**CS257 Project Check List Part 2 Name: Dharahas Tallapally**

**Part 2 is worth 130 points and it is due on 11/16/21.** 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 db.cpp” in either 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.

Which OS (Mac or Windows) \_\_\_\_Mac\_\_\_\_\_\_\_\_ Which compiler did you use \_\_\_\_\_\_\_clang++\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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.

Error test scenarios - error condition must not cause any exception

23. \_\_\_ Check syntax errors in various INSERT 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: