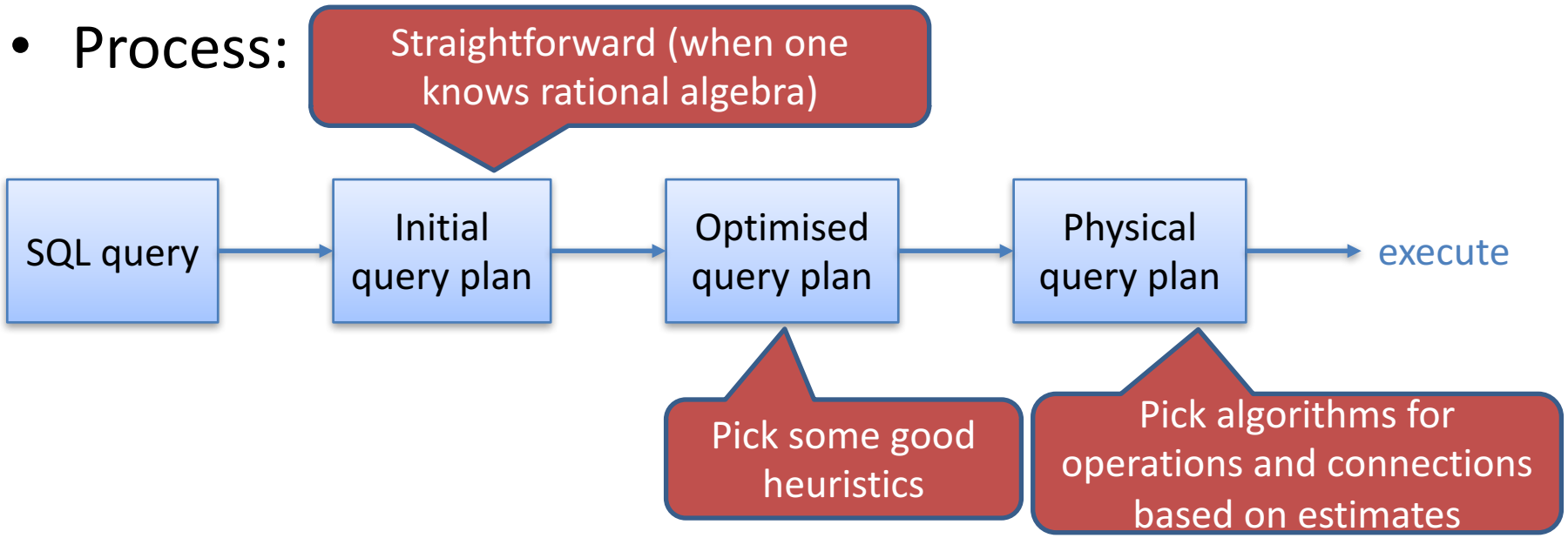
COMP207
Database Development

Lecture 17

Review of query processing

Review

- Query processing is the main task of DBMS
- Process:



- Heuristics: push selections and proje ions down and change to equijoin if possible (j selection)

Joins

- How to estimate R ⋈ S?

Assume A is the only common attribute.

- Simple estimate based on size of R & S and number of distinct values in common attributes

$$\frac{|R| \times |S|}{\text{max. number of distinct values for } A \text{ in } R \text{ or } S}$$

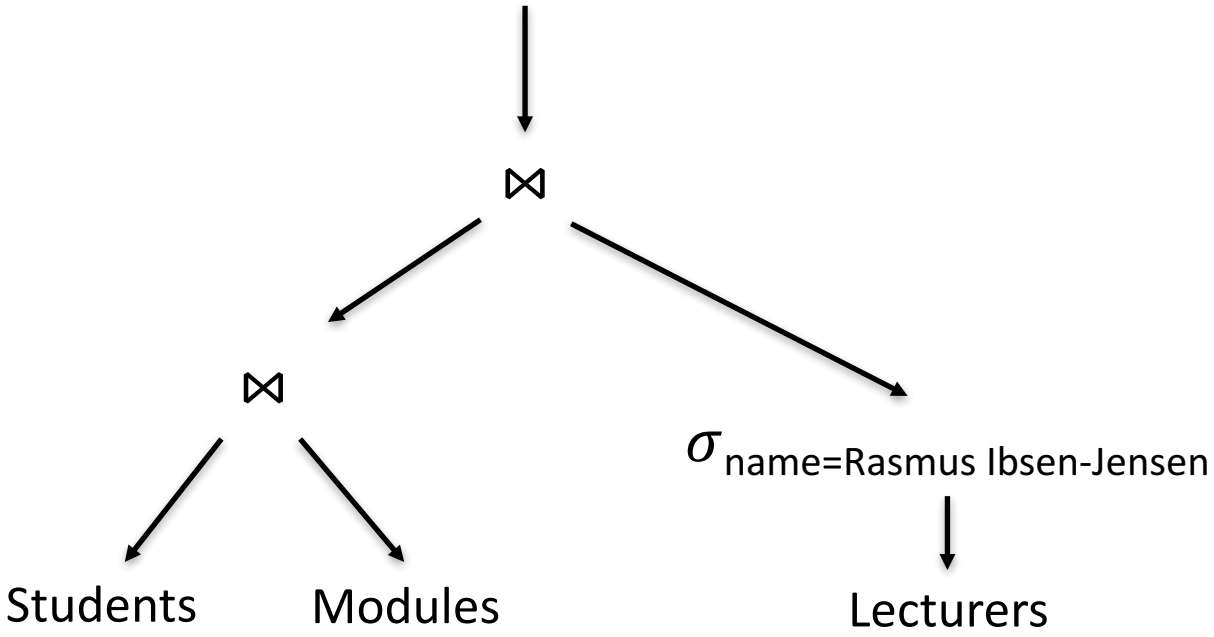
- As for selection, based on assumptions that might not always lead to good estimates
- More sophisticated methods:
  - Many and still a topic of active research
  - See, e.g., SIGMOD/PODS/VLDB conferences

Other Issues

- **How to generate** physical query plans?
  - Explore all?
  - More sensible approaches: top-down/bottom-up
- Selection of a **suitable algorithm** for each operator
  - based on size of intermediate result
- Selection of a **good join order**
  - also based on size of intermediate results
- How to **pass information** from one operator to another?

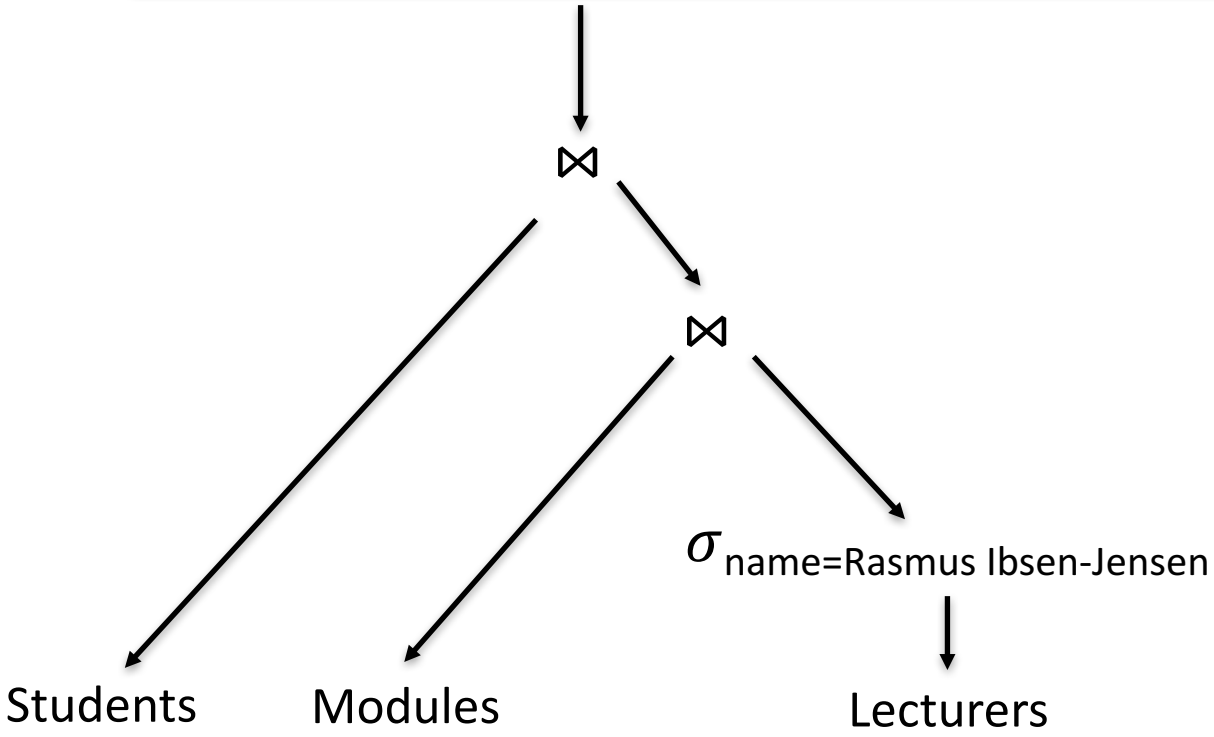
Example where join order matters

```
SELECT *
FROM Lecturers NATURAL JOIN Modules NATURAL JOIN Students
WHERE Lecturers.name = Rasmus Ibsen-Jensen
```



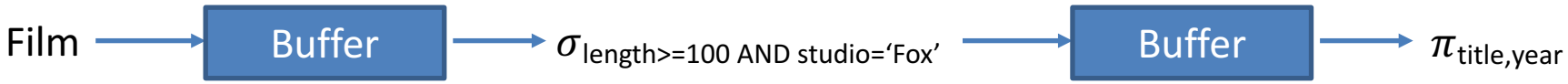
Example where join order matters

```
SELECT *
FROM Lecturers NATURAL JOIN Modules NATURAL JOIN Students
WHERE Lecturers.name = Rasmus Ibsen-Jensen
```

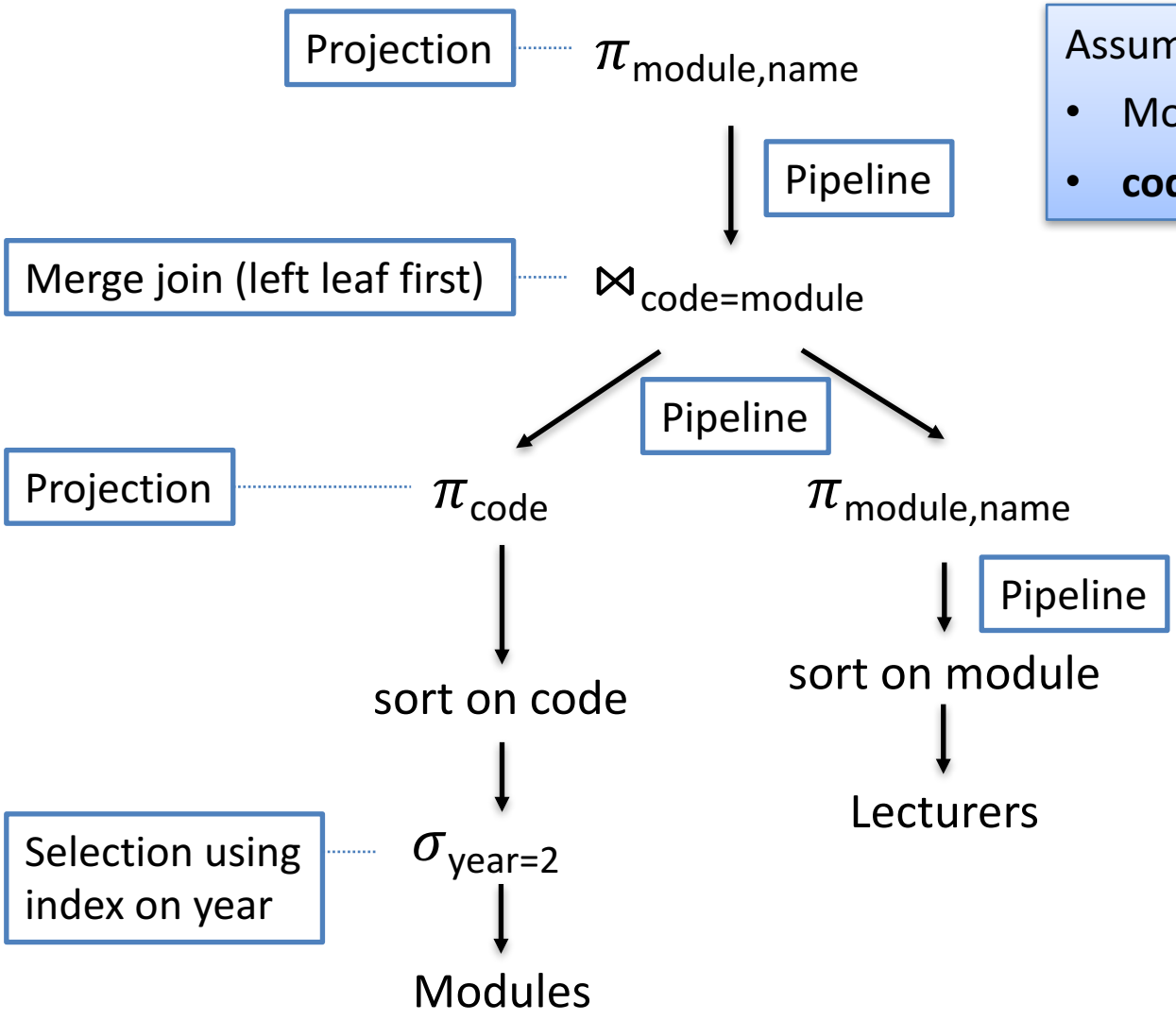


Passing Information

- Materialisation: write intermediate results to disk
- Pipelining ("stream-based processing")
  - Passes the tuples of one operation directly to the next operation without using disk
  - Extra buffer for each pair of adjacent operations to hold tuples passing from one relation to the other
  - Example:
    - $\pi_{\text{title,year}}(\sigma_{\text{length>=100 AND studio='Fox'}}(\text{Film}))$
    - With pipelining, the intermediate result of the selection will be written into a buffer in memory, from which the projection operator will read and process these tuples directly



A Physical Query Plan



Assumption:

- Modules has index on **year**
- **code** is a key of Modules

Sample Exam Question for Chapter 3

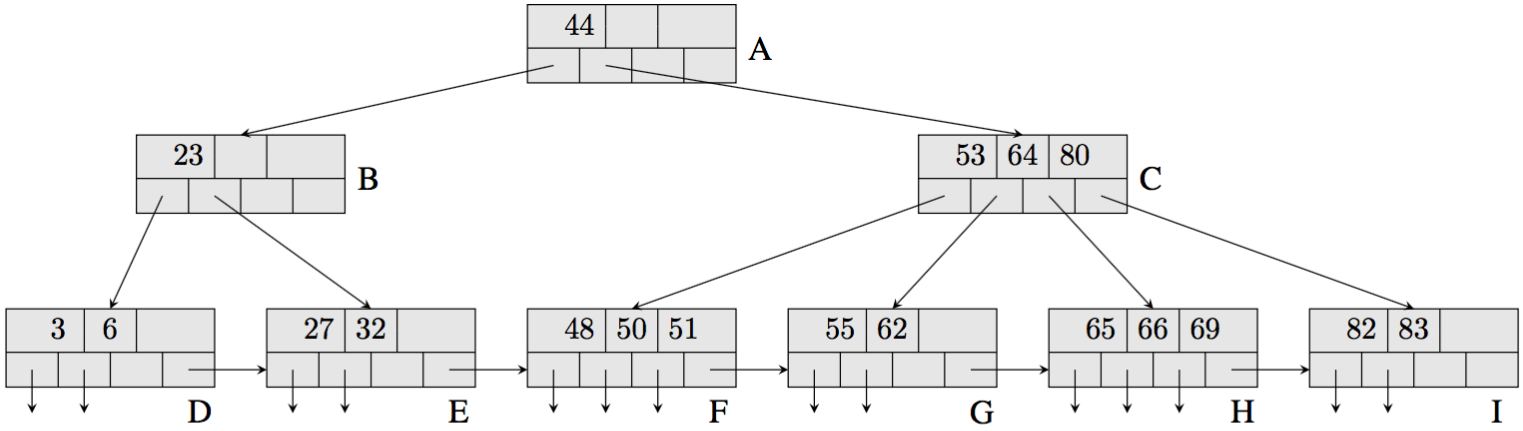


Figure 1: B+ tree for Question 1.

---

1. Consider the B+ tree in 1. Because we use $n = 3$ for each node, we use $x = 2$ as the lower bound on the number of pointers a node can have. What happens if we delete 3? We find and delete 3 in node D and...

- ☐ **A.** do nothing else.
- ☐ **B.** steal a pointer from node E and update the least common ancestor of E and D.
- ☐ **C.** merge with node E.
- ☐ **D.** merge with node E, move a pointer from node C to node B and update node A.
- ☐ **E.** split D.