NATIONAL RESEARCH UNIVERSITY

Faculty of Computer Science

SQLITE RDBMS EXTENSION FOR DATA INDEXING USING B-TREE MODIFICATIONS

Academic English Writing
Individual Home Assignment

Student: Anton Rigin, group BSE153

Moscow, 2018

OUTLINE

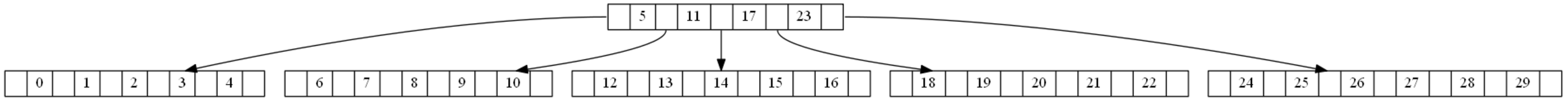
- B-tree
- B-tree operations
- B^+-tree
- SQLite extensions

B-TREE

- Balanced tree
- Search tree [1]
- Nodes contain more than one key and more than two pointers to the children nodes [1]
- If some node contains k keys than it contains $k + 1$ pointers to the children nodes [1]
- **B-tree order** is such a t number that:
  - ✓ for each non-root node the following is true: $t - 1 \leq k \leq 2t - 1$
  - ✓ for root node in the non-empty tree the following is true: $1 \leq k \leq 2t - 1$
  - ✓ for root node in the empty tree: $k = 0$ [1]
- B-tree is usually used as the data index [1]

[1] R. Bayer, E. McCreight, “Organization and maintenance of large ordered indexes,” *Acta Informatica*, vol. 1, no. 3, pp. 173 – 189, 1972.

B-TREE



The B-tree example, t = 6

B-TREE OPERATIONS

- t is the tree order, n is the total keys count
- **Search**
  - ✓ Time complexity: $O(t log_t\ n)$
  - ✓ Memory usage: $O(t)$
  - ✓ Disk operations count: $O(log_t\ n)$ [1]
- **Node split**
  - ✓ Time complexity: $O(t)$
  - ✓ Memory usage: $O(t)$
  - ✓ Disk operations count: $O(1)$ [1]

[1] R. Bayer, E. McCreight, “Organization and maintenance of large ordered indexes,” *Acta Informatica*, vol. 1, no. 3, pp. 173 – 189, 1972.

B-TREE OPERATIONS

- **Insertion (includes nodes split)**
  - ✓ Time complexity: $O(tlog_t\ n)$
  - ✓ Memory usage: $O(tlog_t\ n)$ (for simple recursion), $O(t)$ (for tail recursion)
  - ✓ Disk operations count: $O(log_t\ n)$ [1]
- **Deletion**
  - ✓ Time complexity: $O(tlog_t\ n)$
  - ✓ Memory usage: $O(tlog_t\ n)$
  - ✓ Disk operations count: $O(log_t\ n)$ [1]

[1] R. Bayer, E. McCreight, “Organization and maintenance of large ordered indexes,” *Acta Informatica*, vol. 1, no. 3, pp. 173 – 189, 1972.

B⁺-TREE

- B-tree modification [2]
- Only leaf nodes contain real keys (real data), other nodes contain router keys for searching real keys [2]
- Leaf nodes contain $t \leq k \leq 2t$ keys where t is the tree order [2]
- Deletion is probably faster than in B-tree since it is always performed on the leaf nodes

[2] K. Pollari-Malmi, “B⁺-trees,” *University of Helsinki*. [Online]. Available: https://www.cs.helsinki.fi/u/mluukkai/tirak2010/B-tree.pdf [Accessed: Nov. 18, 2018].

SQLITE EXTENSIONS

- SQLite is the popular open-source embedded relational DBMS, written in the C language
- Uses the B-tree as the default index
- SQLite extensions are the dynamically linked libraries [3]

[3] SQLite, “Run-Time Loadable Extensions,” *SQLite*. [Online]. Available: https://www.sqlite.org/loadext.html [Accessed: Nov. 18, 2018].

SUMMARY

- **B-tree** is the balanced search tree for data indexing
- **B$^+$-tree** is the B-tree modification with better deletion performance
- **SQLite** is the popular open-source embedded RDBMS which supports adding new features using **extensions**

REFERENCES

[1] R. Bayer, E. McCreight, “Organization and maintenance of large ordered indexes,” *Acta Informatica*, vol. 1, no. 3, pp. 173 – 189, 1972.
[2] K. Pollari-Malmi, “B+-trees,” *University of Helsinki*. [Online]. Available: https://www.cs.helsinki.fi/u/mluukkai/tirak2010/B-tree.pdf [Accessed: Nov. 18, 2018].
[3] SQLite, “Run-Time Loadable Extensions,” *SQLite*. [Online]. Available: https://www.sqlite.org/loadext.html [Accessed: Nov. 18, 2018].

NATIONAL RESEARCH
UNIVERSITY

Thank you for your attention!

amrigin@edu.hse.ru
anton19979@yandex.ru
anton19979@yandex-team.ru