Project 1

Due: 5PM 06 Oct 2025 (Mon)

1. **Introduction.** This project is to design and implement the following two components of a database management system, storage and indexing.
   (1) For the storage component, the following settings are assumed.
   - the data is stored on **disk**;
   - to avoid the risk of corrupting the file system on your disk, you can use either a disk image file or a dummy binary file to simulate a disk;

   (2) For the indexing component, the following settings are assumed.
   - a B+ tree is used; the B+ tree should be stored on **disk**; and
   - for building the B+ tree, you can use either the method of iteratively inserting the records **OR** the bulk loading method;

2. **Implementation and Experiments.**
   (1) **Task 1**: Design and implement the storage component based on the settings described in Part 1 and store the data (which is about NBA games and will be described in Part 4).
   - Describe the content of a record, a block, and a database file;
   - Report the following statistics: the size of a record; the number of records; the number of records stored in a block; the number of blocks for storing the data;

   (2) **Task 2**: Design and implement the indexing component based on the settings described in Part 1 and build a B+ tree on the data described in Task 1 with the attribute "FT_PCT_home" as the key.
   - Report the following statistics: the parameter n of the B+ tree; the number of nodes of the B+ tree; the number of levels of the B+ tree; the content of the root node (only the keys);

   (3) **Task 3**: Delete those movies with the attribute “FT_PCT_home” above 0.9 using the B+ tree
   - Report the following statistics: the number of index nodes the process accesses; the number of data blocks the process accesses; the number of games deleted; the average of “FT_PCT_home” of the records that are returned; the running time of the retrieval process; the number of data blocks that would be accessed by a brute-force linear scan method (i.e., it scans the data blocks one by one) and its running time (for comparison);
   - Also report the following statistics of the updated B+ tree: the number of nodes of the B+ tree; the number of levels of the B+ tree; the content of the root node (only the keys);

Note 1. C/C++ is recommended for this project, but other programming languages including Java and C# are also acceptable.

3. **Materials to submit include:**

   (1) **A report** (at most 20 pages long) including:

   - Design of the storage component and the B+ tree component. It is suggested to use some figures to illustrate the designs.
   - Results of the tasks in Part 2;
   - Individual contributions (presented on the first page of the report);

   (2) **A video** (at most 10 mins long) covering:

   - Explanation on the design, implementation, and the codes (e.g., some key functions); and
   - Demonstration of the execution processes in Task 1, 2, and 3;

   (3) **Source code**:

   - You must attach an installation guide to ensure that your code can be run successfully.

Note 2. For the code and video, you need share them via OneDrive or GoogleDrive URLs and include the URLs on the first page of the report. Make sure you grant the users with the URLs the access permission and do **NOT** share the video and the code with the public (e.g., via YouTube and GitHub).

Note 3. For the presentation of the experimental results, please present them in the form of tables for easier checking. Please make the presentation of your report clean, sorted, and consistent as much as you can. These would be considered for assessing the presentation of the report.

4. **Data**.

   - The data could be downloaded via this link: https://www.dropbox.com/scl/fi/s4wgb8uspaq1bog6tbyby/games.txt?rlkey=gmc0i28bs53mmxovcewpc4mlx&dl=0
   - The data contains the statistics of NBA teams in 2003 – 2022. The first line in each file contains headers that describe what is in each column.

5. **Submission policy.**

   (1) All submissions should be uploaded to NTULearn (a submission slot shall be created later on).

   (2) Late submissions will be penalized by 5% deduction per day for at most 7 days. Beyond 7 days after the deadline, no submissions will be accepted.

(3) It is not allowed to copy or refer to public code repositories. Strict plagiarism will be conducted. Any found plagiarism will mean a failing grade and be subject to further disciplinary actions.

(4) Finally please declare if any generative AI tools are used and to what extent (on the first page of the report).