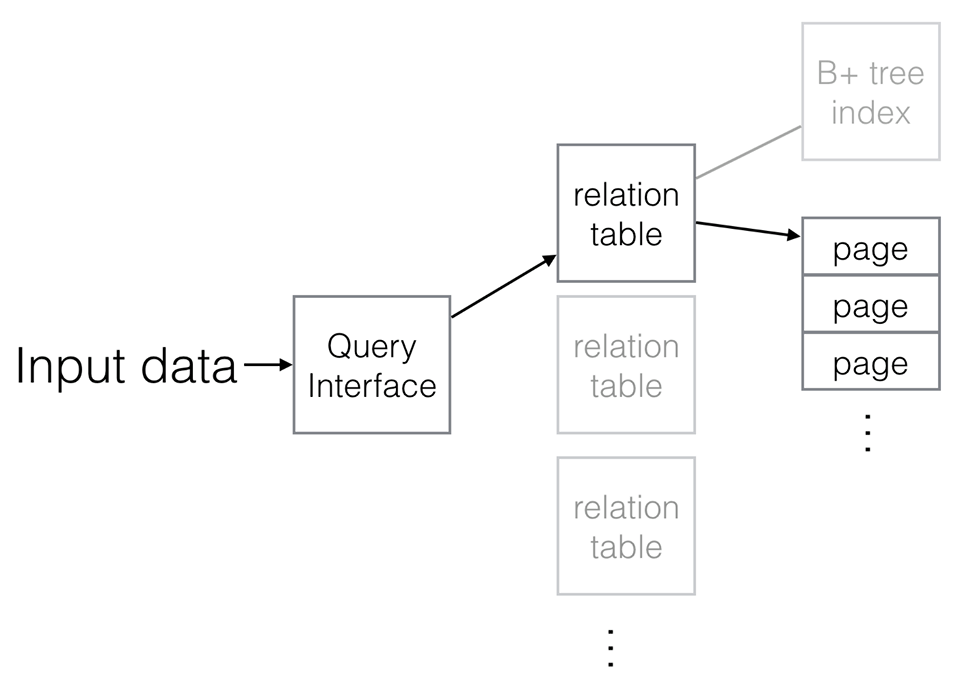
### Overview

This project is using python3 as our programming language. Our design structure is separated by four major parts, so we created four classes, which are “QueryInterface”, “RelationTable”, “Page”, “bPlusTree”, in our code. “QueryIinterface” was designed to help user run the functions to access or edit the relation tables and its pages. User could do all of the work through query interface. “RelationTable” was designed to help us to manage the pages we have. “Page” is to help us to manage the functionality of a page. “bPlusTree” was designed to support equality and range searches efficiently

The whole process is that the query interface gets the command from users and parses the commands to invoke some functions in other class to accomplish the task. Take insertion as example, a user selects a target relation table, and then the corresponding relation table would get the input and store the record into page. After inserting the record to page, it will return the RID and then we could take the RID and the record to rebuild the B+ tree.

We also considered the error users may encounter, so we implemented some error checking mechanisms to prevent the unexpected crash. We would introduce the functionalities and error checking mechanisms for each part next.



### Query interface

We implemented the query interface to help users run the functions to access or edit the relation tables and their pages. We also implemented the command “help” to print out all the support commands and their descriptions in case users don’t know what functionality we have. Users could key in the commands on query interface, and then query interface would invoke the corresponding functions to accomplish the users’ request. In this part we have some error checking mechanisms, we are able to detect some potential problems such as user key in the unsupported command, parameter number mismatch and specified relation table doesn’t exist. If errors described previously have been found, we should inform the users to correct them.

### Relation table

We implemented the class “RelationTable” to help us to manage the pages we have. Before any insertion is invoked, the function “insertable” would check whether the record is valid or not by examining both the record type and size. If the record is valid, the function “insert” would insert it into page and then get the RID to build the B+ tree. The function “search” would check the type of key is string or integer and then search the target key and print out the statistical information. If the target key is not found then it would print out the message “no record found” to inform the user. The function “rangeQuery” would find out the entries whose values between two specific numbers and print out the statistical information. The function “delete” would check the type of key is string or integer and then search the target key, if the target key is found then remove it, else it would print out the message “no record found, no deletion” to inform user. It’s also important to know the status and content of the pages, so the function “showPageContent” could invoke the corresponding function to display the page status and content. The function “showStatistic” is to display the index statistic informations such as number of index pages and number of slotted data pages in memory. The other error checking mechanism in this part is to check whether user specify a page that doesn’t not exist.

### Page

We implemented the class “Page” to help us to manage the functionality of a page. It’s important to check whether the record size is valid or not, so the function “isValidSize” would do that for us. If the record size is not correct, it would inform us what kind of errors emerge. The function “insert” would insert record to an empty slot and return the sid. If empty size is less than record size plus 2, then the insertion is fail and return none. The function “delete” would find the target entry and remove it. If an entry has been deleted from a page, it will leave an empty slot on the page. Next time if user insert a record, the empty slot would be chose to store it. The function “showContent” is to help users to find out the page storage status, including number of slots, occupied slots, empty slots, and the page storage list, including sid and key.

### B+ tree

Implementation of B+ tree is to support equality and range searches efficiently, so we create “bPlusTree” class to maintain indexes. This B+ tree has been simplified in some ways according to the description of the project.

The function “insert” would insert the entry to page of the B+ tree, if the target page doesn’t have enough space, it would cause the page to split, redistribute entries, and then copy up middle key. The function “delete” would find the target entry, says L, and try to remove it, and it work differently on leaf page and non-leaf page. In a leaf page case, after removing L, if the number of remaining entry in that page is more than order, then job is done, else it would try to borrow from sibling. If borrowing from sibling fails, it would merge the sibling page and delete the node from parent of L. In a non-leaf page case, if the number of remaining entry in that page is less than order, the parent page of L would try to borrow for L from L’s sibling and update the node value. If it fails, it would merge the sibling page and delete the node from parent of L, and check whether the number of remaining entry in that page is less than order once again. Error checking mechanism here is if the target entry is not found, user would be informed. The function “rangeQuery” would find out the result between two specific numbers. Finally, the function “pageStatistics” is to help users to find out the current status of the B+ tree, including number of non-leaf page, leaf page and total page.

### Test cases

Case 1 :

R, Student, integer, 80



I, Student, 50, "abcdefghijklmnnnnnnnnnnnnnnnnnnnnnnnnnnnnn "



Scan Student



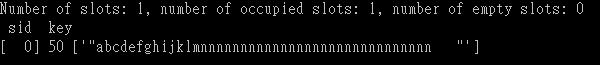
q Student 50



query> q Student 0 70



p Student 0



c Student



(error case)

I, Student



D, Student, 50



Case 2 :

R, News, String, 90



I, News, "Financial Technology", "news article about finanacial technology from New new Times "



Scan News



q News "Financial Technology"



q News "Financial Technology" "Smarter City"



p News 0



c News



(error case)

x News



D, News, "Financial Technology"

