

Test cases for correctness:

Our project deals with a repeating attribute of a relation. For testing purposes, to get a good measure of the performance improvement in both memory and time, we would need to use huge datasets.

For correctness, the given implementation already implements access/search/modify in repeating attributes (non-unique) of a relation by using duplicate values in the B+ tree implementation underneath. But having duplicates in the B+ tree reduces the efficiency of the tree both in terms of memory (space utilization) and time, which we alleviate by creating a secondary index. In one approach the secondary index contains pointers to repeating values of the attribute and in the other approach, it contains an index appended with its primary key.

Since the project is mainly aimed at improving the performance, we will use less number of test cases for checking correctness and more for comparing performances.

Test cases:

NOTE: We write these test cases as SQL queries so as to make the cases easily representable. In actual sense, we would be writing C code to achieve these queries.

1. Basic Query:

- `CREATE TABLE customers(int ID PRIMARY KEY, char bankname, int loanID, int amount);`
- We insert a large number of tuples in this relation with the aim of having 'bankname' as the repeating attribute.
`INSERT INTO customers VALUES(1, SBI, 123, 89776);`
- Now we do a series of queries:
 - `DELETE FROM customers WHERE bankname="Canara" and ID>100;`
 - `UPDATE customers SET amount = 500 WHERE bankname="SBI";`
 - `SELECT FROM customers WHERE bankname="PNB" and amount<50000;`
- We check the output of the above queries and check the correctness of our implementation.
- Since the implementation is done only with regards to the repeating attribute, we perform all queries using that attribute.

2. Complex Queries:

- We can create a relation with multiple repeating attributes and try queries involving both of them.
- `CREATE TABLE banks(int ID PRIMARY KEY, char name, char country, char city);`
- We insert a large number of tuples with repeating values for the attributes country and city.

- We can try queries like:
 - DELETE FROM banks WHERE country="USA" and city<>"Boston";
 - SELECT FROM banks WHERE country="INDIA" and city="Delhi";

These test cases and their subtle variants will be enough to check the correctness of our implementation. The above test cases are strictly for correctness purposes and not for performance measurement. We will devise another set of test cases for the purpose of performance comparisons.

In future, we may devise some more test cases to check correctness as we may get more ideas when we implement. Since, the project deals with pointers we think these test cases will come in quite handy for us to debug and check.

Yeddu Vamsidhar	110050051
M Rajeev Kumar	110050052
Anirudh Vemula	110050055
K Kartik Pranav	110050062