Assignment 2: Develop and Test a Coded Solution

Objective:

To take someone else's coded solution and get it to a successful working state. This will involve resolving any syntax issues as well as logical programming issues.

Instructions:

- This assignment can be completed individually or in a group (there is no size limit for the group).
- The final deliverables will be a report and the corrected code that are submitted to the corresponding Brightspace assignment submission folder. Details on the expectations of this report are provided later in this assignment document.
- You have been given the following pieces of information to work with:
    - A flowchart of what the program must do.
    - Scripts to create the table set structure for this problem as well as initial data for these tables.
    - A copy of what the data looks like before running the program, using the provided data.
    - A copy of what the data should look like after the program has executed, using the provided data.
    - A copy of the coded solution you must test and correct.

Suggestions:

- Make sure to refresh the tables and data between each execution of the source code when correcting logic errors.
- Correct and document any syntax errors.
- Correct and document any logic errors.
- The solution is correct when the provided expected data results are achieved from the execution of your coded solution.
- Code change restrictions:
    - Cannot hard code any values that are not already hard coded.
    - Cannot rewrite/recreate the entire solution from scratch; must fix the provided code.

Assignment Components:

1. Syntax Error Identification, Explanation, and Correction (for each syntax error – there are five):
    - Include in the report (for each syntax error):
        - A screenshot of the code section that contains the syntax error.
        - A screenshot of the error message received from this syntax error.
        - Written explanation identifying the reason for the syntax error (1 – 2 sentences).
        - Written explanation identifying the correction to fix this syntax error (1 – 2 sentences).
        - A screenshot of the code section that contains the corrected code.
    - An example of what this entry would look like for each syntax error is provided in ***Appendix A*** at the end of this document.

Code Section in Error:

2. Logic Error Identification, Explanation, and Correction (for each logic error – there are five):
    - Include in the report (for each logic error):
        - The expected result.
        - The actual result.
        - A screenshot of the code section that contains the logic error.
        - Written explanation identifying the reason for the logic error (1 – 2 sentences).
        - Written explanation identifying the correction to fix this logic error (1 – 2 sentences).
        - A screenshot of the code section that contains the corrected code.
    - An example of what this entry would look like for each logic error is provided in ***Appendix B*** at the end of this document.
3. Additional Recommendations:
    - Include in the report:
        - Provide additional recommendations on what you believe still needs to be improved with this code (think about good programming form).
        - Identify two additional recommendations.
        - Should be half to one-and-a-half pages.
4. Final Coded Solution:
    - Provide your final coded solution with all the corrections.
    - Submit as a separate SQL (text) file.

Deliverables:

- A report in PDF format that includes:
    - Cover page that includes:
        - The name of each group member who participated.
    - Syntax Error Identification, Explanation, and Correction.
    - Logic Error Identification, Explanation, and Correction
    - Additional Recommendations.
- SQL (text) file containing the final corrected code.

Report Format:

- The report should use Times New Roman 11-pt font; double-spaced
- The report should be logically organized and cleanly formatted showing all requested information.
- Make sure to review **Appendices A and B** for examples of how to document syntax and logic error information.

Submission:

- Submit your report as a PDF document and the SQL file to the corresponding assignment submission folder.
- If working in a group of more than one person, only one person in the group needs to submit.

Marking Rubric:

Criteria	2 (Complete)	1 (Incomplete)
Syntax Error Identification, Explanation, and Correction	Syntax errors are identified, explained, and corrected showing this information as requested in this document (See *Appendix A* for an example).	Requested documentation is incomplete as not all requested information is provided.
Logic Error Identification, Explanation, and Correction	Logic errors are identified, explained, and corrected showing this information as requested in this document (See *Appendix B* for an example).	Requested documentation is incomplete as not all requested information is provided.
Additional Recommendations	Provides at least two additional recommendations for this coded solution regarding good programming form.	Provides one additional recommendation for this coded solution regarding good programming form.
Overall Presentation	Well-organized, clear, and free of grammatical errors.	Somewhat organized, with several grammatical errors.
Corrected Coded Solution Provided	SQL file submitted contains the corrected solution.	**There are no part marks for this criterion.**

Total Score Calculation

- **Syntax Error Identification, Explanation, and Correction**: ____ / 10
- **Logical Error Identification, Explanation, and Correction**: ____ / 10
- **Additional Recommendations**: ____ / 2
- **Overall Presentation**: ____ / 2
- **Corrected Coded Solution Provided**: ____ / 2

Total Score: ____ / 26

Note: If a section is missing, this section will receive a zero (0) mark.

Note: If a syntax or logic error is not identified or misidentified, no marks will be received for this error (each error has a maximum value of 2 marks).

Appendix A (Syntax Error Documentation Example):

Syntax Error 1

Code Section in Error:

```
BEGIN
  SELECT SUM(sal) / COUNT(empno)
    INTO v_average
    FROM emp_employee;

  DBMS_OUTPUT.PUT_LINE(v_average);
END;
/
```

Error Message Received:

```
Error at line 2/14: ORA-06550: line 2, column 14:
PL/SQL: ORA-00904: "SAL": invalid identifier
ORA-06512: at "SYS.WWV_DBMS_SQL_APEX_220200", line 828
ORA-06550: line 2, column 3:
PL/SQL: SQL Statement ignored
ORA-06550: line 6, column 24:
PLS-00201: identifier 'V_AVERAGE' must be declared
ORA-06550: line 6, column 3:
PL/SQL: Statement ignored
ORA-06512: at "SYS.DBMS_SYS_SQL", line 1658
ORA-06512: at "SYS.WWV_DBMS_SQL_APEX_220200", line 813
ORA-06512: at "APEX_220200.WWV_FLOW_DYNAMIC_EXEC", line 2046
```

Reason for Error:

- The SELECT…INTO is trying to access a column called SAL, but the database does not recognize this column.

Correction:

- The column name is SALARY, not SAL.

```
BEGIN
  SELECT SUM(salary) / COUNT(empno)
    INTO v_average
    FROM emp_employee;

  DBMS_OUTPUT.PUT_LINE(v_average);
END;
/
```

Appendix B (Logical Error Documentation Example):

Logical Error 1

Expected Result:

- Should output to the screen: 2084.885

Actual Result:

- What is being outputted to the screen: 1925.6

Code Section in Error:

```
DECLARE
  v_average  NUMBER;
BEGIN
  SELECT SUM(salary) / COUNT(empno)
    INTO v_average
    FROM emp_employee;
```

Reason for Error:

- From the value that is expected, it looks like the average salary is being calculated for all departments except for number 40.

Correction:

- Add a WHERE clause excluding department 40.

```
DECLARE
  v_average  NUMBER;

  k_dept_omit  CONSTANT emp_employee.deptno%TYPE := 40;

BEGIN
  SELECT SUM(salary) / COUNT(empno)
    INTO v_average
    FROM emp_employee
   WHERE deptno <> k_dept_omit;

  DBMS_OUTPUT.PUT_LINE(v_average);
END;
/
```