L-Store Database

Team M4A1

Michael Ewing
Amber Lee
Kaito Yoshida

Henghui He
Vadzim Matsiushonak
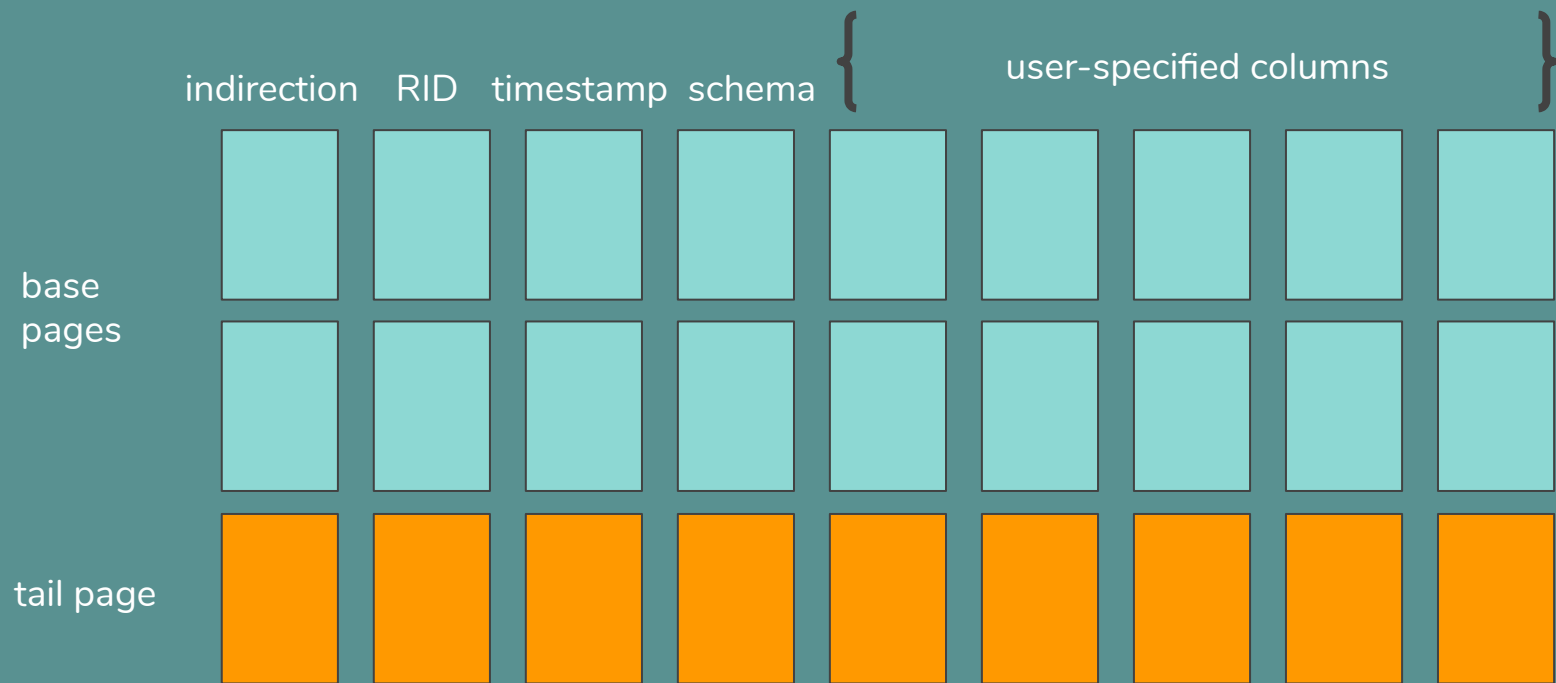
Presentation Breakdown

1. Database Overview
2. Creative Design
3. Q / A
4. Live Demo of Code

Database Overview

General Design Concept

indirection   RID   timestamp   schema   { user-specified columns }

base
pages

tail page

db.py

- Internally saves a dictionary that maps a string name to Table object
- create_table( ): makes a new Table object and saves it into dictionary
- drop_table( ): deletes table from dictionary

table.py

- Internally stores 4 metadata columns in addition to user-given columns
- Implements a dictionary for storing the Page objects
- Capable of storing RID's ranging from 1 to 2^{64}
  - Base Records are numbered 1 and up
  - Tail Records are numbered $2^{64}-1$ and down.

index.py

- Index class stores internal dictionary that is used in 2 ways:
  1. In query.py, as index that maps user-given key to RID

     [key] : [RID]

  2. In table.py, as page directory that maps RID to physical page address

     [RID] : [(base/tail), page_num), row_num]
     - base/tail is specified as 0/1, respectively
     - page_num is the physical page representing the indirection column of the base page containing the record
     - row_num is offset within the page at which the record data is found

query.py

- Contains methods that performs queries within the database including:
  - select( ): uses the index class to create an index mapping key to RID, and then grabs the data from each needed column in the matching record
  - update( ): creates the bitmap schema encoding for the updated column and calls the table to update the record with new data
  - sum( ): selects a range from a column and returns the sum

page.py

- Stores the information in a physical page as a bytearray
- write( ): appends a new value into the page
- read( ): returns the value at the index offset in the page
- overwrite_record( ): updates the data entry at index with new value

Creative design

- Database can handle the input and manipulation of duplicate keys
  - duplicate key selection
  - update multiple columns at once for multiple records

Q&A ?

Live Demo

- Adding
- Modifying
- Querying
- Creative Design