

>>

Selection Overview

- Varieties of Selection
- Implementing Select Efficiently

COMP9315 21T1 ◇ Selection ◇ [0/6]



❖ Varieties of Selection

Selection: `select * from R where C`

- filters a subset of tuples from one relation **R**
- based on a condition **C** on the attribute values

We consider three distinct styles of selection:

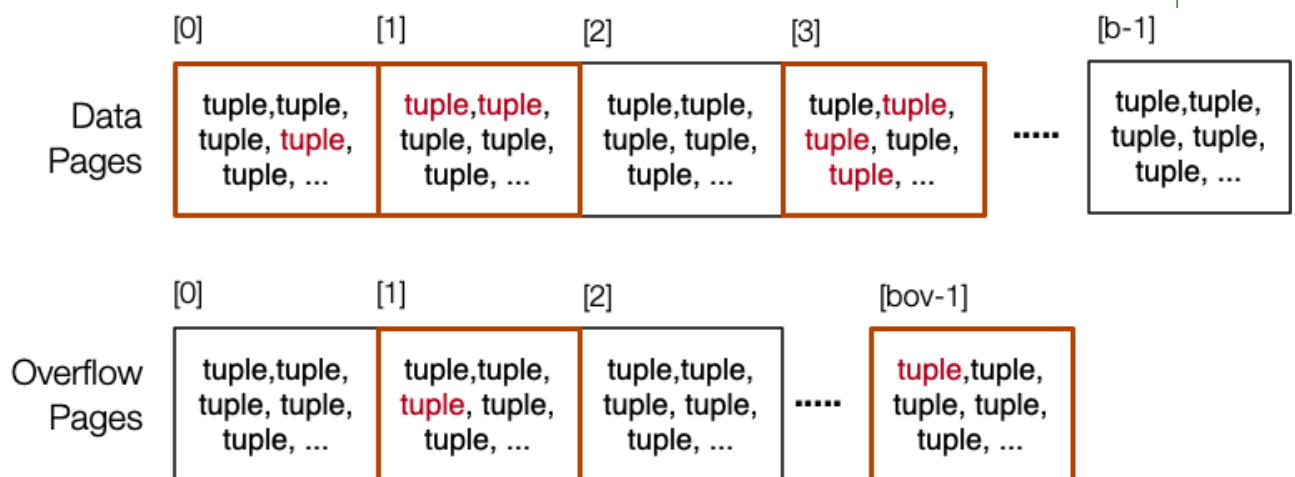
- 1-d (one dimensional) (condition uses only *1* attribute)
- *n*-d (multi-dimensional) (condition uses *>1* attribute)
- similarity (approximate matching, with ranking)

Each style has several possible file-structures/techniques.

❖ Varieties of Selection (cont)

Selection returns a subset of tuples from a table

- r_q = number of tuples that match query q
- b_q = number of pages containing tuples that match query q



In the diagram, $r_q = 8$, $b_q = 5$

❖ Varieties of Selection (cont)

Different categories of selection queries:

one ... queries with at most 1 result ... $0 \leq r_q \leq 1$, $0 \leq b_q \leq 1$

- typically, equality test on primary key attribute, e.g.
- **select * from R where id = 1234**

pmr ... partial match retrieval ... $0 \leq r_q \leq r$, $0 \leq b_q \leq b + b_{ov}$

- conjunction of equality tests on multiple attributes, e.g.
- **select * from R where age=65 (1-d)**
- **select * from R where age=65 and gender='m' (n-d)**

❖ Varieties of Selection (cont)

More categories of selection queries:

rng ... range queries ... $0 \leq r_q \leq r$, $0 \leq b_q \leq b + b_{ov}$

- conjunction of inequalities, on one or more attributes, e.g.
- **select * from R where age ≥ 18 and age ≤ 21 (1-d)**
- **select * from R where 18 ≤ age ≤ 21 and 160 ≤ height ≤ 190 (n-d)**

pat ... pattern-based queries ... $0 \leq r_q \leq r$, $0 \leq b_q \leq b + b_{ov}$

- string-based matching using **like** or regular expressions
- **select * from R where name like '%oo%'**
- **select * from R where name ~ '^Smi'**

❖ Varieties of Selection (cont)

More categories of selection queries:

sim ... similarity matching ... in theory, $r_q = r$... everything matches to some degree

- uses "similarity" measure ($0 \leq sim \leq 1$, 0=different, 1=identical)
- **select * from Images where similar to SampleImage**
- results are ranked by *sim* value, from most to least similar
- can become a filter via
 - threshold ... only items where $sim \geq$ min similarity
 - top-k ... *k* items with highest similarities

We focus on **one**, **pmr** and **rng** queries, but will discuss others

❖ Implementing Select Efficiently

Two basic approaches:

- physical arrangement of tuples
 - sorting (search strategy)
 - hashing (static, dynamic, n -dimensional)
- additional indexing information
 - index files (primary, secondary, trees)
 - signatures (superimposed, disjoint)

Our analysis assumes 1 input buffer available for each relation.

If more buffers are available, most methods benefit.

Produced: 6 Mar 2021