FRIEDRICH-SCHILLER-
UNIVERSITÄT
JENA

Datenbanksystemimplementierung: Buffer Manager

Adnan Alhomssi

Datenbanken und Informationssysteme
Prof. Dr. Viktor Leis

Task 1: Buffer Manager

- Implement a classical Buffer Manager
- You are free to choose the replacement strategy you like
- Ignore segments and free space inventory for now
- Segments are useful for HDD-backed systems
- Free Space Inventory and new page allocation will be needed later

Task 1: Buffer Manager - Interface

- The interface is simple but powerful
- `fix (page_id, is_exclusive)` returns `BufferFrame`
- `unfix (page_id, is_dirty)`
- Upon this interface, we should be able to host different data structures like B-Tree and slotted pages

Task 1: Buffer Manager - Multi-threaded

- SHOULD be thread-safe
- If you tried hard and did not manage it, then a functional single-threaded implementation will be okay
- Your implementation must be efficient:
  - Hold latches as short as possible
  - Don't hold latches while doing I/O

Task 1: Buffer Manager - Hints

- You can synchronize your data structures using: `std::mutex` and `std::shared_mutex`
- For IO, you can use POSIX API: `open, close, pwrite, pread`