Heinrich Heine
University Düsseldorf

Institut für Informatik
Datenbanken und Informationssysteme
Prof. Dr. Stefan Conrad
Thomas Germer

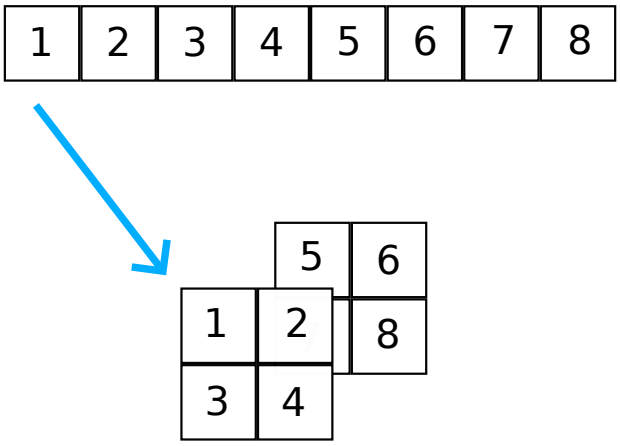
WiSe 24/25
Assignment 9
Due December 17, 8:00 AM

Relational Databases and Data Analysis

- As usual, use `zipme.py` to create the archive to upload in ILIAS.

Exercise 1 *Array Storage* (2 + 1 + 2 Points)
To work with data cubes, we need a way to map between higher-dimensional and single-dimensional data structures, because common memory only supports one-dimensional addressing.

(a) Store an array in an n-dimensional hypercube ($n \in \mathbb{N}$).

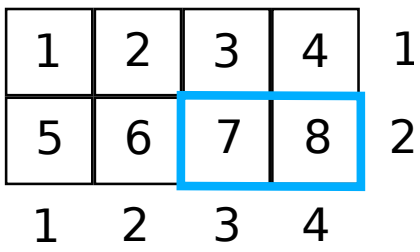


(b) Do the opposite.

(c) Extract a sub-hypercube from the hypercube given the bounds of the sub-hypercube. The sub-hypercube contains the values where the index vector $x_i \in \{1, \ldots, D_i\}$ lies within the two bounds $x_i^{\min}, x_i^{\max} \in \{1, \ldots, D_i\}$:

$$x_i^{\min} \leq x_i \leq x_i^{\max}. \tag{1}$$

For example, the following sub-rectangle $[[7, 8]]$ has bounds $x^{\min} = [3, 2]$ and $x^{\max} = [4, 2]$.



Exercise 2 *GROUP BY Implementations* (2 Points)

The function `group_by_cube_slow` in `test_exercise_2_a.py` computes the following SQL query for relation `mytable(a, b, c, d, e, value)` by iterating over all possible combinations of the first five attributes `(a, b, c, d, e)`, grouping them using a hash table and then summing over the last attribute `value`.

```
SELECT
        a, b, c, d, e, SUM(value)
FROM
        mytable
GROUP BY
        CUBE(a, b, c, d, e)
```

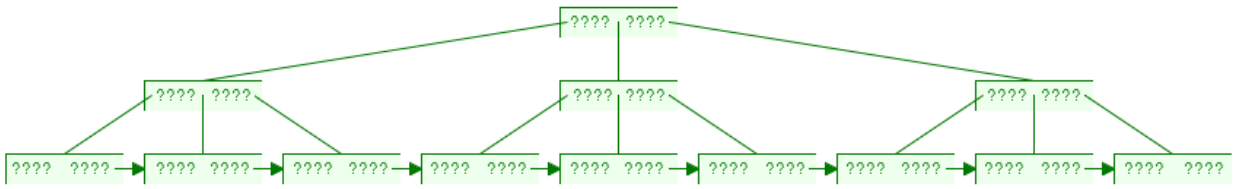
(a) Implement a solution that computes the same result while being at least 30 times faster. You may assume that there are few different values for every individual attribute.

Exercise 3 *B^+ trees* (2 + 1 Points)

The B^+ tree is an associative data structure which is used by almost all database systems. You can find a nice interactive visualization on the following website:

`https://www.cs.usfca.edu/~galles/visualization/BPlusTree.html`[1]

(a) In `test_exercise_3_a.py`, you can find the implementation of such a tree with degree 3, which means that a tree has at most two keys. It would be nice if we could fully utilize all nodes. To do so, come up with a list of 18 keys to insert such that the tree has 13 nodes.



(b) In `exercise_3_b.txt`, write when you would use a B^+ tree over a B-tree. Keep your answer brief.

[1] Mirror: `https://cuuduongthancong.com/~galles/visualization/BPlusTree.html`