Milestone 1

Yingchen Gu, Glenn Chen
Rishika Roy, Kaleb Crans,
Ryan Kim

Overview

Data Model

Database Structure

Table: Page Range

Base and Tail Page

Physical Pages

Bufferpool Management

Table: Page Directory

Indexing

Query Interface

Select

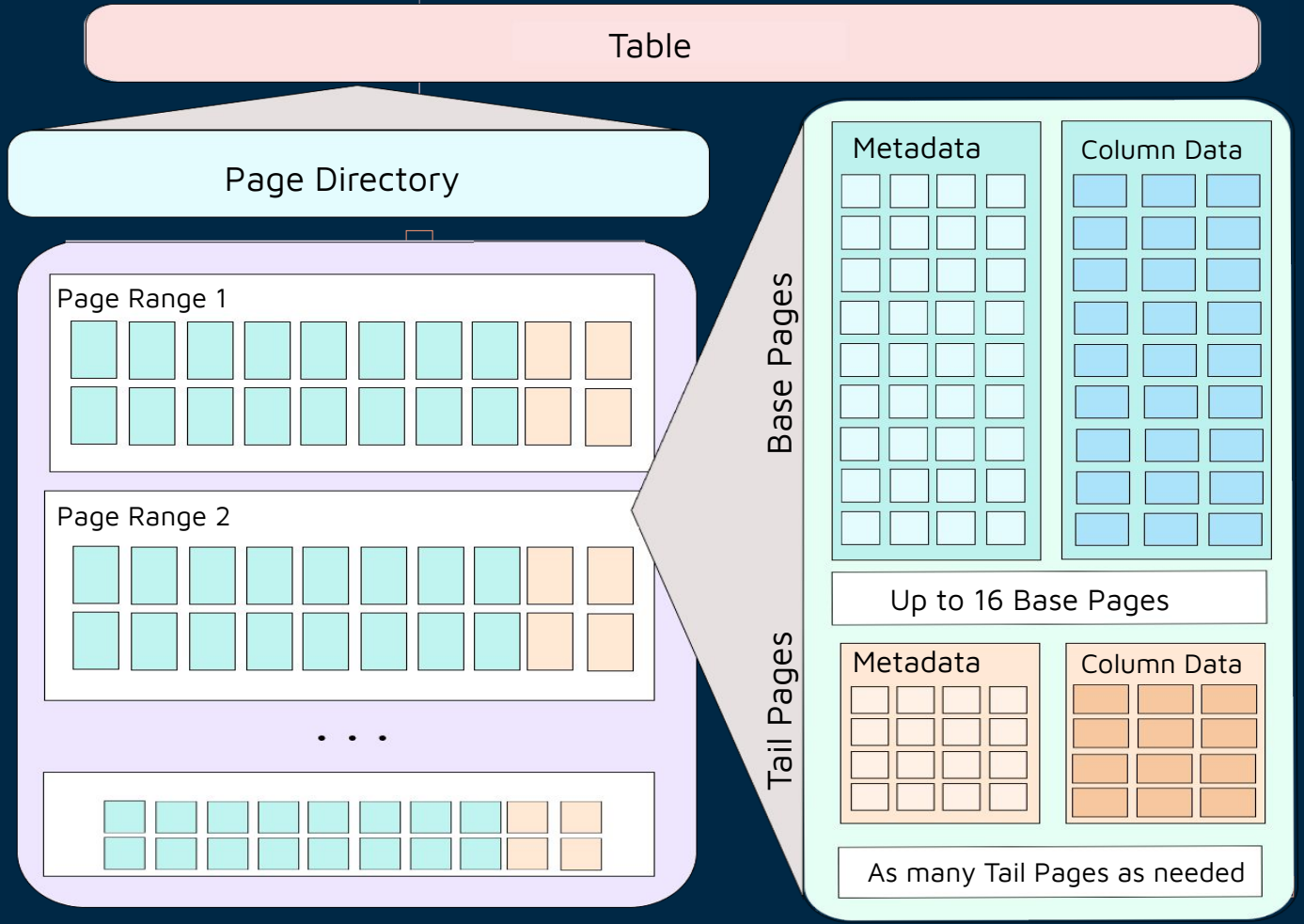
Insert

Update

Delete

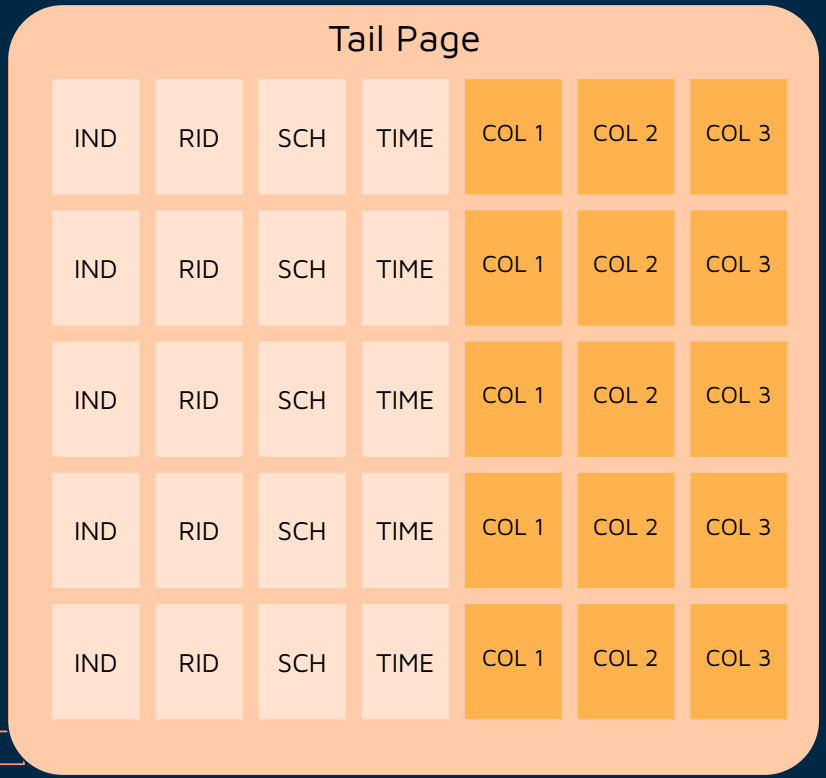
Data Model

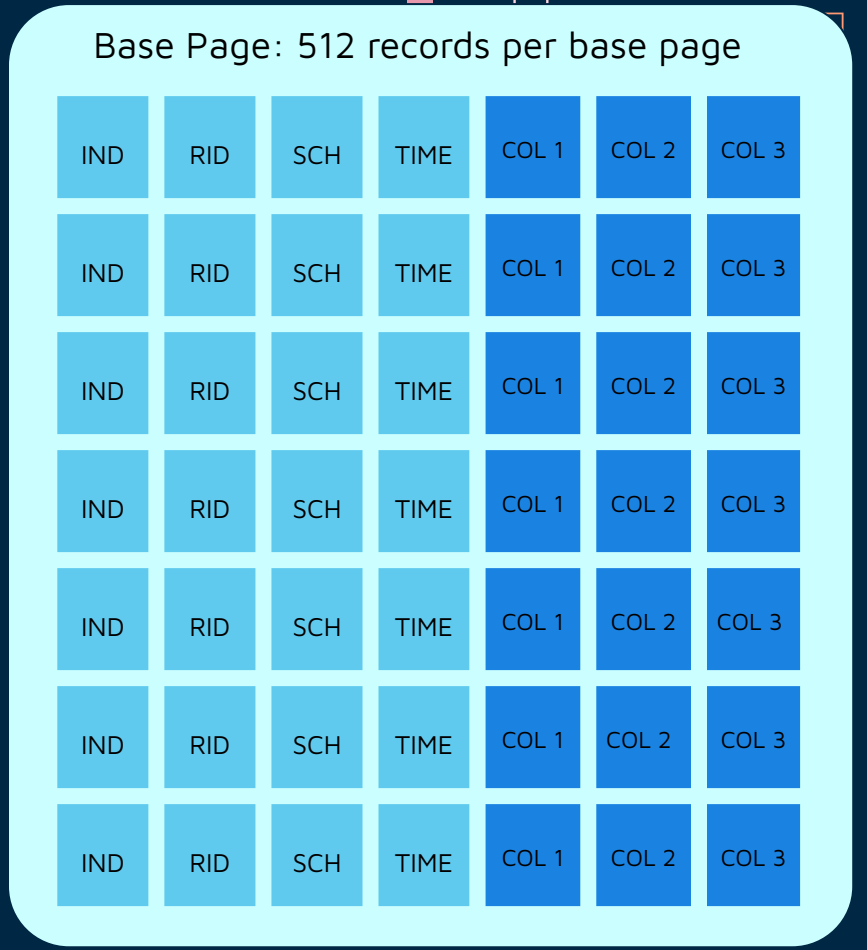
Database Structure



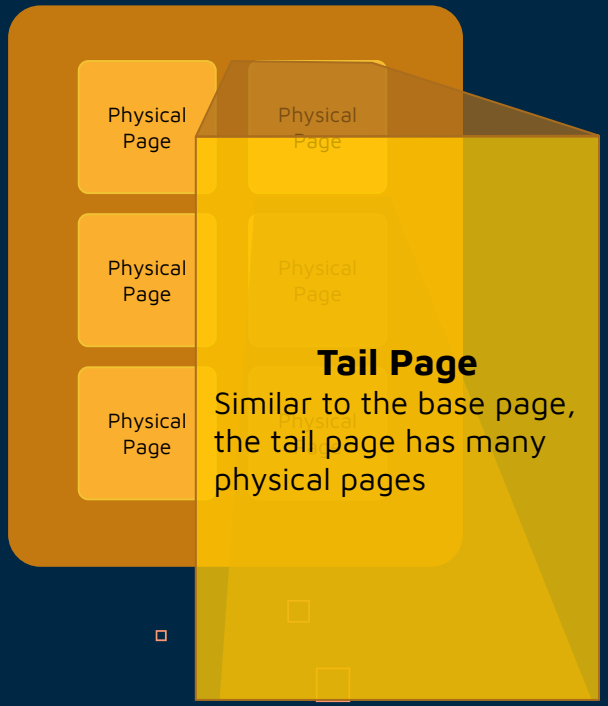
Base Page and Tail Page

Base Page: 4096 kb
8 bytes each



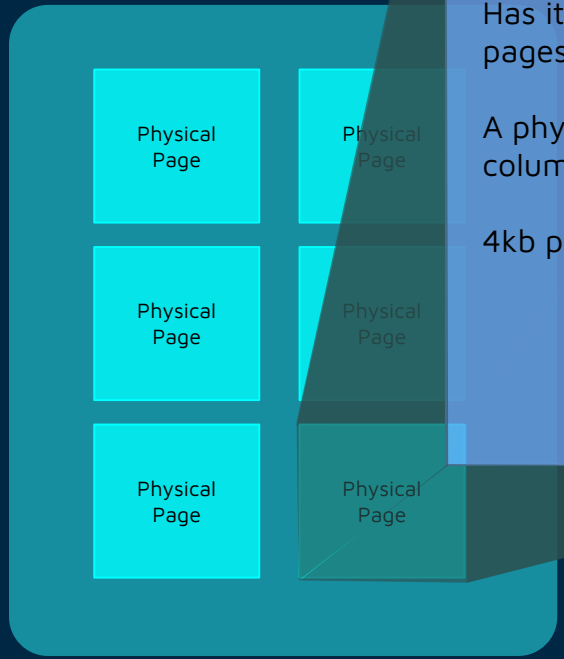


Physical Pages



Tail Page

Similar to the base page, the tail page has many physical pages



Base Page

(Total 16 Base Page)

Has its own set of physical pages

A physical page for each column

4kb per physical page

Table.py: Insert

If the latest page range has space, add a base record to the base pages

Also add a tail record to the tail pages of the same page range

This is the initial tail record for the base record, and prevents issues with indirection and merging for later milestones

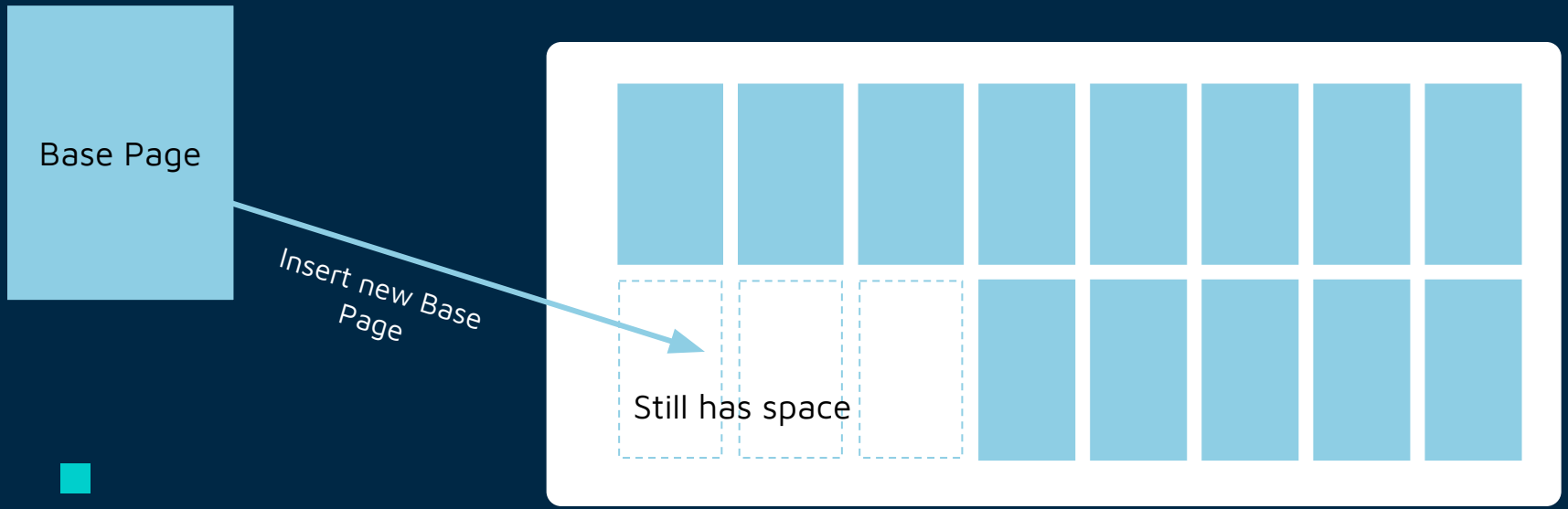


Table.py: Insert (cont.)

If the Page Range is full (has 16 base pages in them already), then a new page range would be added

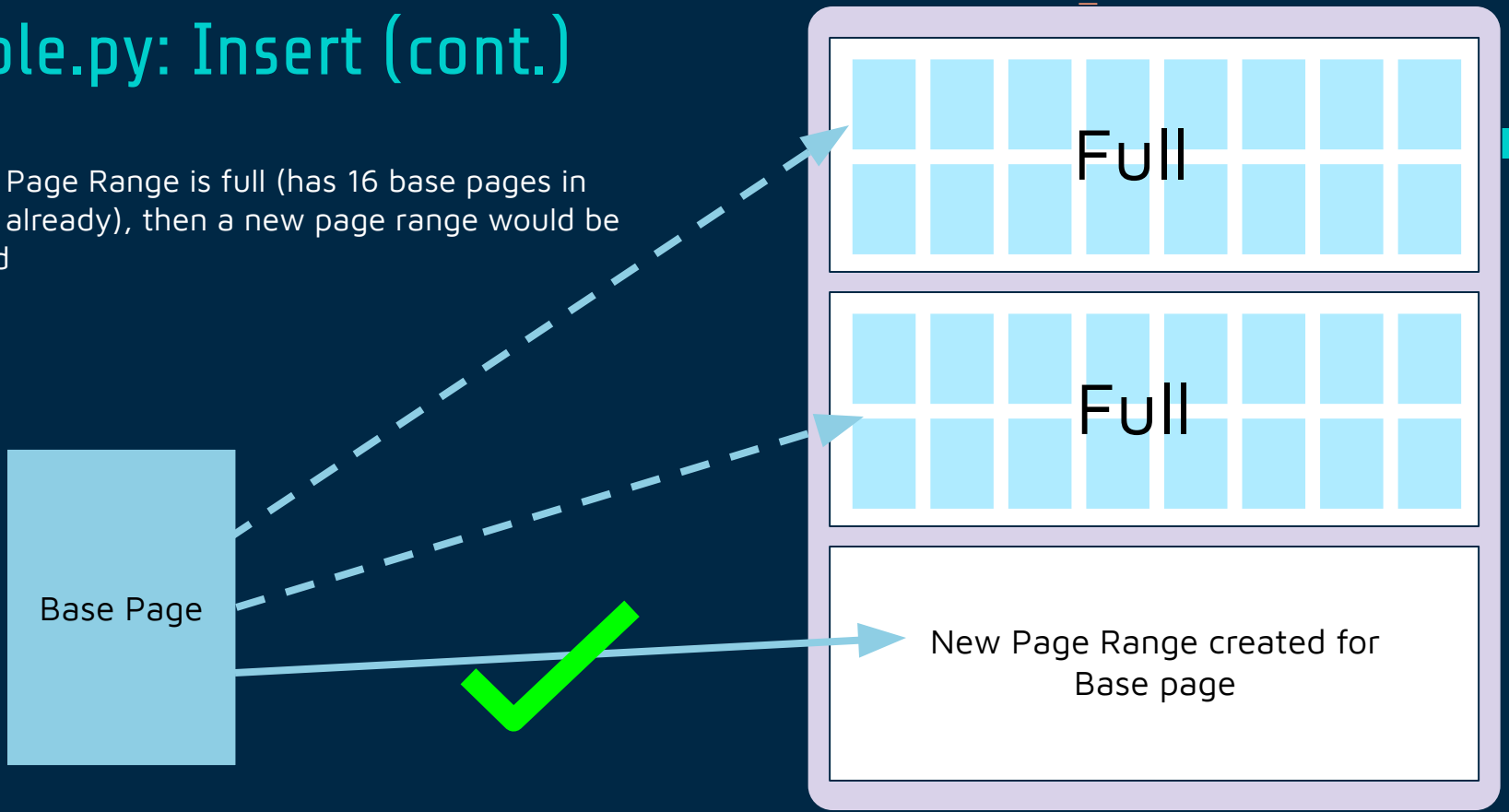


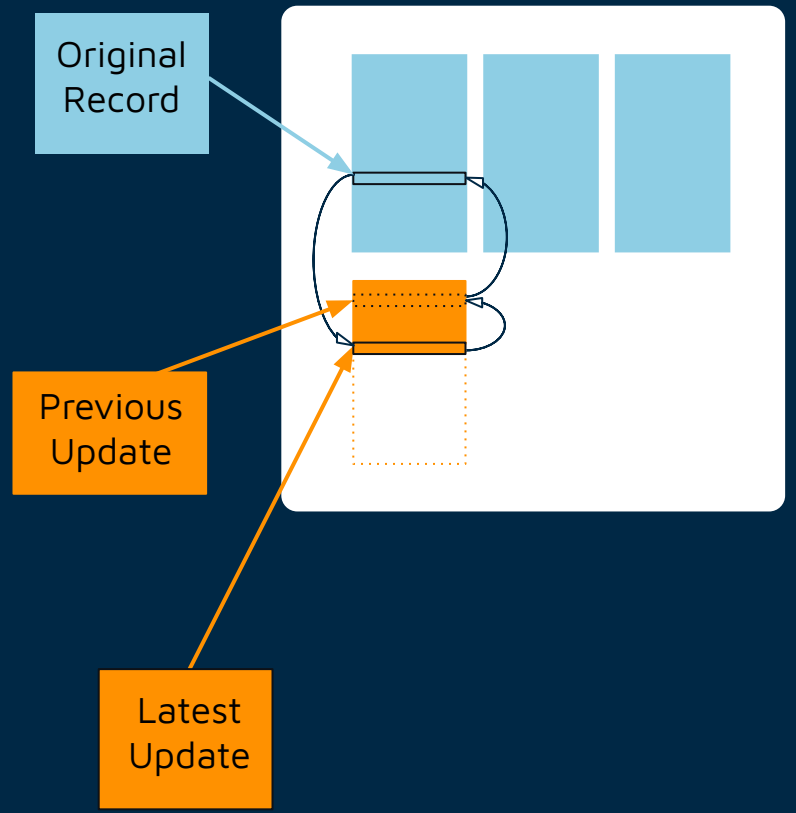
Table.py: Update

Add a tail record to the same page range that the base record that is being updated is in

Update the base record's indirection to point to the new tail record

Make the new tail record's indirection point to the previous latest tail record

Cumulative tail records - each tail record contains the latest updated values



Page.py Functions

def has_capacity(*self*)

- Checks if there are space open for new write functions
- If 512 records (4096kb) are reached, create a new page

def write(*self, value, index*)

- The value will be placed in the specified column of the index
- It will increment each space by 8 bytes so that every number will take a total of 8 bytes

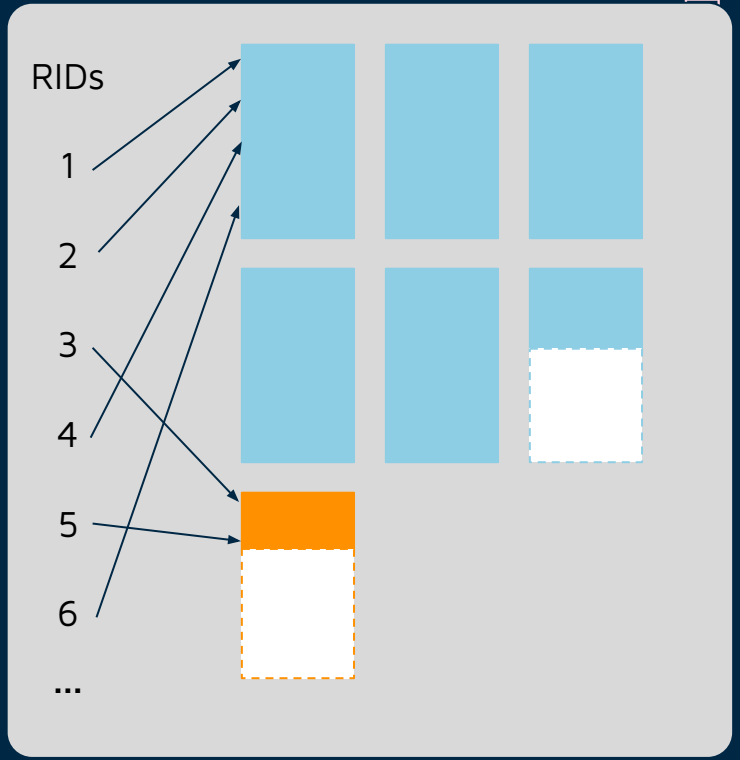
Bufferpool Management

Page Directory

Stores key-value pairs of RIDs and tuples that contain a location in the table

- Which page range
- Base or tail page
- Which physical page
- Offset in physical page

Key-value pairs are created whenever a record is added



Indexing

Multiple dictionaries (key-value pairs)

- one for each data column
- data values stored as keys
- RIDs stored as values

Provides locate and locate_range functions to return RIDs for any data value or range of data values for a specified column

City Dictionary

Data Value	RIDs
Kansas City	0, 2
Davis	1
Houston	3

State Dictionary

Data Value	RIDs
Kansas	0
California	1
Missouri	2
Houston	3

Table

RID	City	State
0	Kansas City	Kansas
1	Davis	California
2	Kansas City	Missouri
3	Houston	Texas

Query Interface

Insert

```
def insert( self, *columns )
    self.table.insert( *columns )
```

Update

```
def update( self, *columns )
    self.table.update( *columns )
```

Both the insert and update functions in Query
uses the insert and update function from Table

Select

Ex:
Select Name column for all with age 19

1. Find RID of all that matches ‘age 19’
2. Get name from those RIDs

RID	NAME	AGE	GRADE
13414	Bob	19	97%
13415	Mary	23	72%
13416	Josh	20	89%
13417	Grace	19	94%
13418	Sabrina	19	73%

- Purpose: show desired records
- Input
  - index_value, index_column, query_columns
- Output
  - Returns a list of record objects upon success
  - False if record locked by 2PL

Returns:

Bob, Grace, Sabrina

Sum

- Purpose: sum record values in a specified columns
- Input
  start_range, end_range,
  Aggregate_column_index
- Output
  Sum of given range upon success
  False if no records exist in given range

def sum(*self, start_range, end_range, aggregate_column_index*)

Start_range: first letter → H
End_range: first letter → N
Aggregate_column_index: column → age (index: 1)

Adds Mary and Josh's age together → 23 + 20 = 43

Output: 43

Assume:

Choose people with the name starting with H to N

Add the age column

0 Name	1 Age	2 Grade	3 Major
Bob	19	97%	CS
Mary	23	72%	Chem
Josh	20	89%	Math

Delete

- Purpose: delete record given primary key
- Input
  - Primary key
- Output
  - True upon successful deletion

def delete(*self* , *primary_key*)

- Find the RID location
- Check to see if it's a Base page or Tail page
- Change the RID to a special value (DELETED_RID_VALUE) using the write function

RID	NAME	AGE	GRADE
13414	Bob	19	97%
13415	Mary	23	72%

Ex:
Delete RID 13415 → change RID to DELETED_RID_VALUE

RID	NAME	AGE	GRADE
13414	Bob	19	97%
~~13415~~ -1	Mary	23	72%

Delete: with Tail Page

Base Records

RID	NAME	AGE	GRADE	INDIRECTION
13414	Bob	19	97%	RID: 15632
~~13415~~ -1	Mary	23	72%	RID: 17482

Check if it has Tail Page

Change Tail Page RID to DELETED_RID_VALUE

Tail Record

RID	NAME	AGE	GRADE	INDIRECTION
~~15632~~ -1	Mary	23	84%	

Questions?

Thank You!