Milestone 1

Group JeLLY-B

Jessica LaCourse
Lingxiao Meng
Lisa Malins
Yinuo Zhang
Benjamin Rauch

Data Model

For our milestone, we've made adjustments to traditional L-store.

1. Use of cumulative updates
2. No schema encoding, but use indirection
3. Use of a timestamp
4. Shift of query functions to Table class

Bufferpool

There were no major modifications to the bufferpool from the traditional L-Store

How are we creating tables?

Create table :

- Add page range
- Create a page directory
- Assign a key column
- Dedicate space for each data column

Indices

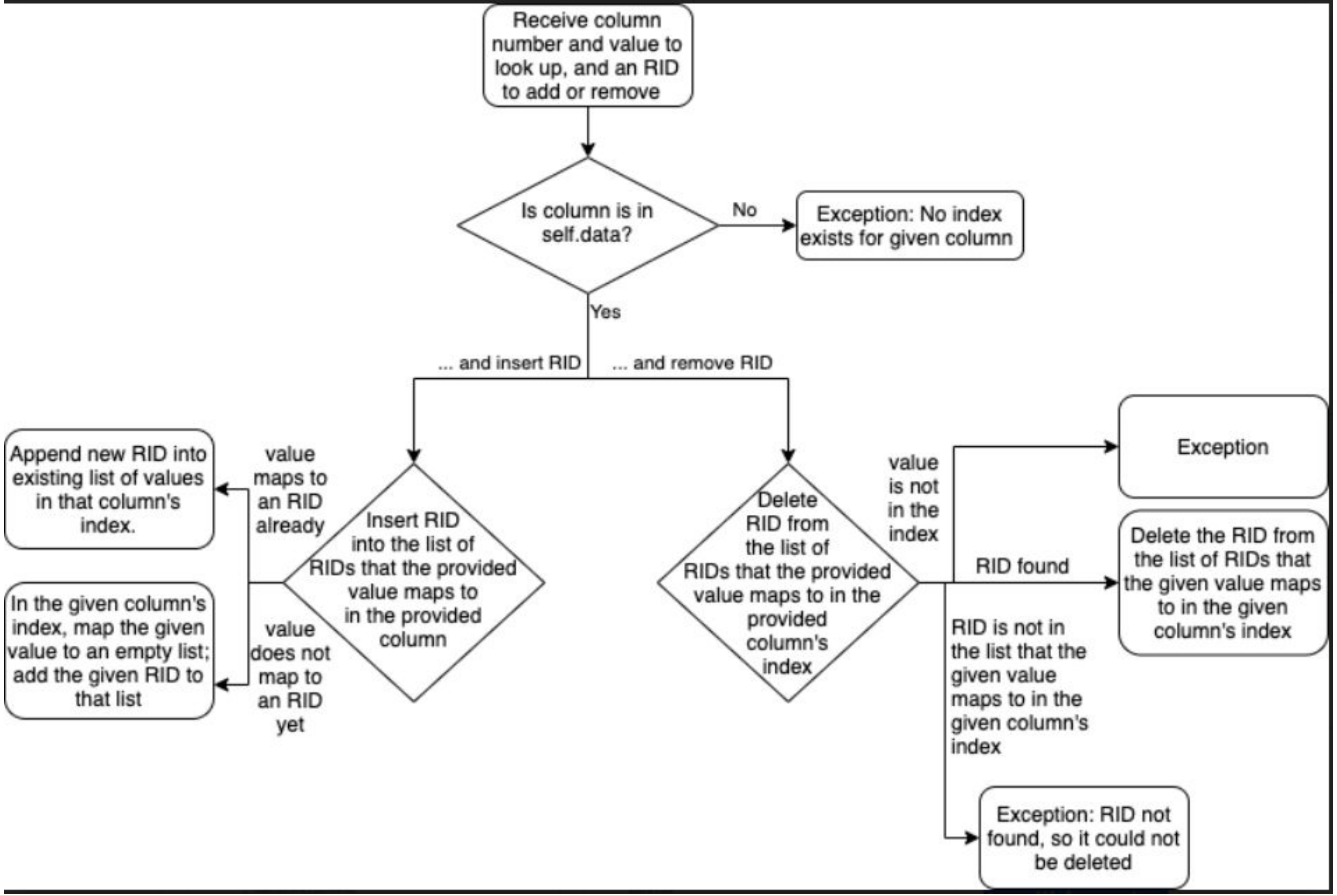
Internal structure: a dictionary of dictionaries

Create_index: indices.data[column_number] = {}

Locate: return indices.data[column_number][value]

Each access method returns a list of RIDs. These RIDs belong to all records with the specified value in the specified column.

Example call: **indices.locate(student_id_column, bens_id) -> [rid_of_bens_record]**


Receive column number and value to look up, and an RID to add or remove
Is column is in self.data?
No
Exception: No index exists for given column
Yes
... and insert RID
... and remove RID
Insert RID into the list of RIDs that the provided value maps to in the provided column
value maps to an RID already
Append new RID into existing list of values in that column's index.
value does not map to an RID yet
In the given column's index, map the given value to an empty list; add the given RID to that list
Delete RID from the list of RIDs that the provided value maps to in the provided column's index
value is not in the index
Exception
RID found
Delete the RID from the list of RIDs that the given value maps to in the given column's index
RID is not in the list that the given value maps to in the given column's index
Exception: RID not found, so it could not be deleted

Query Functions

Insert:

- Collect data columns and append metadata columns
- Metadata: Assign a RID, a timestamp, and an indirection of 0
- Create entry for record in index
- Test capacity of the current page, then assign the record to a slot on a physical page

Query Functions

Select:

Find columns which hold the search key

Table retrieves a list of base RIDs of these columns from Index

Track the location of these base RIDs and find their logical page

If any, get updates from logical page

Get record from last update in logical page

Only return record for user-asked columns

Query Functions

Update :

Find RID of record in index

Find the record's logical page based on RID

Get the last update and record logical page

Assign a RID to the new tail record

Update base record indirection column and set a tail record (update indirection, timestamp, columns)

Query Functions

Delete:

Get record location and metadata from index

Set up deletion flag to push value out of range

Delete flag and values in index

Query Functions

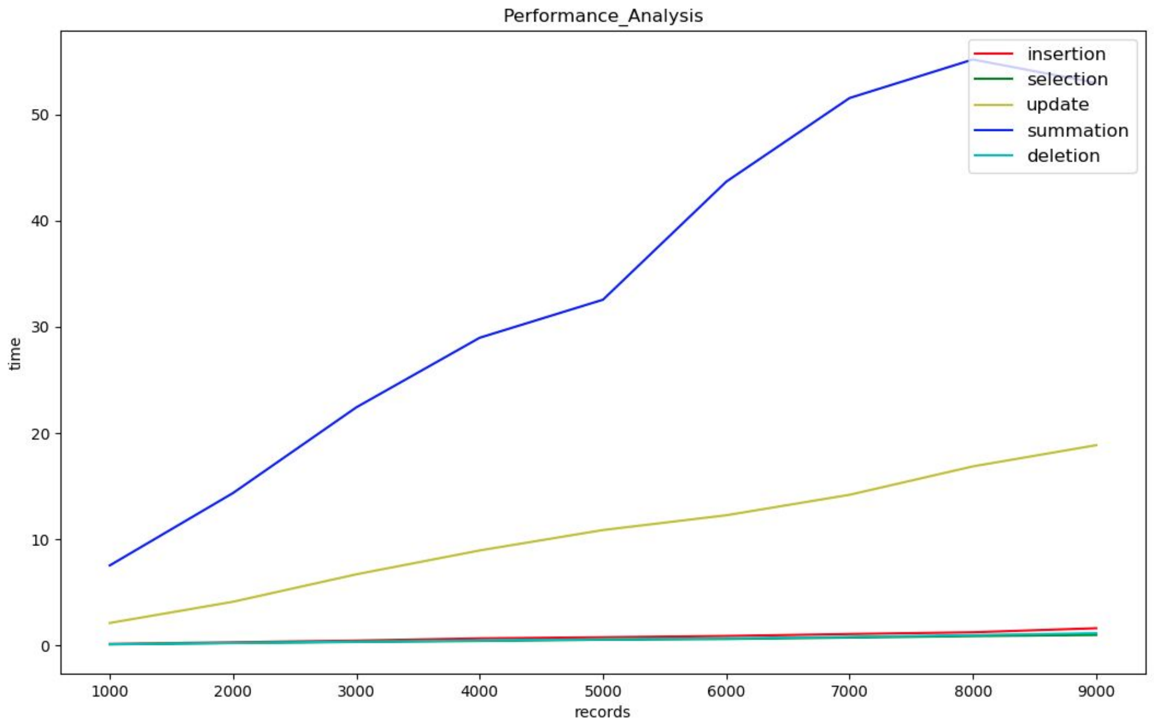
Sum:

- Set user chosen columns as 1, otherwise is 0
- Use select function with these data and append to a list of values to sum
- Calculate the sum of the values

Performance

As the number of total records increases, the time required to complete query functions (insert, select, update, summation, deletion) increase in linear time.

Summation takes the longest time, and followed by update.



Processor Intel(R) Core(TM) i7-6700HQ CPU @ 2.60GHz, 2601 Mhz, 4 Core(s), 8 Logical

Thanks