Database Technology

Topic 12: Query Processing and Optimization

Olaf Hartig
olaf.hartig@liu.se

LINKÖPING UNIVERSITY

General Concepts

Quiz

The purpose of query processing is to ...

(1) … prepare the database such that we can retrieve data from it

(2) … produce a logical plan for a given query

(3) … produce a physical plan for a given query

(4) … produce the result of a given query

Quiz

Only one of the following statements is correct. Which one?

(1) Query validation ensures that the cost of the given query is not too high according to the cost model.

(2) Parsing converts an SQL query string into a program that produces the result of the query.

(3) There are some physical operators without any counterpart among the logical operators.

(4) For every query there exists exactly one logical plan.

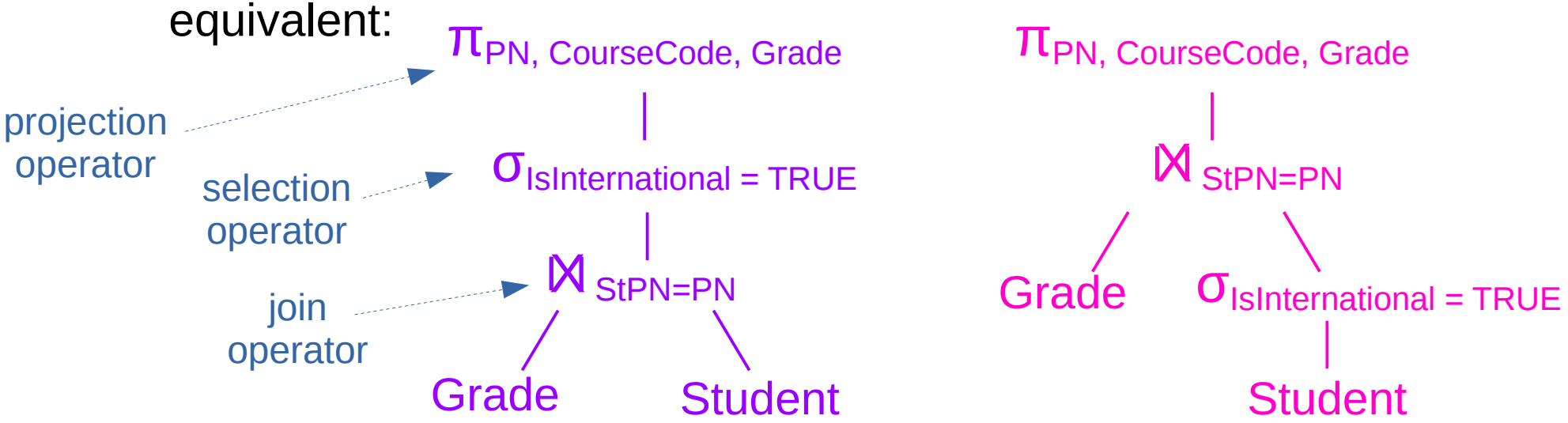
Logical Plans and Logical Optimization

Quiz

- Consider the following SQL query

```
SELECT Student.PN, Grade.CourseCode, Grade.Grade
FROM Grade JOIN Student ON Grade.StPN = Student.PN
WHERE Student.IsInternational = TRUE;
```

- Here are two logical plans for this query, which are semantically equivalent:

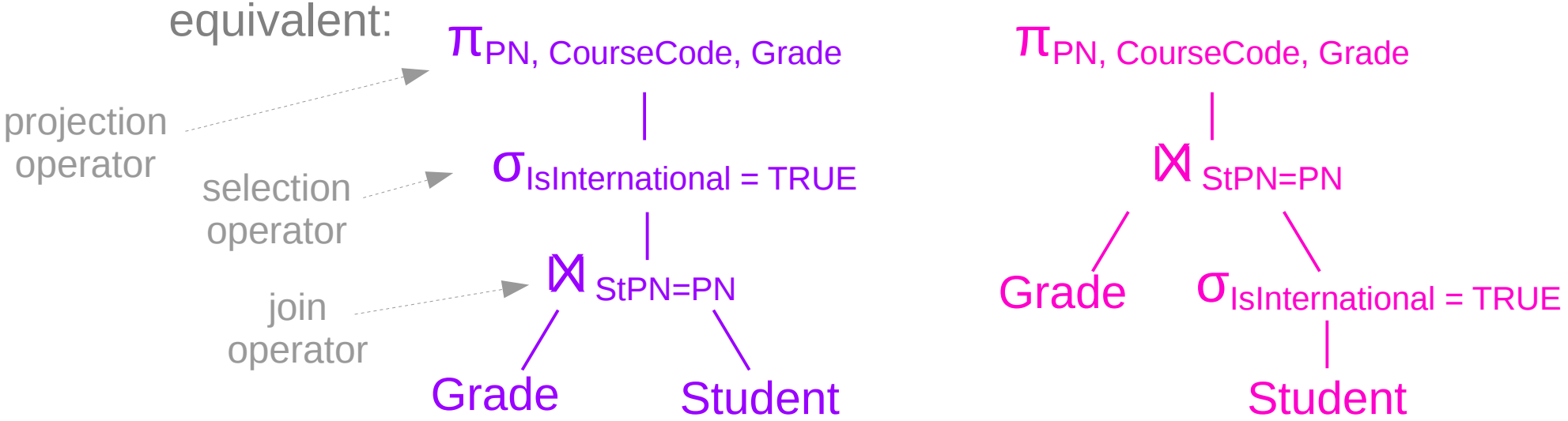


Quiz

- Consider the following SQL query

```
SELECT Student.PN, Grade.CourseCode, Grade.Grade
FROM Grade JOIN Student ON Grade.StPN = Student.PN
WHERE Student.IsInternational = TRUE;
```

- Here are two logical plans for this query, which are semantically equivalent:



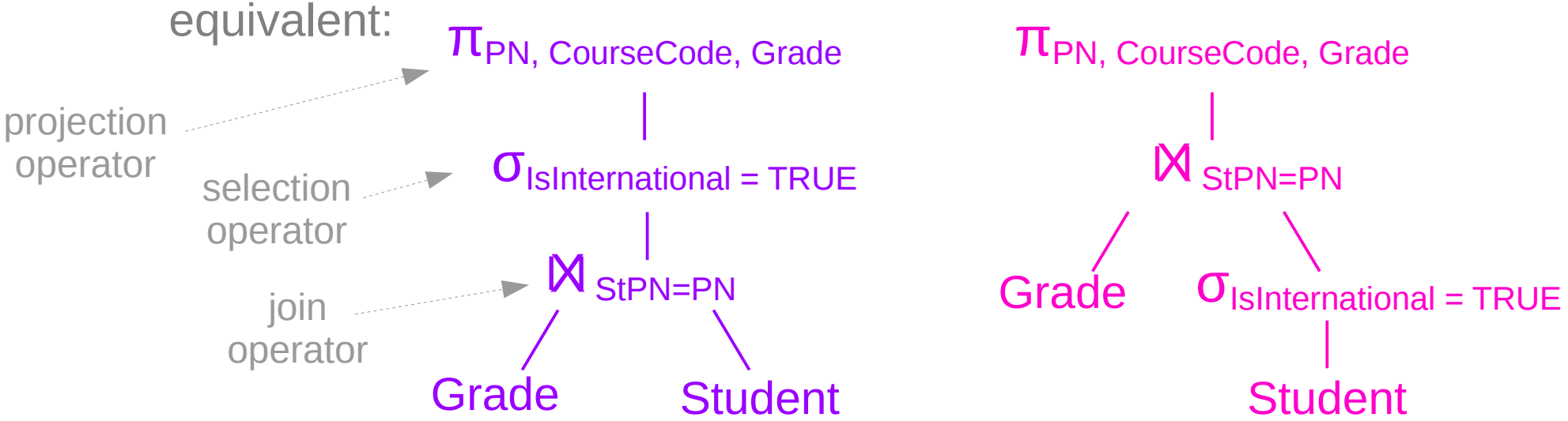
- Which of these two logical plans is likely more efficient?
  (1) the plan on the left (2) the plan on the right

Let's Calculate

- Consider the following SQL query

```
SELECT Student.PN, Grade.CourseCode, Grade.Grade
FROM Grade JOIN Student ON Grade.StPN = Student.PN
WHERE Student.IsInternational = TRUE;
```

- Here are two logical plans for this query, which are semantically equivalent:



- Assume 10,000 students, each of them has a grade in about 10 courses (i.e., ca 100,000 rows in Grade); 1,000 are international

Physical Operators and Physical Query Optimization

Quiz

- Suppose we want to join two relations, *Student* and *Grade*
  - the file for *Student* consists of 100 pages
  - the file for *Grade* consists of 1,000 pages
- Assume we use the nested loops join (basic version) where the outer loop iterates over the *Student* relation
- What is the I/O cost in terms of page reads?

  (1) 1,100

  (2) 100,100

  (3) 101,000

  (4) 110,000

Quiz

- Suppose we want to join two relations, *Student* and *Grade*
  - the file for *Student* consists of 100 pages
  - the file for *Grade* consists of 1,000 pages
- Assume we use the nested loops join (basic version) where the outer loop iterates over the *Student* relation
- What is the I/O cost in terms of page reads?

  (1) 1,100

  (2) **100,100**

  (3) 101,000

  (4) 110,000
- Formula: pages(*outer*) + pages(*outer*) · pages(*inner*)

Block Nested Loops Join

- Suppose we want to join two relations, *Student* and *Grade*
  - the file for *Student* consists of 100 pages
  - the file for *Grade* consists of 1,000 pages
- Assume we use the nested loops join (basic version) where the outer loop iterates over the *Student* relation
- What is the I/O cost in terms of page reads?

  (1) 1,100

  (2) **100,100**

  (3) 101,000

  (4) 110,000
- Formula: pages(*outer*) + pages(*outer*) · pages(*inner*)
- Block-NLJ with 10 buffers for Student: 10,100 page reads

Exercise

- Suppose we want to join two relations, *Student* and *Grade*
  - the file for *Student* consists of 100 pages
  - the file for *Grade* consists of 1,000 pages
- Assume we use the sort-merge join
  - external merge sort for the sorting, i.e., the I/O cost of sorting a relation is $2 \times p \times \lceil \log_m(p) \rceil$ page reads & writes
  - assume we have only 2+1 buffers for it, i.e., m=2
- Calculate the I/O cost needed for the sort-merge join
  - ignore the cost for writing the result

$$\lceil \log_2(100) \rceil = 7$$
$$\lceil \log_2(1{,}000) \rceil = 10$$

Solution

- Suppose we want to join two relations, *Student* and *Grade*
  - the file for *Student* consists of 100 pages
  - the file for *Grade* consists of 1,000 pages
- Assume we use the sort-merge join
  - external merge sort for the sorting, i.e., the I/O cost of sorting a relation is $2 \times p \times \lceil \log_m(p) \rceil$ page reads & writes
  - assume we have only 2+1 buffers for it, i.e., m=2
- Calculate the I/O cost needed for the sort-merge join
  - sorting of *Student* relation: 1,400 page reads and writes
  - sorting of *Grade* relation: 20,000 page reads and writes
  - merge phase: 100 + 1,000 = 1,100 page reads
  - total: 22,500 page reads and writes

$$\lceil \log_2(100) \rceil = 7$$
$$\lceil \log_2(1{,}000) \rceil = 10$$

Let's use more buffers!

- Suppose we want to join two relations, *Student* and *Grade*
  - the file for *Student* consists of 100 pages
  - the file for *Grade* consists of 1,000 pages
- Assume we use the sort-merge join
  - external merge sort for the sorting, i.e., the I/O cost of sorting a relation is $2 \times p \times \lceil \log_m(p) \rceil$ page reads + writes
  - assume we have 10+1 buffers for it, i.e., m=10
- Calculate the I/O cost needed for the sort-merge join
  - sorting of *Student* relation: 400 page reads and writes
  - sorting of *Grade* relation: 6,000 page reads and writes
  - merge phase: 100 + 1,000 = 1,100 page reads
  - total: 7,500 page reads and writes

$\lceil \log_{10}(100) \rceil = 2$

$\lceil \log_{10}(1{,}000) \rceil = 3$

www.liu.se

LINKÖPING UNIVERSITY