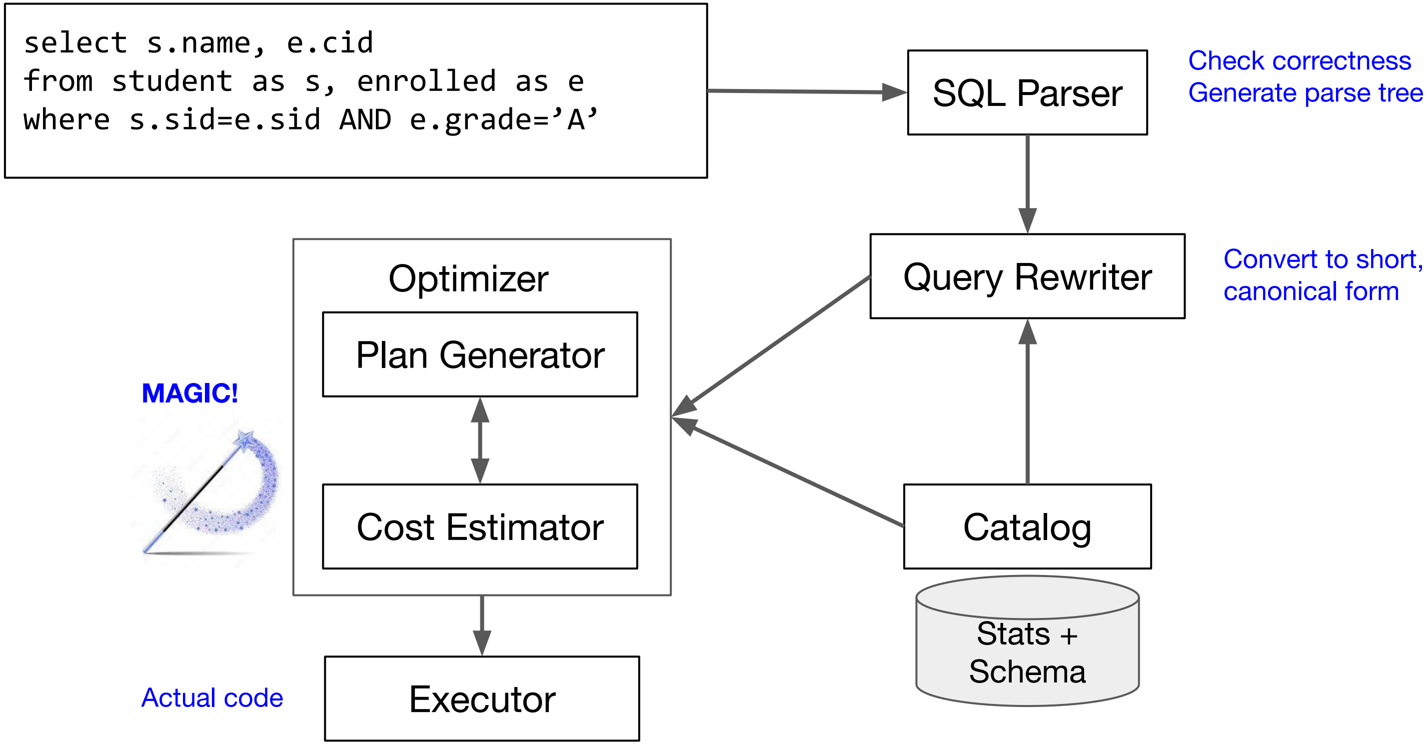
# Database Systems

Lab 8

# Today

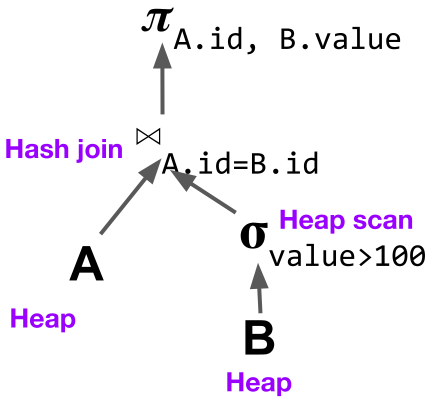
- Recap
- Query Planning and Optimization

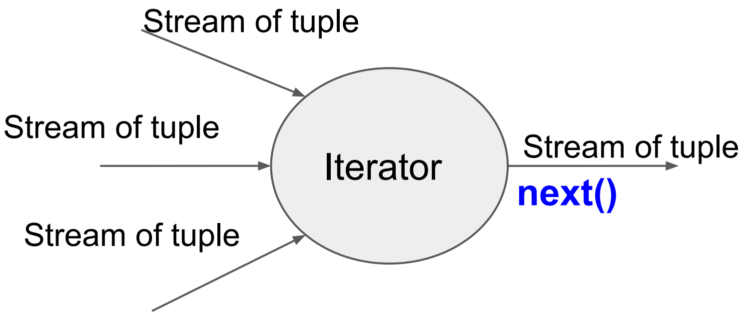
# Query Life Cycle



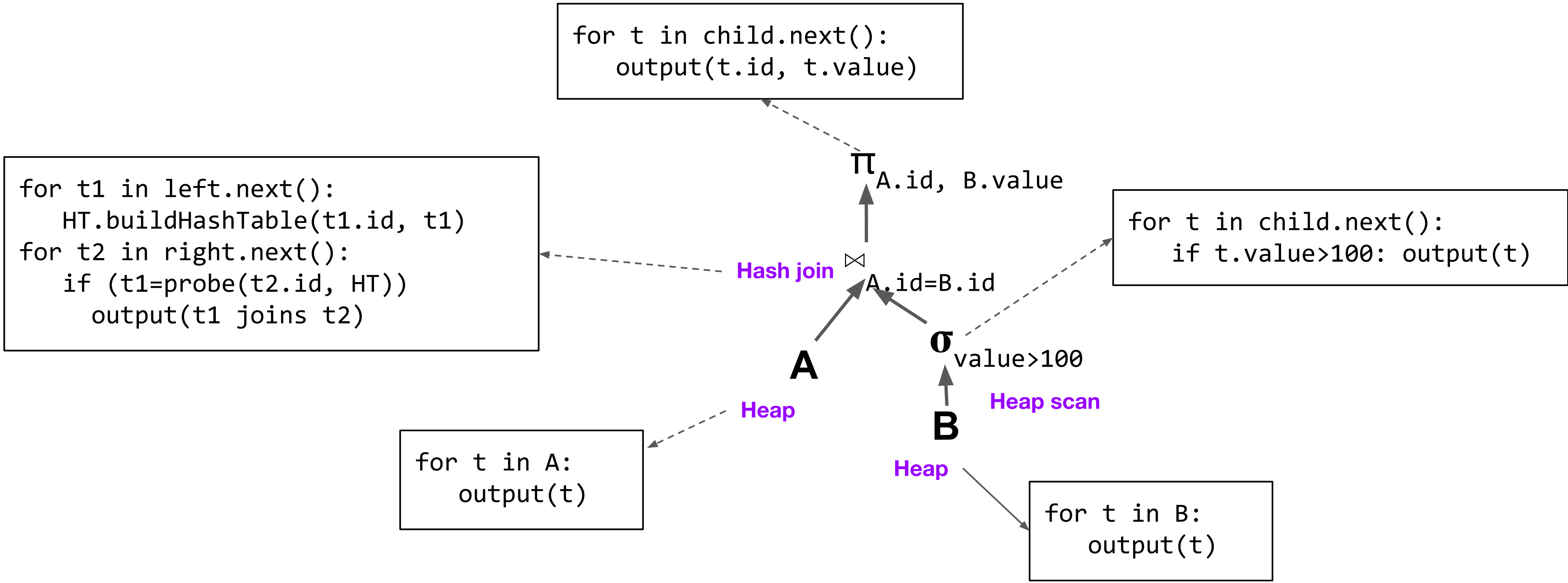
# Execution Model

- Iterator Model
  - Everything in the plan is an **Iterator**
    - Operators are iterators
    - Tables/files are iterator (HeapFile, BTreeFile)
  - An iterator:
    - Like Java Iterator, has a `next()` method
  - Upstream operator calls `next()` of its children



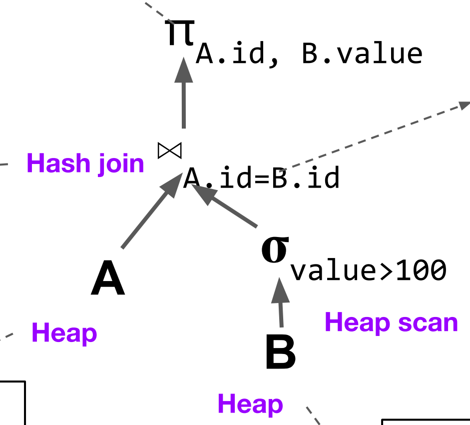
# Iteration Model



# Materialization Model

```
output=[]
for t in child.output():
   output.add(t.id, t.value)
return output
```

```
output=[]
for t1 in left.output():
   HT.buildHashTable(t1.id,t1)
for t2 in right.output():
   if (t1=probe(t2.id, HT))
     output.add(t1 joins t2)
return output
```




```
output=[]
for t in child.output():
   if t.value>100:
      output.add(t)
return output
```

```
output=[]
for t in A:
   output.add(t)
return output
```

```
output=[]
for t in B:
   output.add(t)
return output
```

# Equivalence Rules

Select and Join are commutative

Join are associative

Select distributes over Join

Project cascades

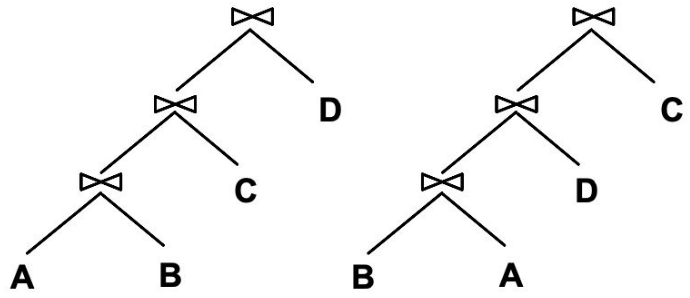
- E, E' are relational expressions
- E, E' are logically equivalent:
  - If for all database instances D, E(D) = E'(D)

# Cardinality Estimation

- The system maintains statistics for each table
  - Inside the catalog (see STATISTICS table)
  - Updated frequently
- Important statistics per relation R
  - N(R): number of tuples
  - V(A,R): number of distinct tuples for attribute A
  - $A_{Max}$, $A_{Min}$: max/min value of attribute A
  - Etc.
- Uniform and independence
- Histogram: better
- Sampling

# Selinger Algorithm

- Selinger dynamic algorithm
  - Only consider left-deep plans
  - Still **n!** Plans
- Idea: best plan to join A,B,C is the best of
  - Best plan to join {A,B}, then join C
  - Best plan to join {A,C}, then join B
  - Best plan to join {B,C}, then join A
- Principle of optimality:
  - Best overall plan consists of best sub-plans



# Search Algorithm

- Pass 1: find best plans for accessing each relation
- Pass 2: find best plans to join any 2 relations
- Pass 3: find best plans to join any 3 relations
- …
- Each pass i:
  - Join best plans at pass i-1 with one more relation

# Exercise 1

Consider relation **People(salary, name, department)**, and the following relational algebra expression:

$$\pi_{name}(\sigma_{salary>100}(\pi_{name,salary}(People)))$$

Which of the following expression is equivalent to the above?

1. $\pi_{name}(\sigma_{salary>100}(People))$
2. $\pi_{name}(\sigma_{salary>100}(\pi_{salary}(People)))$
3. $\sigma_{salary>100}(\pi_{name,salary}(People))$
4. $\sigma_{salary>100}(\pi_{name}(\pi_{name,salary}(People)))$
5. $\pi_{name}(\sigma_{salary>100}(\pi_{name}(\pi_{salary}(People))))$

# Exercise 2

Consider two relation X(A,B) and Y(B,C) with the following statistics:

- $N(X) = 200$, $V(A,X) = 100$, $V(B,X) = 20$
- $N(Y) = 1000$, $V(B,Y) = 10$, $V(C,Y) = 1000$
- $A_{min} = 0$, $A_{max} = 100$

Assume that all values of B that appears Y also appear in X.

[Q1] Estimate the number of tuples of $\boldsymbol{\sigma_{A=10}(X)}$

[Q2] Estimate the number of tuples of $\mathbf{X \bowtie Y}$

[Q3] Estimate $\mathbf{V(A, X \bowtie Y)}$

# Exercise 3

Consider the following relations:

**Sailors(sid, sname, rating, age)**

**Reserves(sid, bid, day)**

Reserves:

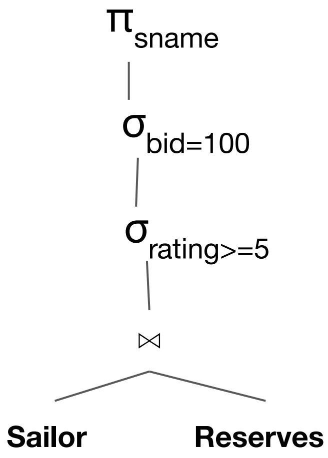
- Each tuple is 40 bytes long, 100 tuples per page, 1000 pages.
- There are 100 boats (each equally likely)

Sailors:

- Each tuple is 50 bytes long, 80 tuples per page, 500 pages.
- Assume there are 10 different ratings (1..10, each equally likely)

Buffer size = 5

[Q0] What does the query on the right return?



# Exercise 3

Consider the following relations:

**Sailors(<u>sid</u>, sname, rating, age)**

**Reserves(<u>sid, bid, day</u>)**

Reserves:

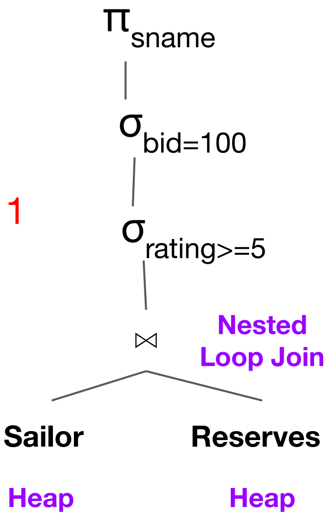
- Each tuple is 40 bytes long, 100 tuples per page, 1000 pages.
- There are 100 boats (each equally likely)

Sailors:

- Each tuple is 50 bytes long, 80 tuples per page, 500 pages.
- Assume there are 10 different ratings (1..10, each equally likely)

Buffer size = 5

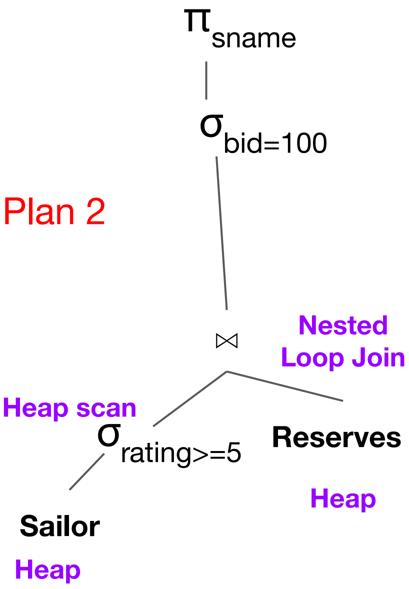
Plan 1




[Q1] What is the cost of Plan 1? Assuming iteration model

# Exercise 3

Plan 2




Consider the following relations:

**Sailors(<u>sid</u>, sname, rating, age)**

**Reserves(<u>sid, bid, day</u>)**

Reserves:

- Each tuple is 40 bytes long, 100 tuples per page, 1000 pages.
- There are 100 boats (each equally likely)

Sailors:

- Each tuple is 50 bytes long, 80 tuples per page, 500 pages.
- Assume there are 10 different ratings (1..10, each equally likely)

Buffer size = 5

[Q2] What is the cost of Plan 2? Assuming iteration model

# Exercise 3

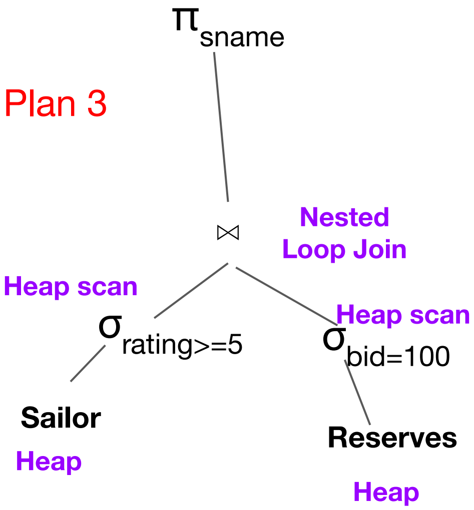
Consider the following relations:
**Sailors(sid, sname, rating, age)**
**Reserves(sid, bid, day)**
Reserves:

- Each tuple is 40 bytes long, 100 tuples per page, 1000 pages.
- There are 100 boats (each equally likely)

Sailors:

- Each tuple is 50 bytes long, 80 tuples per page, 500 pages.
- Assume there are 10 different ratings (1..10, each equally likely)

Buffer size = 5




[Q3] What is the cost of Plan 3? Assuming iteration model

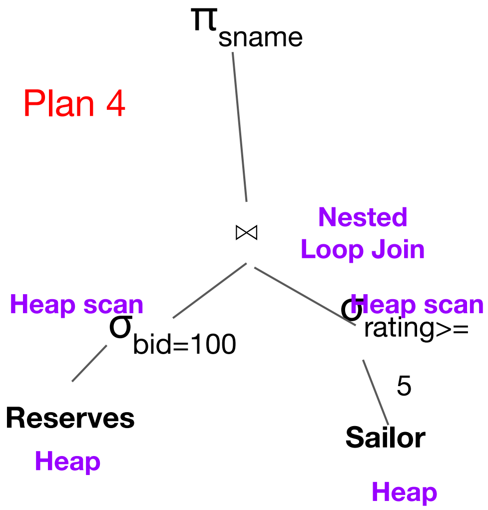
# Exercise 3

Consider the following relations:
**Sailors(sid, sname, rating, age)**
**Reserves(sid, bid, day)**
Reserves:
- Each tuple is 40 bytes long, 100 tuples per page, 1000 pages.
- There are 100 boats (each equally likely)

Sailors:
- Each tuple is 50 bytes long, 80 tuples per page, 500 pages.
- Assume there are 10 different ratings (1..10, each equally likely)

Buffer size = 5

Plan 4




[Q4] What is the cost of Plan 4? Assuming iteration model

# Exercise 3

Consider the following relations:
**Sailors(<u>sid</u>, sname, rating, age)**
**Reserves(<u>sid, bid, day</u>)**
Reserves:

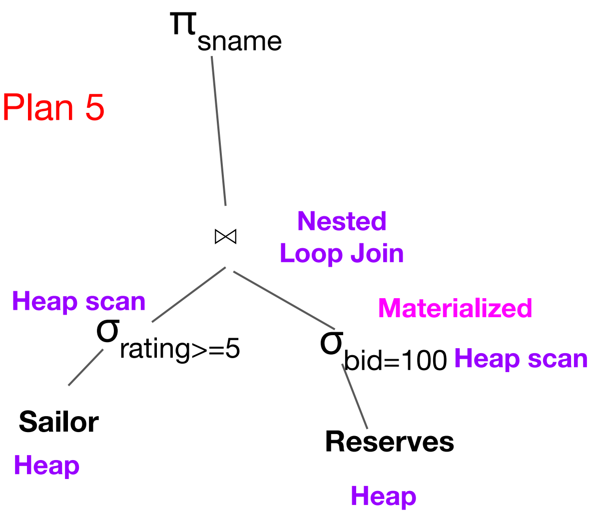
- Each tuple is 40 bytes long, 100 tuples per page, 1000 pages.
- There are 100 boats (each equally likely)

Sailors:

- Each tuple is 50 bytes long, 80 tuples per page, 500 pages.
- Assume there are 10 different ratings (1..10, each equally likely)

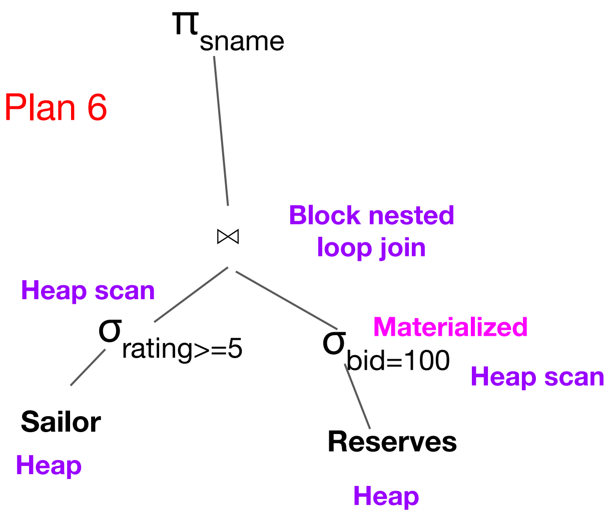
Buffer size = 5

Plan 5




[Q5] What is the cost of Plan 5?

# Exercise 3




Consider the following relations:
**Sailors(<u>sid</u>, sname, rating, age)**
**Reserves(<u>sid, bid, day</u>)**
Reserves:

- Each tuple is 40 bytes long, 100 tuples per page, 1000 pages.
- There are 100 boats (each equally likely)

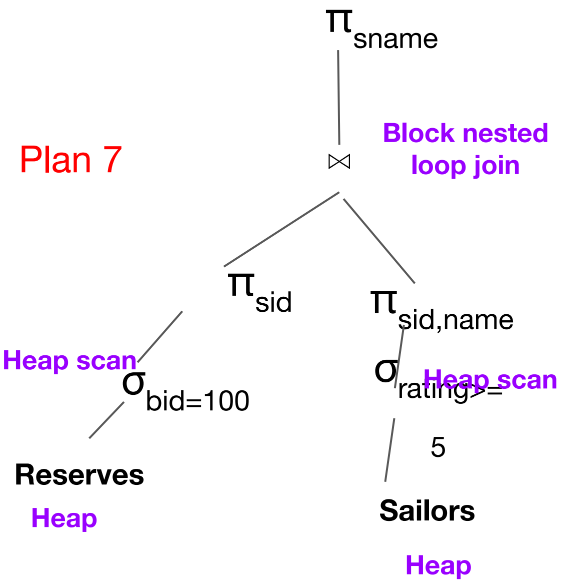
Sailors:

- Each tuple is 50 bytes long, 80 tuples per page, 500 pages.
- Assume there are 10 different ratings (1..10, each equally likely)

Buffer size = 5

[Q6] What is the cost of Plan 6?

# Exercise 3




Consider the following relations:
**Sailors(sid, sname, rating, age)**
**Reserves(sid, bid, day)**
Reserves:
- Each tuple is 40 bytes long, 100 tuples per page, 1000 pages.
- There are 100 boats (each equally likely)

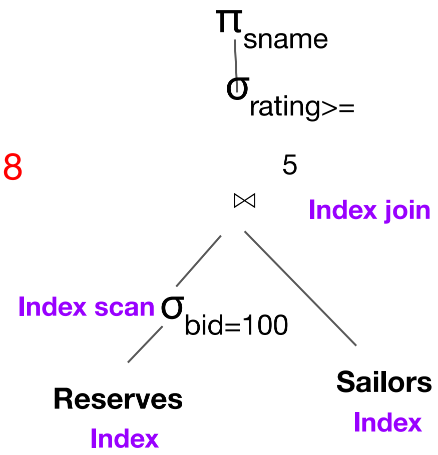
Sailors:
- Each tuple is 50 bytes long, 80 tuples per page, 500 pages.
- Assume there are 10 different ratings (1..10, each equally likely)

Buffer size = 5

[Q7] What is the cost of Plan 7? Assuming sid's size is 4 bytes

# Exercise 3

Plan 8




Consider the following relations:
**Sailors(<u>sid</u>, sname, rating, age)**
**Reserves(<u>sid, bid, day</u>)**
Reserves:

- Each tuple is 40 bytes long, 100 tuples per page, 1000 pages.
- There are 100 boats (each equally likely)

Sailors:

- Each tuple is 50 bytes long, 80 tuples per page, 500 pages.
- Assume there are 10 different ratings (1..10, each equally likely)

Buffer size = 5

**Index join X ⋈ Y**:

For each tuple in X
  Scan for match in Y using Y's index

[Q8] Suppose Reserves has a **clustered B+-tree index on bid**, Sailor has an **unclustered B+-tree index on sid**. Both indices are in memory. What is the cost for Plan 8?