

## Exercise 2

Released: April 27, 2020 · Discussion: May 4, 2020

### 1 Buffer Manager — Replacement Strategies

The lecture discussed the functioning of the *buffer manager*.

- What are the responsibilities of the buffer manager?
- Describe multiple strategies used to fulfill these responsibilities.
- What are their advantages and disadvantages?

### 2 Replacement Strategies — Implementation

In the lecture different replacement strategies for buffer managers were discussed. In this exercise, you have to implement at least one of the following replacement strategies for a simple buffer manager:

- *Least Recently Used (LRU)*
- *Least Frequently Used (LFU)*
- *LRU-k*

Download the file `BeeDB` project from the courses moodle website<sup>1</sup>. The DBMS contains no replacement strategy for the buffer manager right now, because of that it is not possible to do select queries from bigger tables.

- Run query `SELECT * FROM genre;` will work without any strategy because the buffer has enough frames
- Run query `SELECT * FROM movie;` will crash, due to the need to evict pages from the buffer
- Implement method `find_victim()` for LRU (`src/disk/lru_strategy.cpp`), LFU (`src/disk/lfu_strategy.cpp`) and LRU-K (`src/disk/lru_k_strategy.cpp`)

---

<sup>1</sup><https://moodle.tu-dortmund.de/mod/page/view.php?id=566733>

- Configure used strategy and try different k parameter for LRU-K by beedb.ini
- Afterwards the query `SELECT * FROM movie;` should work, because the pages can be evicted from the buffer

### **Build instructions:**

1. Go to the BeeDB project
2. Paste the bee.db into the project
3. Run `cmake` to create a makefile for your system: `cmake .`
4. Run `make` to create an executable binary file: `make`
5. You can now run the queries using your Linux console for example with the following line:  
`./beedb -q "SELECT * FROM movie;"`
6. More information can be found in the README.md