INFO20003 Database Systems

Dr Renata Borovica-Gajic

Lecture 14
Query Optimization Part II

Semester 1 2018, Week 7

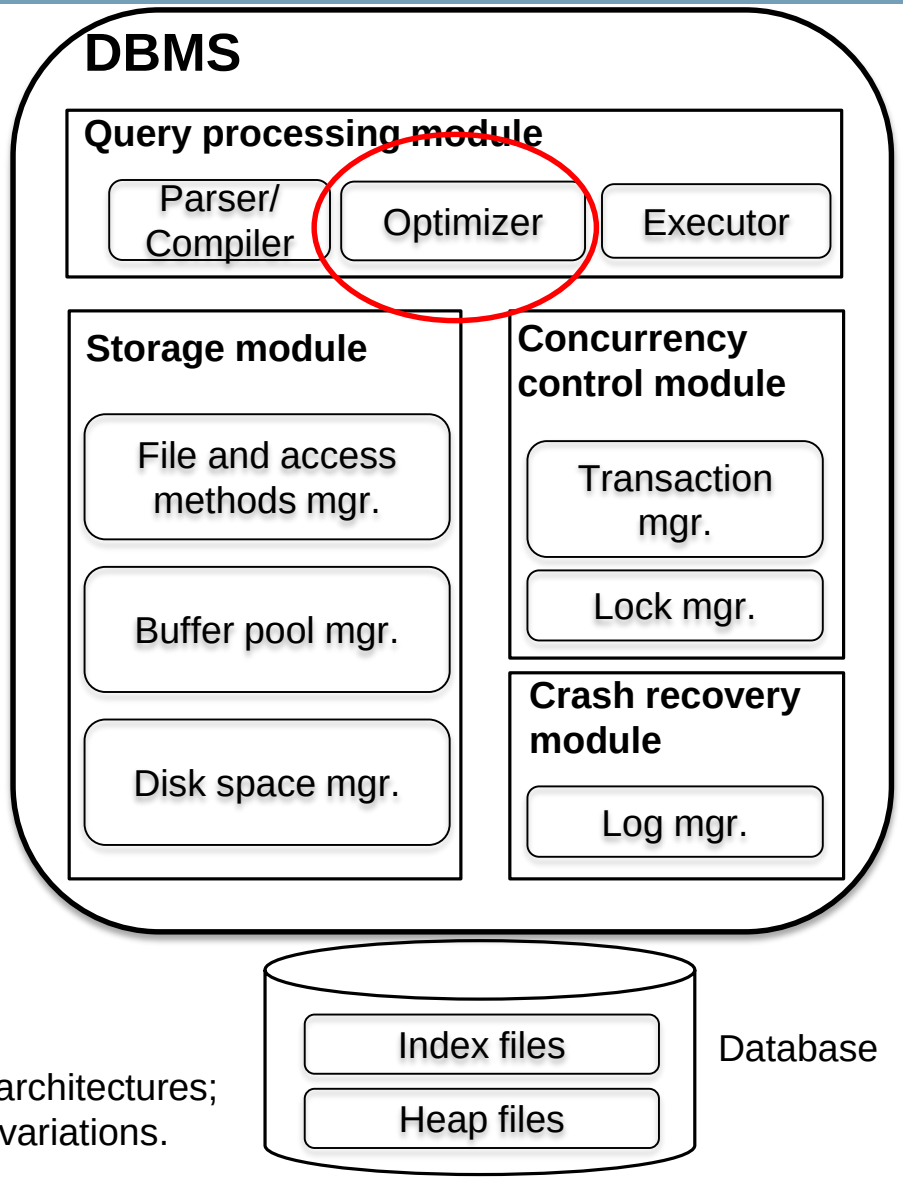
WHY ARE DATABASES COOL ?

Featuring people from industry

Part 1

SNOWFLAKE COMPUTING, USA

Remember this? Components of a DBMS



TODAY
Plan enumeration

This is one of several possible architectures; each system has its own slight variations.

© University of Melbourne 2018

Enumeration of Alternative Plans

- When enumerating alternative plans, there are two main cases:
  - Single-relation plans
  - Multiple-relation plans (joins)

- For queries over a **single relation:**
  - Each *available* access path (file scan / index) is considered, and the one with the lowest estimated cost is chosen
    - Heap scan is always one alternative
    - Each index can be another alternative (if matching selection predicates)
  - Other operations can be performed on top of access paths, but they typically don't incur additional cost since they are done *on the fly* (e.g. projections, additional non-matching predicates)

© University of Melbourne 2018

Cost Estimates for Single-Relation Plans

1. Sequential (heap) scan of data file:
   Cost = NPages(R)

2. Index selection over a primary key (just a single tuple):
   Cost(B+Tree)=Height(I)+1, Height is the index height
   Cost(HashIndex)= ProbeCost(I)+1, ProbeCost(I)~1.2

3. Clustered index matching one or more predicates:
   Cost(B+Tree)=(NPages(I) + NPages(R))* $\prod_{i=1..n} RF_i$
   Cost(HashIndex)= NTuples(R)* $\prod_{i=1..n} RF_i * 2.2$

4. Non-clustered index matching one or more predicates:
   Cost(B+Tree)=(NPages(I) + NTuples(R))* $\prod_{i=1..n} RF_i$
   Cost(HashIndex)= NTuples(R)* $\prod_{i=1..n} RF_i * 2.2$

© University of Melbourne 2018

Examples

Let's say that Sailors(S) has 500 pages, 40000 tuples, NKeys(rating) = 10

```
SELECT  S.sid FROM  Sailors S WHERE  S.rating=8
```

- Result size = (1/NKeys(I)) * NTuples(S) = (1/10)*40000 =4000 tuples

1. If we have I(rating), NPages(I) = 50:
   - Clustered index:

   Cost = (1/NKeys(I))*(NPages(I)+NPages(S))=(1/10)*(50+500) = 55 I/O
   - Unclustered index:

   Cost = (1/NKeys(I))*(NPages(I)+NTuples(S))=(1/10)*(50+40000) = 4005 I/O
2. If we have an I(sid), NPages(I)= 50:
   - Cost = ?, Result size = ?
   - Would have to retrieve all tuples/pages. With a clustered index, the cost is 50+500, with unclustered index, 50+40000
3. Doing a file scan:
   - Cost = NPages(S) = 500

© University of Melbourne 2018

Plan Enumeration

Steps:

1. Select **order** of relations
   - E.g. SxRxB, or SxBxR or RxSxB…
   - maximum possible orderings = N!
2. For each join, select **join algorithm**
   - E.g. Hash join, Sort-Merge Join…
3. For each input relation, select access method
   - Heap Scan, or various index alternatives

Q: How many plans are there for a query over N relations?

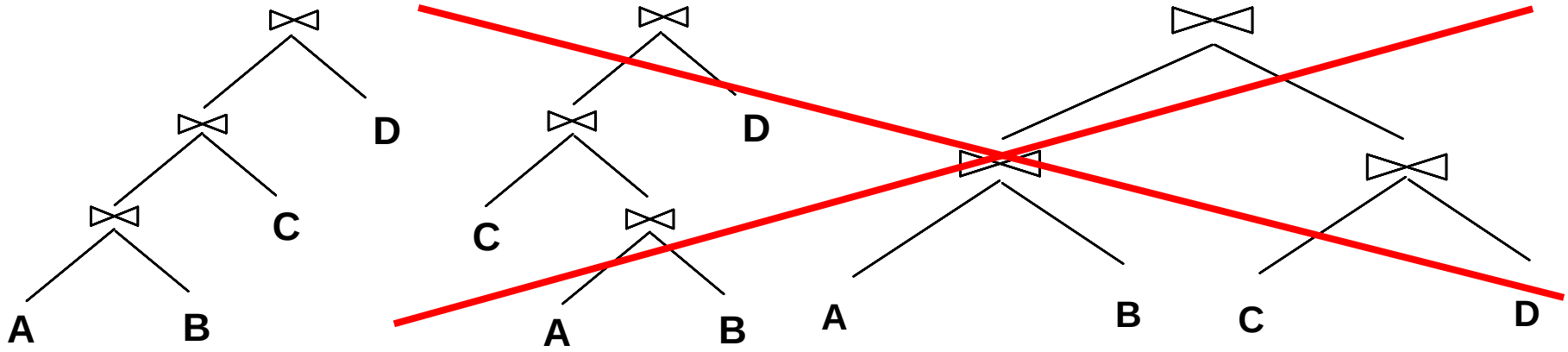
Back-of-envelope calculation:

- With 3 join algorithms, I indexes per relation:
  # plans ≈ $[N!] * [3^{(N-1)}] * [(I + 1)^N]$
- Suppose N = 3, I = 2: # plans ≈ $3! * 3^2 * 3^3 = 1458$ plans
- This is just for illustration – you don't need to remember this

© University of Melbourne 2018

Queries Over Multiple Relations

- As number of joins increases, number of alternative plans grows rapidly → *need to restrict search space*
- Fundamental decision in System R (first DBMS): **only left-deep join trees** are considered
  - Left-deep trees allow us to generate all *fully pipelined* plans
    - Intermediate results are not written to temporary files



© University of Melbourne 2018

Plan Enumeration Example

```
SELECT  S.sname, B.bname, R.day
FROM  Sailors S, Reserves R, Boats B
WHERE  S.sid = R.sid AND R.bid = B.bid
```

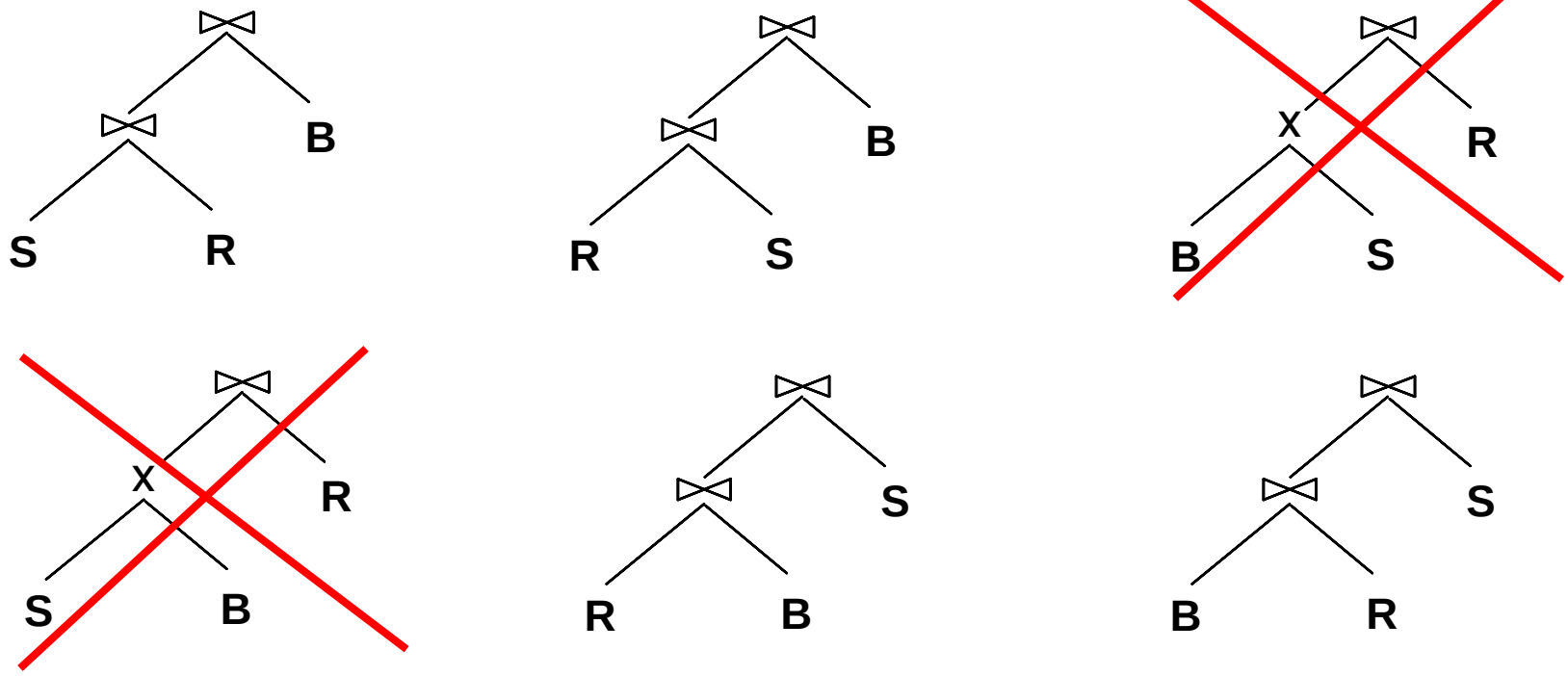
- Let's assume:
  - Two join algorithms to choose from:
    - Hash-Join
    - NL-Join (page-oriented)
  - Unclustered B+Tree index: I(R.sid); NPages(I) = 50
  - No other indexes
  - S: NPages(S) = 500, NTuplesPerPage(S)= 80
  - R: NPages(R) = 1000, NTuplesPerPage(R) = 100
  - B: NPages(B) = 10
  - 100 R $\bowtie$ S tuples fit on a page

© University of Melbourne 2018

Candidate Plans

```
SELECT  S.sname, B.bname, R.day
FROM  Sailors S, Reserves R, Boats B
WHERE  S.sid = R.sid AND R.bid = B.bid
```

1. Enumerate relation orderings:



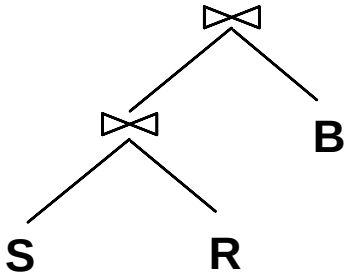
* Prune plans with cross-products immediately!

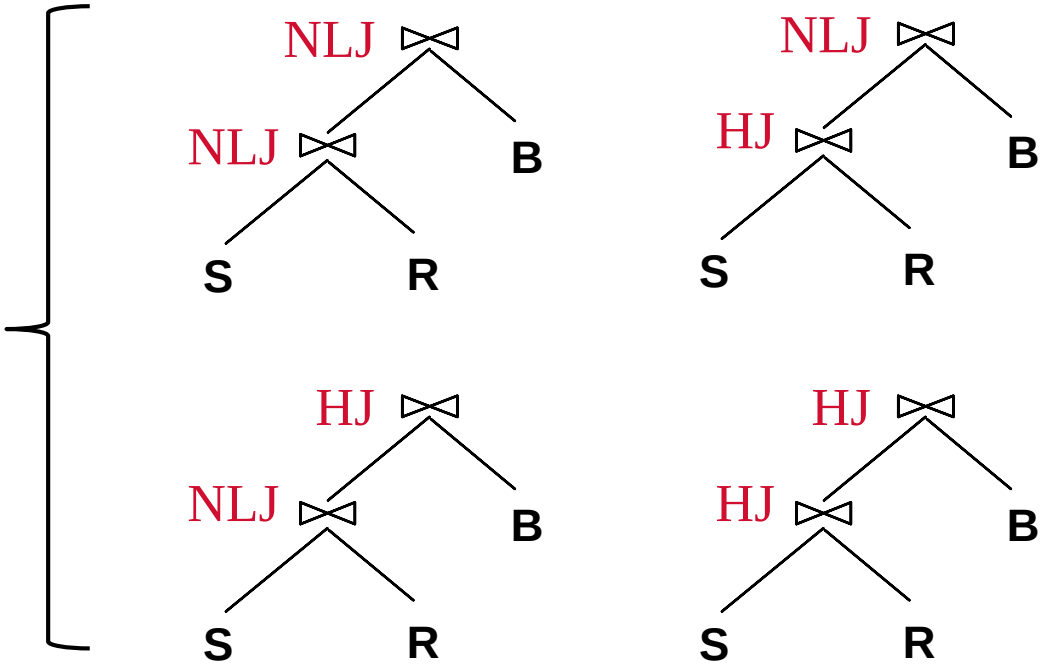
© University of Melbourne 2018

Candidate Plans

```
SELECT  S.sname, B.bname, R.day
FROM  Sailors S, Reserves R, Boats B
WHERE  S.sid = R.sid AND R.bid = B.bid
```

2. Enumerate join algorithm choices:





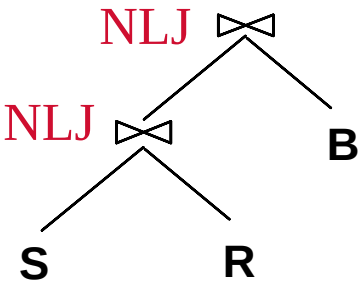
+ do the same for other plans

© University of Melbourne 2018

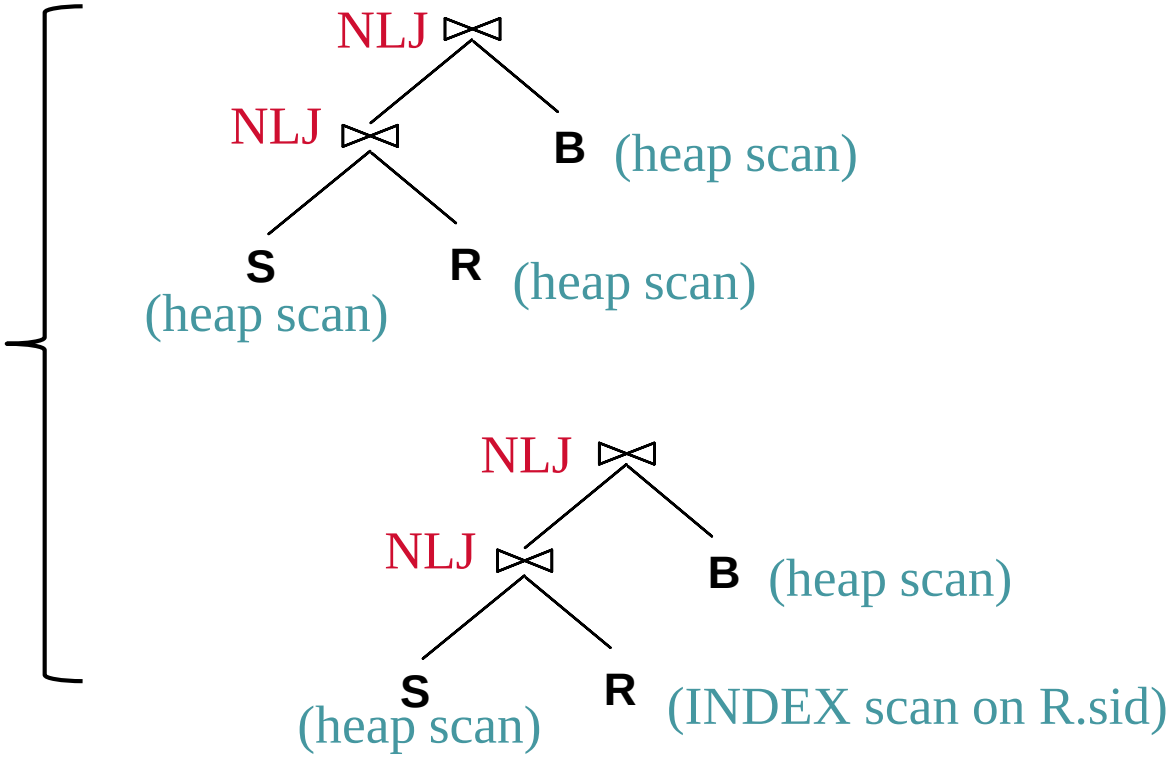
Candidate Plans

```
SELECT  S.sname, B.bname, R.day
FROM  Sailors S, Reserves R, Boats B
WHERE  S.sid = R.sid AND R.bid = B.bid
```

3. Enumerate access method choices:

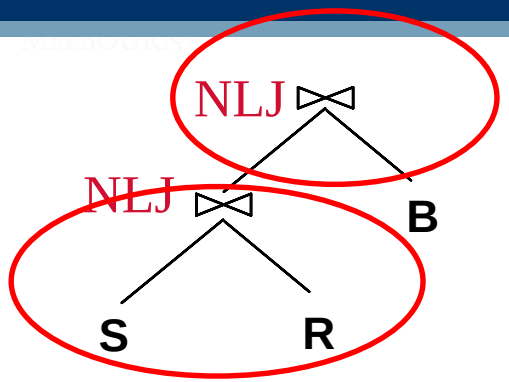


+ do same for other plans



© University of Melbourne 2018

Now estimate the cost of each plan



S: NPages(S) = 500, NTuplesPerPage(S)= 80
R: NPages(R) = 1000, NTuplesPerPage(R) = 100
B: NPages(B) = 10
100 R ⋈ S tuples fit on a page
All 3 relations are Heap Scan

Calculating cost:

SxR

Cost (SxR) = 500 + 500*1000 = 500500

(SxR)xB

Result size (SxR) = 100000*40000 *1/40000 = 100000 tuples = 1000 pages

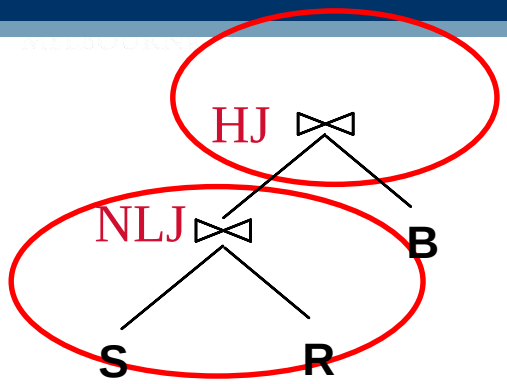
Cost(xB) = ~~1000~~ + 1000*10 = 10000

Already read – left deep plans apply pipelining

Total Cost = 500 + 500*1000 + 1000 * 10 = 510500 I/O

© University of Melbourne 2018

Now estimate the cost of each plan



S: NPages(S) = 500, NTuplesPerPage(S)= 80
R: NPages(R) = 1000, NTuplesPerPage(R) = 100
B: NPages(B) = 10
100 R ⋈ S tuples fit on a page
All 3 relations are Heap Scan

Calculating cost:

SxR

Cost (SxR) = 500 + 500*1000 = 500500

(SxR)xB

Result size (SxR) = 100000*40000 *1/40000 = 100000 tuples = 1000 pages

Cost(xB) = 3*~~1~~2000 + 3*10 = 2*1000 + 3*10 = 2030

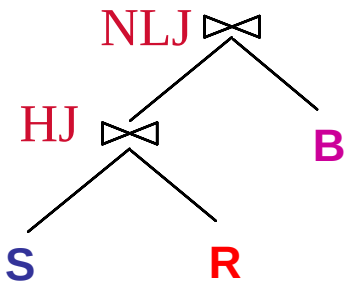
Already read once – left deep plans apply pipelining

Total Cost = 500 + 500*1000 + 2*1000+ 3*10 = 502530 I/O

© University of Melbourne 2018

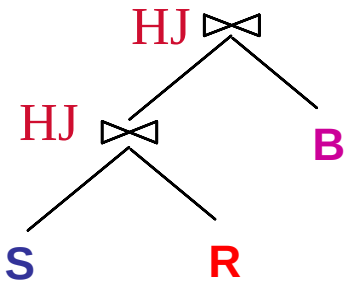
Your turn

Plan 3:



S: NPages(S) = 500, NTuplesPerPage(S)= 80
R: NPages(R) = 1000, NTuplesPerPage(R) = 100
B: NPages(B) = 10
100 R ⋈ S tuples fit on a page
All 3 relations are Heap Scan

Plan 4:



Calculating cost:

Cost (P3) = ?

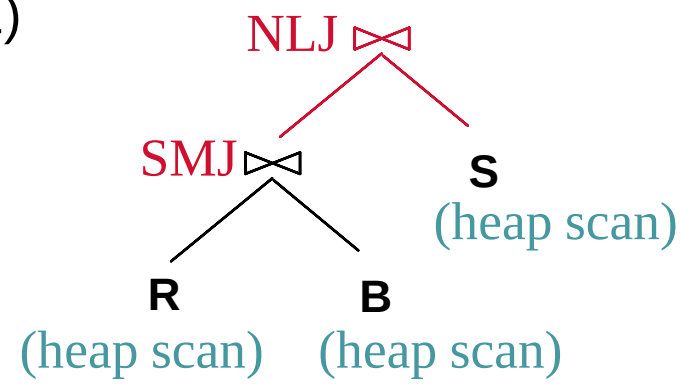
Cost (P4) = ?

© University of Melbourne 2018

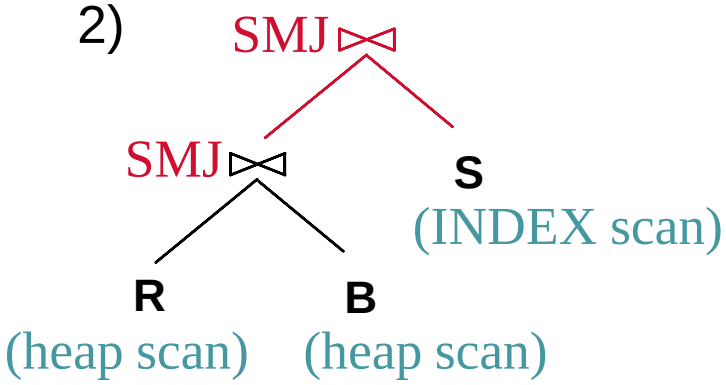
At home

SMJ : 2 passes, RxB: 10 tuples per page

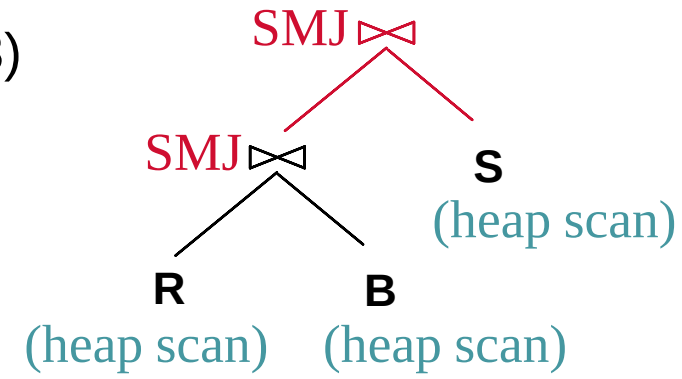
1)



2)



3)



© University of Melbourne 2018

What's examinable

- Understand plan enumeration and cost various plans

- Important for Assignment 3 as well

© University of Melbourne 2018

Next Lecture

- Normalization

© University of Melbourne 2018