

ASSIGNMENT COVER PAGE

Programme	Course Code and Title	
Bachelor of Computer Science (Hons)	CDB3033/N (Database Programming)	
Student's name / student's id	Lecturer's name	
CHAN SEOW FEN / 0207368	Lai Kim Min	
Date issued	Submission Deadline	Indicative Weighting
Week 4 -05/02/2024	Week 7 - 26/02/2024	30%
Assignment [1]	Oracle SQL Queries & Stored Procedures	

This assessment assesses the following course learning outcomes

# as in Course Guide	UOWM KDU Penang University College Learning Outcome
CLO1	Develop scripting for prototyping database applications with predefined functions. (C5, PLO3)
CLO2	Apply database integrity in a concurrent environment (C3, PLO3)
CLO3	
CLO4	
# as in Course Guide	University of Lincoln Learning Outcome
CLO1	Use appropriate tools and techniques to design a database
CLO2	Appraise the structure of a database design using standard evaluation mechanisms
CLO3	
CLO4	

Student's declaration

I certify that the work submitted for this assignment is my own and research sources are fully acknowledged.

Student's signature:



Submission Date:

26/2/24

Assignment 1

ORIGINALITY REPORT

7%

SIMILARITY INDEX

3%

INTERNET SOURCES

0%

PUBLICATIONS

4%

STUDENT PAPERS

PRIMARY SOURCES

1

killarney-h.schools.nsw.gov.au

Internet Source

2%

2

Submitted to UOW Malaysia KDU University
College Sdn. Bhd

Student Paper

2%

3

Submitted to University of West London

Student Paper

2%

4

docplayer.com.br

Internet Source

1%

Exclude quotes Off
Exclude bibliography Off

Exclude matches Off

Table of Contents

1.0 Program Listing	3
1.1 Task 1(a)	3
1.2 Task 1(b)	3
1.3 Task 2	4
2.0 Screenshots and Test Cases	5
2.1 Task 1(a)	5
2.2 Task 1(b)	6
2.3 Task 2	8
2.3.1 Successful Case	8
2.3.2 Unsuccessful Case - Business Rule #1 Violated [Term Format]	9
2.3.3 Unsuccessful Case - Business Rule #2 Violated [Grades Value]	10
2.3.4 Unsuccessful Case – Duplicated Record	11
3.0 Listing of All PL/SQL Statements Processed.....	12
3.1 Task 1(a)	12
3.2 Task 1(b)	13
3.3 Task 2	14
3.4 Screenshots and Test Cases.....	15
4.0 Reference List.....	19

1.0 Program Listing

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)

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

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:\sql\asm_task1a - run script
SQL> SELECT
2   s.sno AS "Subject No",
3   UPPER(s.stitle) AS "Subject Title",
4   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 /
Enter value for term: f96
Enter value for min_weight: 40 3 user Input
old:SELECT
      s.sno AS "Subject No",
      UPPER(s.stitle) AS "Subject Title",
      REPLACE(REPLACE(c.term, 'f', 'Fall-'), 'sp', 'Spring-'), 's', 'Summer-') AS Term,
      INITCAP(com.compname) 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(c.term, 'f', 'Fall-'), 'sp', 'Spring-'), 's', 'Summer-') AS Term,
      INITCAP(com.compname) 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'

Subject No    Subject Title        TERM      Component Name    MAXPOINTS    WEIGHT
csc226       INTRODUCTION TO PROGRAMMING I Fall-96  Final           100      50
csc226       INTRODUCTION TO PROGRAMMING I Fall-96  Final           100      40
csc226       INTRODUCTION TO PROGRAMMING I Fall-96  Programs        400      40
}
3 rows selected. 3 user Input
} 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*.

Formula

$$\frac{\sum \text{Points} \times \text{Weight}}{\sum \text{Weight}}$$

Weighted Average

$$\frac{(90 \times 60) + (85 \times 40)}{60+40}$$

$$= \frac{8800}{100}$$

$$= 88$$

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*.

Formula

$$\frac{\sum \text{Points} \times \text{Weight}}{\sum \text{Weight} \times \text{maxpoints}} \times 100\%$$

Weighted Average

$$\frac{(90 \times 60) + (85 \times 40)}{(60 \times 120) + (40 \times 90)} \times 100\%$$

$$= \frac{8800}{10800} \times 100\%$$

$$= 81.48$$

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*.

Selected Record:

TERM	LINENO	COMPNAME	POINTS
f96	1031	exam1	90

weight and maxpoints to refer:

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

Student 1111:

Points	Weight	Maxpoints
90	30	100
75	20	80
95	50	100

$$\frac{(90 \times 30) + (75 \times 20) + (95 \times 50)}{(30 \times 100) + (20 \times 80) + (50 \times 100)} \times 100 = \frac{8950}{9600} \times 100 = 93.23 \text{ # rounded to integer}$$

Student 2222:

Points	Weight	Maxpoints
70	30	100
40	20	80
82	50	100

$$\frac{(70 \times 30) + (40 \times 20) + (82 \times 50)}{(30 \times 100) + (20 \times 80) + (50 \times 100)} \times 100 = \frac{7000}{9600} \times 100 = 72.92 \text{ # rounded to integer}$$

Student 4444:

Points	Weight	Maxpoints
83	30	100
71	20	80
74	50	100

$$\frac{(83 \times 30) + (71 \times 20) + (74 \times 50)}{(30 \times 100) + (20 \times 80) + (50 \times 100)} \times 100 = \frac{7610}{9600} \times 100 = 79.27 \text{ # rounded to integer}$$

$$= 79 \text{ # rounded to integer}$$

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:\sqlasm_task1b\f96' '1031'
SQL> 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 = 'f1' AND e.lineno = '62'
14 GROUP BY e.term, e.lineno, e.sid, s.lname, s.fname
15 /
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(sc.points * com.weight)/SUM(com.weight*com.maxpoints))*100) AS "Course Average"
FROM enrolls e
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.compname = com.compname
WHERE e.term = 'f96' AND e.lineno = '1031'
GROUP BY e.term, e.lineno, e.sid, s.lname, s.fname
new: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(sc.points * com.weight)/SUM(com.weight*com.maxpoints))*100) AS "Course Average"
FROM enrolls e
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.compname = com.compname
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.
SQL> SELECT * FROM student.course_average;
      TERM   LINENO STUDENT ID  LAST NAME  FIRST NAME  COURSE AVERAGE
      Fall-96 1031 4444        Rajsheshkar Naveen          79
      Fall-96 1031 2222        Corn           Sydney          73
      Fall-96 1031 1111        Rajsheshkar Nandita         93
3 rows selected.
  
```

view created
query the result from the view
} 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.

SQL> SELECT * FROM courses;						
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	
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;
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

```

— select all records from courses table to validate

← New record added 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	Record is not added and prompt out the error message with error code.

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:\sqlplus_task2run
SQL> ACCEPT C.Term CHAR prompt 'Please enter Term (Eg:Sp)';
Please enter Term (Eg:sp?) 22 Invalid term
SQL> ACCEPT C_lineno NUMBER prompt 'Please enter Line No (Eg:1031): ';
SQL> ACCEPT C_Sno CHAR prompt 'Please enter Sno (Eg:1031): ';
SQL> ACCEPT C_Subj CHAR prompt 'Please enter Subject No (Eg:csc227): ';
Please enter Subject No (Eg:csc227): scc227
SQL> ACCEPT C_A NUMBER prompt 'Please enter Grade A Values (Between 0-100): ';
Please enter Grade A Values (Between 0-100): 98
SQL> ACCEPT C_B NUMBER prompt 'Please enter Grade B Values (Between 0-100): ';
Please enter Grade B Values (Between 0-100): 88
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): 66
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('a22', '1031', 'csc227', '98', '88', '70', '60');
END;
Attempt to insert a new course record.
Error: failed alert user record not added
Cause: code=1004
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\class\test2.sql
SQL> ACCEPT c.Term CHAR prompt 'Please enter term (Eg:-97) : '
Please enter term (Eg:sp97): 4999 invalid Term
SQL> ACCEPT c.lineno NUMBER prompt 'Please enter Line No (Eg:1031) : '
Please enter Line No (Eg:1031): 1031
SQL> ACCEPT c.sno NUMBER prompt 'Please enter student number 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): 98
SQL> ACCEPT c.B NUMBER prompt 'Please enter Grade B Values (Between 0-100): '
Please enter Grade B Values (Between 0-100): 88
SQL> ACCEPT c.C NUMBER prompt 'Please enter Grade C Values (Between 0-100): '
Please enter Grade C Values (Between 0-100): 78
SQL> ACCEPT c.D NUMBER prompt 'Please enter Grade D Values (Between 0-100): '
Please enter Grade D Values (Between 0-100): 68
SQL>
SQL> BEGIN
 2   add_course('&c_term', '&c_lineno', '&c_sno', '&c_A', '&c_B', '&c_C', '&c_D');
 4 END;
old:c:BEGIN
  add_course('&c_term', '&c_lineno', '&c_sno', '&c_A', '&c_B', '&c_C', '&c_D');
END;

new:c:BEGIN
  add_course('f999', '1031', 'csc227', '98', '88', '78', '68');
END;
Attempt to insert a new course record.
Attempt failed Alert user record not added
Error code:-10081
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 Error message
PL/SQL procedure successfully completed.

```

Figure 2.11 Output of Test Case – Invalid Digits for Term

```
SQL> SELECT * FROM courses;
TERMINAL>      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 2.12 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	

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\asm_task2run
SQL> ACCEPT c_term CHAR prompt 'Please enter Term (Eg:sp97): '
Please enter Term (Eg:sp97): f23
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): 85
SQL> ACCEPT c_C NUMBER prompt 'Please enter Grade C Values (Between 0-100): '
Please enter Grade C Values (Between 0-100): 84
SQL> ACCEPT c_D NUMBER prompt 'Please enter Grade D Values (Between 0-100): '
Please enter Grade D Values (Between 0-100): 70
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('f23', '1031', 'csc227', '90', '85', '84', '70');
END;
Attempt to insert a new course record.
Attempt failed. Error code:-20001 Alert user record not added
Error code:-20001
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
PL/SQL procedure successfully completed.

```

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

```

SQL> @ c:\sql\asm_task2run
SQL> ACCEPT c_term CHAR prompt 'Please enter Term (Eg:sp97): '
Please enter Term (Eg:sp97): f23
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): 85
SQL> ACCEPT c_C NUMBER prompt 'Please enter Grade C Values (Between 0-100): '
Please enter Grade C Values (Between 0-100): 84
SQL> ACCEPT c_D NUMBER prompt 'Please enter Grade D Values (Between 0-100): '
Please enter Grade D Values (Between 0-100): 70
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('f23', '1031', 'csc227', '-1', '85', '84', '70');
END;
Attempt to insert a new course record.
Attempt failed. Error code:-22998 Alert user record not added
Error code:-22998
Error:ORA-02298: check constraint (SEOWFEN.SYS_C007733) violated
Error Message
PL/SQL procedure successfully completed.

```

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. *Record 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:\sql\asm_task2.sql
SQL> ACCEPT c_term CHAR prompt 'Please enter Term (Eg:sp97): ';Same as
Please enter Term (Eg:sp97): sp97Record ⑤
SQL> ACCEPT c_lineno NUMBER prompt 'Please enter Line No (Eg:1031): ';Same as
Please enter Line No (Eg:1031): 1031Record ⑤
SQL> ACCEPT c_sno CHAR prompt 'Please enter Subject No (Eg:csc227): ';Same as
Please enter Subject No (Eg:csc227): csc498
SQL> ACCEPT c_A NUMBER prompt 'Please enter Grade A Values (Between 0-100): ';Same as
Please enter Grade A Values (Between 0-100): 90
SQL> ACCEPT c_B NUMBER prompt 'Please enter Grade B Values (Between 0-100): ';Same as
Please enter Grade B Values (Between 0-100): 80
SQL> ACCEPT c_C NUMBER prompt 'Please enter Grade C Values (Between 0-100): ';Same as
Please enter Grade C Values (Between 0-100): 70
SQL> ACCEPT c_D NUMBER prompt 'Please enter Grade D Values (Between 0-100): ';Same as
Please enter Grade D Values (Between 0-100): 60
SQL> BEGIN
  2> add_course('&c_term', '&c_lineno', '&c_sno', '&c_A', '&c_B', '&c_C', '&c_D');
  3> END;
old:BEGIN
add_course('&c_term', '&c_lineno', '&c_sno', '&c_A', '&c_B', '&c_C', '&c_D');

new:BEGIN
add_course('sp97', '1031', 'csc498', '90', '80', '70', '60');
END;
Attempt to insert a new course record.Alert user
Attempt failed.record not added
The record for term sp97 and line no of 1031 is found in the database!
Error:-1Error Code
PL/SQL procedure successfully completed.Error Message ↑
```

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(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     8  FROM courses c
15     9  JOIN subject s ON s.sno = c.sno
16    10  JOIN components com ON com.term = c.term AND com.lineno = c.lineno
17    11  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:\sql\asm_task1b_query 'sp97' '1031'
7 SQL> @ c:\sql\asm_task1b 'sp97' '1031'
8 ┏SQL> CREATE OR REPLACE VIEW student_course_average AS
9   2 SELECT
10    3   REPLACE(REPLACE(REPLACE(e.term, 'f', 'Fall-'), 'sp', 'Spring-'), 's', 'Summer-') AS Term,
11    e.lineno AS "Line No",
12    e.sid AS "Student ID",
13    s.lname AS "Last Name",
14    s.fname AS "First Name",
15    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
60 Spring-97      1031 5555      Yam          Elad           85
61 Spring-97      1031 6666      Herring       Lincoln        83
62
63 2 rows selected.
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 200
5  SQL> SET SERVEROUTPUT ON
6  SQL> @c:\sqlasm_task2
7  ┌─SQL> CREATE OR REPLACE PROCEDURE add_course
8  2  (c_term IN VARCHAR2, c_lineno IN NUMBER, c_sno IN VARCHAR2,
9  3  c_A IN NUMBER, c_B IN NUMBER, c_C IN NUMBER, c_D IN NUMBER) AS
10 4
11 5    e_term_error EXCEPTION; --Does not follows Business Rule #1
12 6    PRAGMA EXCEPTION_INIT(e_term_error, -10001); --associate the error code
13 7
14 8    e_grade_error_1 EXCEPTION; --Does not follows Business Rule #2 - A > B > C > D
15 9    PRAGMA EXCEPTION_INIT(e_grade_error_1, -20001); --associate the error code
16 10
17 11    e_grade_error_2 EXCEPTION; --Does not follows Business Rule #2 - Value 0-100
18 12    PRAGMA EXCEPTION_INIT(e_grade_error_2, -20002); --associate the error code
19 13
20 14
21 15 BEGIN
22 16   DBMS_OUTPUT.PUT_LINE('Attempt to insert a new course record.');
23 17
24 18   --Insert the new record
25 19   INSERT INTO courses VALUES(c_term, c_lineno, c_sno, c_A, c_B, c_C, c_D);
26 20
27 21   --Business Rule 1: term must be started with "f" (Fall), "sp" (Spring), "s" (Summer)
28 22   --And followed by 2 digits indicating year
29 23   IF NOT REGEXP_LIKE(c_term, '^([F|sp|s])([0-9]{2})$') THEN
30 24     --raise the EXCEPTION
31 25     RAISE e_term_error;
32 26   END IF;
33 27
34 28   --Business Rule 2: Value of Grade A must bigger than Grade B (minimum gap 10 points)
35 29   --Same for other grades subsequently
36 30   --Grade values must be in the range of 0-100
37 31   IF (c_A < (c_B+10)) OR (c_B < (c_C+10)) OR (c_C < (c_D+10)) THEN
38 32     --raise the EXCEPTION
39 33     RAISE e_grade_error_1;
40 34   END IF;
41 35
42 36   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
43 37     --raise the EXCEPTION
44 38     RAISE e_grade_error_2;
45 39   END IF;
46 40
47 41   DBMS_OUTPUT.PUT_LINE('New course record added!');
48 42
49 43   COMMIT; --Save the record
50 44
51 45 EXCEPTION
52 46
53 47   WHEN e_term_error THEN --Violate Business Rule
54 48     DBMS_OUTPUT.PUT_LINE('Attempt failed.');
55 49     DBMS_OUTPUT.PUT_LINE('Error code: ' || SQLCODE);
56 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');
57 51     ROLLBACK;
58 52
59 53   WHEN e_grade_error_1 THEN --Violate Business Rule
60 54     DBMS_OUTPUT.PUT_LINE('Attempt failed.');
61 55     DBMS_OUTPUT.PUT_LINE('Error code: ' || SQLCODE);
62 56     DBMS_OUTPUT.PUT_LINE('The grade values must be in the range of 0-100. Eg: Grade A: 90 points, Grade B: 80 points, Grade C: 70 points, Grade D: 60 points');
63 57     ROLLBACK;
64 58
65 59   WHEN e_grade_error_2 THEN --Violate Business Rule
66 60     DBMS_OUTPUT.PUT_LINE('Attempt failed.');
67 61     DBMS_OUTPUT.PUT_LINE('Error code: ' || SQLCODE);
68 62     DBMS_OUTPUT.PUT_LINE('The grade values must be in the range of 0-100. Eg: Grade A: 95 points');
69 63     ROLLBACK;
70 64
71 65   WHEN DUP_VAL_ON_INDEX THEN --Inserting Same Record
72 66     DBMS_OUTPUT.PUT_LINE('Attempt failed.');
73 67     DBMS_OUTPUT.PUT_LINE('The record for term #' || c_term || ' and line no of #' || c_lineno || ' is found in the database!');
74 68     DBMS_OUTPUT.PUT_LINE('Error: ' || SQLCODE);
75 69     --DBMS_OUTPUT.PUT_LINE('Message: ' || SQLERRM);
76 70     ROLLBACK;
77 71
78 72   --All error handlers
79 73   WHEN OTHERS THEN
80 74     DBMS_OUTPUT.PUT_LINE('Attempt failed.');
81 75     DBMS_OUTPUT.PUT_LINE('Error code: ' || SQLCODE);
82 76     DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM);
83 77     ROLLBACK; --unto the transaction
84 78 END;
85 79 /
86
87 ┌─Procedure ADD COURSE compiled
88
89 SQL> @c:\sqlasm_task2run
90 SQL> ACCEPT c_term CHAR prompt 'Please enter Term (Eg:sp97): ';
91 Please enter Term (Eg:sp97): f03
92 SQL> ACCEPT c_lineno NUMBER prompt 'Please enter Line No (Eg:1031): ';
93 Please enter Line No (Eg:1031): 1032
94 SQL> ACCEPT c_sno NUMBER prompt 'Please enter Student No (Eg:csc227): ';
95 Please enter Student No (Eg:csc227): csc498
96 SQL> ACCEPT c_A NUMBER prompt 'Please enter Grade A Values (Between 0-100): ';
97 Please enter Grade A Values (Between 0-100): 95
98 SQL> ACCEPT c_B NUMBER prompt 'Please enter Grade B Values (Between 0-100): ';
99 Please enter Grade B Values (Between 0-100): 85
100 SQL> ACCEPT c_C NUMBER prompt 'Please enter Grade C Values (Between 0-100): ';
101 Please enter Grade C Values (Between 0-100): 65
102 SQL> ACCEPT c_D NUMBER prompt 'Please enter Grade D Values (Between 0-100): ';
103 Please enter Grade D Values (Between 0-100): 45
104 SQL>
105 ┌─SQL> BEGIN
106 2   add_course('&c_term', '&c_lineno', '&c_sno', '&c_A', '&c_B', '&c_C', '&c_D');
107 3   END;
108 4 /
109 old:BEGIN
110   add_course('&c_term', '&c_lineno', '&c_sno', '&c_A', '&c_B', '&c_C', '&c_D');
111 END;
112
113 new:BEGIN
114   add_course('f03', '1032', 'csc498', '95', '85', '65', '45');
115 END;
116 Attempt to insert a new course record.
117 New course record added!
118
119 PL/SQL procedure successfully completed.
120
121 SQL> SELECT * FROM courses;
122
123 ┌─SQL> SELECT * FROM courses;
124
125 TERM LINENO SNO A B C D
126 f03 1032 csc498 95 85 65 45
127 f96 1031 csc226 90 80 65 50
128 f96 1032 csc226 90 80 65 50
129 sp97 1031 csc227 90 80 65 50
130
131 4 rows selected.
132
133 SQL> SPOOL OFF
134

```

Figure 3.3 All Statements Processed of Task 2

3.4 Screenshots and Test Cases

```

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(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   8  FROM courses
15   9  JOIN subject s ON s.sno = c.sno
16  10  JOIN components com ON com.term >= c.term AND com.lineno = c.lineno
17  11  WHERE c.term = '&term' AND com.weight >= '&min_weight'
18  12  /
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
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
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 = '&term' 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> @c:\sql\asm_task1b 'f96' '1031'
54 SQL> CREATE OR REPLACE VIEW student_course_average AS
55
56 2  SELECT
57  3      REPLACE(REPLACE(e.term, 'f', 'Fall-'), 'sp', 'Spring-'), 's', 'Summer-') AS Term,
58  4      e.lineno AS "Line No",
59  5      e.sid AS "Student ID",
60  6      s.lname AS "Last Name",
61  7      s.fname AS "First Name",
62  8      ROUND((SUM(sc.points * com.weight)/SUM(com.weight*com.maxpoints))*100) AS "Course Average"
63  9  FROM enrolls e
64 10  JOIN students s ON e.sid = s.sid
65 11  JOIN scores sc ON e.term = sc.term AND e.lineno = sc.lineno AND e.sid = sc.sid
66 12  JOIN components com ON e.term = com.term AND e.lineno = com.lineno AND sc.compname = com.compname
67 13  WHERE e.term = '81' AND e.lineno = '82'
68 14  GROUP BY e.term, e.lineno, e.sid, s.lname, s.fname
69
70 old:CREATE OR REPLACE VIEW student_course_average AS
71
72  SELECT
73      REPLACE(REPLACE(REPLACE(e.term, 'f', 'Fall-'), 'sp', 'Spring-'), 's', 'Summer-') AS Term,
74      e.lineno AS "Line No",
75      e.sid AS "Student ID",
76      s.lname AS "Last Name",
77      s.fname AS "First Name",
78      ROUND((SUM(sc.points * com.weight)/SUM(com.weight*com.maxpoints))*100) AS "Course Average"
79
80 FROM enrolls e
81 JOIN students s ON e.sid = s.sid
82 JOIN scores sc ON e.term = sc.term AND e.lineno = sc.lineno AND e.sid = sc.sid
83 JOIN components com ON e.term = com.term AND e.lineno = com.lineno AND sc.compname = com.compname
84 WHERE e.term = '81' AND e.lineno = '82'
85 GROUP BY e.term, e.lineno, e.sid, s.lname, s.fname
86
87 new:CREATE OR REPLACE VIEW student_course_average AS
88
89  SELECT
90      REPLACE(REPLACE(REPLACE(e.term, 'f', 'Fall-'), 'sp', 'Spring-'), 's', 'Summer-') AS Term,
91      e.lineno AS "Line No",
92      e.sid AS "Student ID",
93      s.lname AS "Last Name",
94      s.fname AS "First Name",
95      ROUND((SUM(sc.points * com.weight)/SUM(com.weight*com.maxpoints))*100) AS "Course Average"
96
97 FROM enrolls e
98 JOIN students s ON e.sid = s.sid
99 JOIN scores sc ON e.term = sc.term AND e.lineno = sc.lineno AND e.sid = sc.sid
100 JOIN components com ON e.term = com.term AND e.lineno = com.lineno AND sc.compname = com.compname
101 WHERE e.term = 'f96' AND e.lineno = '1031'
102 GROUP BY e.term, e.lineno, e.sid, s.lname, s.fname
103
104 View STUDENT_COURSE_AVERAGE created.
105
106 SQL> SELECT * FROM student_course_average;
107
108 TERM Line No Student ID Last Name First Name Course Average
109
110 Fall-96 1031 4444 Rajsheshkar Naveen 79
111 Fall-96 1031 2222 Corn Sydney 73
112 Fall-96 1031 1111 Rajsheshkar Nandita 93
113
114 3 rows selected.
115
116 SQL> SELECT * FROM courses;
117
118 TERM LINENO SNO A B C D
119
120 F03 1032 csc498 95 85 65 45
121 F96 1031 csc226 90 80 65 50
122 F96 1032 csc226 90 80 65 50
123 sp97 1031 csc227 90 80 65 50
124
125 4 rows selected.
126
127 SQL> @c:\sql\asm_task2
128 SQL> CREATE OR REPLACE PROCEDURE add_course
129 2  (c_term IN VARCHAR2, c_lineno IN NUMBER, c_sno IN VARCHAR2,
130 3  c_A IN NUMBER, c_B IN NUMBER, c_C IN NUMBER, c_D IN NUMBER) AS
131 4
132 5  e_term_error EXCEPTION; --Does not follows Business Rule #1
133 6  PRAGMA EXCEPTION_INIT(e_term_error, -10001); --associate the error code
134 7
135 8  e_grade_error_1 EXCEPTION; --Does not follows Business Rule #2 - A > B > C > D
136 9  PRAGMA EXCEPTION_INIT(e_grade_error_1, -20001); --associate the error code
137 10
138 11  e_grade_error_2 EXCEPTION; --Does not follows Business Rule #2 - Value 0-100
139 12  PRAGMA EXCEPTION_INIT(e_grade_error_2, -20002); --associate the error code
140 13

```

```

136 14 BEGIN
137 15   DBMS_OUTPUT.PUT_LINE('Attempt to insert a new course record.');
138 16
139 17   --Insert the new record
140 18   INSERT INTO courses VALUES(c_term, c_lineno, c_sno, c_A, c_B, c_C, c_D);
141 19
142 20   --Business Rule 1: term must be started with "f" (Fall), "sp" (Spring), "s" (Summer)
143 21   --And followed by 2 digits indicating year
144 22   IF NOT REGEXP_LIKE(c_term, '^([f|sp|s]\d{2})$') THEN
145 23     --raise the EXCEPTION
146 24     RAISE e_term_error;
147 25   END IF;
148 26
149 27   --Business Rule 2: Value of Grade A must bigger than Grade B (minimum gap 10 points)
150 28   --Same for other grades subsequently
151 29   --Grade values must be in the range of 0-100
152 30   IF (c_A < c_B+10) OR (c_B < c_C+10) OR (c_C < c_D+10) THEN
153 31     --raise the EXCEPTION
154 32     RAISE e_grade_error_1;
155 33   END IF;
156 34
157 35   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
158 36     --raise the EXCEPTION
159 37     RAISE e_grade_error_2;
160 38   END IF;
161 39
162 40   DBMS_OUTPUT.PUT_LINE('New course record added!');
163 41
164 42   COMMIT --Save the record
165 43
166 44
167 45 EXCEPTION
168 46
169 47   WHEN e_term_error THEN --Violate Business Rule
170 48     DBMS_OUTPUT.PUT_LINE('Attempt failed.');
171 49     DBMS_OUTPUT.PUT_LINE('Error code:' || SQLCODE);
172 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');
173 51     ROLLBACK;
174 52
175 53   WHEN e_grade_error_1 THEN --Violate Business Rule
176 54     DBMS_OUTPUT.PUT_LINE('Attempt failed.');
177 55     DBMS_OUTPUT.PUT_LINE('Error code:' || SQLCODE);
178 56     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');
179 57     ROLLBACK;
180 58
181 59   WHEN e_grade_error_2 THEN --Violate Business Rule
182 60     DBMS_OUTPUT.PUT_LINE('Attempt failed.');
183 61     DBMS_OUTPUT.PUT_LINE('Error code:' || SQLCODE);
184 62     DBMS_OUTPUT.PUT_LINE('The grade values must be in the range of 0-100. Eg: Grade A: 95 points');
185 63     ROLLBACK;
186 64
187 65   WHEN DUP_VAL_ON_INDEX THEN --Inserting Same Record
188 66     DBMS_OUTPUT.PUT_LINE('Attempt failed.');
189 67     DBMS_OUTPUT.PUT_LINE('The record for term #' || c_term || ' and line no of '|| '#' || c_lineno || ' is found in the database!');
190 68     DBMS_OUTPUT.PUT_LINE('Error:' || SQLCODE);
191 69     --DBMS_OUTPUT.PUT_LINE('Message:' || SQLERRM);
192 70     ROLLBACK;
193 71
194 72   --All error handlers
195 73   WHEN OTHERS THEN
196 74     DBMS_OUTPUT.PUT_LINE('Attempt failed.');
197 75     DBMS_OUTPUT.PUT_LINE('Error code:' || SQLCODE);
198 76     DBMS_OUTPUT.PUT_LINE('Error:' || SQLERRM);
199 77     ROLLBACK; --undo the transaction
200 78 END;
201 79 /
202
203 Procedure ADD_COURSE compiled
204
205 SQL> @ c:\sqlasm_task2run
206 SQL> ACCEPT c_term CHAR prompt 'Please enter Term (Eg:sp97): '
207 Please enter Term (Eg:sp97): f01
208 SQL> ACCEPT c_lineno NUMBER prompt 'Please enter Line No (Eg:1031): '
209 Please enter Line No (Eg:1031): 1031
210 SQL> ACCEPT c_sno CHAR prompt 'Please enter Subject No (Eg:csc227): '
211 Please enter Subject (Eg:csc227): csc481
212 SQL> ACCEPT c_A NUMBER prompt 'Please enter Grade A Values (Between 0-100): '
213 Please enter Grade A Values (Between 0-100): 80
214 SQL> ACCEPT c_B NUMBER prompt 'Please enter Grade B Values (Between 0-100): '
215 Please enter Grade B Values (Between 0-100): 70
216 SQL> ACCEPT c_C NUMBER prompt 'Please enter Grade C Values (Between 0-100): '
217 Please enter Grade C Values (Between 0-100): 60
218 SQL> ACCEPT c_D NUMBER prompt 'Please enter Grade D Values (Between 0-100): '
219 Please enter Grade D Values (Between 0-100): 45
220 SQL>
221 SQL> BEGIN
222   :add_course('&c_term', '&c_lineno', '&c_sno', '&c_A', '&c_B', '&c_C', '&c_D');
223   :END;
224   :/
225 old:BEGIN
226   add_course('&c_term', '&c_lineno', '&c_sno', '&c_A', '&c_B', '&c_C', '&c_D');
227 END;
228 new:BEGIN
229   add_course('f01', '1031', 'csc481', '80', '70', '60', '45');
230 END;
231 Attempt to insert a new course record.
232 New course record added!
233
234
235 PL/SQL procedure successfully completed.
236
237 SQL> SELECT * FROM courses;
238
239
240
241 TERM LINENO SNO A B C D
242 f01 1031 csc481 80 70 60 45
243 f03 1032 csc498 95 85 65 45
244 f96 1031 csc226 90 80 65 50
245 f96 1032 csc226 90 80 65 50
246 sp97 1031 csc227 90 80 65 50
247
248 5 rows selected.
249
250 SQL> @ c:\sqlasm_task2run
251 SQL> ACCEPT c_term CHAR prompt 'Please enter Term (Eg:sp97): '
252 Please enter Term (Eg:sp97): a22
253 SQL> ACCEPT c_lineno NUMBER prompt 'Please enter Line No (Eg:1031): '
254 Please enter Line No (Eg:1031): 1031
255 SQL> ACCEPT c_sno CHAR prompt 'Please enter Subject No (Eg:csc227): '
256 Please enter Subject (Eg:csc227): csc227
257 SQL> ACCEPT c_A NUMBER prompt 'Please enter Grade A Values (Between 0-100): '
258 Please enter Grade A Values (Between 0-100): 90
259 SQL> ACCEPT c_B NUMBER prompt 'Please enter Grade B Values (Between 0-100): '
260 Please enter Grade B Values (Between 0-100): 80
261 SQL> ACCEPT c_C NUMBER prompt 'Please enter Grade C Values (Between 0-100): '
262 Please enter Grade C Values (Between 0-100): 70
263 SQL> ACCEPT c_D NUMBER prompt 'Please enter Grade D Values (Between 0-100): '
264 Please enter Grade D Values (Between 0-100): 60
265 SQL>
266 SQL> BEGIN
267   :add_course('&c_term', '&c_lineno', '&c_sno', '&c_A', '&c_B', '&c_C', '&c_D');
268   :END;
269   :/
270 old:BEGIN

```

```
271     add_course('&c_term', '&c_lineno', '&c_sno', '&c_A', '&c_B', '&c_C', '&c_D');
272   END;
273
274 new:BEGIN
275   add_course('a22', '1031', 'csc227', '99', '80', '70', '60');
276 END;
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\asm_task2run
287 SQL> ACCEPT c_term CHAR prompt 'Please enter Term (Eg:sp97): ';
288 Please enter Term (Eg:sp97): f99
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): 99
295 SQL> ACCEPT c_B NUMBER prompt 'Please enter Grade B Values (Between 0-100): ';
296 Please enter Grade B Values (Between 0-100): 88
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 SQL>
302 SQL> BEGIN
303   add_course('&c_term', '&c_lineno', '&c_sno', '&c_A', '&c_B', '&c_C', '&c_D');
304   3 END;
305   4 /
306 old:BEGIN
307   add_course('&c_term', '&c_lineno', '&c_sno', '&c_A', '&c_B', '&c_C', '&c_D');
308 END;
309
310 new:BEGIN
311   add_course('f999', '1031', 'csc227', '99', '80', '70', '60');
312 END;
Attempt to insert a new course record.
313 Attempt failed.
314 Error code:-10001
315 The term format is invalid, it should started with "f"(Fall), "sp"(Spring), "s"(Summer) and followed
316 by 2 digits for year indication. Eg:f96
317
318
319 PL/SQL procedure successfully completed.
320
321 SQL> SELECT * FROM courses;
322
323
324 TERM LINENO SNO A B C D
325
326 F01 1031 csc491 89 79 60 45
327 F03 1032 csc498 95 85 65 45
328 F96 1031 csc226 90 80 65 50
329 F96 1032 csc226 90 80 65 50
330 sp97 1031 csc227 90 80 65 50
331
332 5 rows selected.
333
334 SQL> @ c:\sql\asm_task2run
335 SQL> ACCEPT c_term CHAR prompt 'Please enter Term (Eg:sp97): ';
336 Please enter Term (Eg:sp97): f23
337 SQL> ACCEPT c_lineno NUMBER prompt 'Please enter Line No (Eg:1031): ';
338 Please enter Line No (Eg:1031): 1031
339 SQL> ACCEPT c_sno CHAR prompt 'Please enter Subject No (Eg:csc227): ';
340 Please enter Subject No (Eg:csc227): csc227
341 SQL> ACCEPT c_A NUMBER prompt 'Please enter Grade A Values (Between 0-100): ';
342 Please enter Grade A Values (Between 0-100): 90
343 SQL> ACCEPT c_B NUMBER prompt 'Please enter Grade B Values (Between 0-100): ';
344 Please enter Grade B Values (Between 0-100): 85
345 SQL> ACCEPT c_C NUMBER prompt 'Please enter Grade C Values (Between 0-100): ';
346 Please enter Grade C Values (Between 0-100): 84
347 SQL> ACCEPT c_D NUMBER prompt 'Please enter Grade D Values (Between 0-100): ';
348 Please enter Grade D Values (Between 0-100): 70
349 SQL>
350 SQL> BEGIN
351   2 add_course('&c_term', '&c_lineno', '&c_sno', '&c_A', '&c_B', '&c_C', '&c_D');
352   3 END;
353   4 /
354 old:BEGIN
355   add_course('&c_term', '&c_lineno', '&c_sno', '&c_A', '&c_B', '&c_C', '&c_D');
356 END;
357
358 new:BEGIN
359   add_course('f23', '1031', 'csc227', '99', '85', '84', '70');
360 END;
Attempt to insert a new course record.
361 Attempt failed.
362 Error code:-20001
363 The value for higher grade should be more than the next grade by at least 10 points. Eg: Grade A: 90
364 points, Grade B: 80 points, Grade C: 70 points, Grade D: 60 points
365
366
367 PL/SQL procedure successfully completed.
368
369 SQL> @ c:\sql\asm_task2run
370 SQL> ACCEPT c_term CHAR prompt 'Please enter Term (Eg:sp97): ';
371 Please enter Term (Eg:sp97): f23
372 SQL> ACCEPT c_lineno NUMBER prompt 'Please enter Line No (Eg:1031): ';
373 Please enter Line No (Eg:1031): 1031
374 SQL> ACCEPT c_sno CHAR prompt 'Please enter Subject No (Eg:csc227): ';
375 Please enter Subject No (Eg:csc227): csc227
376 SQL> ACCEPT c_A NUMBER prompt 'Please enter Grade A Values (Between 0-100): ';
377 Please enter Grade A Values (Between 0-100): -1
378 SQL> ACCEPT c_B NUMBER prompt 'Please enter Grade B Values (Between 0-100): ';
379 Please enter Grade B Values (Between 0-100): 85
380 SQL> ACCEPT c_C NUMBER prompt 'Please enter Grade C Values (Between 0-100): ';
381 Please enter Grade C Values (Between 0-100): 84
382 SQL> ACCEPT c_D NUMBER prompt 'Please enter Grade D Values (Between 0-100): ';
383 Please enter Grade D Values (Between 0-100): 70
384 SQL>
385 SQL> BEGIN
386   2 add_course('&c_term', '&c_lineno', '&c_sno', '&c_A', '&c_B', '&c_C', '&c_D');
387   3 END;
388   4 /
389 old:BEGIN
390   add_course('&c_term', '&c_lineno', '&c_sno', '&c_A', '&c_B', '&c_C', '&c_D');
391 END;
392
393 new:BEGIN
394   add_course('f23', '1031', 'csc227', '-1', '85', '84', '70');
395 END;
Attempt to insert a new course record.
396 Attempt failed.
397 Error code:-2298
398 Error:ORA-02298: check constraint (SEOWFEN.SYS_C007733) violated
399
400
401 PL/SQL procedure successfully completed.
402
403 SQL> SELECT * FROM courses;
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
```

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

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


```
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


```
SQL> @ c:\sqlvse\task2run
```

```
SQL> ACCEPT c_term CHAR prompt 'Please enter Term (Eg:sp97): ';
```

```
Please enter Term (Eg:sp97): sp97
```

```
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): csc498
```

```
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): 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> BEGIN
```

```
 2 add_course('&c_term', '&c_lineno', '&c_sno', '&c_A', '&c_B', '&c_C', '&c_D');
```

```
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('sp97', '1031', 'csc498', '80', '80', '70', '60');
```

```
END;
```

```
Attempt to insert a new course record.
```

```
Attempt failed.
```

```
The record for term #sp97 and line no of #1031 is found in the database!
```

```
Error:-1
```



```
PL/SQL procedure successfully completed.
```



```
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.
```



```
SQL> SPOOL OFF
```

Figure 3.4 All Statements Processed of Screenshots and Test Cases

4.0 Reference List

Mamalat, G. (2020) *Weighted Average Definition, Formula & Examples - Lesson | Study.com*, study.com. Available at: <https://study.com/academy/lesson/calculating-weighted-average-method-formula-example.html#:~:text=Simply%2C%20in%20order%20to%20find,equal%20to%201%20or%20100%25>. (Accessed: 1 March 2024).

CDB3033N DATABASE PROGRAMMING

MARKING RUBRIC

ASSIGNMENT [1]

SCRIPTING & STORED PROCEDURE

Section (1)-12%

<p style="text-align: center;">CDB3033N DATABASE PROGRAMMING</p> <p style="text-align: center;">MARKING RUBRIC</p> <p style="text-align: center;">ASSIGNMENT [1]</p> <p style="text-align: center;">SCRIPTING & STORED PROCEDURE</p>											
Section (2)-18%											
LEARNING OUTCOME	MARKING CRITERIA	SCALE							YOUR MARKS/COMMENTS		
		Fail (0-49)	3 rd Class (50-59)	2 nd Lower Class (60-69)	2 nd Upper Class (70-79)	1 st Class (80-100)	100%	Weightage	Actual Marks		
CLO2	Stored Procedure (50%)	The procedure implemented with major flaws	The procedure implemented with some flaws	The procedure implemented with minor flaws	Good implementation in the procedure but not in exceptional way	The procedure implemented with excellent result and fulfil all the assignment requirements			0.5		
	Screen shots & Test Cases (10%)	Minor or no screen shots and test cases provided.	Some screen shots and test cases provided but with some flaws	Appropriate screen shots and test cases provided but with some flaws	Good screen shots and test cases provided but with minor flaws	Excellent screen shots and test cases provided with clear explanation			0.1		
Total (60%)											
Overall Score (100%)											