ECS165A

Milestone 2 Overview

Haley Raizes, Brittany Bates,

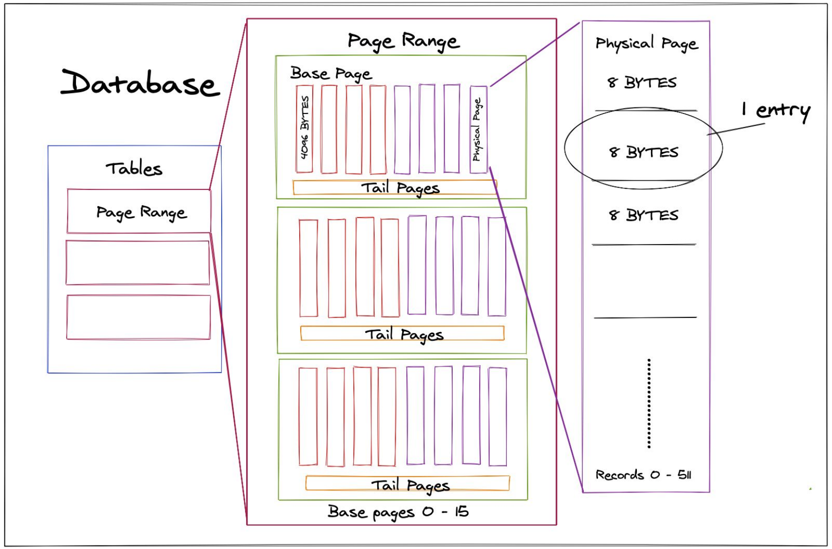
Jim McKerney, Nicholas Chen, Chris Bried

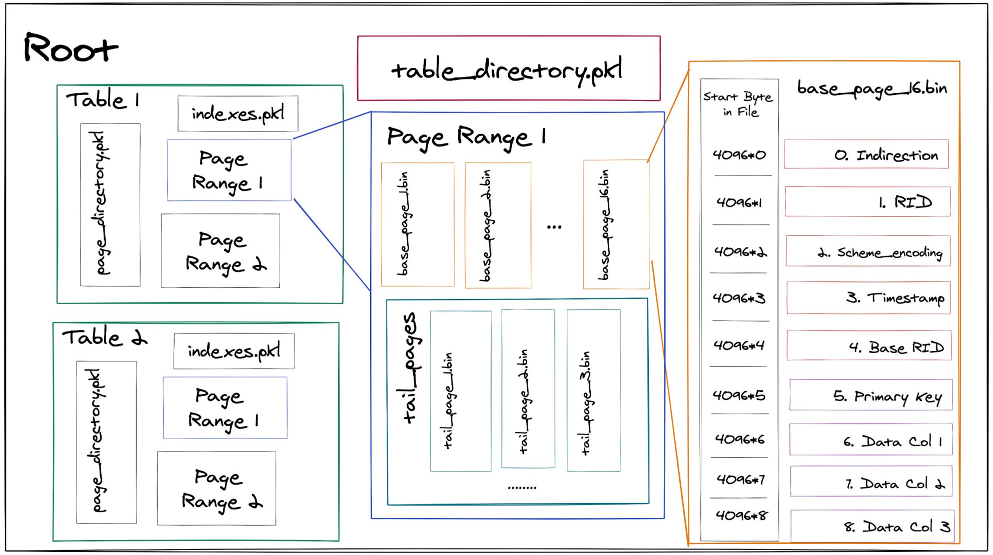
Objectives

- Disk Structure
- Bufferpool Design
- Eviction Policy
- Indexing
- Merge
- Performance
- Questions
- Demo

The Path to Durability

Milestone 1 → Milestone 2



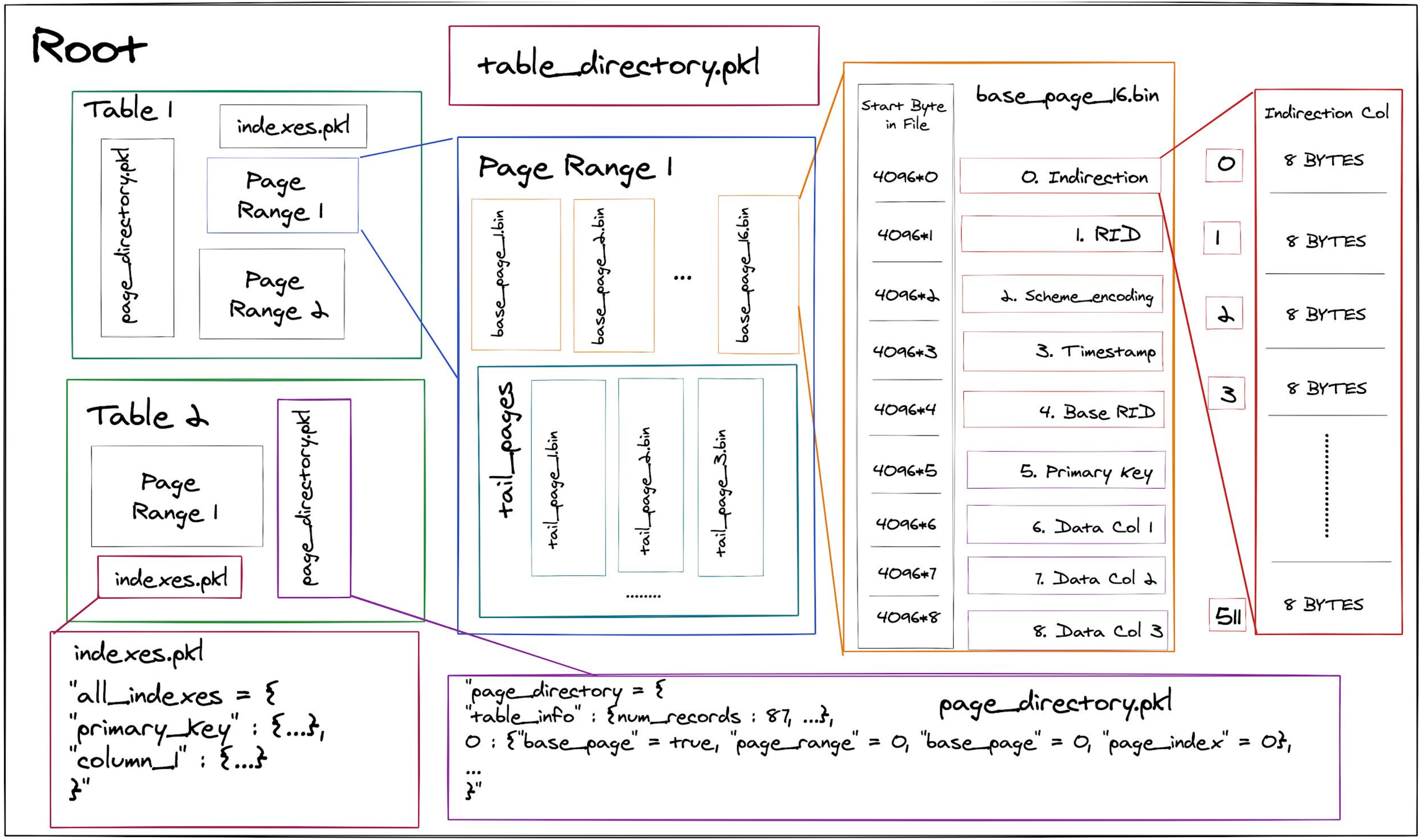


Start Byte in File	base_page_16.bin
4096*0	0. Indirection
4096*1	1. RID
4096*2	2. Scheme_encoding
4096*3	3. Timestamp
4096*4	4. Base RID
4096*5	5. Primary Key
4096*6	6. Data Col 1
4096*7	7. Data Col 2
4096*8	8. Data Col 3

Volatile Memory

Non-Volatile Memory

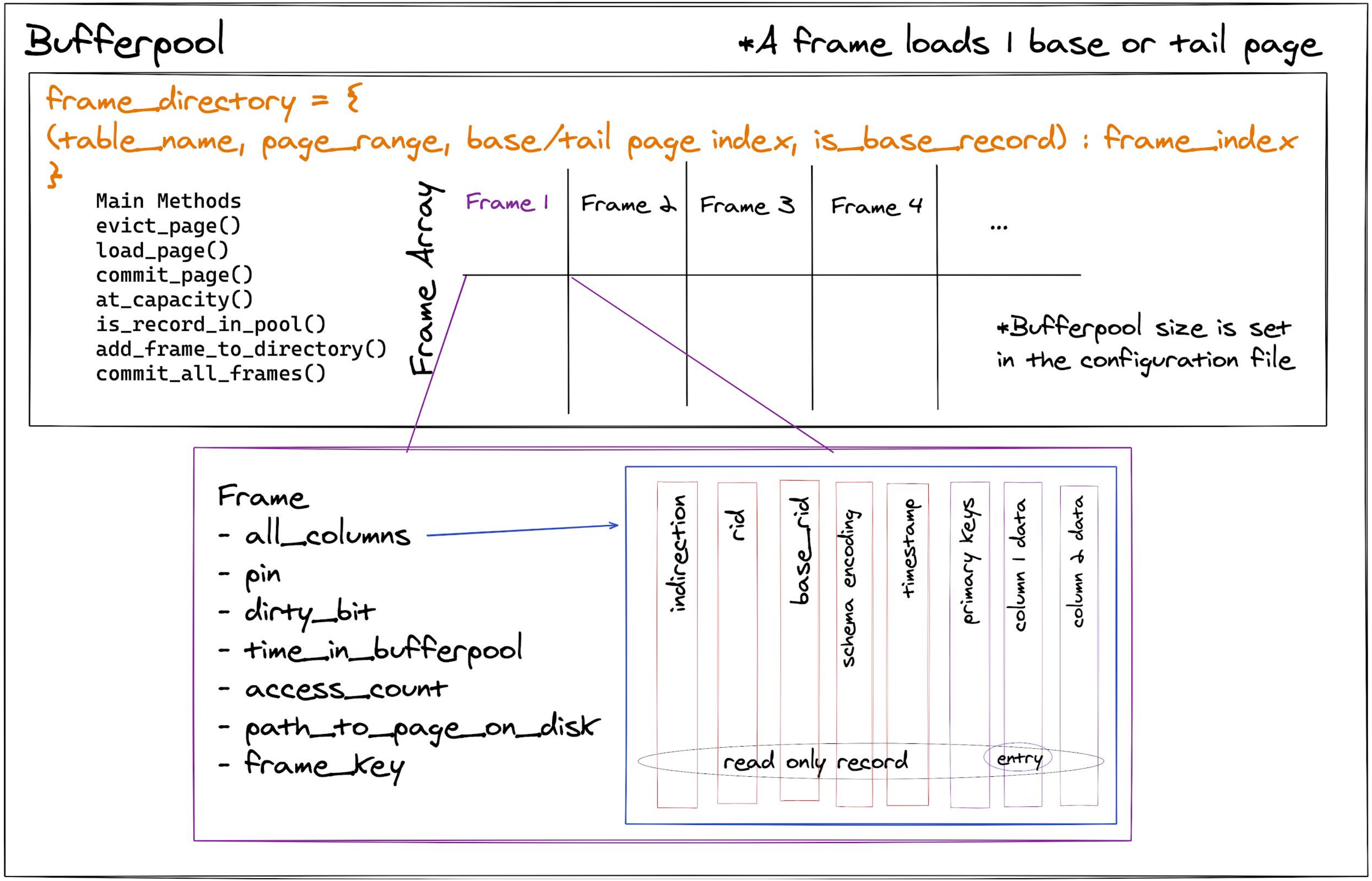
Disk Structure



```
"all_indexes = {
"primary_key" : {...},
"column_1" : {...}
}"
```

```
"page_directory = {
"table_info" : {num_records : 87, ...},
0 : {"base_page" = true, "page_range" = 0, "base_page" = 0, "page_index" = 0},
...
}"
```

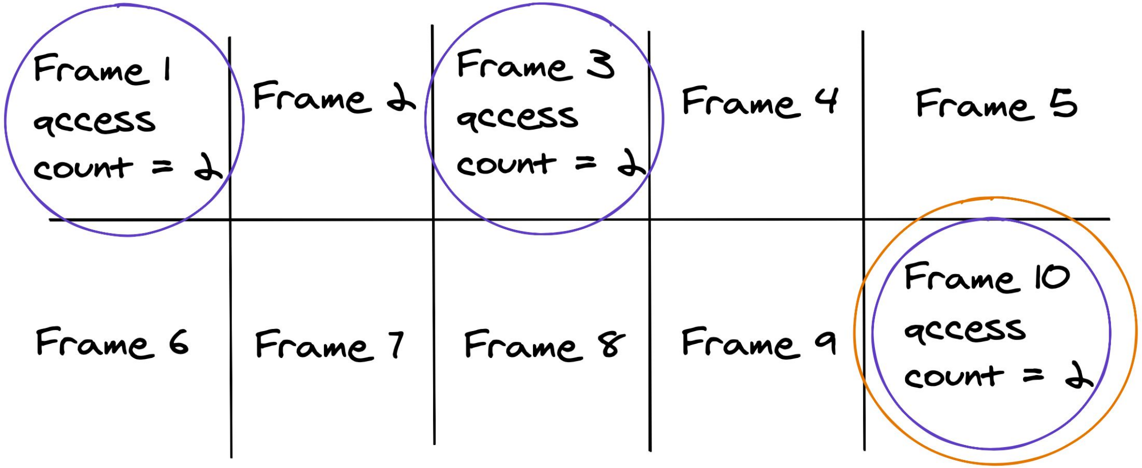
Bufferpool Design



```
Main Methods
evict_page()
load_page()
commit_page()
at_capacity()
is_record_in_pool()
add_frame_to_directory()
commit_all_frames()
```

Eviction Policy

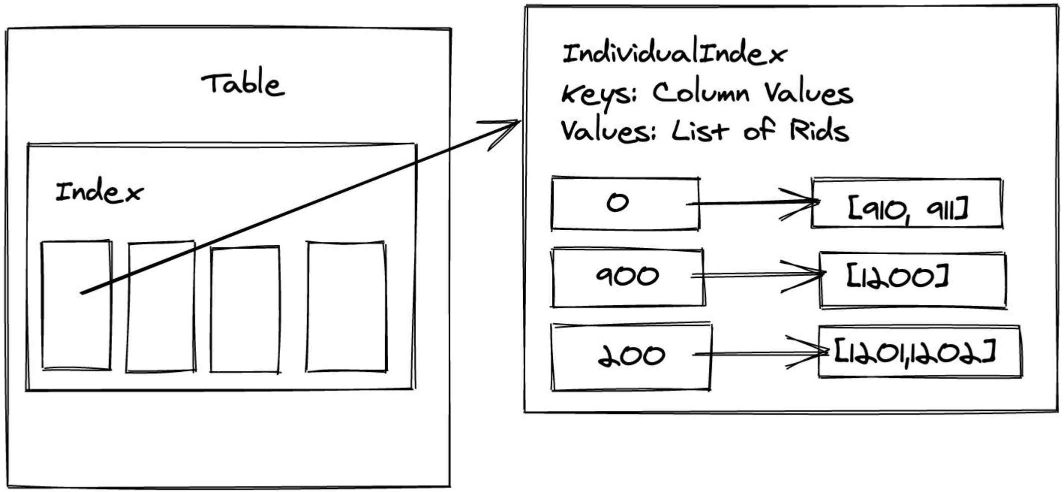
- Synthesis of LRU and LFU policies
  - Among the least frequently used records, evict the least recently used
- Chose this method primarily for speed & simplicity
- Does not distinguish between privileged and unprivileged data



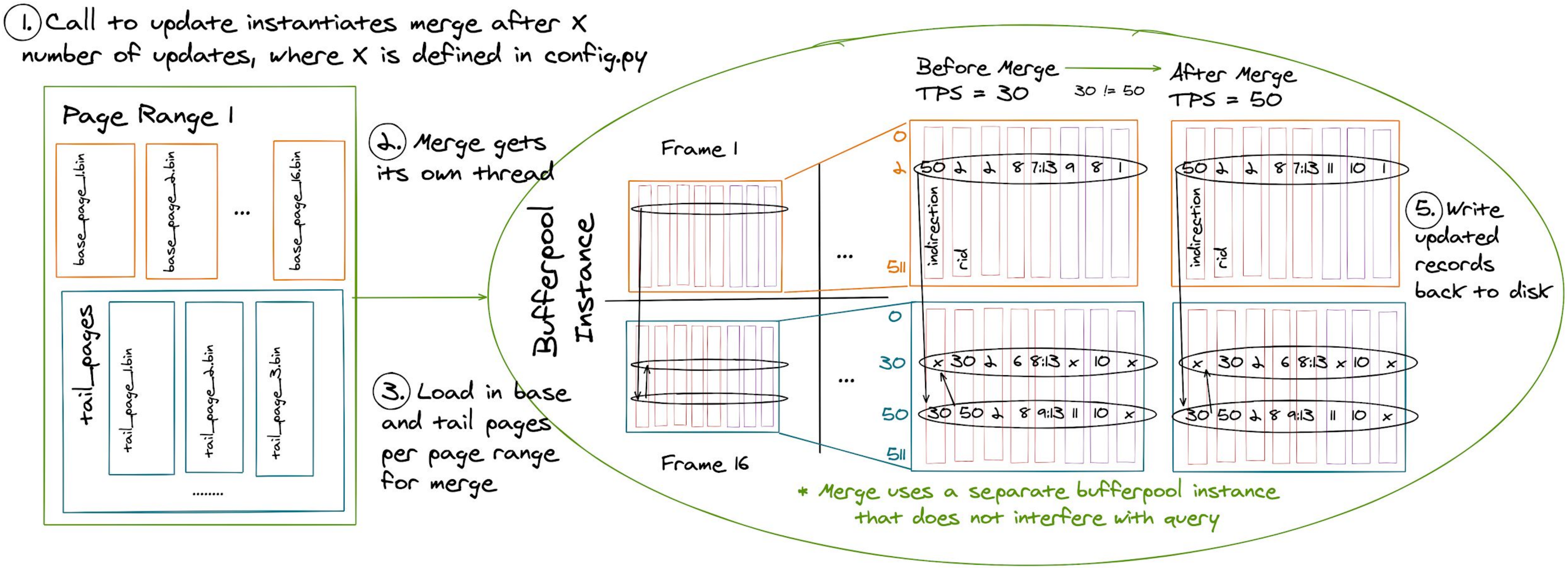
Frame 1 access count = 2	Frame 2	Frame 3 access count = 2	Frame 4	Frame 5
Frame 6	Frame 7	Frame 8	Frame 9	Frame 10 access count = 2

Indexing

- Hash index
- Maps column values to list of RIDs
- Used for select and update
- Index created automatically for primary key
- Bufferpool frames are committed before index creation
- Index persisted as .pkl file



Merge - General Flow



1. Call to update instantiates merge after X number of updates, where X is defined in config.py

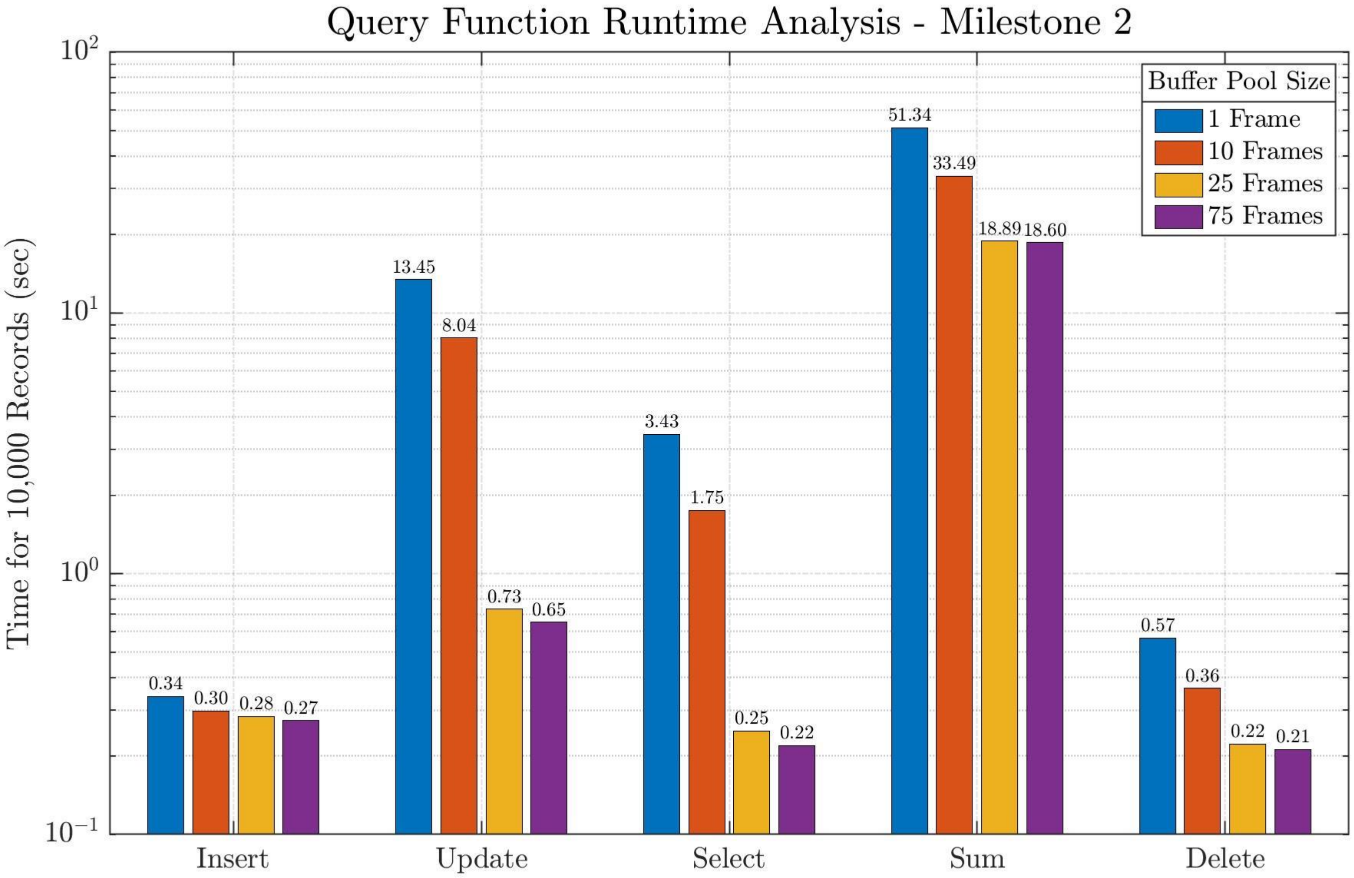
2. Merge gets its own thread

3. Load in base and tail pages per page range for merge

5. Write updated records back to disk

* Merge uses a separate bufferpool instance that does not interfere with query

Optimizations & Performance



*These times are based on 10 run averages using the provided __main__.py

Specs: 6 core Intel Core i7, 2.6 GHz, 256KB l2 cache per core, 12 MB L3 Cache, 16GB Memory

Questions

Demo