

[CS360] Introduction to Database

Term project #1

Hyun, Soon Joo

Term project #1

- Implement a python program that builds a B+ tree for a table

- Version: python 3.6

- Table constraints

You should collect your own data

- A file that contains real-world data
 - 1 table
 - Number of tuples ≥ 20
 - Number of attributes ≥ 4
 - Each tuple has its own identifier

If the original data does not have identifiers of tuples, you can assign the identifiers by yourself

tid	mid	uid	rating	date
1	762	2031826	3	2004-04-06
2	762	515436	5	2005-03-24
3	886	2645160	5	2005-03-24
4	312	2410056	3	2005-09-01
5	312	528423	2	2004-07-14

Ex) Netflix dataset

- *tid: tuple id*
- *mid: movie id*
- *uid: user id*
- *rating: # of 'stars' assigned to a movie by a customer*
- *date: timestamp of a rating in the form YYYY-MM-DD*

Term project #1

▪ Implement a python program that builds a B+ tree for a table

• B+ tree constraints

- Internal nodes contain the keys
- Leaf nodes contain $\langle \text{key}, \text{value} \rangle$ pairs
- Order d of the B+ tree: 3 or 4
- The number of $\langle \text{key}, \text{value} \rangle$ pairs: m
 - ♦ Root node: $1 \leq m \leq d - 1$
 - ♦ Internal node and leaf node: $\lceil d/2 \rceil - 1 \leq m \leq d - 1$
- The key of the B+ tree should use a pair of two attributes
 - ♦ You should decide the order of the keys

Key: A pair of two attributes

Value: A list of identifiers of tuples that have the same key

Ex) Netflix dataset

- *A pair, $\langle \text{rating}, \text{date} \rangle$, is used as the key of the B+ tree index*
- *The order of the keys: Ascending order of ratings and dates*
- *Comparison: Compare the rating first, and then the date*

Term project #1

- Implement a python program that builds a B+ tree for a table
 - B+ tree operations
 - Supports 6 operations
 - ♦ **LOAD, PRINT, INSERT, DELETE, SEARCH, RANGE_SEARCH**
 - ♦ Your program should provide **an user interface** (e.g. a command line interface) to test the operations

Term project #1

- Implement a python program that builds a B+ tree for a table
 - B+ tree operations
 - **LOAD:** Read a table and build a B+ tree for a specific tuples
 - ◆ Input argument: Start tuple ID, End tuple ID
 - ◆ Insert tuples from start tuple ID to end tuple ID into the B+ tree
 - **PRINT:** Print the whole contents of the B+ tree
 - ◆ Show the contents of all B+ tree nodes by the breadth-first manner
 - **INSERT:** Insert a tuple into the B+ tree
 - ◆ Input argument: Tuple ID (identifier)
 - ◆ If the B+ tree has already the key of the tuple, append the tuple ID to the corresponding value
 - ◆ Otherwise, create a new <key, value> pair and insert it into the B+ tree node

Term project #1

- Implement a python program that builds a B+ tree for a table
 - B+ tree operations
 - **DELETE**: Delete a tuple from the B+ tree
 - ◆ Input argument: Tuple ID
 - ◆ Delete a tuple ID from the value corresponding to the key of the tuple
 - ◆ After the deletion, if the key has no tuple ID, delete the key from the B+ tree node
 - **SEARCH**: Search tuples by a given key
 - ◆ Input argument: Searching key (e.g. (3, 2004-04-06))
 - ◆ Return tuple IDs from the value corresponding to the given key
 - ◆ Show the contents of found tuples
 - **RANGE_SEARCH**: Search tuples with a certain range of the key
 - ◆ Input argument: A certain range of the key (e.g. [(3, 2005-09-01), (5, 2005-03-24)])
 - ◆ Return all <key, value> pairs within the range
 - ◆ Show the contents of found tuples

Term project #1

- Example of the user interface
 - Main menu

```
===== B+ tree program =====  
1.  LOAD  
2.  PRINT  
3.  INSERT  
4.  DELETE  
5.  SEARCH  
6.  RANGE_SEARCH  
7.  EXIT  
=====
```

SELECT MENU:

Term project #1

- Examples of the B+ tree operations
 - Example: **LOAD**

tid	mid	uid	rating	date
1	762	2031826	3	2004-04-06
2	762	515436	5	2005-03-24
3	886	2645160	5	2005-03-24
4	312	2410056	3	2005-09-01
5	312	528423	2	2004-07-14

Build a B+ tree
(Insert tuple #1, #2)



Order d: 3

<i>Key:</i> (3,2004-04-06)	(5,2005-03-24)
<i>Value:</i> [1]	[2]

Sample UI

```
=====
SELECT MENU: 1
```

```
===== LOAD =====
```

```
LOAD_START_TID: 1
```

```
LOAD_END_TID: 2
```

```
LOADING ....
```

```
B+ Tree is built.
```


Term project #1

- Examples of the B+ tree operations
 - Example: **INSERT**

*The B+ tree has already
the key: (5,2005-03-24)*

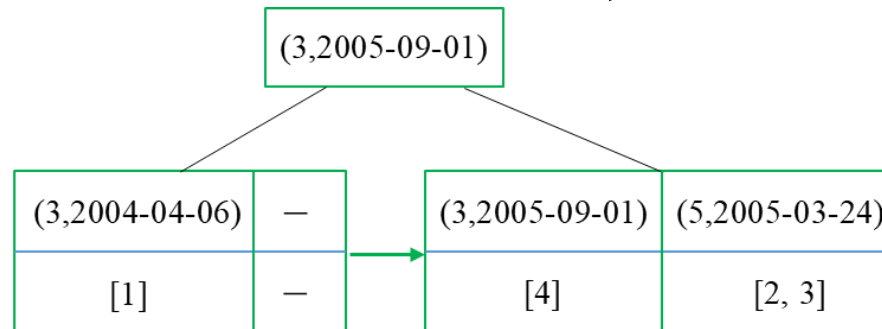
(3,2004-04-06)	(5,2005-03-24)
[1]	[2]

Insert
tuple #3



(3,2004-04-06)	(5,2005-03-24)
[1]	[2, 3]

Insert tuple #4

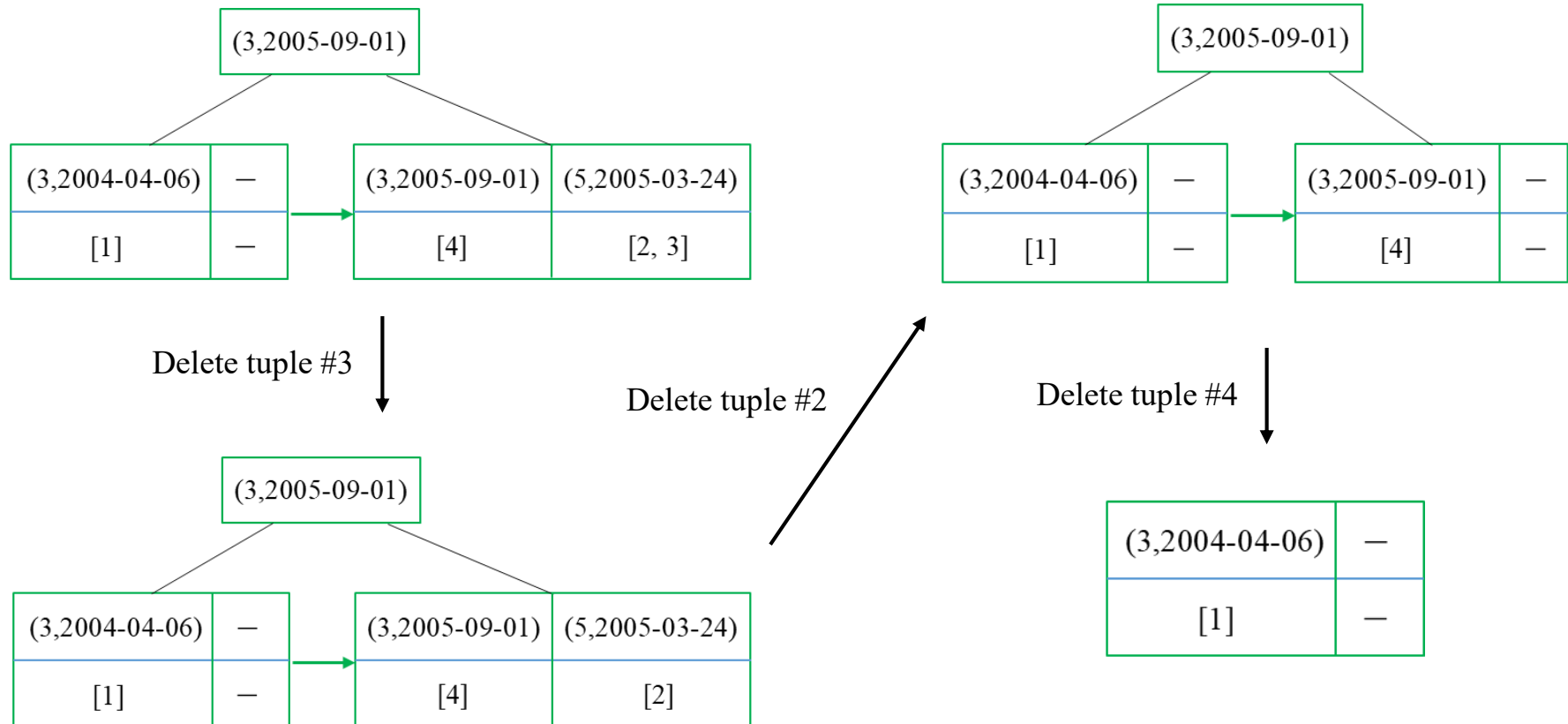


Leaf nodes are connected by pointers (a linked list)

```
=====
SELECT MENU: 3
===== INSERT =====
TUPLE ID: 3
Tuple #3 is inserted.
=====
SELECT MENU: 3
===== INSERT =====
TUPLE ID: 4
Tuple #4 is inserted.
```

Term project #1

- Examples of the B+ tree operations
 - Example: **DELETE**



Term project #1

- Examples of the B+ tree operations
 - Example: **DELETE**

```
=====
SELECT MENU: 4
```

```
===== INSERT =====
```

```
TUPLE ID: 3
```

```
Tuple #3 is deleted.
```

```
=====
SELECT MENU: 4
```

```
===== INSERT =====
```

```
TUPLE ID: 2
```

```
Tuple #2 is deleted.
```

```
=====
SELECT MENU: 4
```

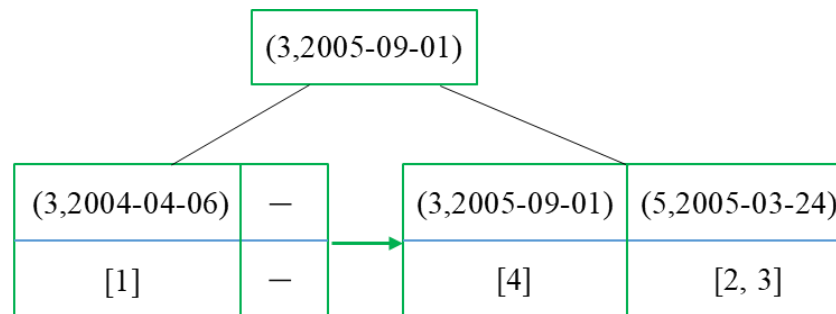
```
===== INSERT =====
```

```
TUPLE ID: 4
```

```
Tuple #4 is deleted.
```

Term project #1

- Examples of the B+ tree operations
 - Example: **PRINT**



Print the tree



```
=====
SELECT MENU: 2
```

```
===== PRINT =====
```

```
Level 1: [(3, 2005-09-01)]
```

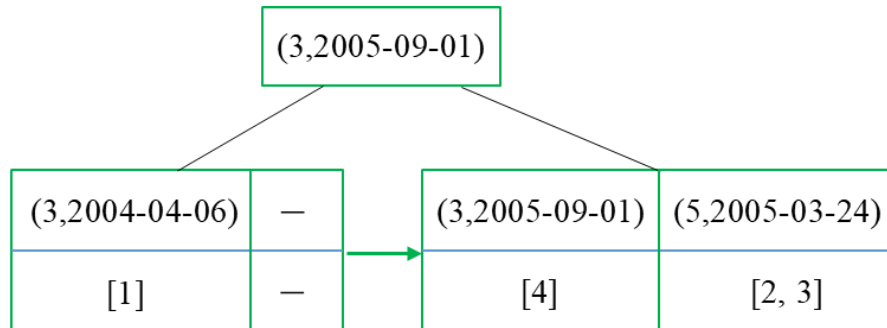
```
Level 2: [ ((3, 2004-04-06), [1]) ] --> [ ((3, 2005-09-01), [4]), ((5,2005-03-24), [2,3]) ]
```

Term project #1

▪ Examples of the B+ tree operations

• Example: **SEARCH**

Search key: (5, 2005-03-24)



tid	mid	uid	rating	date
1	762	2031826	3	2004-04-06
2	762	515436	5	2005-03-24
3	886	2645160	5	2005-03-24
4	312	2410056	3	2005-09-01
5	312	528423	2	2004-07-14

```
=====
SELECT MENU: 5
```

```
===== SEARCH =====
```

```
  SEARCH KEY: (5,2005-03-24)
```

```
Found tuple IDs : [2, 3]
```

```
Attributes: < tid, mid, uid, rating, date >
```

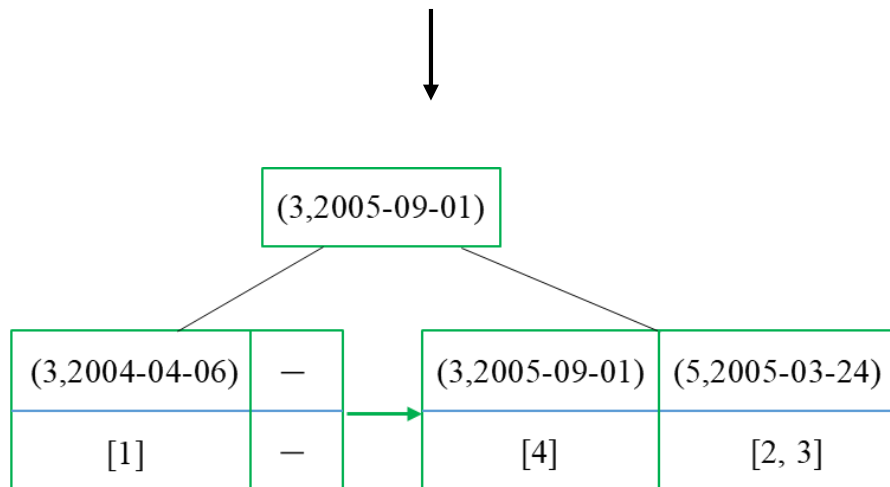
```
Tuple #2 : < 2, 762, 515436, 5, "2005-03-24" >
```

```
Tuple #3 : < 3, 886, 2645160, 5, "2005-03-24" >
```

Term project #1

- Examples of the B+ tree operations
 - Example: **RANGE_SEARCH**

Search range: $[(3, 2005-09-01), (5, 2005-03-24)]$



tid	mid	uid	rating	date
1	762	2031826	3	2004-04-06
2	762	515436	5	2005-03-24
3	886	2645160	5	2005-03-24
4	312	2410056	3	2005-09-01
5	312	528423	2	2004-07-14

Term project #1

- Examples of the B+ tree operations
 - Example: **RANGE_SEARCH**

```
=====
SELECT MENU: 6
```

```
===== RANGE_SEARCH =====
```

```
SEARCH RANGE: [(3, 2005-09-01), (5, 2005-03-24)]
```

```
Found pairs : [ ((3,2005-09-01), [4]), ((5,2005-03-24), [2,3]) ]
```

```
Attributes: < tid, mid, uid, rating, date >
```

```
Tuple #4 : < 4, 312, 2410056, 3, "2005-09-01" >
```

```
Tuple #2 : < 2, 762, 515436, 5, "2005-03-24" >
```

```
Tuple #3 : < 3, 886, 2645160, 5, "2005-03-24" >
```

Term project #1

- Project report (about 1-2 pages)
 - Contents
 - Descriptions of your data and B-tree
 - ◆ Source, schema, number of tuples
 - ◆ Order of the B+ tree, what attributes are used for the key of the B+ tree
 - Instructions of your program
 - ◆ How to run the program
 - ◆ How to test 6 operations (**LOAD, PRINT, INSERT, DELETE, SEARCH, RANGE_SEARCH**)

Submission

- Due

- October. 1, 11:59 p.m.
- Delay is not accepted

- Contact

- Contact to your assigned TA if you have any question (see another uploaded file)

Name	Email
Taewook Ha	twha@dbserver.kaist.ac.kr
Wonhyuk Ahn	whahnize@kaist.ac.kr
Sumin Han	smhan@dbserver.kaist.ac.kr

Submission

- Submission standard

- Compress your python files, a data file, and a report file into a .zip file with the filename;
‘CS360_TP1_studentID_name.zip’
- Submit the .zip file to the course homepage (<http://klms.kaist.ac.kr>)

- Evaluation

- You will get points if your program produce the right results by following the instructions in your report
- Do not cheat others. Both of them will get no point