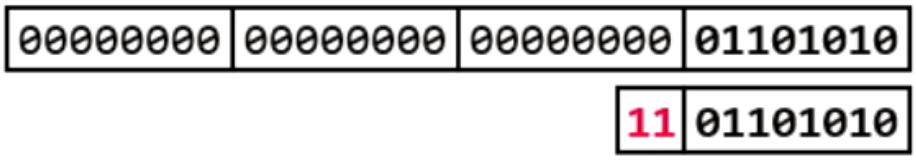
DB Compression Overview

- fit larger datasets in memory
- less I/O
- better cache uitilization
- some DBs allow query processing directly on compressed data
  - **#1** Page-level compression (general-purpose GZIP, Snappy, LZ4)
  - **#2** Row-level heavyweight/lightweight compression (e.g., Huffman)
  - **#3** Column-level lightweight compression (NS, RLE, DICT, Delta, FOR → next slide)
  - **#4** Specialized log and index compression

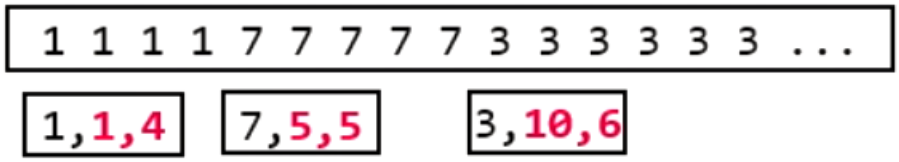
[Patrick Damme et al: Ligh Data Compression Algoritl Experimental Survey. **EDB**

Lightweight Database Compression Schemes

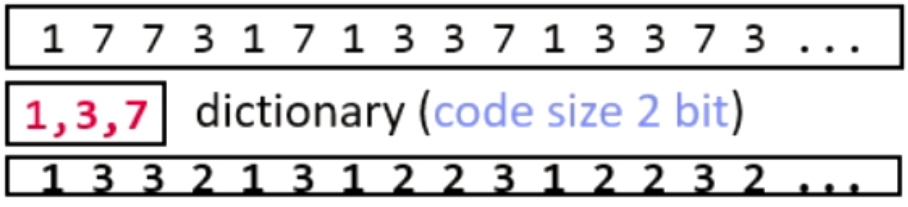
- null suppresion
  - compress integers with leading zeros



- run-length encoding
  - compress sequences of equal values by “runs”
  - each run consists of
    - value
    - start
    - length

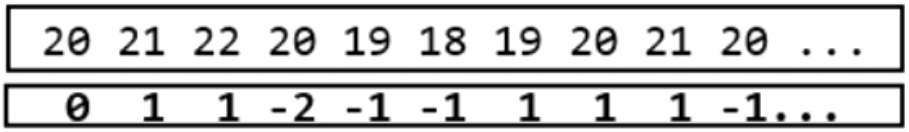


- dictionary encoding
  - compress column with few distinct values
  - create dictionary with all values
  - store pos in dictionary instead of actual value

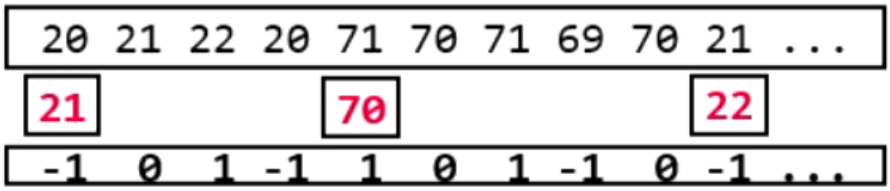


    - compression would be more effective if values were strings instead

- delta encoding
  - compress sequence with small changes
  - store delta/change to previous value



- frame-of-reference encoding
  - compress values by storing delta to reference value
  - outlier handling



[[Physical Design]]