17.02.2025 Week 1 exercises: Storage

Exercise 1:

Given relations R and S, and the following query:

```
select Ra from R, S
where Rc = Sb and 5<Ra<20 and 40<Rb<50 and 49<Sa<65
```

Give the execution steps of the query in a row store, and in a column store. Specifically, provide the output of each operator given a row store or a column store.

R

Ra	Rb	Rc
3	12	12
16	34	34
56	75	53
9	45	23
11	49	78
27	58	65
8	97	33
41	75	21
19	42	29
35	55	0

S

Sa	Sb
17	11
49	35
58	62
99	44
64	29
37	78
53	19
61	81
32	26
50	23

Solution

Answer: Row store

a) Filtering tables

R

Ra	Rb	Rc
3	12	12
16	34	34
56	75	53
9	45	23
11	49	78
27	58	65
8	97	33
41	75	21
19	42	29
35	55	0

S

Sa	Sb
17	11
49	35
58	62
99	44
64	29
37	78
53	19
61	81
32	26
50	23

filtering ⇒

R

Ra	Rb	Rc
9	45	23
11	49	78
19	42	29

S

Sa	Sb
58	62
64	29
53	19
61	81
50	23

b) Joining tables

Ra	Rb	Rc	Sa	Sb
9	45	23	50	23
19	42	29	64	29

c) Projecting Ra ⇒ **9, 19**

Answer: Column store

R

Ra
3
16
56
9
11
27
8
41
19
35

Rb
12
34
75
45
49
58
97
75
42
55

Rc
12
34
53
23
78
65
33
21
29
0

S

Sa
17
49
58
99
64
37
53
61
32
50

Sb
11
35
62
44
29
78
19
81
26
23

Naive solution for column store

We assume that early materialization is used. As all columns are accessed, it reconstructs the full tuples

Reconstruct tuple

Ra	Rb	Rc
3	12	12
16	34	34
56	75	53
9	45	23
11	49	78
27	58	65
8	97	33
41	75	21
19	42	29
35	55	0

Select(Ra, 5, 20)

Ra	Rb	Rc
3	12	12
16	34	34
56	75	53
9	45	23
11	49	78
27	58	65
8	97	33
41	75	21
19	42	29
35	55	0

Select(Rb, 40, 50)

Ra	Rb	Rc
16	34	34
9	**45**	23
11	**49**	78
8	97	33
19	**42**	29

Reconstruct tuple

Sa	Sb
17	11
49	35
58	62
99	44
64	29
37	78
53	19
61	81
32	26
50	23

Select(Sa, 49, 65)

Sa	Sb
17	11
49	35
58	62
99	44
64	29
37	78
53	19
61	81
32	26
50	23

Join as in the solution for the row store, followed by projection on Ra.

Optimized solution for column store

By pushing a selection down to the storage, this solutions prunes the virtual id set involved in reconstruction.

Select(Ra, 5, 20)

id	Ra
1	3
2	**16**
3	56
4	**9**
5	**11**
6	27
7	**8**
8	41
9	**19**
10	35

id
2
4
5
7
9

Select(Rb, 40, 50)

id	Rb
2	34
4	**45**
5	**49**
7	97
9	**42**

id
4
5
9

Reconstruct tuple

Ra	Rb	Rc
9	45	23
11	49	78
19	42	29

Select(Sa, 49, 65)

id	Sa
1	17
2	49
3	**58**
4	99
5	**64**
6	37
7	**53**
8	**61**
9	32
10	**50**

id
3
5
7
8
10

Reconstruct tuple

Sa	Sb
58	62
64	29
53	19
61	81
50	23

Join as in the solution for the row store, followed by projection on Ra.

Exercise 2:

Assume a single-relation schema with a relation R(A0, ..., A127), with 128 4-byte integer columns and ||R|| number of records. Consider a columnar layout. The page size is 8kB. Assume page metadata consumes no space. Consider select-project queries (no joins, no aggregates) which use a total of k columns (by using them in the select predicate theta or choosing them for output). Compute the cost measured in number of pages read; disregard seeks and in-core computation time.

Solution

In total we have $k \cdot ||R|| \cdot 4$ Bytes
Number of pages:

$$\frac{\text{columns} \cdot \text{records} \cdot 4B}{\text{page-size}} = \frac{k \cdot ||R|| \cdot 4B}{8kB}$$

Exercise 3:

You have a 100 Gb table of the following format:
<int col1, int col2, int col3, int col4, int col5>.

The following queries are very frequent on this table:

Q1: `SELECT col1 FROM tbl`

Q2: `SELECT col3,col5 FROM tbl`

Q3: `SELECT col2 FROM tbl`

Q4: `SELECT col4 FROM tbl`

Design a good storage layout for this table that would help on the performance of these four queries. Explain your answer briefly.

Solution

Columnar storage would be the best choice for this table, as only individual columns are accesed with each query.

Exercise 4:

Assume the tables are stored in column layout. Which compression techniques presented in the lecture would you choose for the following tables? You are allowed to combine compression techniques. Assume that you can bit-pack the keys of dictionary keys.

(**Note**: do not think about possible future data for this table, just try to compress the given data as well as possible.)

visitors

ID	Downloads	IP_address
13	217	138.92.122.175
81	0	138.92.122.195
42	6	138.92.122.182
25	4	138.92.122.181
21	52	138.92.122.177
56	4	138.92.122.188
78	2	138.92.122.191
30	1	138.92.122.185
23	0	138.92.122.179
80	2	138.92.122.193
27	3	138.92.122.183
82	0	138.92.122.197

issues

ID	Status	Subject
1	In Progress	Migrate Moodle site to Turnkey VM
2	In Progress	Come up with backup strategy
5	New	Test recovery of Moodle site
6	Resolved	Drink fifth coffee
7	Resolved	Buy next coffee
8	Resolved	Test group selection for students
9	Resolved	Set up Moodle site

Solution

Visitors: Dictionary on `IP_address` column to reduce the size of the IP addresses
Issues:
Dictionary on `Subject` column in form of tokenization: mapping from each word/token to an int and represent a text as a list of integers corresponding to each word/token.
For the `Status` column there are two options that offer a good compression:

a) Dictionary + RLE on `Status`:

Size of dictionary: $3 \cdot$ (string size + 2bits)
Column size: $3 \cdot$ 2bits for the ids and $3 \cdot$ 3bits for the cardinalities (see below). Total: 15bits

New representation of `Status` in memory:
(1, 2)

(2, 1)
(3, 4)

b) Bit-vector:

$3 \cdot$ string size for storing the distinct values.
$3 \cdot 7$bits for the bit vector.