line('The Total of available seat courses is:');
12. **OPEN** c\_cursor;
13. LOOP
14. **FETCH** c\_cursor **INTO** c\_course\_no, c\_course\_enrl;
15. EXIT **WHEN** c\_cursor%NOTFOUND;
16. seat\_number := seat\_number + c\_course\_enrl;
17. c\_litstr := substr(c\_course\_no, INSTR(c\_course\_no, ' ') + 1);
18. c\_code := substr(c\_course\_no, INSTR(c\_course\_no, ' ', -1, 1) + 1);
19. dbms\_output.put\_line('The Course Inf ' || c\_course\_no
20. || ', Course Literal String ' || c\_litstr
21. || ', Course Code ' || c\_code
22. || ' Course Seats ' || c\_course\_enrl);
23. **END** LOOP;
24. **CLOSE** c\_cursor;
25. dbms\_output.put\_line('The Total of Course Numbering is: ' || seat\_number);
27. **END**;



3. Using multiple Explicit Cursors and Joining Tables.a) Using explicit cursor and the anchored declaration %TYPE when declaring variable, edit  
PL/SQL program to display all Faculties, their supervised students and to which  
department they belong. Display the output as shown in Figure 4

1. **DECLARE**
2. current\_f\_id faculty.f\_id%TYPE;
3. current\_f\_last faculty.f\_last%TYPE;
4. current\_f\_first faculty.f\_first%TYPE;
5. current\_deptid faculty.deptid%TYPE;
6. **CURSOR** f\_cursor **IS**
7. **SELECT** f\_id, f\_last, f\_first, deptid
8. **FROM** faculty;
9. **BEGIN**
10. **OPEN** f\_cursor;
11. LOOP
12. **FETCH** f\_cursor **INTO** current\_f\_id, current\_f\_last,
13. current\_f\_first, current\_deptid;
14. EXIT **WHEN** f\_cursor%NOTFOUND;
15. **DECLARE**
16. current\_s\_last student.s\_last%TYPE;
17. current\_s\_first student.s\_first%TYPE;
18. current\_s\_class student.s\_class%TYPE;
19. d\_deptname department.deptname%TYPE;
20. d\_location department.location%TYPE;
21. s\_f\_id student.f\_id%TYPE;
22. **CURSOR** s\_cursor **IS**
23. **SELECT** s\_last, s\_first, s\_class, f\_id
24. **FROM** student;
25. **BEGIN**
26. **OPEN** s\_cursor;
27. LOOP
28. **FETCH** s\_cursor **INTO** current\_s\_last, current\_s\_first,
29. current\_s\_class, s\_f\_id;
30. EXIT **WHEN** s\_cursor%NOTFOUND;
31. IF s\_f\_id = current\_f\_id **THEN**
32. dbms\_output.put\_line('The Faculty Inf: ' || current\_f\_id || ', '
33. || current\_f\_last || ' ' || current\_f\_first);
34. dbms\_output.put\_line('  Student Name:' || current\_s\_last
35. || ' ' || current\_s\_first
36. || ', St Class Type:' || current\_s\_class);
37. **SELECT** deptname, location
38. **INTO** d\_deptname, d\_location
39. **FROM** department d
40. **WHERE** d.deptid = current\_deptid;
41. dbms\_output.put\_line('  Dep Info:' || d\_deptname
42. || ', ' || d\_location);
43. **END** IF;
44. **END** LOOP;
45. **CLOSE** s\_cursor;
46. **END**;
47. **END** LOOP;
48. **CLOSE** f\_cursor;
49. **END**;



b) Using explicit cursor, edit PL/SQL program to display all departments, and its faculties.Display the output shown in Figure 5. Use **%ROWTYPE** variable to display explicit cursorvalues.

1. **DECLARE**
2. d\_dept department%ROWTYPE;
3. **CURSOR** cur\_dept **is** **SELECT** \* **FROM** department;
4. f\_facu faculty%ROWTYPE;
5. **CURSOR** cur\_facu **is** **SELECT** \* **FROM** faculty;
6. i **INTEGER**;
7. **BEGIN**
8. **OPEN** cur\_dept;
9. dbms\_output.put\_line('The List of Departments and its Faculties:');
10. LOOP
11. **FETCH** cur\_dept **into** d\_dept;
12. EXIT **WHEN** cur\_dept%NOTFOUND;
13. dbms\_output.put\_line('Dep ID:' || d\_dept.deptid
14. || ', Dep Name: ' || d\_dept.deptname
15. || ', Dep Loc: ' || d\_dept.location);
16. **OPEN** cur\_facu;
17. i := 1;
18. LOOP
19. **FETCH** cur\_facu **into** f\_facu;
20. EXIT **WHEN** cur\_facu%NOTFOUND;
21. IF f\_facu.deptid = d\_dept.deptid **THEN**
22. dbms\_output.put\_line(i || ')       '
23. || f\_facu.f\_last || ' ' || f\_facu.f\_first
24. || ', who''s SOCIAL INS is ' || f\_facu.soc\_ins);
25. i := i + 1;
26. **END** IF;
27. **END** LOOP;
28. **CLOSE** cur\_facu;
29. IF i>1 **THEN**
30. dbms\_output.put\_line('');
31. **ELSE**
32. dbms\_output.put\_line('-----No Faculty Assigned for this Department so far--');
33. **END** IF;
34. **END** LOOP;
35. **CLOSE** cur\_dept;
36. **END**;

