L-Store: Milestone 1

ECS 165A: Database Systems

Yiling Chen, Tina Young, Olivia Tobin

3 Main Parts

Data Model

Columnar Data Storage

Base Page vs Tail Page

Page Range

Bufferpool Management

Page Directory

Index Directory

Query Interface

INSERT

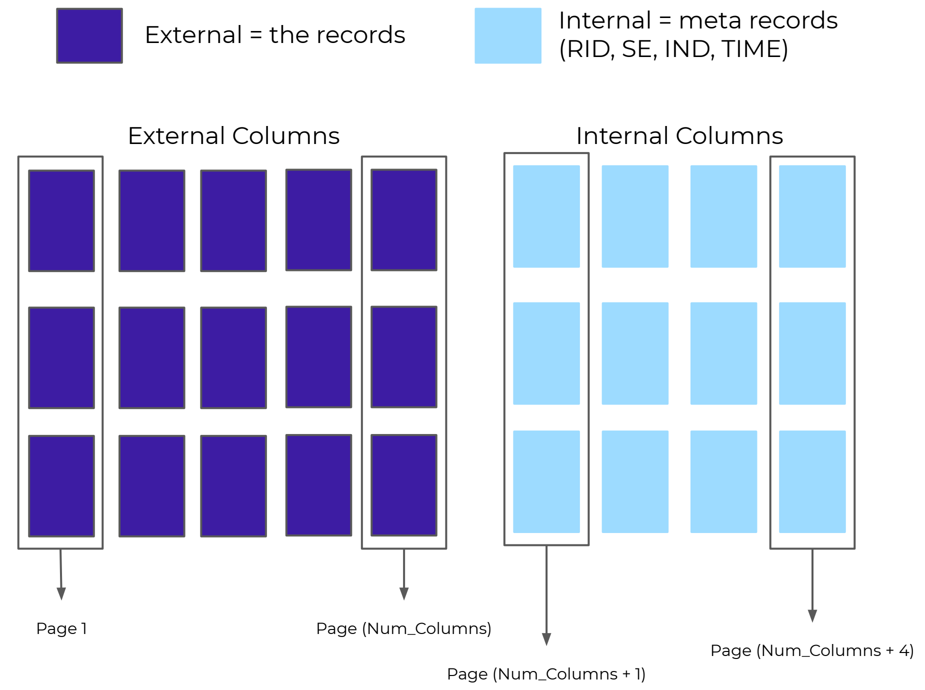
UPDATE

SELECT

DELETE & SUM

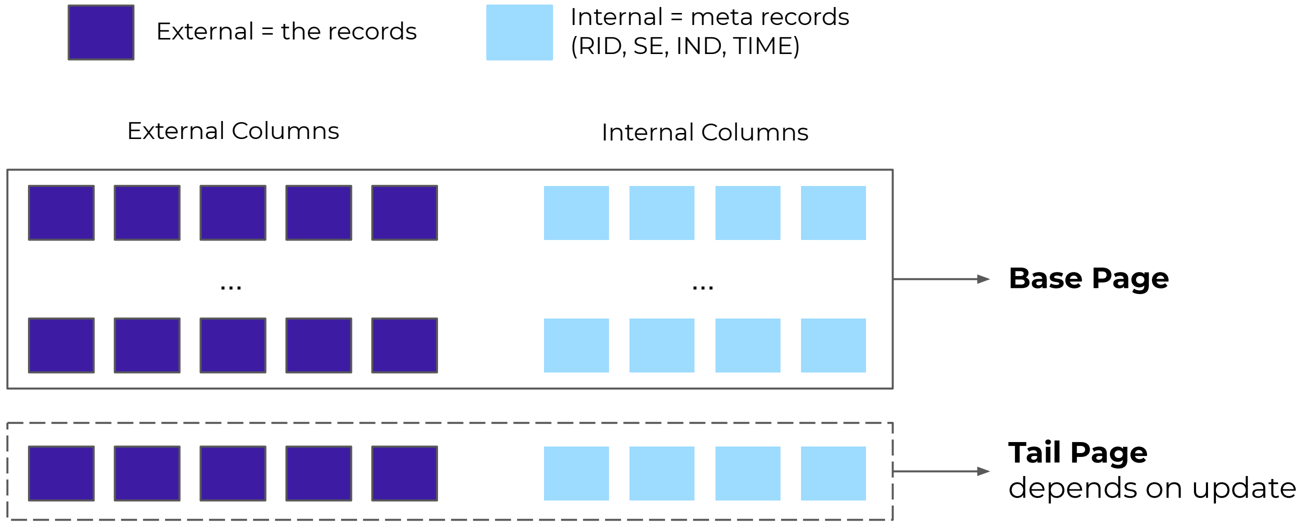
Data Model

Columnar Data Storage

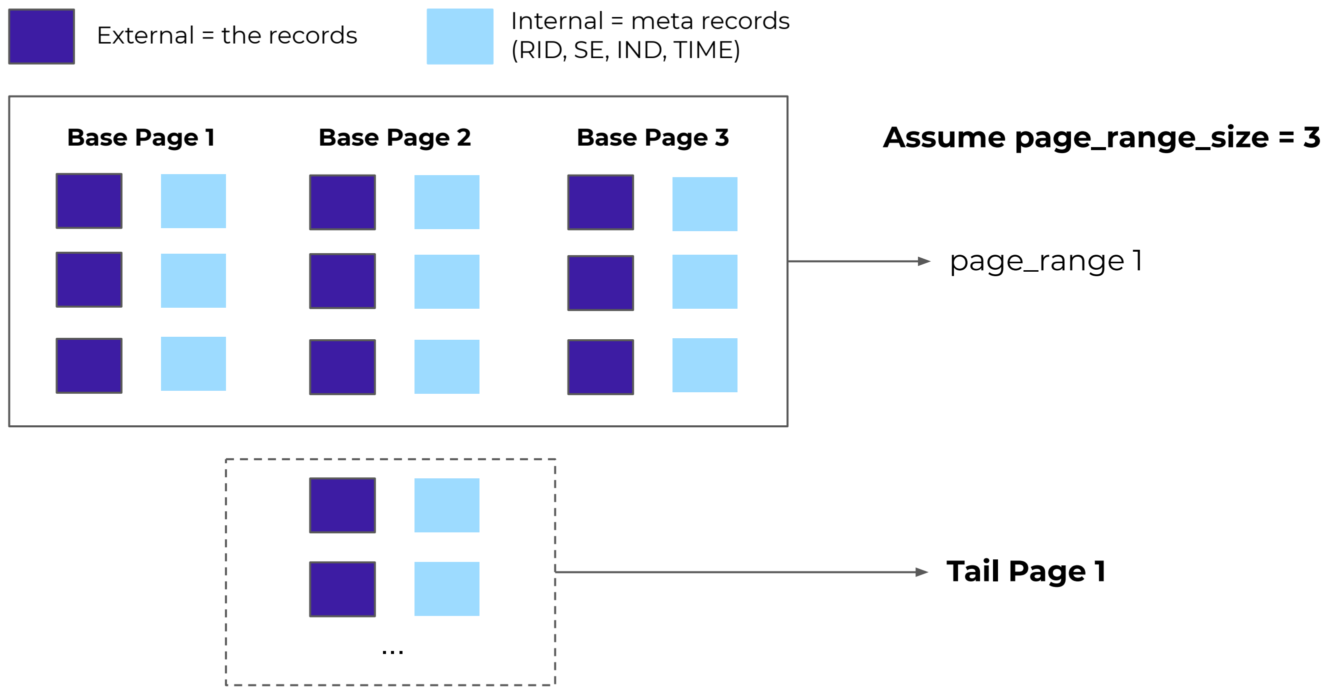


Num_Total_Columns = Num_Columns + 4

Base Page vs Tail Page



Page Range Implementation



Helper Functions: Page

- In `page.py`:
  - `has_capacity()`: This function make sure the page still has capacity to add with increment of 8 (bytes).

- In `table.py`:
  - `checker()`: This function uses `has_capacity()` in order to add pages when capacity of particular page is full.

Helper Functions: Read, Write and Edit

```
def read(offset)
    # use in get_schema_encoding and get_indirection
    return data[offset*8: (offset+1)*8]

def write(offset, value)
    # use to write into both base and tail pages
    data[self.num_records * 8: (self.num_records + 1) * 8] =
    value.to_bytes(8, byteorder='big')
    num_records += 1

def edit(offset, value)
    # use in set_schema_encoding and set_indirection
    data[offset*8: (offset+1)*8] = value.to_bytes(8, byteorder = 'big')
```

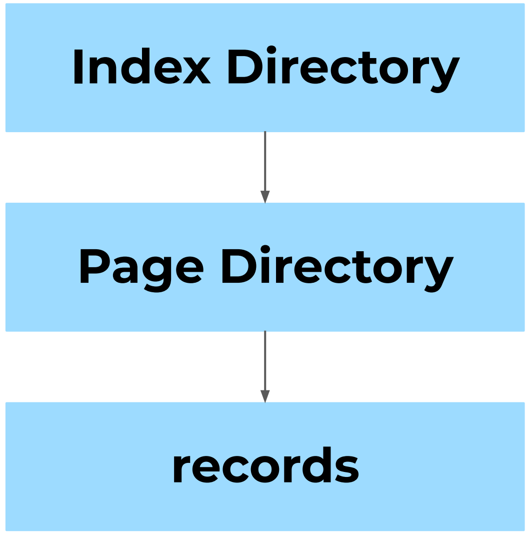
Bufferpool Management

Page vs Index Directory

	Page Directory	Index Directory
Data Structure	Hashmap (Dict)	Hashmap (Dict)
Key	RID	KEY
Value	PageID, Offsets	RID

Page vs Index Directory

In **Milestone 1**, we did not use `column` parameter since we assume that the key is at column 0



Query Interface

Helper Functions: Get and Set + Exist

Getters:

- get_schema_encoding_base(pageID, offset)
- get_indirection_base(pageId, offset)
- get_record_element(pageId, offset, col):

Setters:

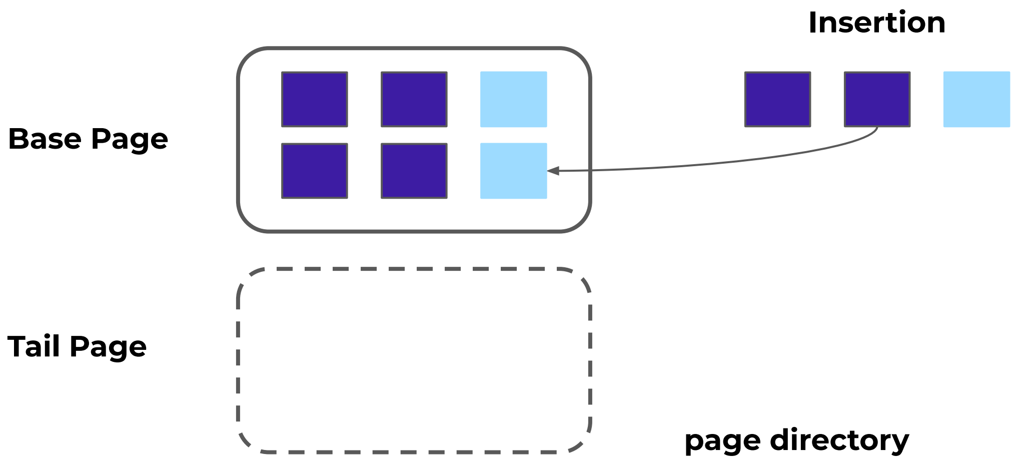
- set_indirection_base(pageId, offset, new_indirection)
- set_schema_encoding_base(pageId, offset, new_schema_encoding)

key_Exists:

- Check if record exists or has been deleted

Query Interface: `INSERT`

```
INSERT(*columns)
```

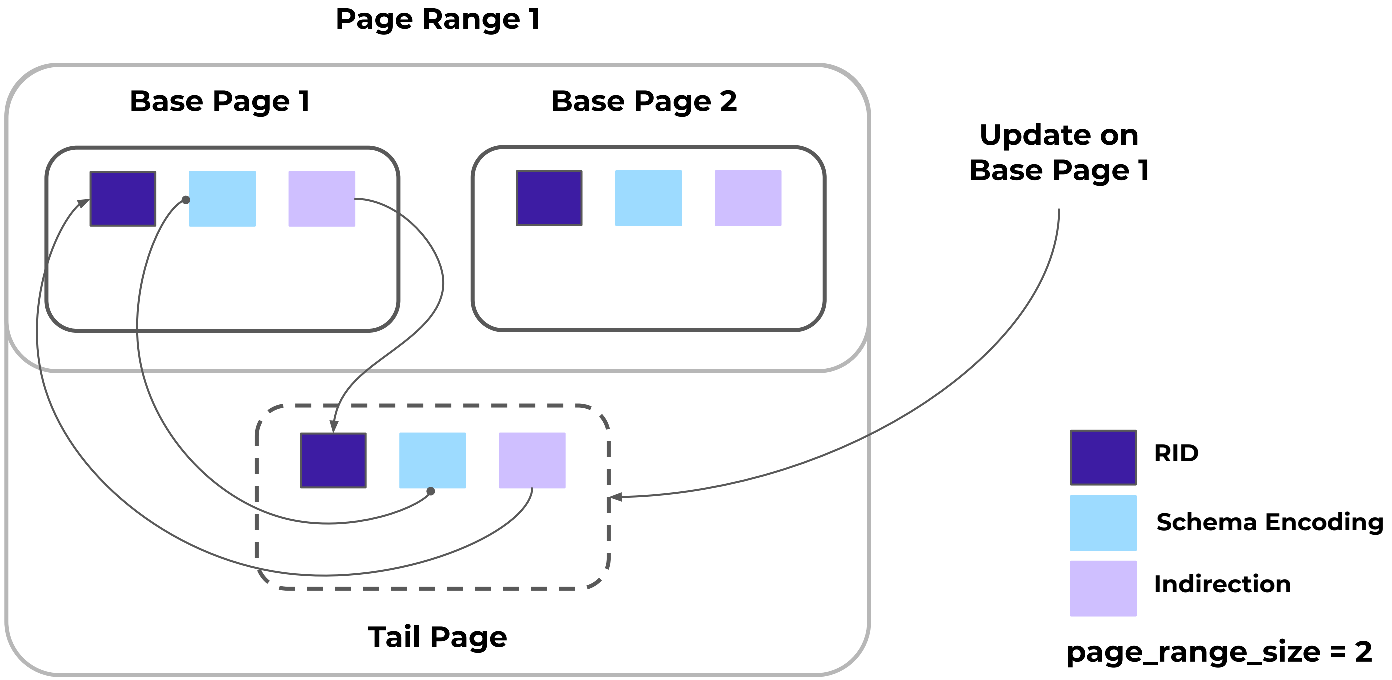


page directory
append (key = RID, value = (pageID, offset))

index directory
append (key = KEY, value = RID)

Query Interface: UPDATE

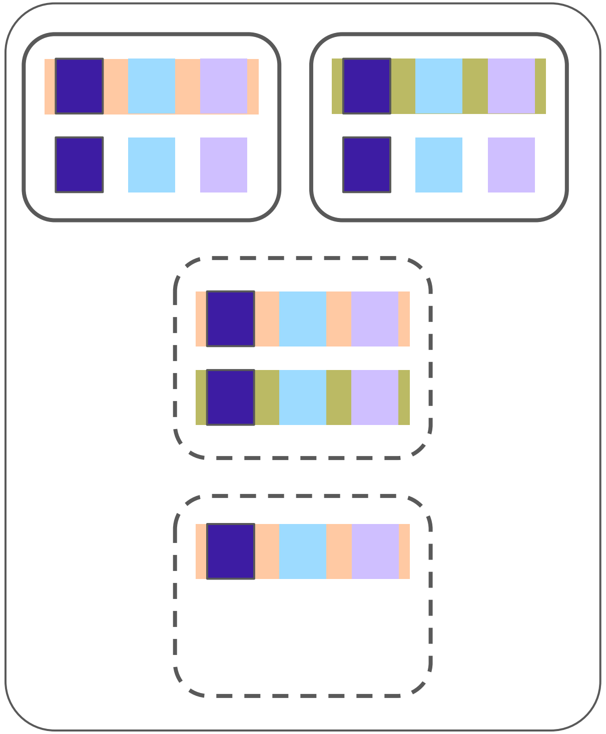
```
UPDATE(KEY, *columns)
```



Query Interface: SELECT

```
SELECT(KEY, column, query_columns)
```

RID	Schema Encoding	Indirection	Value1	Value2	Value3
b1	110	t2	9	5	18
t1	010	b1	maxInt	3	maxInt
t2	110	t1	2	3	maxInt



Query Interface: DELETE, SUM

DELETE(KEY)

Index Directory	RID
Schema_Encoding	'0' * num_columns (for base page)
UPDATE	update(key, *([None] * num_columns))

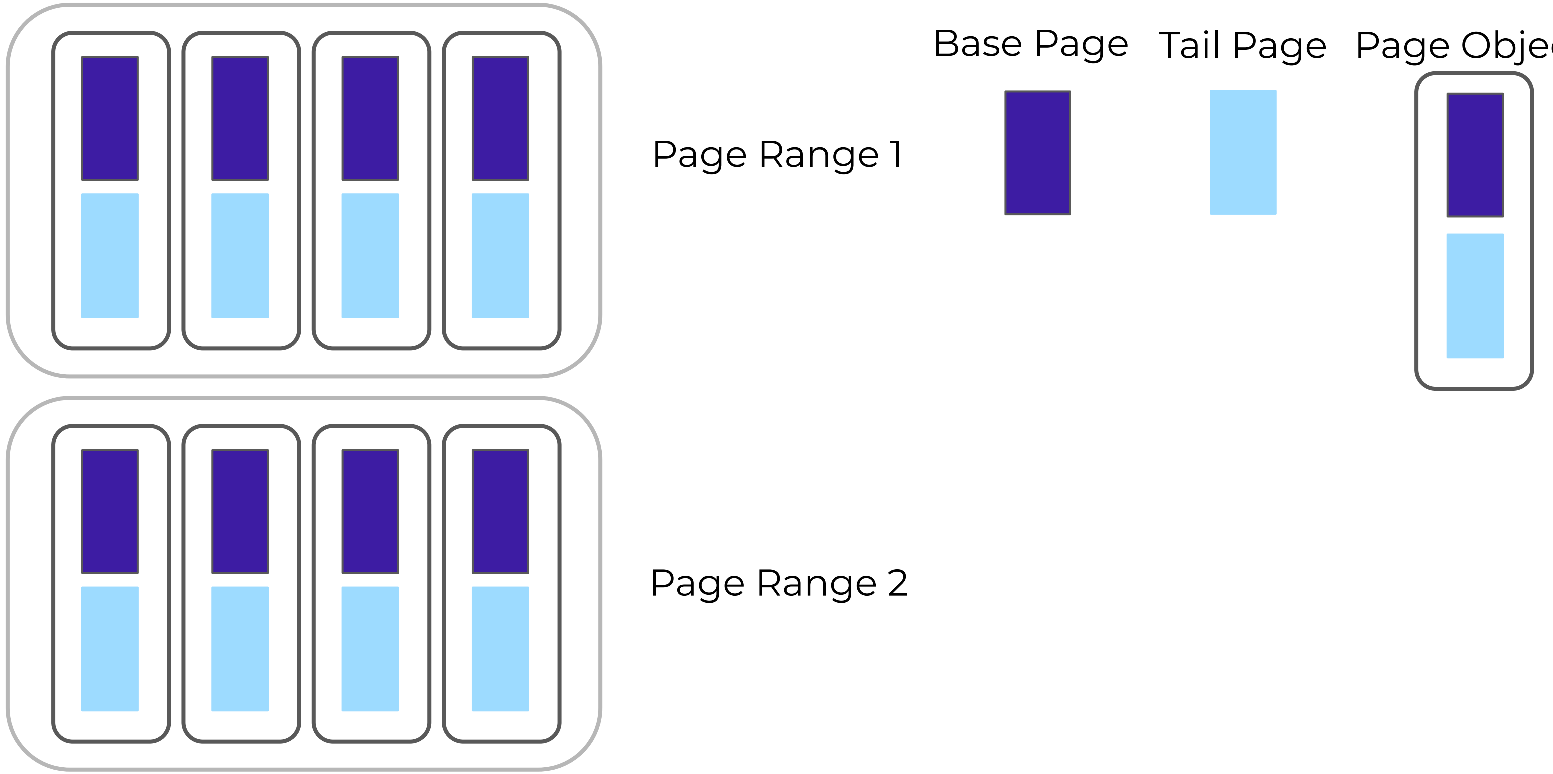
SUM(start_range, end_range, aggregate_column_index))

key_list	[start_range, ... , end_range]
query_column	query_column[aggregate_column_index] = 1
SUM(SELECT)	sum += select(key, 0, query_column)[0].columns[aggregate_column_index]

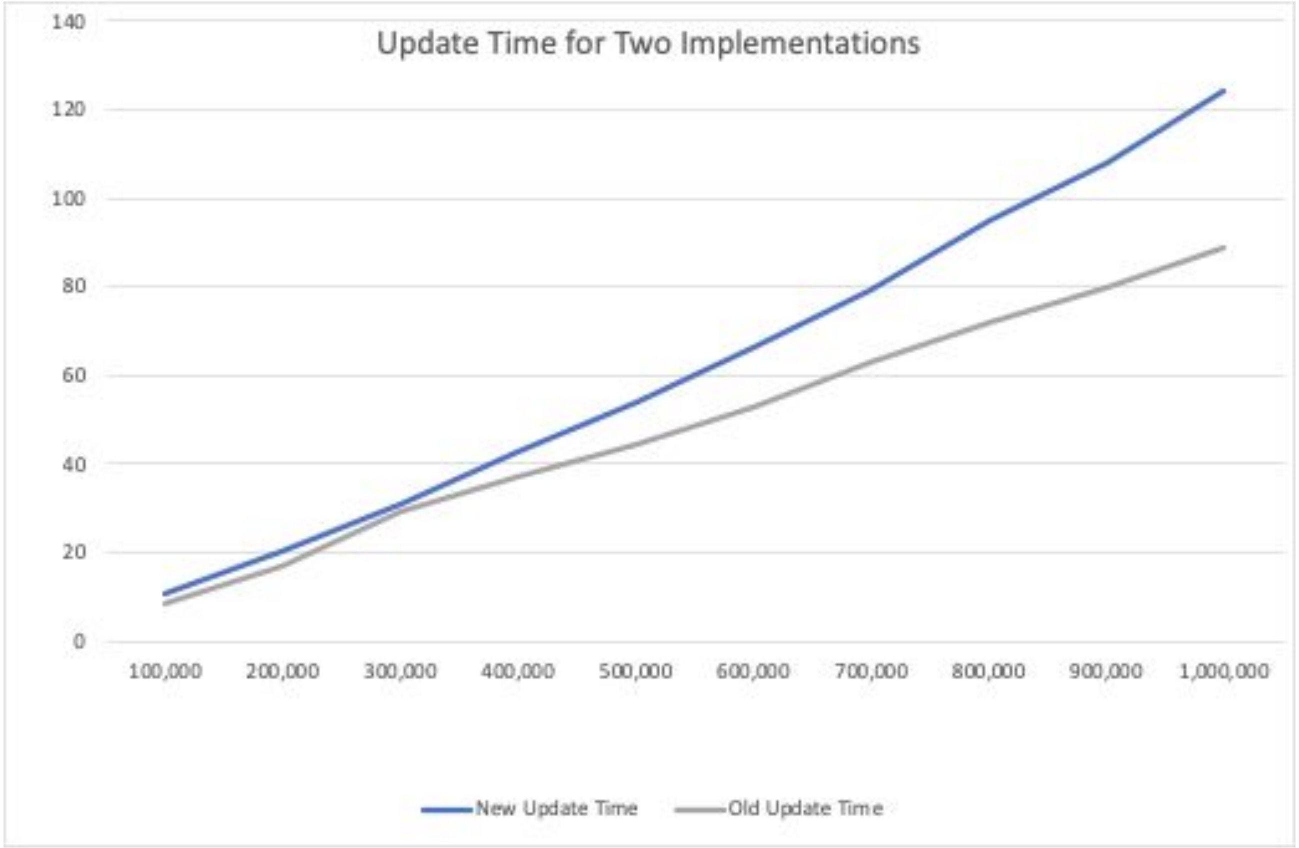
Things to Improve Upon for M2

- Expand query capabilities
    - More functionalities
- Non-int values conversion
- Indexing columns
    - Page Range
    - locate() and locate_range()
    - B-Tree
- Clean up the CODE

Alternate Implementation: Base Page and Tail Page in One Page Object



Alternate Implementation Speed



Thank You!