COL362/662 Assignment 2
Operations of Relational Algebra

Due Date: 22 Feb 2016

Feb 3, 2016

updated doc: 18th Feb 2016

Assignment In-charge: Ovia

Programming

Questions:

1. Implement Creation, Deletion, Insertion, modification using b/b+ trees.
2. Implement relational algebra operations, select, project, union, intersection and set-difference on b/b+ tree structures.

Guidelines

- Assignment is to be done individually.
- Copying from web or from each other will result in penalty as per institute rules. Moss will be run on submitted code.
- All operations discussed above are set theoretic
- Document your code properly and make it modular for readability and debugging convenience.
- Use piazza for doubts and clarification. TAs will clarify doubts on the assignment till 9th Feb only.
- Marking scheme of the assignment is 20 marks. Evaluation is based on code efficiency during demo.
- You may use C/C++, java or python for implementation.
- You should be strictly following the input specification written below, otherwise you will be penalized.
- Submit your code in a single zipped file

We will be checking :-

- Program which will save the B/B+ tree Data Structure on Hard drive
- Program which will read the file above and queries will be executed on that input.
- The Program will be tested for Efficiency.
- You will be penalized for not following the input format given.

You may assume the following:

- The first attribute of every entity will be a primary key and is of integer type
- Remaining may be string or integer.

Input and Output Specification

Input File Format:

#number of tables
Table1_name
#number of attributes in the Table1_name
attribute1_name,attribute2_name,attribute3_name,attribute4_name,..
#number of tuples in Table1_name
val1,val2,val3,val4, ...
val1,val2,val3,val4, ...
val1,val2,val3,val4, ...
.
.
.
.
.
.
val1, val2, val3, val4, ...
$ --- end table data
Table2_name
#number of attributes in the Table2_name
attribute1_name,attribute2_name,attribute3_name,attribute4_name, ...
#number of tuples in Table2_name
val1,val2,val3,val4, ...
val1,val2,val3,val4, ...
val1,val2,val3,val4, ...
.
.
.
.
.
.
val1, val2, val3, val4, ...
$
.
.
.
.
.
.

$# ---end file

Output File Format:

#number of attributes in the result
attribute1_name,attribute2_name,attribute3_name,attribute4_name, ...
#number of tuples in result
val1,val2,val3,val4, ...
val1,val2,val3,val4, ...
val1,val2,val3,val4, ...
.
.
.
val1, val2, val3, val4, ...
$ --- end table data

Query Input Format:

Type 1:
Individual row operation
QUERY→ <KEYWORD> <table> (opt <attribute_list separated by ','> <conditions separated by ','> <values separated by ‘,’> <set value>)
KEYWORD → insert|delete|modify

Eg: insert student name,age raju,21
delete student age<18
update student name=’rani’ age=25 –sets age 25 where name is rani

Type 2:

QUERY--> project (<SUB_QUERY>)
SUB_QUERY --> select <attribute_list separated by ','> <table_list separated by ','> <conditions separated by ','> |
(<SUB_QUERY>) <KEYWORD> (<SUBQUERY>)
Keywords - union|intersection|difference

Eg: project (
(Select name,age student cgpa>8)
Union
(select name,age student age<20)
)