CS257 Project Check List Part 2   **Name: _________Kushagra_Bainsla_________**

Part 2 is worth 150 point. You must submit db.h, db.cpp, and this completed checklist. Remember that your program MUST be able to compile (gcc compilers) in a command prompt using a command like “gcc -o db db.cpp” in a GCC container in Windows or Mac. Below are the basic test scenarios which you should have performed. Put an "X" next to the each test scenario which your program did NOT handle successfully. Leave it blank ONLY if it is 100% successful. This list does NOT include all the tests which I will perform. Therefore you can still get points taken off even you pass all the tests in this list. If you have any special instructions for the TA, put your comments at the Notes section.

Normal test scenarios - no error input, validate dbfile.bin size and table contents after each step

01. ___ Create 1 table, “create table class(Student_Name char(20) NOT NULL, Gender char(1), Exams int, Quiz_Total int, Total int NOT NULL)”. Insert 15 rows of good data in random order base on Student_Name (check data file content). Test simple select * statement, “e.g. select * from class”. Make sure select output is formatted with headings. Strings are left justified. Integers are right justified. Test simple single column SELECT, “select Student_Name from class”. Test simple multi-column SELECT, “select Student_Name, Total from class”. Verify file content.

02. ___ Test a single row delete, e.g. delete from class where Student_Name = ’Bad_Student’.

03. ___ Test delete where no rows is found, e.g. “0 rows deleted”.

04. ___ Test multi-row delete (3 rows), e.g. delete from class where Total < 100.

05. ___ Test single row update, e.g. update class set Quiz_Total=350 where Student_Name=’David’.

06. ___ Test update when no rows is found, e.g. “0 rows updated”.

07. ___ Test multi-row update (4 rows), e.g. update class set Quiz_Total = 350 where Quiz_Total > 350.

08. ___ Test SELECT with WHERE clause with a single condition.

09. ___ Test SELECT with WHERE clause for case sensitive comparison, e.g. Student_Name < ‘Good_Student’.

10. ___ Test SELECT with WHERE clause with columns that is NULL and NOT NULL.

11. ___ Test SELECT with two conditions separated by the AND keyword.

12. ___ Test SELECT with two conditions separated by the OR keyword.

13. ___ Test SELECT with ORDER BY clause.

14. ___ Test SELECT with WHERE and ORDER BY clause.

15. ___ Test SELECT SUM() function.

16. ___ Test SELECT SUM() function with WHERE clause.

17. ___ Test SELECT AVG() function.

18. ___ Test SELECT AVG() function with WHERE clause.

19. ___ Test SELECT COUNT() function.

20. ___ Test SELECT COUNT() function with WHERE clause.

21. ___ Test SUM(), AVG() when there are NULLs.

22. ___ Test COUNT(*), COUNT(Quiz_Total) when there are NULLs.

Repeat the above SELECT statement tests with NATURAL JOIN as well

Error test scenarios - error condition must not cause any exception

23. ___ Check syntax errors in various DELETE statement combinations

24. ___ Check syntax errors in various UPDATE statement combinations

25. ___ Check syntax errors in various SELECT statement combinations

26. ___ Catch “data type mismatch” errors on INSERT.

27. ___ Enforce NOT NULL condition on INSERT.

28. ___ Enforce NOT NULL condition on UPDATE.

29. ___ Catch “data type mismatch” errors on WHERE clauses of update, delete, select.

30. ___ Catch “invalid data value” errors.

31. ___ Catch “invalid relational operator” and “invalid aggregate function parameter” errors.

Additional confession/information about your project: