COMP9315 22T1 Final Exam

The University of New South Wales

COMP9315 DBMS Implementation 22T1 Final Exam

DBMS Implementation

[Instructions] [PostgreSQL] [C] [Q1-3 Info]
[Q1] [Q2] **[Q3]** [Q4] [Q5] [Q6] [Q7] [Q8]

Question 3 (25 marks)

The `q3` directory contains an incomplete program `q3.c`. You must complete this program so that it reads tuples from a file and inserts them into newly-created "no-frills" data file (see [Q1-3 Info] for details on what a "no-frills" data file is). The program takes two command-line arguments: the first is the name of the "no-frills" data file to be created, the second is the name of the file containing the tuples, e.g.

```
$ ./q3 myData < data/insert200
```

Insert each tuple into a page buffer, which is initially all zeroes. Tuples are inserted directly one after another, where the next tuple occurs immediately after the `'\0'` that terminates the previous tuple. When there is insufficient space to add the next tuple, write out the page buffer, clear it, and start inserting tuples into it. Note that the last tuple in a page should be followed by a `'\0'` character, which follows immediately after the tuple's own terminating `'\0'` character.

If the tuples file is empty, you must write a data page containing no tuples; you cannot leave the file totally empty (i.e. it must have at least PAGESIZE bytes).

The command-line argument processing is already handled in `q3.c`. By the time it reaches the code you need to write, you will have a valid file descriptor for the "no-frills" data file and a valid file pointer for the tuple file. There are examples of using `./q3` in the `tests` directory. You should also be able to devise your own test cases easily enough.

Note that you need to maintain the tuple count in each page as you add tuples. The data file you build should be able to be examined by the `q1` command and return the correct number of tuples.

To help you check whether your program is working correctly, there is a script called `run_tests.sh` which will run the program against the test cases in the `tests` directory and report the results. It will also put the data file created by your program in the current directory (i.e. your `q3` directory). The example `data/inputX` files create the `data/DataY` files in the `data` directory. The testing script then compares the created data file against the relevant data file and reports whether it matches this file. If the created data file is not identical to the actual data file, then the script reports a failure.

You can examine the test cases by looking at the `tests/*.sh` files. You can run the testing script as:

```
$ sh run_tests.sh
```

Once your function is working (passes all tests), follow the submission instructions below. Even if it fails some (or even all) tests, you should submit because you can get *some* marks. If your program does not compile, or if you simply submit the supplied code, then your "answer" is worth zero marks.

Submission Instructions:

- Type your answer to this question into the file called `q3.c`
- Submit via: **`give cs9315 exam_q3 q3.c`**
  or via: Webcms3 > exams > Final Exam > Q3 submission > Make Submission

End of Question