

# Assignment 1

*by Chan Seow Fen*

---

**Submission date:** 01-Mar-2024 10:47PM (UTC+0800)

**Submission ID:** 2308867180

**File name:** New\_Microsoft\_Word\_Document.docx (4.73M)

**Word count:** 1209

**Character count:** 5584

## 1.0 Program Listing

### 3 1.1 Task 1(a)

```
1  SELECT
2    s.sno AS "Subject No",
3    UPPER(s.stitle) AS "Subject Title",
4    REPLACE(REPLACE(REPLACE(c.term, 'f', 'Fall-'), 'sp', 'Spring-'), 's', 'Summer-') AS Term,
5    INITCAP(com.compname) AS "Component Name",
6    com.maxpoints AS Maxpoints,
7    com.weight AS Weight
8  FROM courses c
9  JOIN subject s ON s.sno = c.sno
10 JOIN components com ON com.term = c.term AND com.lineno = c.lineno
11 WHERE c.term = '&term' AND com.weight >= '&min_weight'
12 /
```

Figure 1.1 Source Code of Task1 (a)

### 3 1.2 Task 1(b)

```
1 CREATE OR REPLACE VIEW student_course_average AS
2 SELECT
3   REPLACE(REPLACE(REPLACE(e.term, 'f', 'Fall-'), 'sp', 'Spring-'), 's', 'Summer-') AS Term,
4   e.lineno AS "Line No",
5   e.sid AS "Student ID",
6   s.lname AS "Last Name",
7   s.fname AS "First Name",
8   ROUND((SUM(sc.points * com.weight)/SUM(com.weight*com.maxpoints))*100) AS "Course Average"
9 FROM enrolls e
10 JOIN students s ON e.sid = s.sid
11 JOIN scores sc ON e.term = sc.term AND e.lineno = sc.lineno AND e.sid = sc.sid
12 JOIN components com ON e.term = com.term AND e.lineno = com.lineno AND sc.compname = com.compname
13 WHERE e.term = '&1' AND e.lineno = '&2'
14 GROUP BY e.term, e.lineno, e.sid, s.lname, s.fname
15 /
```

Figure 1.2 Source Code of Task1 (b)

### 1.3 Task 2

```
1  CREATE OR REPLACE PROCEDURE add_course
2  (
3      c_term IN VARCHAR2, c_lineno IN NUMBER, c_sno IN VARCHAR2,
4      c_A IN NUMBER, c_B IN NUMBER, c_C IN NUMBER, c_D IN NUMBER)
5  AS
6
7      e_term_error EXCEPTION; --Does not follows Business Rule #1
8      PRAGMA EXCEPTION_INIT(e_term_error, -10001); --associate the error code
9
10     e_grade_error_1 EXCEPTION; --Does not follows Business Rule #2 - A > B > C > D
11     PRAGMA EXCEPTION_INIT(e_grade_error_1, -20001); --associate the error code
12
13     e_grade_error_2 EXCEPTION; --Does not follows Business Rule #2 - Value 0-100
14     PRAGMA EXCEPTION_INIT(e_grade_error_2, -20002); --associate the error code
15
16 BEGIN
17     DBMS_OUTPUT.PUT_LINE('Attempt to insert a new course record.');
18
19     --Insert the new record
20     INSERT INTO courses VALUES(c_term, c_lineno, c_sno, c_A, c_B, c_C, c_D);
21
22     --Business Rule 1: term must be started with "f" (Fall), "sp" (Spring), "s" (Summer)
23     --and followed by 2 digits indicating year
24     IF NOT REGEXP_LIKE(c_term, '^f{1}{0}(s|sp){1}{0}(\d{2})$') THEN
25         --raise the EXCEPTION
26         RAISE e_term_error;
27     END IF;
28
29     --Business Rule 2: Value of Grade A must bigger than Grade B (minimum gap 10 points)
30     --Some for other grades subsequently
31     --Grade values must be in the range of 0-100
32     IF (c_A < 0 OR c_A > 100) OR (c_B < 0 OR c_B > 100) OR (c_C < 0 OR c_C > 100) OR (c_D < 0 OR c_D > 100) THEN
33         --raise the EXCEPTION
34         RAISE e_grade_error_1;
35     END IF;
36
37     IF (c_A < 0 OR c_A > 100) OR (c_B < 0 OR c_B > 100) OR (c_C < 0 OR c_C > 100) OR (c_D < 0 OR c_D > 100) THEN
38         --raise the EXCEPTION
39         RAISE e_grade_error_2;
40     END IF;
41
42     DBMS_OUTPUT.PUT_LINE('New course record added!');
43
44     COMMIT; --Save the record
45
46 EXCEPTION
47
48     WHEN e_term_error THEN --Violate Business Rule
49         DBMS_OUTPUT.PUT_LINE('Attempt failed.');
50         DBMS_OUTPUT.PUT_LINE('Error code: ' || SQLCODE);
51         DBMS_OUTPUT.PUT_LINE('The term format is invalid, it should started with "f"(Fall), "sp"(Spring), "s"(Summer) and followed by 2 digits for year indication. Eg:f96');
52         ROLLBACK;
53
54     WHEN e_grade_error_1 THEN --Violate Business Rule
55         DBMS_OUTPUT.PUT_LINE('Attempt failed.');
56         DBMS_OUTPUT.PUT_LINE('Error code: ' || SQLCODE);
57         DBMS_OUTPUT.PUT_LINE('The value for higher grade should be more than the next grade by at least 10 points. Eg: Grade A: 90 points, Grade B: 80 points, Grade C: 70 points, Grade D: 60');
58         ROLLBACK;
59
60     WHEN e_grade_error_2 THEN --Violate Business Rule
61         DBMS_OUTPUT.PUT_LINE('Attempt failed.');
62         DBMS_OUTPUT.PUT_LINE('Error code: ' || SQLCODE);
63         DBMS_OUTPUT.PUT_LINE('The grade values must be in the range of 0-100. Eg: Grade A: 95 points');
64         ROLLBACK;
65
66     WHEN DUP_VAL_ON_INDEX THEN --Inserting Some Record
67         DBMS_OUTPUT.PUT_LINE('Attempt failed.');
68         DBMS_OUTPUT.PUT_LINE('Error code: ' || SQLCODE);
69         DBMS_OUTPUT.PUT_LINE('The term ' || c_term || ' and line no of ' || '#' || c_lineno || ' is found in the database!');
70         DBMS_OUTPUT.PUT_LINE('Message: ' || SQLERRM);
71         ROLLBACK;
72
73     --All error handlers
74     WHEN OTHERS THEN
75         DBMS_OUTPUT.PUT_LINE('Attempt failed.');
76         DBMS_OUTPUT.PUT_LINE('Error code: ' || SQLCODE);
77         DBMS_OUTPUT.PUT_LINE('Error message: ' || SQLERRM);
78         ROLLBACK; --write the transaction
79     END;
80 
```

Figure 1.4 Source Code of Task2

## 2.0 Screenshots and Test Cases

### 2.1 Task 1(a)

REI@orcl-> SELECT * FROM components;				
TERM	LINENO	COMPNAME	MAXPOINTS	WEIGHT
F96	1031	exam1	100	30
F96	1031	quizzes	80	20
F96	1031	final	100	50
F96	1032	programs	400	40
F96	1032	midterm	100	20
F96	1032	final	100	40
sp97	1031	paper	100	50
sp97	1031	project	100	50

Figure 2.1 All Components Record

	Input	Expected Outcome
Term	f96	Only record 3, record 4, record 6 will be selected along with the subject no, capitalised subject title, expanded term, component name with first letter capitalised, maxpoints and weight.
Minimum Weight	40	

Table 1.1 Test Case for Task 1(a)

According to *Figure 2.1*, listing out all of the record in components, it is expected that only record 3, record 4 and record 6 will be selected as the output of the query when user key in f96 as term and 40 as minimum weight. The reason is that, records other than having term of f96 and weight greater or equal to 40 should be filtered out.

```
SQL> @ c:\sqlasm_task1a - run script
SQL> SELECT
2   s.sno AS "Subject No",
3   UPPER(s.stitle) AS "Subject Title",
4   REPLACE(REPLACE(REPLACE(c.term, 'f', 'Fall-'), 'sp', 'Spring-'), 's', 'Summer-') AS Term,
5   INITCAP(com.componentname) AS "Component Name",
6   com.maxpoints AS Maxpoints,
7   com.weight AS Weight
8  FROM courses c
9  JOIN subject s ON s.sno = c.sno
10 JOIN components com ON com.term = c.term AND com.lineno = c.lineno
11 WHERE c.term = '&term' AND com.weight >= '&min_weight'
12 /
Enter value for term: f96
Enter value for min_weight: 40 } user input
old:SELECT
      s.sno AS "Subject No",
      UPPER(s.stitle) AS "Subject Title",
      REPLACE(REPLACE(REPLACE(c.term, 'f', 'Fall-'), 'sp', 'Spring-'), 's', 'Summer-') AS Term,
      INITCAP(com.componentname) AS "Component Name",
      com.maxpoints AS Maxpoints,
      com.weight AS Weight
    FROM courses c
   JOIN subject s ON s.sno = c.sno
  JOIN components com ON com.term = c.term AND com.lineno = c.lineno
 WHERE c.term = '&term' AND com.weight >= '&min_weight'

new:SELECT
      s.sno AS "Subject No",
      UPPER(s.stitle) AS "Subject Title",
      REPLACE(REPLACE(REPLACE(c.term, 'f', 'Fall-'), 'sp', 'Spring-'), 's', 'Summer-') AS Term,
      INITCAP(com.componentname) AS "Component Name",
      com.maxpoints AS Maxpoints,
      com.weight AS Weight
    FROM courses c
   JOIN subject s ON s.sno = c.sno
  JOIN components com ON com.term = c.term AND com.lineno = c.lineno
 WHERE c.term = 'F96' AND com.weight >= '40'
}
3 rows selected. } output
```

Figure 2.2 Output of Test Case

*Figure 2.2* shows that the program works perfectly which its output is same as the expected outcome of the test case in *Table 1.1*.

## 2.2 Task 1(b)

The course average is the sum of the weighted averages of the individual component scores. According to (Mamalat, 2020), the formula of weighted average is defined by divide the sum of the products of points with its corresponded weight by the total of the weights as shown in *Figure 2.3*.

$$\begin{array}{l}
 \text{Formula} \\
 \frac{\sum \text{Points} \times \text{Weight}}{\sum \text{Weight}}
 \end{array}
 \quad \text{Eg:}
 \begin{array}{|c|c|} \hline
 \text{Points} & \text{Weight} \\ \hline
 90 & 60 \\ \hline
 85 & 40 \\ \hline
 \end{array}$$

$$\begin{aligned}
 &\text{Weighted Average} \\
 &\frac{(90 \times 60) + (85 \times 40)}{60+40} \\
 &= \frac{8800}{100} \\
 &= 88
 \end{aligned}$$

*Figure 2.3 Formula of Weighted Average and Example of Calculation*

However, due to the reason that, some of the components having maximum marks that are greater than 100 or not equal to 100, for example, programs with the maxpoints of 400, and quizzes with the maxpoints of 80 as shown in *Figure 2.4*.

REI@orcl-> SELECT * FROM components;				
TERM	LINENO	COMPNAME	MAXPOINTS	WEIGHT
f96	1031	exam1	100	30
f96	1031	quizzes	80	20
f96	1031	final	100	50
f96	1032	programs	400	40
f96	1032	midterm	100	20
f96	1032	final	100	40
sp97	1031	paper	100	50
sp97	1031	project	100	50

*Figure 2.4 Components Maxpoints not 100 for All Components*

Hence, when applying the formula for weighted average, one more value has to be considered, which is its maxpoints. Hence, the formula used in this program is shown in *Figure 2.5*.

$$\begin{array}{l}
 \text{Formula} \\
 \frac{\sum \text{Points} \times \text{Weight}}{\sum \text{Weight} \times \text{maxpoints}} \times 100\%
 \end{array}
 \quad \text{Eg:}
 \begin{array}{|c|c|c|} \hline
 \text{Points} & \text{Weight} & \text{Max points} \\ \hline
 90 & 60 & 120 \\ \hline
 85 & 40 & 90 \\ \hline
 \end{array}$$

$$\begin{aligned}
 &\text{Weighted Average} \\
 &\frac{(90 \times 60) + (85 \times 40)}{(60 \times 120) + (40 \times 90)} \times 100\% \\
 &= \frac{8800}{10800} \times 100\% \\
 &= 81.48
 \end{aligned}$$

*Figure 2.5 Formula Used in Program Calculating Weighted Average with Example*

Hence, using this formula, the expected course average for the records selected in *Table 1.2* test case is shown in *Figure 2.6*.

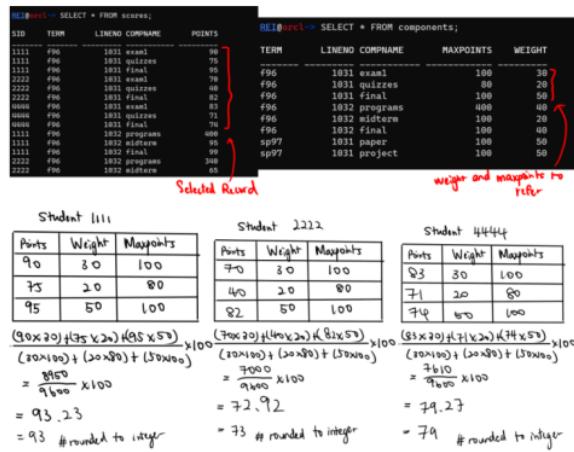


Figure 2.6 Expected Course Average for Selected Records

	Input	Expected Outcome
Term	f96	Only select record with the term of f96, and lineno of 1031. Display the query result with the term, lineno, student ID, last name, first name and course average.
Line No	1031	

Table 1.2 Test Case for Task 1(b)

Hence, based on the output in *Figure 2.7*, it is clear that the program executed well without any mistake as it is same as the expected outcome as stated in *Table 1.2* test case.

run script and pass parameter to create view

```

SQL> @c:\sql\lasm_task1b 'f96' '1031'
SQL> CREATE OR REPLACE VIEW student_course_average AS
  1> SELECT *
  2>   REPLACE(REPLACE(REPLACE(e.term, 'f', 'Fall--'), 'sp', 'Spring--'), 's', 'Summer--') AS Term,
  3>   e.lineno AS "Line No",
  4>   e.sid AS "Student ID",
  5>   s.lname AS "Last Name",
  6>   s.fname AS "First Name",
  7>   ROUND((SUM(lsc.points * com.weight)/SUM(com.weight*com.maxpoints))*100) AS "Course Average"
  8>   FROM enrolled lsc
  9> JOIN students s ON e.sid = s.sid
 10> JOIN scores sc ON e.term = sc.term AND e.lineno = sc.lineno AND e.sid = sc.sid
 11> JOIN components com ON e.term = com.term AND e.lineno = com.lineno AND sc.cmpname = com.cmpname
 12> WHERE e.term = 'f96' AND e.lineno = '1031'
 13> GROUP BY e.term, e.lineno, e.sid, s.lname, s.fname
 14>
old:CREATE OR REPLACE VIEW student_course_average AS
SELECT *
  REPLACE(REPLACE(REPLACE(e.term, 'f', 'Fall--'), 'sp', 'Spring--'), 's', 'Summer--') AS Term,
  e.lineno AS "Line No",
  e.sid AS "Student ID",
  s.lname AS "Last Name",
  s.fname AS "First Name",
  ROUND((SUM(lsc.points * com.weight)/SUM(com.weight*com.maxpoints))*100) AS "Course Average"
FROM enrolled lsc
JOIN students s ON e.sid = s.sid
JOIN scores sc ON e.term = sc.term AND e.lineno = sc.lineno AND e.sid = sc.sid
JOIN components com ON e.term = com.term AND e.lineno = com.lineno AND sc.cmpname = com.cmpname
WHERE e.term = 'f96' AND e.lineno = '1031'
GROUP BY e.term, e.lineno, e.sid, s.lname, s.fname
  View STUDENT_COURSE_AVERAGE created. - view created
  - query the result from the view
SQL> SELECT * FROM student_course_average;

```

View Result

Figure 2.7 Output of Test Case

## 2.3 Task 2

### 2.3.1 Successful Case

Figure 2.8 shows the initial records in courses table.

TERM	LINENO	SNO	A	B	C	D
f03	1032	csc498	95	85	65	45
f96	1031	csc226	90	80	65	50
f96	1032	csc226	90	80	65	50
sp97	1031	csc227	90	80	65	50

4 rows selected.

Figure 2.8 Initial Records in Courses Table

Figure 2.9 shows that after compiled the file for procedure and run the script for prompting user input, then, enter all valid input, it shows that the new record is added to the courses table which indicates the program runs well as expected corresponding to the test case Table 1.3.

	Input	Expected Outcome
Term	f01	
Line No	1031	
Subject No	csc481	New record added to the courses table with correct details entered.
Grade A	80	
Grade B	70	
Grade C	60	
Grade D	45	

Table 1.3 Test Case for Task 2 – Successful Case

```

Procedure ADD_COURSE compiled - compiled procedure file
SQL> @ c:\sql\asm_task2run - run prompting script
SQL> ACCEPT c_term CHAR prompt 'Please enter Term (Eg:sp97): ';
Please enter Term (Eg:sp97): f01
SQL> ACCEPT c_lineno NUMBER prompt 'Please enter Line No (Eg:1031): ';
Please enter Line No (Eg:1031): 1031
SQL> ACCEPT c_sno CHAR prompt 'Please enter Subject No (Eg:csc227): ';
Please enter Subject No (Eg:csc227): csc481
SQL> ACCEPT c_A NUMBER prompt 'Please enter Grade A Values (Between 0-100): ';
Please enter Grade A Values (Between 0-100): 80
SQL> ACCEPT c_B NUMBER prompt 'Please enter Grade B Values (Between 0-100): ';
Please enter Grade B Values (Between 0-100): 70
SQL> ACCEPT c_C NUMBER prompt 'Please enter Grade C Values (Between 0-100): ';
Please enter Grade C Values (Between 0-100): 60
SQL> ACCEPT c_D NUMBER prompt 'Please enter Grade D Values (Between 0-100): ';
Please enter Grade D Values (Between 0-100): 45
SQL>
SQL> BEGIN
  2   add_course('&c_term', '&c_lineno', '&c_sno', '&c_A', '&c_B', '&c_C', '&c_D');
  3 END;
  4 /
old:BEGIN
      add_course('&c_term', '&c_lineno', '&c_sno', '&c_A', '&c_B', '&c_C', '&c_D');
END;
new:BEGIN
      add_course('f01', '1031', 'csc481', '80', '70', '60', '45');
END;
Attempt to insert a new course record.
New course record added!

PL/SQL procedure successfully completed.
SQL> SELECT * FROM courses; - select all records from courses
table to validate
TERM      LINENO    SNO      A      B      C      D
f01       1031     csc481   80     70     60     45 ← New record added
f03       1032     csc498   95     85     65     45
f96       1031     csc226   90     80     65     50
f96       1032     csc226   90     80     65     50
sp97      1031     csc227   90     80     65     50
with correct
details

```

Figure 2.9 Output of Test Case

### 2.3.2 Unsuccessful Case - Business Rule #1 Violated [Term Format]

As stated in the assignment instructions, the first business rule of the program is that the format of term should be started with the character either 'f' (Fall), 'sp' (Spring) or 's' (Summer) and followed by 2 digits to indicate year (Eg:sp97). Hence, the expected outcome is shown in test case *Table 1.4*. Note that slash (/) indicates the input for second test.

	Input	Expected Outcome
Term	a22 / f999	
Line No	1031	
Subject No	csc227	
Grade A	90	
Grade B	80	
Grade C	70	
Grade D	60	

Table 1.4 Test Case for Task 2 – Unsuccessful Case [Business Rule #1 Violated]

*Figure 2.10* and *Figure 2.11* show that the program can capture the action of user violating the business rule and prompt the error message with error code. The result has not been added to the courses table as shown in *Figure 2.12*.

```
SQL> # c:\sql\ass_task2run
SQL> ACCEPT c_ano CHAR prompt 'Please enter Term (Eg:sp97): '
Please enter Term (Eg:sp97): a22
SQL> ACCEPT c_lineno NUMBER prompt 'Please enter Line No (Eg:1031): '
Please enter Line No (Eg:1031): 1031
SQL> ACCEPT c_sno CHAR prompt 'Please enter Subject No (Eg:csc227): '
Please enter Subject No (Eg:csc227): csc227
SQL> ACCEPT c_a NUMBER prompt 'Please enter Grade A Values (Between 0-100): '
Please enter Grade A Values (Between 0-100): 90
SQL> ACCEPT c_b NUMBER prompt 'Please enter Grade B Values (Between 0-100): '
Please enter Grade B Values (Between 0-100): 80
SQL> ACCEPT c_c NUMBER prompt 'Please enter Grade C Values (Between 0-100): '
Please enter Grade C Values (Between 0-100): 70
SQL> ACCEPT c_d NUMBER prompt 'Please enter Grade D Values (Between 0-100): '
Please enter Grade D Values (Between 0-100): 60
SQL>
SQL> BEGIN
2   add_course('ac_term', 'ac_lineno', 'ac_sno', 'ac_A', 'ac_B', 'ac_C', 'ac_D');
3 END;
4 /
old:BEGIN
      add_course('ac_term', 'ac_lineno', 'ac_sno', 'ac_A', 'ac_B', 'ac_C', 'ac_D');

END;
new:BEGIN
      add_course('a22', '1031', 'csc227', '90', '80', '70', '60');
END;
Attempt to insert a new course record.
Error: ORA-01403: 
The term format is invalid, it should started with "f"(Fall), "sp"(Spring), "s"(Summer) and followed by 2 digits for year indication. Eg:f96
-- Error message
PL/SQL procedure successfully completed.
```

Figure 2.10 Output of Test Case – Invalid Character for Term

```
SQL> # c:\sql\ass_task2run
SQL> ACCEPT c_ano CHAR prompt 'Please enter Term (Eg:sp97): '
Please enter Term (Eg:sp97): sp99
SQL> ACCEPT c_lineno NUMBER prompt 'Please enter Line No (Eg:1031): '
Please enter Line No (Eg:1031): 1031
SQL> ACCEPT c_sno CHAR prompt 'Please enter Subject No (Eg:csc227): '
Please enter Subject No (Eg:csc227): csc227
SQL> ACCEPT c_a NUMBER prompt 'Please enter Grade A Values (Between 0-100): '
Please enter Grade A Values (Between 0-100): 90
SQL> ACCEPT c_b NUMBER prompt 'Please enter Grade B Values (Between 0-100): '
Please enter Grade B Values (Between 0-100): 80
SQL> ACCEPT c_c NUMBER prompt 'Please enter Grade C Values (Between 0-100): '
Please enter Grade C Values (Between 0-100): 70
SQL> ACCEPT c_d NUMBER prompt 'Please enter Grade D Values (Between 0-100): '
Please enter Grade D Values (Between 0-100): 60
SQL>
SQL> BEGIN
2   add_course('ac_term', 'ac_lineno', 'ac_sno', 'ac_A', 'ac_B', 'ac_C', 'ac_D');
3 END;
4 /
old:BEGIN
      add_course('ac_term', 'ac_lineno', 'ac_sno', 'ac_A', 'ac_B', 'ac_C', 'ac_D');

END;
new:BEGIN
      add_course('sp99', '1031', 'csc227', '90', '80', '70', '60');
END;
Attempt to insert a new course record.
Error: ORA-01403: 
The term format is invalid, it should started with "f"(Fall), "sp"(Spring), "s"(Summer) and followed by 2 digits for year indication. Eg:f96
-- Error code
PL/SQL procedure successfully completed.
```

Figure 2.11 Output of Test Case – Invalid Digits for Term

SQL> SELECT * FROM courses;						
TERM	LINENO	SNO	A	B	C	D
f01	1031	csc481	80	70	60	45
f03	1032	csc498	95	85	65	45
f96	1031	csc226	90	80	65	50
f96	1032	csc226	90	80	65	50
sp97	1031	csc227	90	80	65	50

Figure 2.12 Result is Not Added

### 2.3.3 Unsuccessful Case - Business Rule #2 Violated [Grades Value]

As stated in the assignment instructions, the second business rule of the program is that the grades value should be in the range of 0-100 and the higher-grade value should be larger than the subsequent lower grade at least 10 points (A:90, B:80, C:70, D:60). Hence, the expected outcome is shown in test case *Table 1.5*. Note that slash (/) indicates the input for second test.

	Input	Expected Outcome
Term	f23	
Line No	1031	
Subject No	csc227	
Grade A	90 / -1	
Grade B	85	
Grade C	84	
Grade D	70	Record is not added and prompt out the error message with error code.

Table 1.5 Test Case for Task 2 – Unsuccessful Case [Business Rule #2 Violated]

*Figure 2.13* and *Figure 2.14* show that the program can capture the action of user violating the business rule and prompt the error message with error code. The result has not been added to the courses table as shown in *Figure 2.15*.

```
SQL> @ c:\sql\vw\vw_Task2Run
SQL> ACCEPT c NUMBER prompt 'Please enter Term (Eg:sp97): ';
Please enter Term (Eg:sp97): f23
SQL> ACCEPT l NUMBER prompt 'Please enter Line No (Eg:1031): ';
Please enter Line No (Eg:1031): 1031
SQL> ACCEPT s NUMBER prompt 'Please enter Subject No (Eg:csc227): ';
Please enter Subject No (Eg:csc227): csc227
SQL> ACCEPT g1 NUMBER prompt 'Please enter Grade A Values (Between 0-100): ' 90-85=5
Please enter Grade A Values (Between 0-100): 90
SQL> ACCEPT g2 NUMBER prompt 'Please enter Grade B Values (Between 0-100): ' 85-84=1
Please enter Grade B Values (Between 0-100): 85
SQL> ACCEPT g3 NUMBER prompt 'Please enter Grade C Values (Between 0-100): ' which is lesser
Please enter Grade C Values (Between 0-100): 84
SQL> ACCEPT g4 NUMBER prompt 'Please enter Grade D Values (Between 0-100): ' than '0'
Please enter Grade D Values (Between 0-100): 70
SQL>
SQL> BEGIN
 2   add_course('Ac_term', 'Ac_lineno', 'Ac_sno', 'Ac_A', 'Ac_B', 'Ac_C', 'Ac_D');
 3 END;
 4 /
old:BEGIN
add_course('Ac_term', 'Ac_lineno', 'Ac_sno', 'Ac_A', 'Ac_B', 'Ac_C', 'Ac_D');

new:BEGIN
add_course('f23', '1031', 'csc227', '90', '85', '84', '70');
END;
Attempt to insert a new course record.
Attempt failed.
Error code: -2290
The value for higher grade should be more than the next grade by at least 10 points. Eg: Grade A: 90
points, Grade B: 80 points, Grade C: 70 points, Grade D: 60 points
Error Message
```

Figure 2.13 Output of Test Case – The Higher Grade is Not At Least 10 Points Greater Than Subsequent Lower Grade

```
SQL> @ c:\sql\vw\vw_Task2Run
SQL> ACCEPT c NUMBER prompt 'Please enter Term (Eg:sp97): ';
Please enter Term (Eg:sp97): f23
SQL> ACCEPT l NUMBER prompt 'Please enter Line No (Eg:1031): ';
Please enter Line No (Eg:1031): 1031
SQL> ACCEPT s NUMBER prompt 'Please enter Subject No (Eg:csc227): ';
Please enter Subject No (Eg:csc227): csc227
SQL> ACCEPT g1 NUMBER prompt 'Please enter Grade A Values (Between 0-100): ' Not between
Please enter Grade A Values (Between 0-100): 90
SQL> ACCEPT g2 NUMBER prompt 'Please enter Grade B Values (Between 0-100): ' range
Please enter Grade B Values (Between 0-100): 85
SQL> ACCEPT g3 NUMBER prompt 'Please enter Grade C Values (Between 0-100): ' 0-100
Please enter Grade C Values (Between 0-100): 84
SQL> ACCEPT g4 NUMBER prompt 'Please enter Grade D Values (Between 0-100): ' Error code
Please enter Grade D Values (Between 0-100): 70
SQL>
SQL> BEGIN
 2   add_course('Ac_term', 'Ac_lineno', 'Ac_sno', 'Ac_A', 'Ac_B', 'Ac_C', 'Ac_D');
 3 END;
 4 /
old:BEGIN
add_course('Ac_term', 'Ac_lineno', 'Ac_sno', 'Ac_A', 'Ac_B', 'Ac_C', 'Ac_D');

new:BEGIN
add_course('f23', '1031', 'csc227', '-1', '85', '84', '70');
END;
Attempt to insert a new course record.
Attempt failed.
Error code: -2290
Error: ORA-02390: check constraint (SOMFEN.SYS_C097733) violated
Error Message
```

Figure 2.14 Output of Test Case – Grade Value Not in The Range of 0-100

```
SQL> SELECT * FROM courses;
-----+-----+-----+-----+-----+-----+
TERM    | LINENO | SNO   | A     | B     | C     | D
-----+-----+-----+-----+-----+-----+
f01    | 1031  | csc481| 80    | 70    | 60    | 45
f03    | 1032  | csc498| 95    | 85    | 65    | 45
f96    | 1031  | csc226| 90    | 80    | 65    | 50
f96    | 1032  | csc226| 90    | 80    | 65    | 50
sp97   | 1031  | csc227| 90    | 80    | 65    | 50
-----+-----+-----+-----+-----+-----+
5 rows selected.      Error Not Added
```

Figure 2.15 Result is Not Added

#### 2.3.4 Unsuccessful Case – Duplicated Record

In order to ensure the database integrity, it is necessary to ensure that there is no duplicated data or record. Hence, test case *Table 1.6* shows the expected outcome when user entered duplicated data on the term and lineno, which act as the primary key and having the unique constraint.

	Input	Expected Outcome
Term	sp97	
Line No	1031	
Subject No	csc498	
Grade A	90	
Grade B	80	
Grade C	70	
Grade D	60	Record is not added and prompt out the error message with error code.

*Table 1.6 Test Case for Task 2 – Unsuccessful Case [Duplicated Data]*

*Figure 2.16* shows the initial courses table and *Figure 2.17* shows that the program can capture the action of user violating the system integrity practices as they entered duplicated term and lineno, then the program prompts the error message with error code. The result has not been added to the courses table as shown in *Figure 2.18*.

```
SQL> SELECT * FROM courses;
TERM      LINENO SNO      A      B      C      D
-----  -----  -----  -----  -----  -----  -----
f01        1031 csc481    80     70     60     45
f03        1032 csc498    95     85     65     45
f96        1031 csc226   90     80     65     50
f96        1032 csc226   90     80     65     50
sp97       1031 csc227   90     80     65     50
5 rows selected.
```

*Figure 2.16 Initial Record*

```
SQL> c:\V:\U:\csc\csc498>
SQL> ACCEPT <c>Term CHAR prompt 'Please enter Term (Eg:sp97) : '
SQL> Please enter Term (Eg:sp97) : sp97
SQL> ACCEPT <c>LineNo NUMBER prompt 'Please enter Line No (Eg:1031) : '
SQL> Please enter Line No (Eg:1031) : 1031
SQL> ACCEPT <c>Sno CHAR prompt 'Please enter Subject No (Eg:csc227) : '
SQL> Please enter Subject No (Eg:csc227) : csc498
SQL> ACCEPT <c>A NUMBER prompt 'Please enter Grade A Values (Between 0-100) : '
SQL> Please enter Grade A Values (Between 0-100) : 90
SQL> ACCEPT <c>B NUMBER prompt 'Please enter Grade B Values (Between 0-100) : '
SQL> Please enter Grade B Values (Between 0-100) : 80
SQL> ACCEPT <c>C NUMBER prompt 'Please enter Grade C Values (Between 0-100) : '
SQL> Please enter Grade C Values (Between 0-100) : 70
SQL> ACCEPT <c>D NUMBER prompt 'Please enter Grade D Values (Between 0-100) : '
SQL> Please enter Grade D Values (Between 0-100) : 60
SQL> BEGIN
 2>   add_course('sc_term', 'sc_lineno', 'sc_sno', 'sc_A', 'sc_B', 'sc_C', 'sc_D');
 3> END;
old:BEGIN
add_course('sc_term', 'sc_lineno', 'sc_sno', 'sc_A', 'sc_B', 'sc_C', 'sc_D');

new:BEGIN
add_course('sp97', '1031', 'csc498', '90', '80', '70', '60');

END;
Attempt to add a new course record.
Record called < Select user record not added
The record for term sp97 and line no of #1031 is found in the database
Error: 1453
< Error like error Message >
PL/SQL procedure successfully completed.
```

*Figure 2.17 Output of Test Case*

```
SQL> SELECT * FROM courses;
TERM      LINENO SNO      A      B      C      D
-----  -----  -----  -----  -----  -----  -----
f01        1031 csc481    80     70     60     45
f03        1032 csc498    95     85     65     45
f96        1031 csc226   90     80     65     50
f96        1032 csc226   90     80     65     50
sp97       1031 csc227   90     80     65     50
5 rows selected.      Record Not Added
```

*Figure 2.18 Result is Not Added*

### 3.0 Listing of All PL/SQL Statements Processed

#### 3.1 Task 1(a)

```
1  SQL> SET ECHO ON
2  SQL> SET FEEDBACK ON
3  SQL> SET LINESIZE 100
4  SQL> SET PAGESIZE 200
5  SQL> SET SERVEROUTPUT ON
6  SQL> @ c:\sql\asm_task1a
7  SQL> SELECT
8    2    s.sno AS "Subject No",
9    3    UPPER(s.stitle) AS "Subject Title",
10   4    REPLACE(REPLACE(c.term, 'f', 'Fall-'), 'sp', 'Spring-'), 's', 'Summer-') AS Term,
11   5    INITCAP(com.compname) AS "Component Name",
12   6    com.maxpoints AS Maxpoints,
13   7    com.weight AS Weight
14   FROM courses c
15   JOIN subject s ON s.sno = c.sno
16   JOIN components com ON com.term = c.term AND com.lineno = c.lineno
17   WHERE c.term = '&term' AND com.weight >= '&min_weight'
18  /
19 Enter value for term: f96
20 Enter value for min_weight: 40
21 old:SELECT
22    s.sno AS "Subject No",
23    UPPER(s.stitle) AS "Subject Title",
24    REPLACE(REPLACE(REPLACE(c.term, 'f', 'Fall-'), 'sp', 'Spring-'), 's', 'Summer-') AS Term,
25    INITCAP(com.compname) AS "Component Name",
26    com.maxpoints AS Maxpoints,
27    com.weight AS Weight
28   FROM courses c
29   JOIN subject s ON s.sno = c.sno
30   JOIN components com ON com.term = c.term AND com.lineno = c.lineno
31  WHERE c.term = '&term' AND com.weight >= '&min_weight'
32
33 new:SELECT
34    s.sno AS "Subject No",
35    UPPER(s.stitle) AS "Subject Title",
36    REPLACE(REPLACE(REPLACE(c.term, 'f', 'Fall-'), 'sp', 'Spring-'), 's', 'Summer-') AS Term,
37    INITCAP(com.compname) AS "Component Name",
38    com.maxpoints AS Maxpoints,
39    com.weight AS Weight
40   FROM courses c
41   JOIN subject s ON s.sno = c.sno
42   JOIN components com ON com.term = c.term AND com.lineno = c.lineno
43  WHERE c.term = 'f96' AND com.weight >= '40'
44
45 Subject No      Subject Title          TERM      Component Name      MAXPOINTS      WEIGHT
46 -----
47 csc226        INTRODUCTION TO PROGRAMMING I Fall-96    Final           100          50
48 csc226        INTRODUCTION TO PROGRAMMING I Fall-96    Final           100          40
49 csc226        INTRODUCTION TO PROGRAMMING I Fall-96    Programs        400          40
50
51 3 rows selected.
52
53 SQL> SPOOL OFF
```

Figure 3.1 All Statements Processed of Task 1(a)

### 3.2 Task 1(b)

```
1 SQL> SET ECHO ON
2 SQL> SET FEEDBACK ON
3 SQL> SET LINESIZE 100
4 SQL> SET PAGESIZE 200
5 SQL> SET SERVEROUTPUT ON
6 SQL> @ c:\sqlasm_task1b\sp97' '1031'
7 SQL> @ c:\sqlasm_task1b\sp97' '1031'
8 SQL> CREATE OR REPLACE VIEW student_course_average AS
9  SELECT
10   3 REPLACE(REPLACE(REPLACE(e.term, 'f', 'Fall-'), 'sp', 'Spring-'), 's', 'Summer-') AS Term,
11   4 e.lineno AS "Line No",
12   5 e.sid AS "Student ID",
13   6 s.lname AS "Last Name",
14   7 s.fname AS "First Name",
15   8 ROUND((SUM(sc.points * com.weight)/SUM(com.weight*com.maxpoints))*100) AS "Course Average"
16  FROM enrolls e
17 JOIN students s ON e.sid = s.sid
18 JOIN scores sc ON e.term = sc.term AND e.lineno = sc.lineno AND e.sid = sc.sid
19 JOIN components com ON e.term = com.term AND e.lineno = com.lineno AND sc.compname = com.compname
20 WHERE e.term = '&1' AND e.lineno = '&2'
21 GROUP BY e.term, e.lineno, e.sid, s.lname, s.fname
22 /
23 old:CREATE OR REPLACE VIEW student_course_average AS
24 SELECT
25   REPLACE(REPLACE(REPLACE(e.term, 'f', 'Fall-'), 'sp', 'Spring-'), 's', 'Summer-') AS Term,
26   e.lineno AS "Line No",
27   e.sid AS "Student ID",
28   s.lname AS "Last Name",
29   s.fname AS "First Name",
30   ROUND((SUM(sc.points * com.weight)/SUM(com.weight*com.maxpoints))*100) AS "Course Average"
31  FROM enrolls e
32 JOIN students s ON e.sid = s.sid
33 JOIN scores sc ON e.term = sc.term AND e.lineno = sc.lineno AND e.sid = sc.sid
34 JOIN components com ON e.term = com.term AND e.lineno = com.lineno AND sc.compname = com.compname
35 WHERE e.term = '&1' AND e.lineno = '&2'
36 GROUP BY e.term, e.lineno, e.sid, s.lname, s.fname
37
38 new:CREATE OR REPLACE VIEW student_course_average AS
39 SELECT
40   REPLACE(REPLACE(REPLACE(e.term, 'f', 'Fall-'), 'sp', 'Spring-'), 's', 'Summer-') AS Term,
41   e.lineno AS "Line No",
42   e.sid AS "Student ID",
43   s.lname AS "Last Name",
44   s.fname AS "First Name",
45   ROUND((SUM(sc.points * com.weight)/SUM(com.weight*com.maxpoints))*100) AS "Course Average"
46  FROM enrolls e
47 JOIN students s ON e.sid = s.sid
48 JOIN scores sc ON e.term = sc.term AND e.lineno = sc.lineno AND e.sid = sc.sid
49 JOIN components com ON e.term = com.term AND e.lineno = com.lineno AND sc.compname = com.compname
50 WHERE e.term = 'sp97' AND e.lineno = '1031'
51 GROUP BY e.term, e.lineno, e.sid, s.lname, s.fname
52
53 View STUDENT.Course_Average created.
54
55 SQL> SELECT * FROM student_course_average
56  2 /
57
58 TERM          Line No Student ID    Last Name   First Name   Course Average
59 Spring-97      1031  5555        Yam          Elad           85
60 Spring-97      1031  6666        Herring      Lincoln       83
61
62 2 rows selected.
63
64
65 SQL> SPOOL OFF
```

Figure 3.2 All Statements Processed of Task 1(b)

### 3.3 Task 2

```

1  SQL> SET ECHO ON
2  SQL> SET FEEDBACK ON
3  SQL> SET LINESIZE 100
4  SQL> SET PAGESIZE 100
5  SQL> SET SERVEROUTPUT ON
6  SQL> @ c:\sql\asm_task2
7  PL/SQL CREATE OR REPLACE PROCEDURE add_course
8  (
9    c_term IN VARCHAR2, c_lineno IN NUMBER, c_sno IN VARCHAR2,
10   c_A IN NUMBER, c_B IN NUMBER, c_C IN NUMBER, c_D IN NUMBER) AS
11
12   e_term_error EXCEPTION; -- Does not follows Business Rule #1
13   PRAGMA EXCEPTION_INIT(e_term_error, -10001); --associate the error code
14
15   e_grade_error_1 EXCEPTION; --Does not follows Business Rule #2 - A > B > C > D
16   PRAGMA EXCEPTION_INIT(e_grade_error_1, -20001); --associate the error code
17
18   e_grade_error_2 EXCEPTION; --Does not follows Business Rule #3 - Value 0-100
19   PRAGMA EXCEPTION_INIT(e_grade_error_2, -20002); --associate the error code
20
21
22  BEGIN
23    DBMS_OUTPUT.PUT_LINE('Attempt to insert a new course record.');
24
25    --Insert the new record
26    INSERT INTO courses VALUES(c_term, c_lineno, c_sno, c_A, c_B, c_C, c_D);
27
28    --Business Rule 1: term must be started with "F" (Fall), "sp" (Spring), "s" (Summer)
29    --And followed by 2 digits indicating year
30    IF (c_term NOT LIKE 'F%' OR c_term NOT LIKE 'sp%' OR c_term NOT LIKE 's%') THEN
31      --raise the EXCEPTION;
32      RAISE e_term_error;
33    END IF;
34
35    --Business Rule 2: Value of Grade A must bigger than Grade B (minimum gap 10 points)
36    --Same for other grades subsequently
37    --Grade values must be in the range of 0-100
38    IF (c_A <= 0 OR c_A > 100) OR (c_B <= 0 OR c_B > 100) OR (c_C <= 0 OR c_C > 100) OR (c_D <= 0 OR c_D > 100) THEN
39      --raise the EXCEPTION;
40      RAISE e_grade_error_1;
41    END IF;
42
43    IF (c_A < 0 OR c_A > 100) OR (c_B < 0 OR c_B > 100) OR (c_C < 0 OR c_C > 100) OR (c_D < 0 OR c_D > 100) THEN
44      --raise the EXCEPTION;
45      RAISE e_grade_error_2;
46    END IF;
47
48    DBMS_OUTPUT.PUT_LINE('New course record added!');
49
50    COMMIT; --Save the record
51
52  EXCEPTION
53  WHEN e_term_error THEN --Violate Business Rule
54    DBMS_OUTPUT.PUT_LINE('Attempt failed.');
55    DBMS_OUTPUT.PUT_LINE('Error code: ' || SQLCODE);
56    DBMS_OUTPUT.PUT_LINE('The term format is invalid, it should started with "F"(Fall), "sp"(Spring), "s"(Summer) and followed by 2 digits for year indication. Eg:F96');
57    ROLLBACK;
58
59  WHEN e_grade_error_1 THEN --Violate Business Rule
60    DBMS_OUTPUT.PUT_LINE('Attempt failed.');
61    DBMS_OUTPUT.PUT_LINE('Error code: ' || SQLCODE);
62    DBMS_OUTPUT.PUT_LINE('The value for higher grade should be more than the next grade by at least 10 points. Eg: Grade A: 90 points, Grade B: 80 points, Grade C: 70 points, Grade D: 60 points');
63    ROLLBACK;
64
65  WHEN e_grade_error_2 THEN --Violate Business Rule
66    DBMS_OUTPUT.PUT_LINE('Attempt failed.');
67    DBMS_OUTPUT.PUT_LINE('Error code: ' || SQLCODE);
68    DBMS_OUTPUT.PUT_LINE('The grade values must be in the range of 0-100. Eg: Grade A: 95 points');
69    ROLLBACK;
70
71  WHEN DUP_VAL_ON_INDEX THEN --Inserting Same Record
72    DBMS_OUTPUT.PUT_LINE('Attempt failed.');
73    DBMS_OUTPUT.PUT_LINE('The record for term #' || c_term || ' and line no of '||'s' || c_lineno || ' is found in the database!');
74    DBMS_OUTPUT.PUT_LINE('Error: ' || SQLCODE);
75    DBMS_OUTPUT.PUT_LINE('Message: ' || SQLERRM);
76    ROLLBACK;
77
78  END;
79 /
80
81  Procedure ADD.Course compiled
82
83
84  SQL> @ c:\sql\asm_task2run
85  SQL> ACCEPT c_term CHAR prompt 'Please enter Term (Eg:sp07): ';
86  Please enter Term (Eg:sp07): sp07
87  SQL> ACCEPT c_lineno NUMBER prompt 'Please enter Line No (Eg:1031): ';
88  Please enter Line No (Eg:1031): 1032
89  SQL> ACCEPT c_sno CHAR prompt 'Please enter Subject No (Eg:csc227): ';
90  Please enter Subject No (Eg:csc227): csc408
91  SQL> ACCEPT c_A NUMBER prompt 'Please enter Grade A Values (Between 0-100): ';
92  Please enter Grade A Values (Between 0-100): 95
93  SQL> ACCEPT c_B NUMBER prompt 'Please enter Grade B Values (Between 0-100): ';
94  Please enter Grade B Values (Between 0-100): 85
95  SQL> ACCEPT c_C NUMBER prompt 'Please enter Grade C Values (Between 0-100): ';
96  Please enter Grade C Values (Between 0-100): 65
97  SQL> ACCEPT c_D NUMBER prompt 'Please enter Grade D Values (Between 0-100): ';
98  Please enter Grade D Values (Between 0-100): 45
99
100
101  SQL> BEGIN
102    2    add_course('&c_term', '&c_lineno', '&c_sno', '&c_A', '&c_B', '&c_C', '&c_D');
103    3  END;
104
105  old:BEGIN
106    add_course('&c_term', '&c_lineno', '&c_sno', '&c_A', '&c_B', '&c_C', '&c_D');
107  END;
108
109  new:BEGIN
110    add_course('F03', '1032', 'csc408', '95', '85', '65', '45');
111  END;
112
113  Attempt to insert a new course record.
114  New course record added!
115
116  PL/SQL procedure successfully completed.
117
118
119  SQL> SELECT * FROM courses;
120
121
122  SQL> SELECT * FROM courses;
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
559
560
561
562
563
564
565
566
567
568
569
569
570
571
572
573
574
575
576
577
578
579
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
617
618
619
619
620
621
622
623
623
624
625
625
626
626
627
627
628
628
629
629
630
630
631
631
632
632
633
633
634
634
635
635
636
636
637
637
638
638
639
639
640
640
641
641
642
642
643
643
644
644
645
645
646
646
647
647
648
648
649
649
650
650
651
651
652
652
653
653
654
654
655
655
656
656
657
657
658
658
659
659
660
660
661
661
662
662
663
663
664
664
665
665
666
666
667
667
668
668
669
669
670
670
671
671
672
672
673
673
674
674
675
675
676
676
677
677
678
678
679
679
680
680
681
681
682
682
683
683
684
684
685
685
686
686
687
687
688
688
689
689
690
690
691
691
692
692
693
693
694
694
695
695
696
696
697
697
698
698
699
699
700
700
701
701
702
702
703
703
704
704
705
705
706
706
707
707
708
708
709
709
710
710
711
711
712
712
713
713
714
714
715
715
716
716
717
717
718
718
719
719
720
720
721
721
722
722
723
723
724
724
725
725
726
726
727
727
728
728
729
729
730
730
731
731
732
732
733
733
734
734
735
735
736
736
737
737
738
738
739
739
740
740
741
741
742
742
743
743
744
744
745
745
746
746
747
747
748
748
749
749
750
750
751
751
752
752
753
753
754
754
755
755
756
756
757
757
758
758
759
759
760
760
761
761
762
762
763
763
764
764
765
765
766
766
767
767
768
768
769
769
770
770
771
771
772
772
773
773
774
774
775
775
776
776
777
777
778
778
779
779
780
780
781
781
782
782
783
783
784
784
785
785
786
786
787
787
788
788
789
789
790
790
791
791
792
792
793
793
794
794
795
795
796
796
797
797
798
798
799
799
800
800
801
801
802
802
803
803
804
804
805
805
806
806
807
807
808
808
809
809
810
810
811
811
812
812
813
813
814
814
815
815
816
816
817
817
818
818
819
819
820
820
821
821
822
822
823
823
824
824
825
825
826
826
827
827
828
828
829
829
830
830
831
831
832
832
833
833
834
834
835
835
836
836
837
837
838
838
839
839
840
840
841
841
842
842
843
843
844
844
845
845
846
846
847
847
848
848
849
849
850
850
851
851
852
852
853
853
854
854
855
855
856
856
857
857
858
858
859
859
860
860
861
861
862
862
863
863
864
864
865
865
866
866
867
867
868
868
869
869
870
870
871
871
872
872
873
873
874
874
875
875
876
876
877
877
878
878
879
879
880
880
881
881
882
882
883
883
884
884
885
885
886
886
887
887
888
888
889
889
890
890
891
891
892
892
893
893
894
894
895
895
896
896
897
897
898
898
899
899
900
900
901
901
902
902
903
903
904
904
905
905
906
906
907
907
908
908
909
909
910
910
911
911
912
912
913
913
914
914
915
915
916
916
917
917
918
918
919
919
920
920
921
921
922
922
923
923
924
924
925
925
926
926
927
927
928
928
929
929
930
930
931
931
932
932
933
933
934
934
935
935
936
936
937
937
938
938
939
939
940
940
941
941
942
942
943
943
944
944
945
945
946
946
947
947
948
948
949
949
950
950
951
951
952
952
953
953
954
954
955
955
956
956
957
957
958
958
959
959
960
960
961
961
962
962
963
963
964
964
965
965
966
966
967
967
968
968
969
969
970
970
971
971
972
972
973
973
974
974
975
975
976
976
977
977
978
978
979
979
980
980
981
981
982
982
983
983
984
984
985
985
986
986
987
987
988
988
989
989
990
990
991
991
992
992
993
993
994
994
995
995
996
996
997
997
998
998
999
999
1000
1000
1001
1001
1002
1002
1003
1003
1004
1004
1005
1005
1006
1006
1007
1007
1008
1008
1009
1009
1010
1010
1011
1011
1012
1012
1013
1013
1014
1014
1015
1015
1016
1016
1017
1017
1018
1018
1019
1019
1020
1020
1021
1021
1022
1022
1023
1023
1024
1024
1025
1025
1026
1026
1027
1027
1028
1028
1029
1029
1030
1030
1031
1031
1032
1032
1033
1033
1034
1034
1035
1035
1036
1036
1037
1037
1038
1038
1039
1039
1040
1040
1041
1041
1042
1042
1043
1043
1044
1044
1045
1045
1046
1046
1047
1047
1048
1048
1049
1049
1050
1050
1051
1051
1052
1052
1053
1053
1054
1054
1055
1055
1056
1056
1057
1057
1058
1058
1059
1059
1060
1060
1061
1061
1062
1062
1063
1063
1064
1064
1065
1065
1066
1066
1067
1067
1068
1068
1069
1069
1070
1070
1071
1071
1072
1072
1073
1073
1074
1074
1075
1075
1076
1076
1077
1077
1078
1078
1079
1079
1080
1080
1081
1081
1082
1082
1083
1083
1084
1084
1085
1085
1086
1086
1087
1087
1088
1088
1089
1089
1090
1090
1091
1091
1092
1092
1093
1093
1094
1094
1095
1095
1096
1096
1097
1097
1098
1098
1099
1099
1100
1100
1101
1101
1102
1102
1103
1103
1104
1104
1105
1105
1106
1106
1107
1107
1108
1108
1109
1109
1110
1110
1111
1111
1112
1112
1113
1113
1114
1114
1115
1115
1116
1116
1117
1117
1118
1118
1119
1119
1120
1120
1121
1121
1122
1122
1123
1123
1124
1124
1125
1125
1126
1126
1127
1127
1128
1128
1129
1129
1130
1130
1131
1131
1132
1132
1133
1133
1134
1134
1135
1135
1136
1136
1137
1137
1138
1138
1139
1139
1140
1140
1141
1141
1142
1142
1143
1143
1144
1144
1145
1145
1146
1146
1147
1147
1148
1148
1149
1149
1150
1150
1151
1151
1152
1152
1153
1153
1154
1154
1155
1155
1156
1156
1157
1157
1158
1158
1159
1159
1160
1160
1161
1161
1162
1162
1163
1163
1164
1164
1165
1165
1166
1166
1167
1167
1168
1168
1169
1169
1170
1170
1171
1171
1172
1172
1173
1173
1174
1174
1175
1175
1176
1176
1177
1177
1178
1178
1179
1179
1180
1180
1181
1181
1182
1182
1183
1183
1184
1184
1185
1185
1186
1186
1187
1187
1188
1188
1189
1189
1190
1190
1191
1191
1192
1192
1193
1193
1194
1194
1195
1195
1196
1196
1197
1197
1198
1198
1199
1199
1200
1200
1201
1201
1202
1202
1203
1203
1204
1204
1205
1205
1206
1206
1207
1207
1208
1208
1209
1209
1210
1210
1211
1211
1212
1212
1213
1213
1214
1214
1215
1215
1216
1216
1217
1217
1218
1218
1219
1219
1220
1220
1221
1221
1222
1222
1223
1223
1224
1224
1225
1225
1226
1226
1227
1227
1228
1228
1229
1229
1230
1230
1231
1231
1232
1232
1233
1233
1234
1234
1235
1235
1236
1236
1237
1237
1238
1238
1239
1239
1240
1240
1241
1241
1242
1242
1243
1243
1244
1244
1245
1245
1246
1246
1247
1247
1248
1248
1249
1249
1250
1250
1251
1251
1252
1252
1253
1253
1254
1254
1255
1255
1256
1256
1257
1257
1258
1258
1259
1259
1260
1260
1261
1261
1262
1262
1263
1263
1264
1264
1265
1265
1266
1266
1267
1267
1268
1268
1269
1269
1270
1270
1271
1271
1272
1272
1273
1273
1274
1274
1275
1275
1276
1276
1277
1277
1278
1278
1279
1279
1280
1280
1281
1281
1282
1282
1283
1283
1284
1284
1285
1285
1286
1286
1287
1287
1288
1288
1289
1289
1290
1290
1291
1291
1292
1292
1293
1293
1294
1294
1295
1295
1296
1296
1297
1297
1298
1298
1299
1299
1300
1300
1301
1301
1302
1302
1303
1303
1304
1304
1305
1305
1306
1306
1307
1307
1308
1308
1309
1309
1310
1310
1311
1311
1312
1312
1313
1313
1314
1314
1315
1315
1316
1316
1317
1317
1318
1318
1319
1319
1320
1320
1321
1321
1322
1322
1323
1323
1324
1324
1325
1325
1326
1326
1327
1327
1328
1328
1329
1329
1330
1330
1331
1331
1332
1332
1333
1333
1334
1334
1335
1335
1336
1336
1337
1337
1338
1338
1339
1339
1340
1340
1341
1341
1342
1342
1343
1343
1344
1344
1345
1345
1346
1346
1347
1347
1348
1348
1349
1349
1350
1350
1351
1351
1352
1352
1353
1353
1354
1354
1355
1355
1356
1356
1357
1357
1358
1358
1359
1359
1360
1360
1361
1361
1362
1362
1363
1363
1364
1364
1365
1365
1366
1366
1367
1367
1368
1368
1369
1369
1370
1370
1371
1371
1372
1372
1373
1373
1374
1374
1375
1375
1376
1376
1377
1377
1378
1378
1379
1379
1380
1380
1381
1381
1382
1382
1383
1383
1384
1384
1385
1385
1386
1386
1387
1387
1388
1388
1389
1389
1390
1390
1391
1391
1392
1392
1393
1393
1394
1394
1395
1395
1396
1396
1397
1397
1398
1398
1399
1399
1400
1400
1401
1401
1402
1402
1403
1403
1404
1404
1405
1405
1406
1406
1407
1407
1408
1408
1409
1409
1410
1410
1411
1411
1412
1412
1413
1413
1414
1414
1415
1415
1416
1416
1417
1417
1418
1418
1419
1419
1420
1420
1421
1421
1422
1422
1423
1423
1424
1424
1425
1425
1426
1426
1427
1427
1428
1428
1429
1429
1430
1430
1431
1431
1432
1432
1433
1433
1434
1434
1435
1435
1436
1436
1437
1437
1438
1438
1439
1439
1440
1440
1441
1441
1442
1
```

### **3.4 Screenshots and Test Cases**

```

SQL> SET ECHO ON
SQL> SET FEEDBACK ON
SQL> SET PAGESIZE 100
SQL> SET SERVEROUTPUT ON
SQL> @c:\val\use_taska
SQL> SELECT *
  2   FROM (
  3     SELECT c.sno AS "Subject No",
  4           c.title AS "Subject Title",
  5           REPLACE(REPLACE(REPLACE(c.term, 'f', 'Fall-'), 'sp', 'Spring-'), 's', 'Summer-') AS Term,
  6           INITCAP(c.com_name) AS "Component Name",
  7           com.mapoints AS Mapoints,
  8           com.weight AS Weight
  9        FROM courses c
 10      JOIN subject s ON c.sno = s.sno
 11      JOIN components com ON com.term = c.term AND com.lineno = c.lineno
 12      WHERE c.term = 'Term' AND com.weight >= 'Min_weight'
 13    )
 14
 15 Enter value for term: F96
 16 Enter value for min_weight: 40
 17 old SELECT *
 18   2   FROM (
 19     3     SELECT c.sno AS "Subject No",
 20           c.title AS "Subject Title",
 21           REPLACE(REPLACE(REPLACE(c.term, 'f', 'Fall-'), 'sp', 'Spring-'), 's', 'Summer-') AS Term,
 22           INITCAP(c.com_name) AS "Component Name",
 23           com.mapoints AS Mapoints,
 24           com.weight AS Weight
 25        FROM courses c
 26      JOIN subject s ON c.sno = s.sno
 27      JOIN components com ON com.term = c.term AND com.lineno = c.lineno
 28      WHERE c.term = 'F96' AND com.weight >= '40'
 29    )
 30
 31 new SELECT *
 32   2   FROM (
 33     3     SELECT c.sno AS "Subject No",
 34           c.title AS "Subject Title",
 35           REPLACE(REPLACE(REPLACE(c.term, 'f', 'Fall-'), 'sp', 'Spring-'), 's', 'Summer-') AS Term,
 36           INITCAP(c.com_name) AS "Component Name",
 37           com.mapoints AS Mapoints,
 38           com.weight AS Weight
 39        FROM courses c
 40      JOIN subject s ON c.sno = s.sno
 41      JOIN components com ON com.term = c.term AND com.lineno = c.lineno
 42      WHERE c.term = 'F96' AND com.weight >= '40'
 43    )
 44
 45 Subject No Subject Title TERM Component Name MAPPOINTS WEIGHT
 46
 47 csc26 INTRODUCTION TO PROGRAMMING I Fall-96 Final 100 50
 48 csc26 INTRODUCTION TO PROGRAMMING I Fall-96 Final 100 40
 49 csc26 INTRODUCTION TO PROGRAMMING I Fall-96 Programs 400 40
 50
 51 3 rows selected.
 52
 53
 54 SQL> @c:\val\use_taskb 'F96' '303'
 55
 56 SQL> CREATE OR REPLACE VIEW student_course_average AS
 57 2  SELECT *
 58 3   FROM (
 59    4     SELECT e.sno AS "Line No",
 60          e.sid AS "Student ID",
 61          s.lname AS "Last Name",
 62          s.fname AS "First Name",
 63          (ROUND((SUM(sc.points * com.weight)/SUM(com.weight+com.mapoints))*100)) AS "Course Average"
 64        FROM enrolle e
 65      JOIN students s ON e.sid = s.sid
 66      JOIN components com ON e.term = com.term AND e.lineno = sc.lineno AND e.sid = sc.sid
 67      JOIN course c ON e.term = com.term AND e.lineno = com.lineno AND sc.comname = com.comname
 68      WHERE e.term = 'A1' AND e.lineno = '82'
 69    GROUP BY e.term, e.lineno, e.sid, s.lname, s.fname
 70  )
 71
 72 old CREATE OR REPLACE VIEW student_course_average AS
 73
 74 2  SELECT *
 75 3   FROM (
 76    4     REPLACE(REPLACE(REPLACE(e.term, 'f', 'Fall-'), 'sp', 'Spring-'), 's', 'Summer-') AS Term,
 77    5       l.sname AS "Line No",
 78    6       e.sid AS "Student ID",
 79    7       s.lname AS "Last Name",
 80    8       s.fname AS "First Name",
 81    9       (ROUND((SUM(sc.points * com.weight)/SUM(com.weight+com.mapoints))*100)) AS "Course Average"
 82  FROM enrolle e
 83  JOIN students s ON e.sid = s.sid
 84  JOIN courses c ON e.term = com.term AND e.lineno = sc.lineno AND e.sid = sc.sid
 85  JOIN components com ON e.term = com.term AND e.lineno = com.lineno AND sc.comname = com.comname
 86  WHERE e.term = 'F96' AND e.lineno = '103'
 87  GROUP BY e.term, e.lineno, e.sid, s.lname, s.fname
 88
 89 View STUDENT_COURSE_AVERAGE created.
 90
 91 SQL> SELECT * FROM student_course_average
 92
 93 TERM Line No Student ID Last Name First Name Course Average
 94
 95 Fall-96 1031 4444 Rajeshkhan Neven 79
 96 Fall-96 1031 2223 Corp Sydney 73
 97 Fall-96 1031 1111 Rajeshkhan Nandita 53
 98
 99 3 rows selected.
100
101 SQL> SELECT * FROM courses;
102
103 TERM LINENO SNO A B C D
104
105 F96 1032 csc498 95 85 65 45
106 F96 1032 csc26 90 80 65 50
107 F96 1032 csc26 90 80 65 50
108 F96 1031 csc27 90 80 65 50
109
110 4 rows selected.
111
112 SQL> @c:\val\use_task2
113
114 SQL> CREATE OR REPLACE PROCEDURE add_course
115 2   (c_term IN VARCHAR2, c_lineno IN NUMBER, c_sno IN VARCHAR2,
116 3   c_d IN NUMBER, c_a IN NUMBER, c_b IN NUMBER, c_c IN NUMBER)
117 AS
118
119   BEGIN
120     IF c_term NOT LIKE 'Term%' THEN
121       RAISE_APPLICATION_ERROR(-20001, 'Term must start with Term');
122     END IF;
123
124     IF c_lineno < 1000 OR c_lineno > 10000 THEN
125       RAISE_APPLICATION_ERROR(-20002, 'Line number must be between 1000 and 10000');
126     END IF;
127
128     IF c_sno NOT LIKE 'S%' THEN
129       RAISE_APPLICATION_ERROR(-20003, 'Student number must start with S');
130     END IF;
131
132     IF c_d < 0 OR c_d > 4 THEN
133       RAISE_APPLICATION_ERROR(-20004, 'Degree must be between 0 and 4');
134     END IF;
135
136     IF c_a < 0 OR c_a > 100 THEN
137       RAISE_APPLICATION_ERROR(-20005, 'A must be between 0 and 100');
138     END IF;
139
140     IF c_b < 0 OR c_b > 100 THEN
141       RAISE_APPLICATION_ERROR(-20006, 'B must be between 0 and 100');
142     END IF;
143
144     IF c_c < 0 OR c_c > 100 THEN
145       RAISE_APPLICATION_ERROR(-20007, 'C must be between 0 and 100');
146     END IF;
147
148     IF c_a + c_b + c_c > 100 THEN
149       RAISE_APPLICATION_ERROR(-20008, 'Sum of A, B, C must be less than or equal to 100');
150     END IF;
151
152     IF c_a < 0 OR c_a > 100 THEN
153       RAISE_APPLICATION_ERROR(-20009, 'Grade A must be between 0 and 100');
154     END IF;
155
156     IF c_b < 0 OR c_b > 100 THEN
157       RAISE_APPLICATION_ERROR(-20010, 'Grade B must be between 0 and 100');
158     END IF;
159
160     IF c_c < 0 OR c_c > 100 THEN
161       RAISE_APPLICATION_ERROR(-20011, 'Grade C must be between 0 and 100');
162     END IF;
163
164     IF c_d < 0 OR c_d > 4 THEN
165       RAISE_APPLICATION_ERROR(-20012, 'Degree must be between 0 and 4');
166     END IF;
167
168     IF c_sno NOT LIKE 'S%' THEN
169       RAISE_APPLICATION_ERROR(-20013, 'Student number must start with S');
170     END IF;
171
172     IF c_lineno < 1000 OR c_lineno > 10000 THEN
173       RAISE_APPLICATION_ERROR(-20014, 'Line number must be between 1000 and 10000');
174     END IF;
175
176     IF c_term NOT LIKE 'Term%' THEN
177       RAISE_APPLICATION_ERROR(-20015, 'Term must start with Term');
178     END IF;
179
180     IF c_d < 0 OR c_d > 4 THEN
181       RAISE_APPLICATION_ERROR(-20016, 'Degree must be between 0 and 4');
182     END IF;
183
184     IF c_a < 0 OR c_a > 100 THEN
185       RAISE_APPLICATION_ERROR(-20017, 'A must be between 0 and 100');
186     END IF;
187
188     IF c_b < 0 OR c_b > 100 THEN
189       RAISE_APPLICATION_ERROR(-20018, 'B must be between 0 and 100');
190     END IF;
191
192     IF c_c < 0 OR c_c > 100 THEN
193       RAISE_APPLICATION_ERROR(-20019, 'C must be between 0 and 100');
194     END IF;
195
196     IF c_a + c_b + c_c > 100 THEN
197       RAISE_APPLICATION_ERROR(-20020, 'Sum of A, B, C must be less than or equal to 100');
198     END IF;
199
200     IF c_a < 0 OR c_a > 100 THEN
201       RAISE_APPLICATION_ERROR(-20021, 'Grade A must be between 0 and 100');
202     END IF;
203
204     IF c_b < 0 OR c_b > 100 THEN
205       RAISE_APPLICATION_ERROR(-20022, 'Grade B must be between 0 and 100');
206     END IF;
207
208     IF c_c < 0 OR c_c > 100 THEN
209       RAISE_APPLICATION_ERROR(-20023, 'Grade C must be between 0 and 100');
210     END IF;
211
212     IF c_d < 0 OR c_d > 4 THEN
213       RAISE_APPLICATION_ERROR(-20024, 'Degree must be between 0 and 4');
214     END IF;
215
216     IF c_sno NOT LIKE 'S%' THEN
217       RAISE_APPLICATION_ERROR(-20025, 'Student number must start with S');
218     END IF;
219
220     IF c_lineno < 1000 OR c_lineno > 10000 THEN
221       RAISE_APPLICATION_ERROR(-20026, 'Line number must be between 1000 and 10000');
222     END IF;
223
224     IF c_term NOT LIKE 'Term%' THEN
225       RAISE_APPLICATION_ERROR(-20027, 'Term must start with Term');
226     END IF;
227
228     IF c_d < 0 OR c_d > 4 THEN
229       RAISE_APPLICATION_ERROR(-20028, 'Degree must be between 0 and 4');
230     END IF;
231
232     IF c_a < 0 OR c_a > 100 THEN
233       RAISE_APPLICATION_ERROR(-20029, 'A must be between 0 and 100');
234     END IF;
235
236     IF c_b < 0 OR c_b > 100 THEN
237       RAISE_APPLICATION_ERROR(-20030, 'B must be between 0 and 100');
238     END IF;
239
240     IF c_c < 0 OR c_c > 100 THEN
241       RAISE_APPLICATION_ERROR(-20031, 'C must be between 0 and 100');
242     END IF;
243
244     IF c_a + c_b + c_c > 100 THEN
245       RAISE_APPLICATION_ERROR(-20032, 'Sum of A, B, C must be less than or equal to 100');
246     END IF;
247
248     IF c_a < 0 OR c_a > 100 THEN
249       RAISE_APPLICATION_ERROR(-20033, 'Grade A must be between 0 and 100');
250     END IF;
251
252     IF c_b < 0 OR c_b > 100 THEN
253       RAISE_APPLICATION_ERROR(-20034, 'Grade B must be between 0 and 100');
254     END IF;
255
256     IF c_c < 0 OR c_c > 100 THEN
257       RAISE_APPLICATION_ERROR(-20035, 'Grade C must be between 0 and 100');
258     END IF;
259
260     IF c_d < 0 OR c_d > 4 THEN
261       RAISE_APPLICATION_ERROR(-20036, 'Degree must be between 0 and 4');
262     END IF;
263
264     IF c_sno NOT LIKE 'S%' THEN
265       RAISE_APPLICATION_ERROR(-20037, 'Student number must start with S');
266     END IF;
267
268     IF c_lineno < 1000 OR c_lineno > 10000 THEN
269       RAISE_APPLICATION_ERROR(-20038, 'Line number must be between 1000 and 10000');
270     END IF;
271
272     IF c_term NOT LIKE 'Term%' THEN
273       RAISE_APPLICATION_ERROR(-20039, 'Term must start with Term');
274     END IF;
275
276     IF c_d < 0 OR c_d > 4 THEN
277       RAISE_APPLICATION_ERROR(-20040, 'Degree must be between 0 and 4');
278     END IF;
279
280     IF c_a < 0 OR c_a > 100 THEN
281       RAISE_APPLICATION_ERROR(-20041, 'A must be between 0 and 100');
282     END IF;
283
284     IF c_b < 0 OR c_b > 100 THEN
285       RAISE_APPLICATION_ERROR(-20042, 'B must be between 0 and 100');
286     END IF;
287
288     IF c_c < 0 OR c_c > 100 THEN
289       RAISE_APPLICATION_ERROR(-20043, 'C must be between 0 and 100');
290     END IF;
291
292     IF c_a + c_b + c_c > 100 THEN
293       RAISE_APPLICATION_ERROR(-20044, 'Sum of A, B, C must be less than or equal to 100');
294     END IF;
295
296     IF c_a < 0 OR c_a > 100 THEN
297       RAISE_APPLICATION_ERROR(-20045, 'Grade A must be between 0 and 100');
298     END IF;
299
300     IF c_b < 0 OR c_b > 100 THEN
301       RAISE_APPLICATION_ERROR(-20046, 'Grade B must be between 0 and 100');
302     END IF;
303
304     IF c_c < 0 OR c_c > 100 THEN
305       RAISE_APPLICATION_ERROR(-20047, 'Grade C must be between 0 and 100');
306     END IF;
307
308     IF c_d < 0 OR c_d > 4 THEN
309       RAISE_APPLICATION_ERROR(-20048, 'Degree must be between 0 and 4');
310     END IF;
311
312     IF c_sno NOT LIKE 'S%' THEN
313       RAISE_APPLICATION_ERROR(-20049, 'Student number must start with S');
314     END IF;
315
316     IF c_lineno < 1000 OR c_lineno > 10000 THEN
317       RAISE_APPLICATION_ERROR(-20050, 'Line number must be between 1000 and 10000');
318     END IF;
319
320     IF c_term NOT LIKE 'Term%' THEN
321       RAISE_APPLICATION_ERROR(-20051, 'Term must start with Term');
322     END IF;
323
324     IF c_d < 0 OR c_d > 4 THEN
325       RAISE_APPLICATION_ERROR(-20052, 'Degree must be between 0 and 4');
326     END IF;
327
328     IF c_a < 0 OR c_a > 100 THEN
329       RAISE_APPLICATION_ERROR(-20053, 'A must be between 0 and 100');
330     END IF;
331
332     IF c_b < 0 OR c_b > 100 THEN
333       RAISE_APPLICATION_ERROR(-20054, 'B must be between 0 and 100');
334     END IF;
335
336     IF c_c < 0 OR c_c > 100 THEN
337       RAISE_APPLICATION_ERROR(-20055, 'C must be between 0 and 100');
338     END IF;
339
340     IF c_a + c_b + c_c > 100 THEN
341       RAISE_APPLICATION_ERROR(-20056, 'Sum of A, B, C must be less than or equal to 100');
342     END IF;
343
344     IF c_a < 0 OR c_a > 100 THEN
345       RAISE_APPLICATION_ERROR(-20057, 'Grade A must be between 0 and 100');
346     END IF;
347
348     IF c_b < 0 OR c_b > 100 THEN
349       RAISE_APPLICATION_ERROR(-20058, 'Grade B must be between 0 and 100');
350     END IF;
351
352     IF c_c < 0 OR c_c > 100 THEN
353       RAISE_APPLICATION_ERROR(-20059, 'Grade C must be between 0 and 100');
354     END IF;
355
356     IF c_d < 0 OR c_d > 4 THEN
357       RAISE_APPLICATION_ERROR(-20060, 'Degree must be between 0 and 4');
358     END IF;
359
360     IF c_sno NOT LIKE 'S%' THEN
361       RAISE_APPLICATION_ERROR(-20061, 'Student number must start with S');
362     END IF;
363
364     IF c_lineno < 1000 OR c_lineno > 10000 THEN
365       RAISE_APPLICATION_ERROR(-20062, 'Line number must be between 1000 and 10000');
366     END IF;
367
368     IF c_term NOT LIKE 'Term%' THEN
369       RAISE_APPLICATION_ERROR(-20063, 'Term must start with Term');
370     END IF;
371
372     IF c_d < 0 OR c_d > 4 THEN
373       RAISE_APPLICATION_ERROR(-20064, 'Degree must be between 0 and 4');
374     END IF;
375
376     IF c_a < 0 OR c_a > 100 THEN
377       RAISE_APPLICATION_ERROR(-20065, 'A must be between 0 and 100');
378     END IF;
379
380     IF c_b < 0 OR c_b > 100 THEN
381       RAISE_APPLICATION_ERROR(-20066, 'B must be between 0 and 100');
382     END IF;
383
384     IF c_c < 0 OR c_c > 100 THEN
385       RAISE_APPLICATION_ERROR(-20067, 'C must be between 0 and 100');
386     END IF;
387
388     IF c_a + c_b + c_c > 100 THEN
389       RAISE_APPLICATION_ERROR(-20068, 'Sum of A, B, C must be less than or equal to 100');
390     END IF;
391
392     IF c_a < 0 OR c_a > 100 THEN
393       RAISE_APPLICATION_ERROR(-20069, 'Grade A must be between 0 and 100');
394     END IF;
395
396     IF c_b < 0 OR c_b > 100 THEN
397       RAISE_APPLICATION_ERROR(-20070, 'Grade B must be between 0 and 100');
398     END IF;
399
400     IF c_c < 0 OR c_c > 100 THEN
401       RAISE_APPLICATION_ERROR(-20071, 'Grade C must be between 0 and 100');
402     END IF;
403
404     IF c_d < 0 OR c_d > 4 THEN
405       RAISE_APPLICATION_ERROR(-20072, 'Degree must be between 0 and 4');
406     END IF;
407
408     IF c_sno NOT LIKE 'S%' THEN
409       RAISE_APPLICATION_ERROR(-20073, 'Student number must start with S');
410     END IF;
411
412     IF c_lineno < 1000 OR c_lineno > 10000 THEN
413       RAISE_APPLICATION_ERROR(-20074, 'Line number must be between 1000 and 10000');
414     END IF;
415
416     IF c_term NOT LIKE 'Term%' THEN
417       RAISE_APPLICATION_ERROR(-20075, 'Term must start with Term');
418     END IF;
419
420     IF c_d < 0 OR c_d > 4 THEN
421       RAISE_APPLICATION_ERROR(-20076, 'Degree must be between 0 and 4');
422     END IF;
423
424     IF c_a < 0 OR c_a > 100 THEN
425       RAISE_APPLICATION_ERROR(-20077, 'A must be between 0 and 100');
426     END IF;
427
428     IF c_b < 0 OR c_b > 100 THEN
429       RAISE_APPLICATION_ERROR(-20078, 'B must be between 0 and 100');
430     END IF;
431
432     IF c_c < 0 OR c_c > 100 THEN
433       RAISE_APPLICATION_ERROR(-20079, 'C must be between 0 and 100');
434     END IF;
435
436     IF c_a + c_b + c_c > 100 THEN
437       RAISE_APPLICATION_ERROR(-20080, 'Sum of A, B, C must be less than or equal to 100');
438     END IF;
439
440     IF c_a < 0 OR c_a > 100 THEN
441       RAISE_APPLICATION_ERROR(-20081, 'Grade A must be between 0 and 100');
442     END IF;
443
444     IF c_b < 0 OR c_b > 100 THEN
445       RAISE_APPLICATION_ERROR(-20082, 'Grade B must be between 0 and 100');
446     END IF;
447
448     IF c_c < 0 OR c_c > 100 THEN
449       RAISE_APPLICATION_ERROR(-20083, 'Grade C must be between 0 and 100');
450     END IF;
451
452     IF c_d < 0 OR c_d > 4 THEN
453       RAISE_APPLICATION_ERROR(-20084, 'Degree must be between 0 and 4');
454     END IF;
455
456     IF c_sno NOT LIKE 'S%' THEN
457       RAISE_APPLICATION_ERROR(-20085, 'Student number must start with S');
458     END IF;
459
460     IF c_lineno < 1000 OR c_lineno > 10000 THEN
461       RAISE_APPLICATION_ERROR(-20086, 'Line number must be between 1000 and 10000');
462     END IF;
463
464     IF c_term NOT LIKE 'Term%' THEN
465       RAISE_APPLICATION_ERROR(-20087, 'Term must start with Term');
466     END IF;
467
468     IF c_d < 0 OR c_d > 4 THEN
469       RAISE_APPLICATION_ERROR(-20088, 'Degree must be between 0 and 4');
470     END IF;
471
472     IF c_a < 0 OR c_a > 100 THEN
473       RAISE_APPLICATION_ERROR(-20089, 'A must be between 0 and 100');
474     END IF;
475
476     IF c_b < 0 OR c_b > 100 THEN
477       RAISE_APPLICATION_ERROR(-20090, 'B must be between 0 and 100');
478     END IF;
479
480     IF c_c < 0 OR c_c > 100 THEN
481       RAISE_APPLICATION_ERROR(-20091, 'C must be between 0 and 100');
482     END IF;
483
484     IF c_a + c_b + c_c > 100 THEN
485       RAISE_APPLICATION_ERROR(-20092, 'Sum of A, B, C must be less than or equal to 100');
486     END IF;
487
488     IF c_a < 0 OR c_a > 100 THEN
489       RAISE_APPLICATION_ERROR(-20093, 'Grade A must be between 0 and 100');
490     END IF;
491
492     IF c_b < 0 OR c_b > 100 THEN
493       RAISE_APPLICATION_ERROR(-20094, 'Grade B must be between 0 and 100');
494     END IF;
495
496     IF c_c < 0 OR c_c > 100 THEN
497       RAISE_APPLICATION_ERROR(-20095, 'Grade C must be between 0 and 100');
498     END IF;
499
500     IF c_d < 0 OR c_d > 4 THEN
501       RAISE_APPLICATION_ERROR(-20096, 'Degree must be between 0 and 4');
502     END IF;
503
504     IF c_sno NOT LIKE 'S%' THEN
505       RAISE_APPLICATION_ERROR(-20097, 'Student number must start with S');
506     END IF;
507
508     IF c_lineno < 1000 OR c_lineno > 10000 THEN
509       RAISE_APPLICATION_ERROR(-20098, 'Line number must be between 1000 and 10000');
510     END IF;
511
512     IF c_term NOT LIKE 'Term%' THEN
513       RAISE_APPLICATION_ERROR(-20099, 'Term must start with Term');
514     END IF;
515
516     IF c_d < 0 OR c_d > 4 THEN
517       RAISE_APPLICATION_ERROR(-20100, 'Degree must be between 0 and 4');
518     END IF;
519
520     IF c_a < 0 OR c_a > 100 THEN
521       RAISE_APPLICATION_ERROR(-20101, 'A must be between 0 and 100');
522     END IF;
523
524     IF c_b < 0 OR c_b > 100 THEN
525       RAISE_APPLICATION_ERROR(-20102, 'B must be between 0 and 100');
526     END IF;
527
528     IF c_c < 0 OR c_c > 100 THEN
529       RAISE_APPLICATION_ERROR(-20103, 'C must be between 0 and 100');
530     END IF;
531
532     IF c_a + c_b + c_c > 100 THEN
533       RAISE_APPLICATION_ERROR(-20104, 'Sum of A, B, C must be less than or equal to 100');
534     END IF;
535
536     IF c_a < 0 OR c_a > 100 THEN
537       RAISE_APPLICATION_ERROR(-20105, 'Grade A must be between 0 and 100');
538     END IF;
539
540     IF c_b < 0 OR c_b > 100 THEN
541       RAISE_APPLICATION_ERROR(-20106, 'Grade B must be between 0 and 100');
542     END IF;
543
544     IF c_c < 0 OR c_c > 100 THEN
545       RAISE_APPLICATION_ERROR(-20107, 'Grade C must be between 0 and 100');
546     END IF;
547
548     IF c_d < 0 OR c_d > 4 THEN
549       RAISE_APPLICATION_ERROR(-20108, 'Degree must be between 0 and 4');
550     END IF;
551
552     IF c_sno NOT LIKE 'S%' THEN
553       RAISE_APPLICATION_ERROR(-20109, 'Student number must start with S');
554     END IF;
555
556     IF c_lineno < 1000 OR c_lineno > 10000 THEN
557       RAISE_APPLICATION_ERROR(-20110, 'Line number must be between 1000 and 10000');
558     END IF;
559
560     IF c_term NOT LIKE 'Term%' THEN
561       RAISE_APPLICATION_ERROR(-20111, 'Term must start with Term');
562     END IF;
563
564     IF c_d < 0 OR c_d > 4 THEN
565       RAISE_APPLICATION_ERROR(-20112, 'Degree must be between 0 and 4');
566     END IF;
567
568     IF c_a < 0 OR c_a > 100 THEN
569       RAISE_APPLICATION_ERROR(-20113, 'A must be between 0 and 100');
570     END IF;
571
572     IF c_b < 0 OR c_b > 100 THEN
573       RAISE_APPLICATION_ERROR(-20114, 'B must be between 0 and 100');
574     END IF;
575
576     IF c_c < 0 OR c_c > 100 THEN
577       RAISE_APPLICATION_ERROR(-20115, 'C must be between 0 and 100');
578     END IF;
579
580     IF c_a + c_b + c_c > 100 THEN
581       RAISE_APPLICATION_ERROR(-20116, 'Sum of A, B, C must be less than or equal to 100');
582     END IF;
583
584     IF c_a < 0 OR c_a > 100 THEN
585       RAISE_APPLICATION_ERROR(-20117, 'Grade A must be between 0 and 100');
586     END IF;
587
588     IF c_b < 0 OR c_b > 100 THEN
589       RAISE_APPLICATION_ERROR(-20118, 'Grade B must be between 0 and 100');
590     END IF;
591
592     IF c_c < 0 OR c_c > 100 THEN
593       RAISE_APPLICATION_ERROR(-20119, 'Grade C must be between 0 and 100');
594     END IF;
595
596     IF c_d < 0 OR c_d > 4 THEN
597       RAISE_APPLICATION_ERROR(-20120, 'Degree must be between 0 and 4');
598     END IF;
599
600     IF c_sno NOT LIKE 'S%' THEN
601       RAISE_APPLICATION_ERROR(-20121, 'Student number must start with S');
602     END IF;
603
604     IF c_lineno < 1000 OR c_lineno > 10000 THEN
605       RAISE_APPLICATION_ERROR(-20122, 'Line number must be between 1000 and 10000');
606     END IF;
607
608     IF c_term NOT LIKE 'Term%' THEN
609       RAISE_APPLICATION_ERROR(-20123, 'Term must start with Term');
610     END IF;
611
612     IF c_d < 0 OR c_d > 4 THEN
613       RAISE_APPLICATION_ERROR(-20124, 'Degree must be between 0 and 4');
614     END IF;
615
616     IF c_a < 0 OR c_a > 100 THEN
617       RAISE_APPLICATION_ERROR(-20125, 'A must be between 0 and 100');
618     END IF;
619
620     IF c_b < 0 OR c_b > 100 THEN
621       RAISE_APPLICATION_ERROR(-20126, 'B must be between 0 and 100');
622     END IF;
623
624     IF c_c < 0 OR c_c > 100 THEN
625       RAISE_APPLICATION_ERROR(-20127, 'C must be between 0 and 100');
626     END IF;
627
628     IF c_a + c_b + c_c > 100 THEN
629       RAISE_APPLICATION_ERROR(-20128, 'Sum of A, B, C must be less than or equal to 100');
630     END IF;
631
632     IF c_a < 0 OR c_a > 100 THEN
633       RAISE_APPLICATION_ERROR(-20129, 'Grade A must be between 0 and 100');
634     END IF;
635
636     IF c_b < 0 OR c_b > 100 THEN
637       RAISE_APPLICATION_ERROR(-20130, 'Grade B must be between 0 and 100');
638     END IF;
639
640     IF c_c < 0 OR c_c > 100 THEN
641       RAISE_APPLICATION_ERROR(-20131, 'Grade C must be between 0 and 100');
642     END IF;
643
644     IF c_d < 0 OR c_d > 4 THEN
645       RAISE_APPLICATION_ERROR(-20132, 'Degree must be between 0 and 4');
646     END IF;
647
648     IF c_sno NOT LIKE 'S%' THEN
649       RAISE_APPLICATION_ERROR(-20133, 'Student number must start with S');
650     END IF;
651
652     IF c_lineno < 1000 OR c_lineno > 10000 THEN
653       RAISE_APPLICATION_ERROR(-20134, 'Line number must be between 1000 and 10000');
654     END IF;
655
656     IF c_term NOT LIKE 'Term%' THEN
657       RAISE_APPLICATION_ERROR(-20135, 'Term must start with Term');
658     END IF;
659
660     IF c_d < 0 OR c_d > 4 THEN
661       RAISE_APPLICATION_ERROR(-20136, 'Degree must be between 0 and 4');
662     END IF;
663
664     IF c_a < 0 OR c_a > 100 THEN
665       RAISE_APPLICATION_ERROR(-20137, 'A must be between 0 and 100');
666     END IF;
667
668     IF c_b < 0 OR c_b > 100 THEN
669       RAISE_APPLICATION_ERROR(-20138, 'B must be between 0 and 100');
670     END IF;
671
672     IF c_c < 0 OR c_c > 100 THEN
673       RAISE_APPLICATION_ERROR(-20139, 'C must be between 0 and 100');
674     END IF;
675
676     IF c_a + c_b + c_c > 100 THEN
677       RAISE_APPLICATION_ERROR(-20140, 'Sum of A, B, C must be less than or equal to 100');
678     END IF;
679
680     IF c_a < 0 OR c_a > 100 THEN
681       RAISE_APPLICATION_ERROR(-20141, 'Grade A must be between 0 and 100');
682     END IF;
683
684     IF c_b < 0 OR c_b > 100 THEN
685       RAISE_APPLICATION_ERROR(-20142, 'Grade B must be between 0 and 100');
686     END IF;
687
688     IF c_c < 0 OR c_c > 100 THEN
689       RAISE_APPLICATION_ERROR(-20143, 'Grade C must be between 0 and 100');
690     END IF;
691
692     IF c_d < 0 OR c_d > 4 THEN
693       RAISE_APPLICATION_ERROR(-20144, 'Degree must be between 0 and 4');
694     END IF;
695
696     IF c_sno NOT LIKE 'S%' THEN
697       RAISE_APPLICATION_ERROR(-20145, 'Student number must start with S');
698     END IF;
699
700     IF c_lineno < 1000 OR c_lineno > 10000 THEN
701       RAISE_APPLICATION_ERROR(-20146, 'Line number must be between 1000 and 10000');
702     END IF;
703
704     IF c_term NOT LIKE 'Term%' THEN
705       RAISE_APPLICATION_ERROR(-20147, 'Term must start with Term');
706     END IF;
707
708     IF c_d < 0 OR c_d > 4 THEN
709       RAISE_APPLICATION_ERROR(-20148, 'Degree must be between 0 and 4');
710     END IF;
711
712     IF c_a < 0 OR c_a > 100 THEN
713       RAISE_APPLICATION_ERROR(-20149, 'A must be between 0 and 100');
714     END IF;
715
716     IF c_b < 0 OR c_b > 100 THEN
717       RAISE_APPLICATION_ERROR(-20150, 'B must be between 0 and 100');
718     END IF;
719
720     IF c_c < 0 OR c_c > 100 THEN
721       RAISE_APPLICATION_ERROR(-20151, 'C must be between 0 and 100');
722     END IF;
723
724     IF c_a + c_b + c_c > 100 THEN
725       RAISE_APPLICATION_ERROR(-20152, 'Sum of A, B, C must be less than or equal to 100');
726     END IF;
727
728     IF c_a < 0 OR c_a > 100 THEN
729       RAISE_APPLICATION_ERROR(-20153, 'Grade A must be between 0 and 100');
730     END IF;
731
732     IF c_b < 0 OR c_b > 100 THEN
733       RAISE_APPLICATION_ERROR(-20154, 'Grade B must be between 0 and 100');
734     END IF;
735
736     IF c_c < 0 OR c_c > 100 THEN
737       RAISE_APPLICATION_ERROR(-20155, 'Grade C must be between 0 and 100');
738     END IF;
739
740     IF c_d < 0 OR c_d > 4 THEN
741       RAISE_APPLICATION_ERROR(-20156, 'Degree must be between 0 and 4');
742     END IF;
743
744     IF c_sno NOT LIKE 'S%' THEN
745       RAISE_APPLICATION_ERROR(-20157, 'Student number must start with S');
746     END IF;
747
748     IF c_lineno < 1000 OR c_lineno > 10000 THEN
749       RAISE_APPLICATION_ERROR(-20158, 'Line number must be between 1000 and 10000');
750     END IF;
751
752     IF c_term NOT LIKE 'Term%' THEN
753       RAISE_APPLICATION_ERROR(-20159, 'Term must start with Term');
754     END IF;
755
756     IF c_d < 0 OR c_d > 4 THEN
757       RAISE_APPLICATION_ERROR(-20160, 'Degree must be between 0 and 4');
758     END IF;
759
760     IF c_a < 0 OR c_a > 100 THEN
761       RAISE_APPLICATION_ERROR(-20161, 'A must be between 0 and 100');
762     END IF;
763
764     IF c_b < 0 OR c_b > 100 THEN
765       RAISE_APPLICATION_ERROR(-20162, 'B must be between 0 and 100');
766     END IF;
767
768     IF c_c < 0 OR c_c > 100 THEN
769       RAISE_APPLICATION_ERROR(-20163, 'C must be between 0 and 100');
770     END IF;
771
772     IF c_a + c_b + c_c > 100 THEN
773       RAISE_APPLICATION_ERROR(-20164, 'Sum of A, B, C must be less than or equal to 100');
774     END IF;
775
776     IF c_a < 0 OR c_a > 100 THEN
777       RAISE_APPLICATION_ERROR(-20165, 'Grade A must be between 0 and 100');
778     END IF;
779
780     IF c_b < 0 OR c_b > 100 THEN
781       RAISE_APPLICATION_ERROR(-20166, 'Grade B must be between 0 and 100');
782     END IF;
783
784     IF c_c < 0 OR c_c > 100 THEN
785       RAISE_APPLICATION_ERROR(-20167, 'Grade C must be between 0 and 100');
786     END IF;
787
788     IF c_d < 0 OR c_d > 4 THEN
789       RAISE_APPLICATION_ERROR(-20168, 'Degree must be between 0 and 4');
790     END IF;
791
792     IF c_sno NOT LIKE 'S%' THEN
793       RAISE_APPLICATION_ERROR(-20169, 'Student number must start with S');
794     END IF;
795
796     IF c_lineno < 1000 OR c_lineno > 10000 THEN
797       RAISE_APPLICATION_ERROR(-20170, 'Line number must be between 1000 and 10000');
798     END IF;
799
800     IF c_term NOT LIKE 'Term%' THEN
801       RAISE_APPLICATION_ERROR(-20171, 'Term must start with Term');
802     END IF;
803
804     IF c_d < 0 OR c_d > 4 THEN
805       RAISE_APPLICATION_ERROR(-20172, 'Degree must be between 0 and 4');
806     END IF;
807
808     IF c_a < 0 OR c_a > 100 THEN
809       RAISE_APPLICATION_ERROR(-20173, 'A must be between 0 and 100');
810     END IF;
811
812     IF c_b < 0 OR c
```

```

130 18 BEGIN
131 19 DBMS_OUTPUT.PUT_LINE('Attempt to insert a new course record.');
132 20
133 21 ...Insert the new record
134 22 INSERT INTO courses VALUES(:c_term, :c_linenr, :c_sno, :c_A, :c_B, :c_C, :c_D);
135 23
136 24 --Business Rule 1: term must be started with "F" (Fall), "sp" (Spring), "s" (Summer)
137 25 --And followed by 2 digits indicating year
138 26 IF NOT REGEXP_LIKE(c_term, '^([F|sp|s])\d{2}$') THEN
139 27 --raise the EXCEPTION
140 28 RAISE e_term_error;
141 29 END IF;
142
143 30 --Business Rule 2: Value of Grade A must bigger than Grade B (minimum gap 10 points)
144 31 --Same for other grades subsequently
145 32 --Grade C must be in the range of 0-100
146 33 IF (c_A <= c_B) OR (c_B <= c_C) OR (c_C <= c_D) THEN
147 34 --raise the EXCEPTION
148 35 RAISE e_grade_error;
149 36 END IF;
150
151 37 --Business Rule 3: term must be started with "F" (Fall), "sp" (Spring), "s" (Summer)
152 38 --And followed by 2 digits indicating year
153 39 IF NOT REGEXP_LIKE(c_term, '^([F|sp|s])\d{2}$') THEN
154 40 --raise the EXCEPTION
155 41 RAISE e_term_error;
156 42 END IF;
157
158 43 DBMS_OUTPUT.PUT_LINE('New course record added!');
159 44 COMMIT; --Save the record
160 45
161 46 EXCEPTION
162 47 WHEN e_term_error THEN --Violate Business Rule
163 48 DBMS_OUTPUT.PUT_LINE('Attempt failed..');
164 49 DBMS_OUTPUT.PUT_LINE('Error code: ' || SQLCODE);
165 50 DBMS_OUTPUT.PUT_LINE('The term format is invalid, it should started with "F"(fall), "sp"(spring), "s"(Summer) and followed by 2 digits for year indication. Eg:F96');
166 51 ROLLBACK;
167
168 52 WHEN e_grade_error THEN --Violate Business Rule
169 53 DBMS_OUTPUT.PUT_LINE('Attempt failed..');
170 54 DBMS_OUTPUT.PUT_LINE('Error code: ' || SQLCODE);
171 55 DBMS_OUTPUT.PUT_LINE('The value for higher grade should be more than the next grade by at least 10 points. Eg: Grade A: 90 points, Grade B: 80 points, Grade C: 70 points, Grade D: 60 points');
172 56
173 57 ROLLBACK;
174
175 58 WHEN DUP_VAL_ON_INDEX THEN --Inserting Same Record
176 59 DBMS_OUTPUT.PUT_LINE('The record for term #' || c_term || ' and line no. of '||#'|| c_linenr || ' is found in the database!');
177 60 DBMS_OUTPUT.PUT_LINE('Attempt failed..');
178 61 DBMS_OUTPUT.PUT_LINE('Error code: ' || SQLCODE);
179 62 DBMS_OUTPUT.PUT_LINE('The grade values must be in the range of 0-100. Eg: Grade A: 95 points');
180 63 ROLLBACK;
181
182 64 WHEN DUP_VAL_ON_INDEX THEN --Violate Business Rule
183 65 DBMS_OUTPUT.PUT_LINE('Attempt failed..');
184 66 DBMS_OUTPUT.PUT_LINE('The record for term #' || c_term || ' and line no. of '||#'|| c_linenr || ' is found in the database!');
185 67 DBMS_OUTPUT.PUT_LINE('Attempt failed..');
186 68 DBMS_OUTPUT.PUT_LINE('Error code: ' || SQLCODE);
187 69 DBMS_OUTPUT.PUT_LINE('Message: ' || SQLERRM);
188 70 ROLLBACK;
189
190 71 --All error handlers
191 72 WHEN OTHERS THEN
192 73 DBMS_OUTPUT.PUT_LINE('Attempt failed..');
193 74 DBMS_OUTPUT.PUT_LINE('Error code: ' || SQLCODE);
194 75 DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM);
195 76 DBMS_OUTPUT.PUT_LINE('Message: ' || SQLERRM);
196 77 ROLLBACK; --Rollback until the transaction
197 78 END;
198 79 /
199
200 80
201 81 ┌ Procedure ADD_COURSE compiled
202 82
203 83 SQL # C:\sql\lson\task2run
204 84
205 85 SQL> ACCEPT c_term CHAR prompt 'Please enter Term (Eg:sp07): '
206 86 Please enter Term (Eg:sp07): f01
207 87 SQL> ACCEPT c_linenr NUMBER prompt 'Please enter Line No (Eg:1031): '
208 88 Please enter Line No (Eg:1031): 1031
209 89 SQL> ACCEPT c_sno NUMBER prompt 'Please enter Subject No (Eg:csc227): '
210 90 Please enter Subject No (Eg:csc227): exc081
211 91 SQL> ACCEPT c_A NUMBER prompt 'Please enter Grade A Values (Between 0-100): '
212 92 Please enter Grade A Values (Between 0-100): 80
213 93 SQL> ACCEPT c_B NUMBER prompt 'Please enter Grade B Values (Between 0-100): '
214 94 Please enter Grade B Values (Between 0-100): 70
215 95 SQL> ACCEPT c_C NUMBER prompt 'Please enter Grade C Values (Between 0-100): '
216 96 Please enter Grade C Values (Between 0-100): 70
217 97 SQL> ACCEPT c_D NUMBER prompt 'Please enter Grade D Values (Between 0-100): '
218 98 Please enter Grade D Values (Between 0-100): 60
219 99
220 100 BEGIN
221 101 2 add_course(:c_term, :c_linenr, :c_sno, :c_A, :c_B, :c_C, :c_D);
222 102
223 103 END;
224 104
225 105 ROLLBACK;
226
227 106 add_course(:c_term, :c_linenr, :c_sno, :c_A, :c_B, :c_C, :c_D);
228
229 107
230 108 new BEGIN
231 109 add_course('f01', '1031', 'exc081', '80', '70', '60', '65');
232 110
233 111 END;
234 112
235 113 Attempt to insert a new course record.
236 114 New course record added!
237
238
239 115 PL/SQL procedure successfully completed.
240
241
242 116 SQL> SELECT * FROM courses;
243
244 117
245 118
246 119
247 120
248 121
249 122
250 123
251 124
252 125
253 126
254 127
255 128
256 129
257 130
258 131
259 132
260 133
261 134
262 135
263 136
264 137
265 138
266 139
267 140
268 141
269 142
270 143
271 144
272 145
273 146
274 147
275 148
276 149
277 150
278 151
279 152
280 153
281 154
282 155
283 156
284 157
285 158
286 159
287 160
288 161
289 162
290 163
291 164
292 165
293 166
294 167
295 168
296 169
297 170
298 171
299 172
300 173
301 174
302 175
303 176
304 177
305 178
306 179
307 180
308 181
309 182
310 183
311 184
312 185
313 186
314 187
315 188
316 189
317 190
318 191
319 192
320 193
321 194
322 195
323 196
324 197
325 198
326 199
327 200
328 201
329 202
330 203
331 204
332 205
333 206
334 207
335 208
336 209
337 210
338 211
339 212
340 213
341 214
342 215
343 216
344 217
345 218
346 219
347 220
348 221
349 222
350 223
351 224
352 225
353 226
354 227
355 228
356 229
357 230
358 231
359 232
360 233
361 234
362 235
363 236
364 237
365 238
366 239
367 240
368 241
369 242
370 243
371 244
372 245
373 246
374 247
375 248
376 249
377 250
378 251
379 252
380 253
381 254
382 255
383 256
384 257
385 258
386 259
387 260
388 261
389 262
390 263
391 264
392 265
393 266
394 267
395 268
396 269
397 270
398 271
399 272
400 273
401 274
402 275
403 276
404 277
405 278
406 279
407 280
408 281
409 282
410 283
411 284
412 285
413 286
414 287
415 288
416 289
417 290
418 291
419 292
420 293
421 294
422 295
423 296
424 297
425 298
426 299
427 300
428 301
429 302
430 303
431 304
432 305
433 306
434 307
435 308
436 309
437 310
438 311
439 312
440 313
441 314
442 315
443 316
444 317
445 318
446 319
447 320
448 321
449 322
450 323
451 324
452 325
453 326
454 327
455 328
456 329
457 330
458 331
459 332
460 333
461 334
462 335
463 336
464 337
465 338
466 339
467 340
468 341
469 342
470 343
471 344
472 345
473 346
474 347
475 348
476 349
477 350
478 351
479 352
480 353
481 354
482 355
483 356
484 357
485 358
486 359
487 360
488 361
489 362
490 363
491 364
492 365
493 366
494 367
495 368
496 369
497 370
498 371
499 372
500 373
501 374
502 375
503 376
504 377
505 378
506 379
507 380
508 381
509 382
510 383
511 384
512 385
513 386
514 387
515 388
516 389
517 390
518 391
519 392
520 393
521 394
522 395
523 396
524 397
525 398
526 399
527 400
528 401
529 402
530 403
531 404
532 405
533 406
534 407
535 408
536 409
537 410
538 411
539 412
540 413
541 414
542 415
543 416
544 417
545 418
546 419
547 420
548 421
549 422
550 423
551 424
552 425
553 426
554 427
555 428
556 429
557 430
558 431
559 432
560 433
561 434
562 435
563 436
564 437
565 438
566 439
567 440
568 441
569 442
570 443
571 444
572 445
573 446
574 447
575 448
576 449
577 450
578 451
579 452
580 453
581 454
582 455
583 456
584 457
585 458
586 459
587 460
588 461
589 462
590 463
591 464
592 465
593 466
594 467
595 468
596 469
597 470
598 471
599 472
600 473
601 474
602 475
603 476
604 477
605 478
606 479
607 480
608 481
609 482
610 483
611 484
612 485
613 486
614 487
615 488
616 489
617 490
618 491
619 492
620 493
621 494
622 495
623 496
624 497
625 498
626 499
627 500
628 501
629 502
630 503
631 504
632 505
633 506
634 507
635 508
636 509
637 510
638 511
639 512
640 513
641 514
642 515
643 516
644 517
645 518
646 519
647 520
648 521
649 522
650 523
651 524
652 525
653 526
654 527
655 528
656 529
657 530
658 531
659 532
660 533
661 534
662 535
663 536
664 537
665 538
666 539
667 540
668 541
669 542
670 543
671 544
672 545
673 546
674 547
675 548
676 549
677 550
678 551
679 552
680 553
681 554
682 555
683 556
684 557
685 558
686 559
687 560
688 561
689 562
690 563
691 564
692 565
693 566
694 567
695 568
696 569
697 570
698 571
699 572
700 573
701 574
702 575
703 576
704 577
705 578
706 579
707 580
708 581
709 582
710 583
711 584
712 585
713 586
714 587
715 588
716 589
717 590
718 591
719 592
720 593
721 594
722 595
723 596
724 597
725 598
726 599
727 600
728 601
729 602
730 603
731 604
732 605
733 606
734 607
735 608
736 609
737 610
738 611
739 612
740 613
741 614
742 615
743 616
744 617
745 618
746 619
747 620
748 621
749 622
750 623
751 624
752 625
753 626
754 627
755 628
756 629
757 630
758 631
759 632
760 633
761 634
762 635
763 636
764 637
765 638
766 639
767 640
768 641
769 642
770 643
771 644
772 645
773 646
774 647
775 648
776 649
777 650
778 651
779 652
780 653
781 654
782 655
783 656
784 657
785 658
786 659
787 660
788 661
789 662
790 663
791 664
792 665
793 666
794 667
795 668
796 669
797 670
798 671
799 672
800 673
801 674
802 675
803 676
804 677
805 678
806 679
807 680
808 681
809 682
810 683
811 684
812 685
813 686
814 687
815 688
816 689
817 690
818 691
819 692
820 693
821 694
822 695
823 696
824 697
825 698
826 699
827 700
828 701
829 702
830 703
831 704
832 705
833 706
834 707
835 708
836 709
837 710
838 711
839 712
840 713
841 714
842 715
843 716
844 717
845 718
846 719
847 720
848 721
849 722
850 723
851 724
852 725
853 726
854 727
855 728
856 729
857 730
858 731
859 732
860 733
861 734
862 735
863 736
864 737
865 738
866 739
867 740
868 741
869 742
870 743
871 744
872 745
873 746
874 747
875 748
876 749
877 750
878 751
879 752
880 753
881 754
882 755
883 756
884 757
885 758
886 759
887 760
888 761
889 762
890 763
891 764
892 765
893 766
894 767
895 768
896 769
897 770
898 771
899 772
900 773
901 774
902 775
903 776
904 777
905 778
906 779
907 780
908 781
909 782
910 783
911 784
912 785
913 786
914 787
915 788
916 789
917 790
918 791
919 792
920 793
921 794
922 795
923 796
924 797
925 798
926 799
927 800
928 801
929 802
930 803
931 804
932 805
933 806
934 807
935 808
936 809
937 810
938 811
939 812
940 813
941 814
942 815
943 816
944 817
945 818
946 819
947 820
948 821
949 822
950 823
951 824
952 825
953 826
954 827
955 828
956 829
957 830
958 831
959 832
960 833
961 834
962 835
963 836
964 837
965 838
966 839
967 840
968 841
969 842
970 843
971 844
972 845
973 846
974 847
975 848
976 849
977 850
978 851
979 852
980 853
981 854
982 855
983 856
984 857
985 858
986 859
987 860
988 861
989 862
990 863
991 864
992 865
993 866
994 867
995 868
996 869
997 870
998 871
999 872
1000 873
1001 874
1002 875
1003 876
1004 877
1005 878
1006 879
1007 880
1008 881
1009 882
1010 883
1011 884
1012 885
1013 886
1014 887
1015 888
1016 889
1017 890
1018 891
1019 892
1020 893
1021 894
1022 895
1023 896
1024 897
1025 898
1026 899
1027 900
1028 901
1029 902
1030 903
1031 904
1032 905
1033 906
1034 907
1035 908
1036 909
1037 910
1038 911
1039 912
1040 913
1041 914
1042 915
1043 916
1044 917
1045 918
1046 919
1047 920
1048 921
1049 922
1050 923
1051 924
1052 925
1053 926
1054 927
1055 928
1056 929
1057 930
1058 931
1059 932
1060 933
1061 934
1062 935
1063 936
1064 937
1065 938
1066 939
1067 940
1068 941
1069 942
1070 943
1071 944
1072 945
1073 946
1074 947
1075 948
1076 949
1077 950
1078 951
1079 952
1080 953
1081 954
1082 955
1083 956
1084 957
1085 958
1086 959
1087 960
1088 961
1089 962
1090 963
1091 964
1092 965
1093 966
1094 967
1095 968
1096 969
1097 970
1098 971
1099 972
1100 973
1101 974
1102 975
1103 976
1104 977
1105 978
1106 979
1107 980
1108 981
1109 982
1110 983
1111 984
1112 985
1113 986
1114 987
1115 988
1116 989
1117 990
1118 991
1119 992
1120 993
1121 994
1122 995
1123 996
1124 997
1125 998
1126 999
1127 1000
1128 1001
1129 1002
1130 1003
1131 1004
1132 1005
1133 1006
1134 1007
1135 1008
1136 1009
1137 1010
1138 1011
1139 1012
1140 1013
1141 1014
1142 1015
1143 1016
1144 1017
1145 1018
1146 1019
1147 1020
1148 1021
1149 1022
1150 1023
1151 1024
1152 1025
1153 1026
1154 1027
1155 1028
1156 1029
1157 1030
1158 1031
1159 1032
1160 1033
1161 1034
1162 1035
1163 1036
1164 1037
1165 1038
1166 1039
1167 1040
1168 1041
1169 1042
1170 1043
1171 1044
1172 1045
1173 1046
1174 1047
1175 1048
1176 1049
1177 1050
1178 1051
1179 1052
1180 1053
1181 1054
1182 1055
1183 1056
1184 1057
1185 1058
1186 1059
1187 1060
1188 1061
1189 1062
1190 1063
1191 1064
1192 1065
1193 1066
1194 1067
1195 1068
1196 1069
1197 1070
1198 1071
1199 1072
1200 1073
1201 1074
1202 1075
1203 1076
1204 1077
1205 1078
1206 1079
1207 1080
1208 1081
1209 1082
1210 1083
1211 1084
1212 1085
1213 1086
1214 1087
1215 1088
1216 1089
1217 1090
1218 1091
1219 1092
1220 1093
1221 1094
1222 1095
1223 1096
1224 1097
1225 1098
1226 1099
1227 1100
1228 1101
1229 1102
1230 1103
1231 1104
1232 1105
1233 1106
1234 1107
1235 1108
1236 1109
1237 1110
1238 1111
1239 1112
1240 1113
1241 1114
1242 1115
1243 1116
1244 1117
1245 1118
1246 1119
1247 1120
1248 1121
1249 1122
1250 1123
1251 1124
1252 1125
1253 1126
1254 1127
1255 1128
1256 1129
1257 1130
1258 1131
1259 1132
1260 1133
1261 1134
1262 1135
1263 1136
1264 1137
1265 1138
1266 1139
1267 1140
1268 1141
1269 1142
1270 1143
1271 1144
1272 1145
1273 1146
1274 1147
1275 1148
1276 1149
1277 1150
1278 1151
1279 1152
1280 1153
1281 1154
1282 1155
1283 1156
1284 1157
1285 1158
1286 1159
1287 1160
1288 1161
1289 1162
1290 1163
1291 1164
1292 1165
1293 1166
1294 1167
1295 1168
1296 1169
1297 1170
1298 1171
1299 1172
1300 1173
1301 1174
1302 1175
1303 1176
1304 1177
1305 1178
1306 1179
1307 1180
1308 1181
1309 1182
1310 1183
1311 1184
1312 1185
1313 1186
1314 1187
1315 1188
1316 1189
1317 1190
1318 1191
1319 1192
1320 1193
1321 1194
1322 1195
1323 1196
1324 1197
1325 1198
1326 1199
1327 1200
1328 1201
1329 1202
1330 1203
1331 1204
1332 1205
1333 1206
1334 1207
1335 1208
1336 1209
1337 1210
1338 1211
1339 1212
1340 1213
1341 1214
1342 1215
1343 1216
1344 1217
1345 1218
1346 1219
1347 1220
1348 1221
1349 1222
1350 1223
135
```

```

271      add_course('Ac_term', 'Ac_lineno', 'Ac_sno', 'Ac_A', 'Ac_B', 'Ac_C', 'Ac_D');
272  ]
273  [
274 new BEGIN
275   add_course('Ac22', '1031', 'csc227', '90', '80', '70', '60');
276 END;
277 Attempt to insert a new course record.
278 Attempt failed.
279 Error code:10001
280 The term format is invalid, it should started with "f"(Fall), "sp"(Spring), "s"(Summer) and followed
281 by 2 digits for year indication. Eg:F96
282
283
284 PL/SQL procedure successfully completed.
285
286 SQL> @c:\sql\last_task2run
287 SQL> ACCEPT c_term CHAR prompt 'Please enter Term (Eg:sp07): '
288 Please enter Term (Eg:sp07): 9999
289 SQL> ACCEPT c_lineno NUMBER prompt 'Please enter Line No (Eg:1031): '
290 Please enter Line No (Eg:1031): 1031
291 SQL> ACCEPT c_sno CHAR prompt 'Please enter Subject No (Eg:csc227): '
292 Please enter Subject No (Eg:csc227): csc227
293 SQL> ACCEPT c_A NUMBER prompt 'Please enter Grade A Values (Between 0-100): '
294 Please enter Grade A Values (Between 0-100): 90
295 SQL> ACCEPT c_B NUMBER prompt 'Please enter Grade B Values (Between 0-100): '
296 Please enter Grade B Values (Between 0-100): 80
297 SQL> ACCEPT c_C NUMBER prompt 'Please enter Grade C Values (Between 0-100): '
298 Please enter Grade C Values (Between 0-100): 70
299 SQL> ACCEPT c_D NUMBER prompt 'Please enter Grade D Values (Between 0-100): '
300 Please enter Grade D Values (Between 0-100): 60
301
302 SQL> BEGIN
303   2 add_course('Ac_term', 'Ac_lineno', 'Ac_sno', 'Ac_A', 'Ac_B', 'Ac_C', 'Ac_D');
304 3 END;
305 [
306 old BEGIN
307   add_course('Ac_term', 'Ac_lineno', 'Ac_sno', 'Ac_A', 'Ac_B', 'Ac_C', 'Ac_D');
308 ]
309
310 new BEGIN
311   add_course('F999', '1031', 'csc227', '90', '80', '70', '60');
312 END;
313 Attempt to insert a new course record.
314 Attempt failed.
315 Error code:10001
316 The term format is invalid, it should started with "f"(Fall), "sp"(Spring), "s"(Summer) and followed
317 by 2 digits for year indication. Eg:F96
318
319
320 PL/SQL procedure successfully completed.
321
322
323 SQL> SELECT * FROM courses;
324
325
326 TERM LINENO SNO A B C D
327
328 F91 1031 csc481 80 70 60 45
329 F91 1032 csc498 95 85 65 45
330 F96 1031 csc226 90 80 65 50
331 F96 1032 csc228 90 80 65 50
332 F97 1033 csc227 90 80 65 50
333
334 5 rows selected.
335
336 SQL> @c:\sql\last_task2run
337 SQL> ACCEPT c_term CHAR prompt 'Please enter Term (Eg:sp07): '
338 Please enter Term (Eg:sp07): F23
339 SQL> ACCEPT c_lineno NUMBER prompt 'Please enter Line No (Eg:1031): '
340 Please enter Line No (Eg:1031): 1031
341 SQL> ACCEPT c_sno CHAR prompt 'Please enter Subject No (Eg:csc227): '
342 Please enter Subject No (Eg:csc227): csc227
343 SQL> ACCEPT c_A NUMBER prompt 'Please enter Grade A Values (Between 0-100): '
344 Please enter Grade A Values (Between 0-100): 90
345 SQL> ACCEPT c_B NUMBER prompt 'Please enter Grade B Values (Between 0-100): '
346 Please enter Grade B Values (Between 0-100): 85
347 SQL> ACCEPT c_C NUMBER prompt 'Please enter Grade C Values (Between 0-100): '
348 Please enter Grade C Values (Between 0-100): 84
349 SQL> ACCEPT c_D NUMBER prompt 'Please enter Grade D Values (Between 0-100): '
350 Please enter Grade D Values (Between 0-100): 70
351
352 SQL> BEGIN
353   2 add_course('Ac_term', 'Ac_lineno', 'Ac_sno', 'Ac_A', 'Ac_B', 'Ac_C', 'Ac_D');
354 3 END;
355 [
356 old BEGIN
357   add_course('Ac_term', 'Ac_lineno', 'Ac_sno', 'Ac_A', 'Ac_B', 'Ac_C', 'Ac_D');
358 ]
359
360 new BEGIN
361   add_course('F23', '1031', 'csc227', '90', '85', '84', '70');
362 ]
363
364 Attempt to insert a new course record.
365 Attempt failed.
366 Error code:2290
367 The grade for higher grade should be more than the next grade by at least 10 points. Eg: Grade A: 90
368 points, Grade B: 80 points, Grade C: 70 points, Grade D: 60 points
369
370
371 PL/SQL procedure successfully completed.
372
373 SQL> @c:\sql\last_task2run
374 SQL> ACCEPT c_term CHAR prompt 'Please enter Term (Eg:sp07): '
375 Please enter Term (Eg:sp07): F23
376 SQL> ACCEPT c_lineno NUMBER prompt 'Please enter Line No (Eg:1031): '
377 Please enter Line No (Eg:1031): 1031
378 SQL> ACCEPT c_sno CHAR prompt 'Please enter Subject No (Eg:csc227): '
379 Please enter Subject No (Eg:csc227): csc227
380 SQL> ACCEPT c_A NUMBER prompt 'Please enter Grade A Values (Between 0-100): '
381 Please enter Grade A Values (Between 0-100): 90
382 SQL> ACCEPT c_B NUMBER prompt 'Please enter Grade B Values (Between 0-100): '
383 Please enter Grade B Values (Between 0-100): 85
384 SQL> ACCEPT c_C NUMBER prompt 'Please enter Grade C Values (Between 0-100): '
385 Please enter Grade C Values (Between 0-100): 84
386 SQL> ACCEPT c_D NUMBER prompt 'Please enter Grade D Values (Between 0-100): '
387 Please enter Grade D Values (Between 0-100): 70
388
389 SQL> BEGIN
390   2 add_course('Ac_term', 'Ac_lineno', 'Ac_sno', 'Ac_A', 'Ac_B', 'Ac_C', 'Ac_D');
391 3 END;
392 [
393 old BEGIN
394   add_course('Ac_term', 'Ac_lineno', 'Ac_sno', 'Ac_A', 'Ac_B', 'Ac_C', 'Ac_D');
395 ]
396
397 new BEGIN
398   add_course('F23', '1031', 'csc227', '-1', '85', '84', '70');
399 ]
400
401 Attempt to insert a new course record.
402 Attempt failed.
403 Error code:2290
404 Error (ORA-02290): check constraint (SEDFER.SYS_C007733) violated
405
406
407 PL/SQL procedure successfully completed.
408
409 SQL> SELECT * FROM courses;

```

```

406
407      TERM    LINENO SNO      A   B   C   D
408
409      F91    1031 csc481   80   70   60   45
410      F93    1032 csc498   95   85   65   45
411      F96    1031 csc226   90   80   65   50
412      F96    1032 csc226   90   80   65   50
413      F97    1031 csc227   90   80   65   50
414
415  5 rows selected.
416
417  SQL> SELECT * FROM courses;
418
419      TERM    LINENO SNO      A   B   C   D
420
421      F91    1031 csc481   80   70   60   45
422      F93    1032 csc498   95   85   65   45
423      F96    1031 csc226   90   80   65   50
424      F96    1032 csc226   90   80   65   50
425      F97    1031 csc227   90   80   65   50
426
427  5 rows selected.
428
429  SQL> @c:\sql\java_task\run
430  SQL> ACCEPT c_term CHAR prompt 'Please enter Term (Eg:sp97): '
431  Please enter Term (Eg:sp97): sp97
432  SQL> ACCEPT c_lineno NUMBER prompt 'Please enter Line No (Eg:1031): '
433  Please enter Line No (Eg:1031): 1031
434  SQL> ACCEPT c_sno CHAR prompt 'Please enter Subject No (Eg:csc227): '
435  Please enter Subject No (Eg:csc227): csc498
436  SQL> ACCEPT c_A NUMBER prompt 'Please enter Grade A Values (Between 0-100): '
437  Please enter Grade A Values (Between 0-100): 80
438  SQL> ACCEPT c_B NUMBER prompt 'Please enter Grade B Values (Between 0-100): '
439  Please enter Grade B Values (Between 0-100): 80
440  SQL> ACCEPT c_C NUMBER prompt 'Please enter Grade C Values (Between 0-100): '
441  Please enter Grade C Values (Between 0-100): 60
442  SQL> ACCEPT c_D NUMBER prompt 'Please enter Grade D Values (Between 0-100): '
443  Please enter Grade D Values (Between 0-100): 60
444
445  SQL> BEGIN
446    2    add_course('Ac_term', 'Ac_lineno', 'Ac_sno', 'Ac_A', 'Ac_B', 'Ac_C', 'Ac_D');
447  3  END;
448
449  old:BEGIN
450    add_course('Ac_term', 'Ac_lineno', 'Ac_sno', 'Ac_A', 'Ac_B', 'Ac_C', 'Ac_D');
451
452  END;
453
454  new:BEGIN
455    add_course('sp97', '1031', 'csc498', '95', '85', '65', '45');
456  END;
457  Attempt to insert a new course record.
458  Attempt failed.
459  The record for term #sp97 and line no of #1031 is found in the database!
460
461  Error:1
462
463  PL/SQL procedure successfully completed.
464
465  SQL> SELECT * FROM courses;
466
467      TERM    LINENO SNO      A   B   C   D
468
469      F91    1031 csc481   80   70   60   45
470      F93    1032 csc498   95   85   65   45
471      F96    1031 csc226   90   80   65   50
472      F96    1032 csc226   90   80   65   50
473      F97    1031 csc227   90   80   65   50
474
475  5 rows selected.
476
477  SQL> SPOOL OFF

```

Figure 3.4 All Statements Processed of Screenshots and Test Cases

# Assignment 1

---

## ORIGINALITY REPORT

---

**7%**  
SIMILARITY INDEX

**3%**  
INTERNET SOURCES

**0%**  
PUBLICATIONS

**4%**  
STUDENT PAPERS

---

## PRIMARY SOURCES

---

- |          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>killarney-h.schools.nsw.gov.au</b><br>Internet Source                          | <b>2%</b> |
| <b>2</b> | <b>Submitted to UOW Malaysia KDU University College Sdn. Bhd</b><br>Student Paper | <b>2%</b> |
| <b>3</b> | <b>Submitted to University of West London</b><br>Student Paper                    | <b>2%</b> |
| <b>4</b> | <b>docplayer.com.br</b><br>Internet Source  | <b>1%</b> |
- 

Exclude quotes      Off

Exclude bibliography      Off

Exclude matches      Off