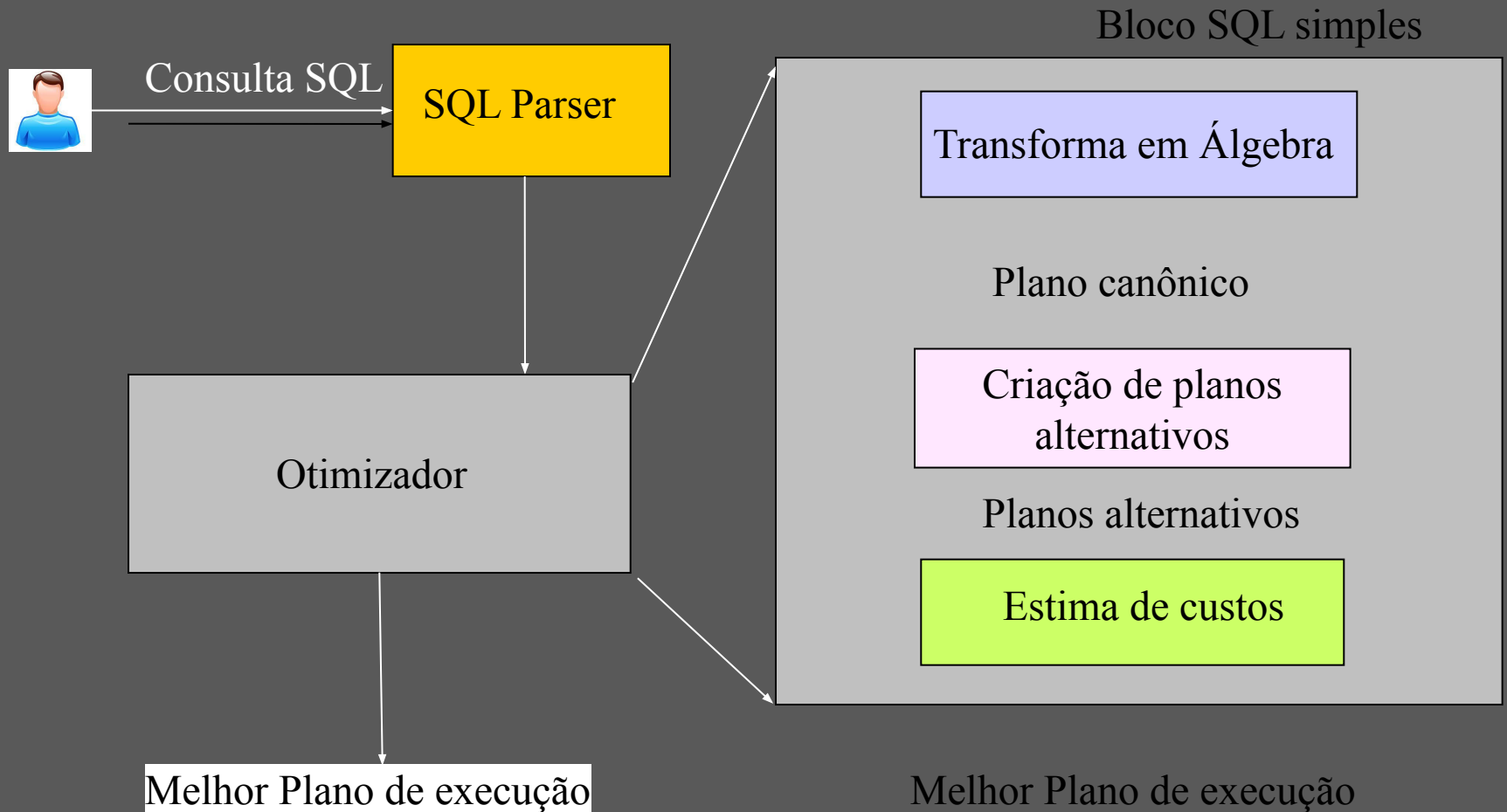


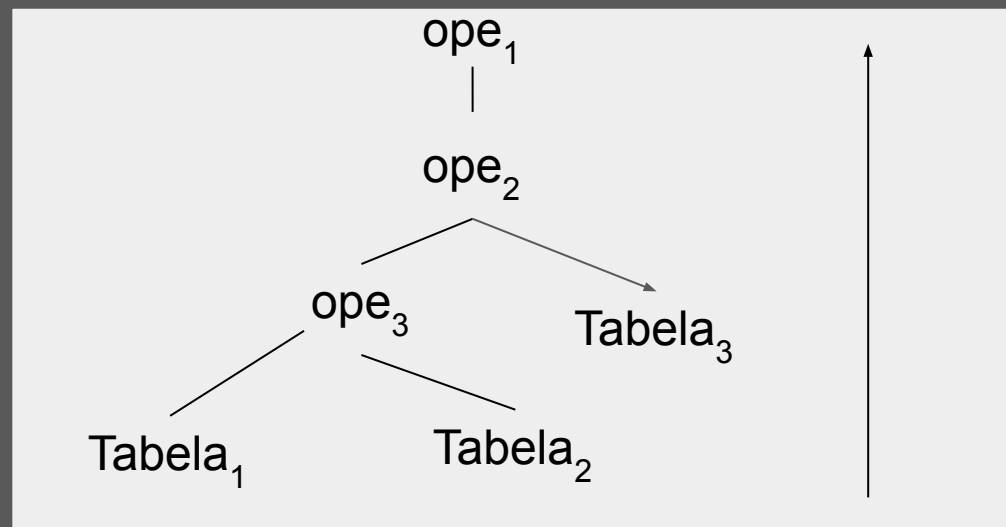
Planos de Consulta

Esquema Geral do Otimizador



Plano de Consulta

- Estratégia que o banco utiliza para executar uma consulta
- Normalmente, vários planos são proposto e um deles é escolhido
- O plano é um pseudocódigo em forma de árvore e álgebra relacional

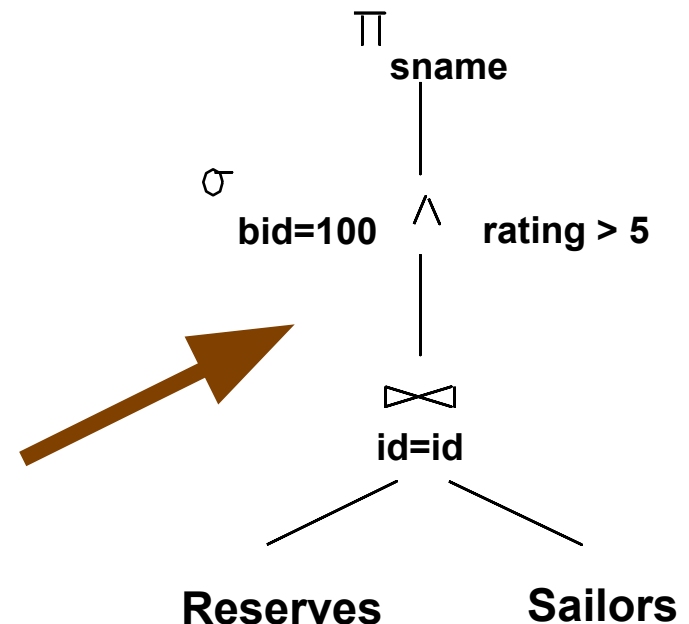


Otimização de consultas

- A consulta é convertida em álgebra relacional
- Álgebra relacional é convertida em uma árvore
- Cada operador pode ser alterado
- Operadores podem ser aplicados em diferentes ordens

```
SELECT S.sname  
FROM Reserves R, Sailors S  
WHERE R.id=S.id AND  
      R.id=100 AND S.rating>5
```

$\pi_{(sname)} \sigma_{(id=100 \wedge rating > 5)} (R \bowtie S)$



Plano de Consulta

- Estratégias para processar consulta
 - Qual tabela processar primeiro
 - Mais ou menos volumosa?
 - Utilizar índice
 - Ordenar tabela
 - Tratamento junção
 - Melhor decomposição
 - Quantos planos propor
 - Como escolher o melhor plano
 - Algoritmo de tratamento dos operadores

Esquema de exemplo

Sailors (id: integer, *sname: string*, *rating: integer*, *age: real*)

Reserves (id: integer, bid: integer, day: dates, *rname: string*)

Reserves:

- Cada tupla com 40 bytes, 100 tuplas por página , 1000 páginas.
- Assumir que existe 100 barcos diferentes, **distribuídos uniformemente**;

Sailors:

- Cada tupla com 50 bytes, 80 tuplas por página, 500 páginas;
- Assumir que existem 10 diferentes ratings, **distribuídos uniformemente**;

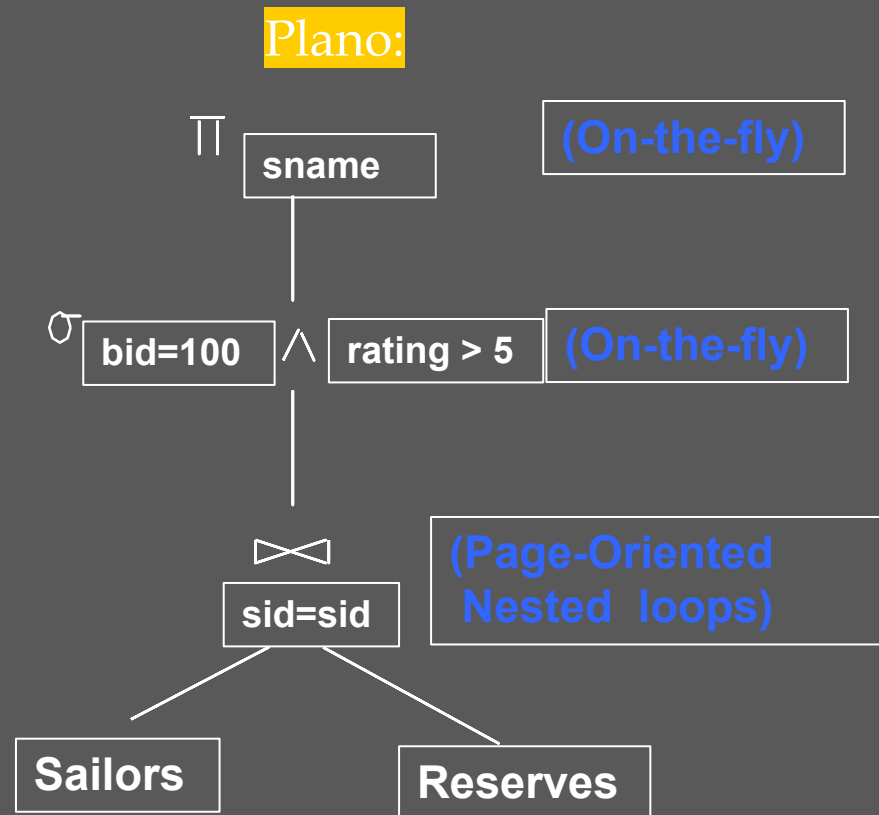
- Assumir que temos 5 páginas no buffer

Exemplo

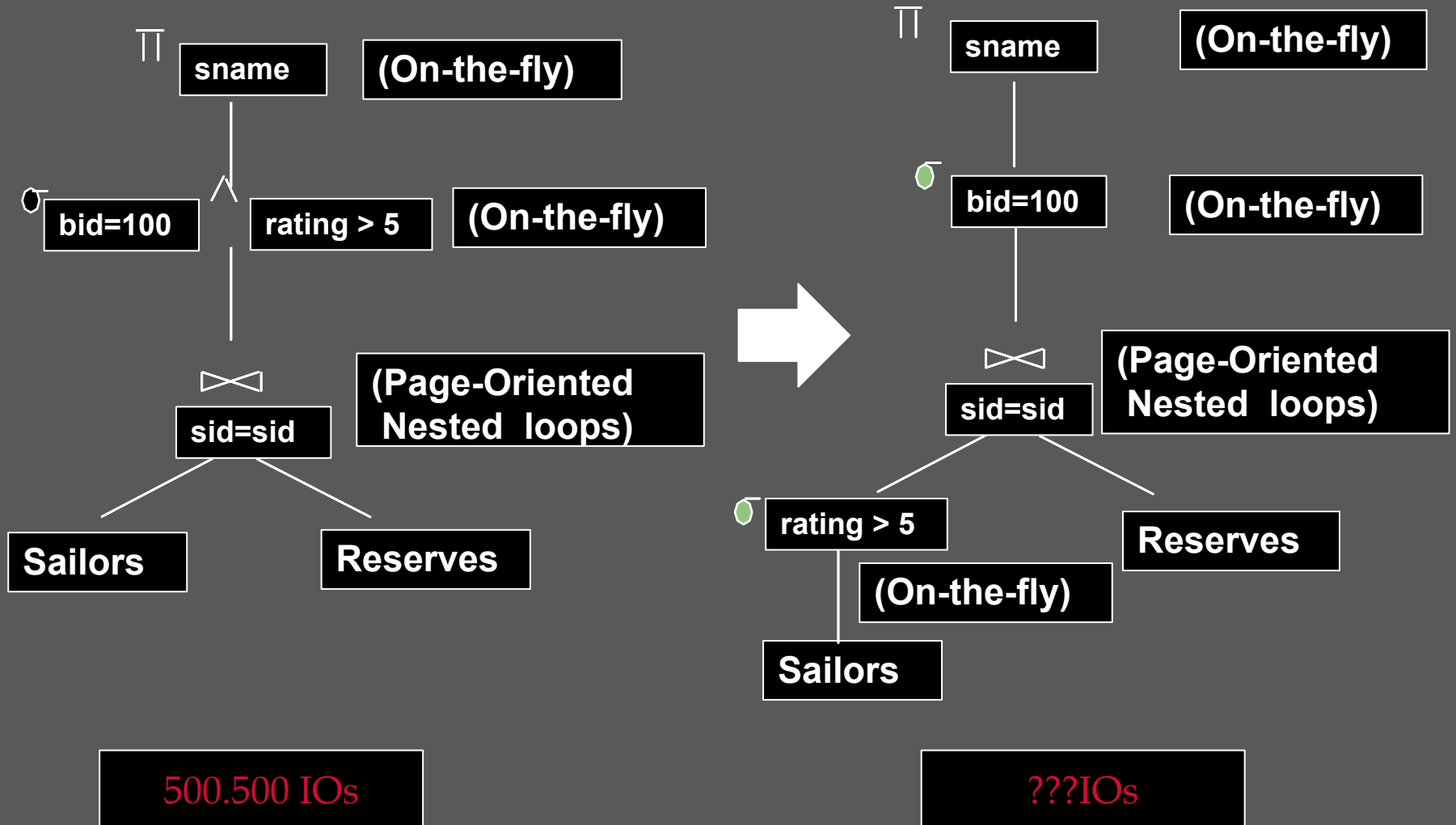
```
SELECT  S.sname
FROM    Reserves R, Sailors S
WHERE   R.sid=S.sid AND
        R.bid=100 AND S.rating>5
```

- R=1000 pg
 - R=100 Tuplas/pg
- S=500 pg
 - S=80 Tuplas/pg
- Custo: $(Pg\ S) + (Pg\ S) * (Pg\ R)$

Custo:

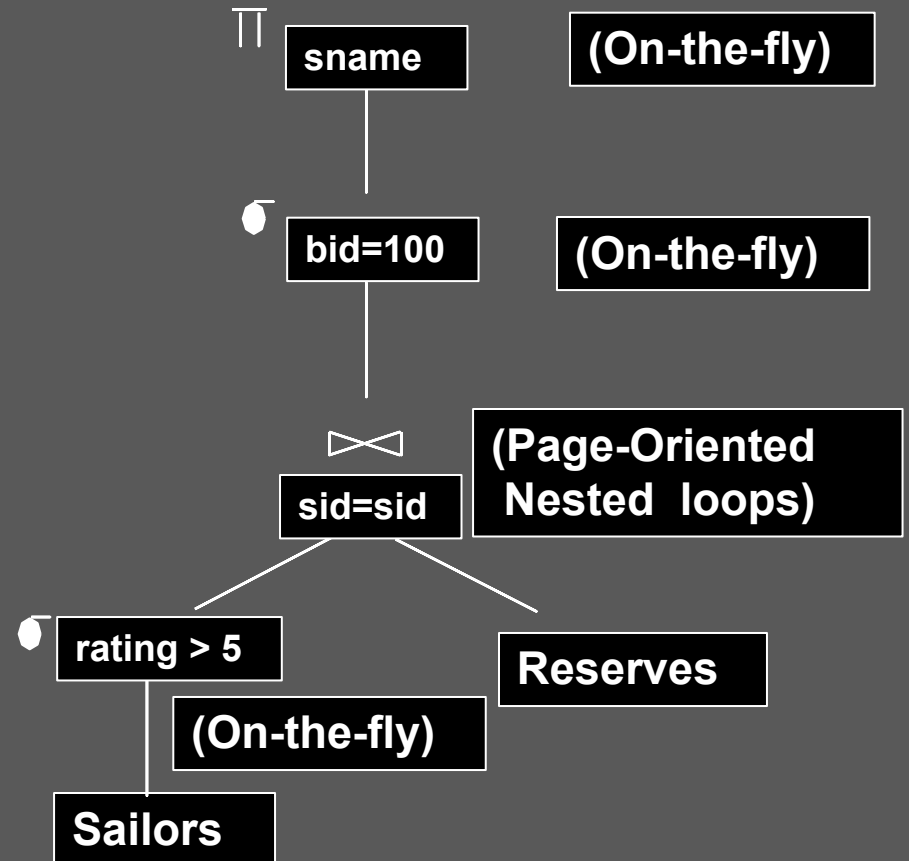


Planos alternativos

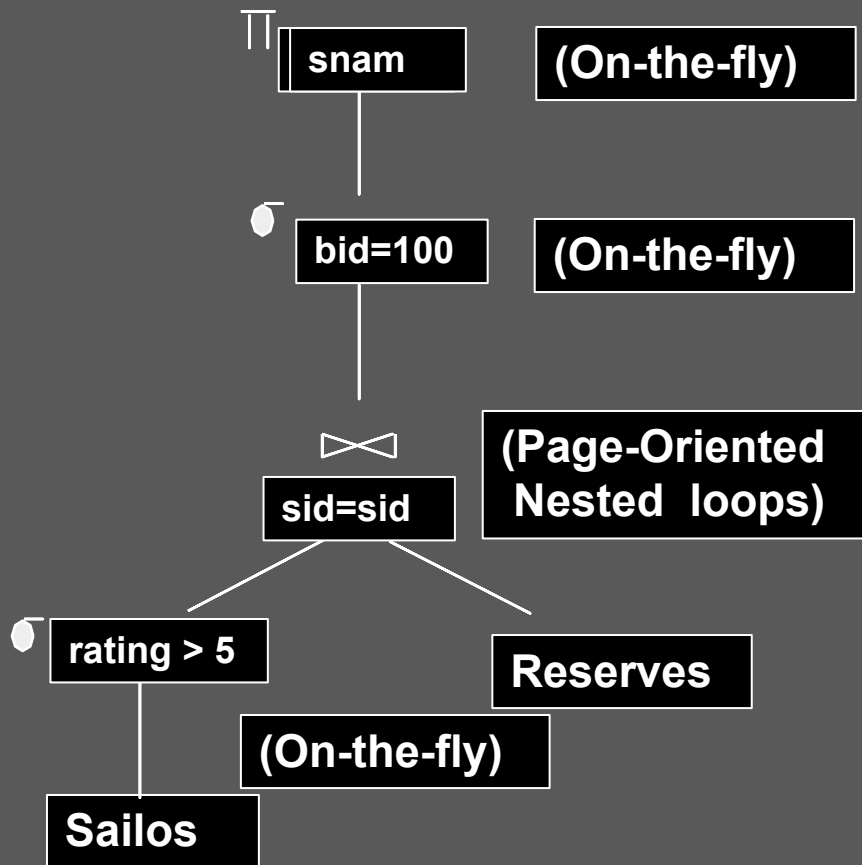


Planos alternativos

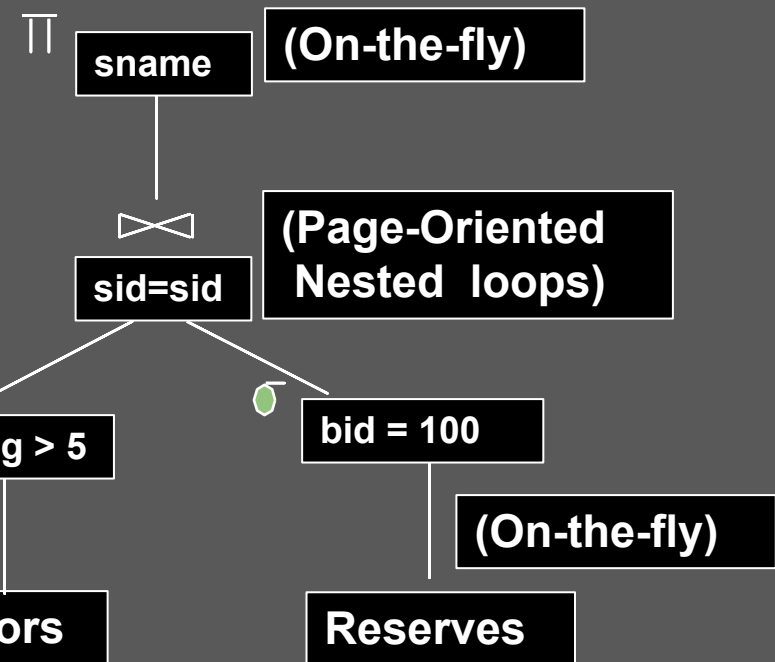
- R=1000 pg
 - R=100 Tuplas/pg
- S=500 pg
 - S=80 Tuplas/pg
- Custo:



Planos alternativos



250,500 IOs

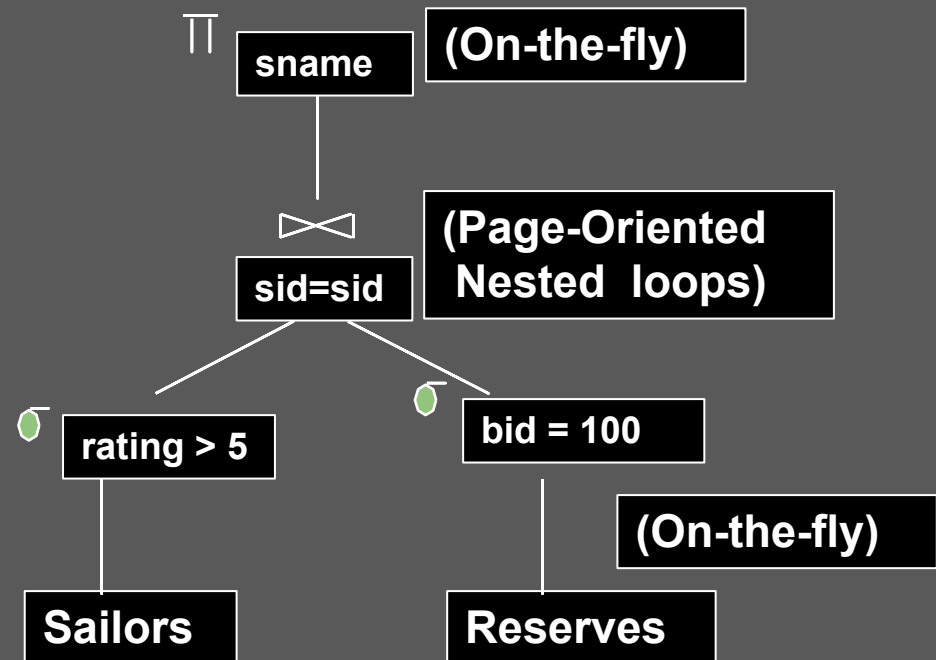


???

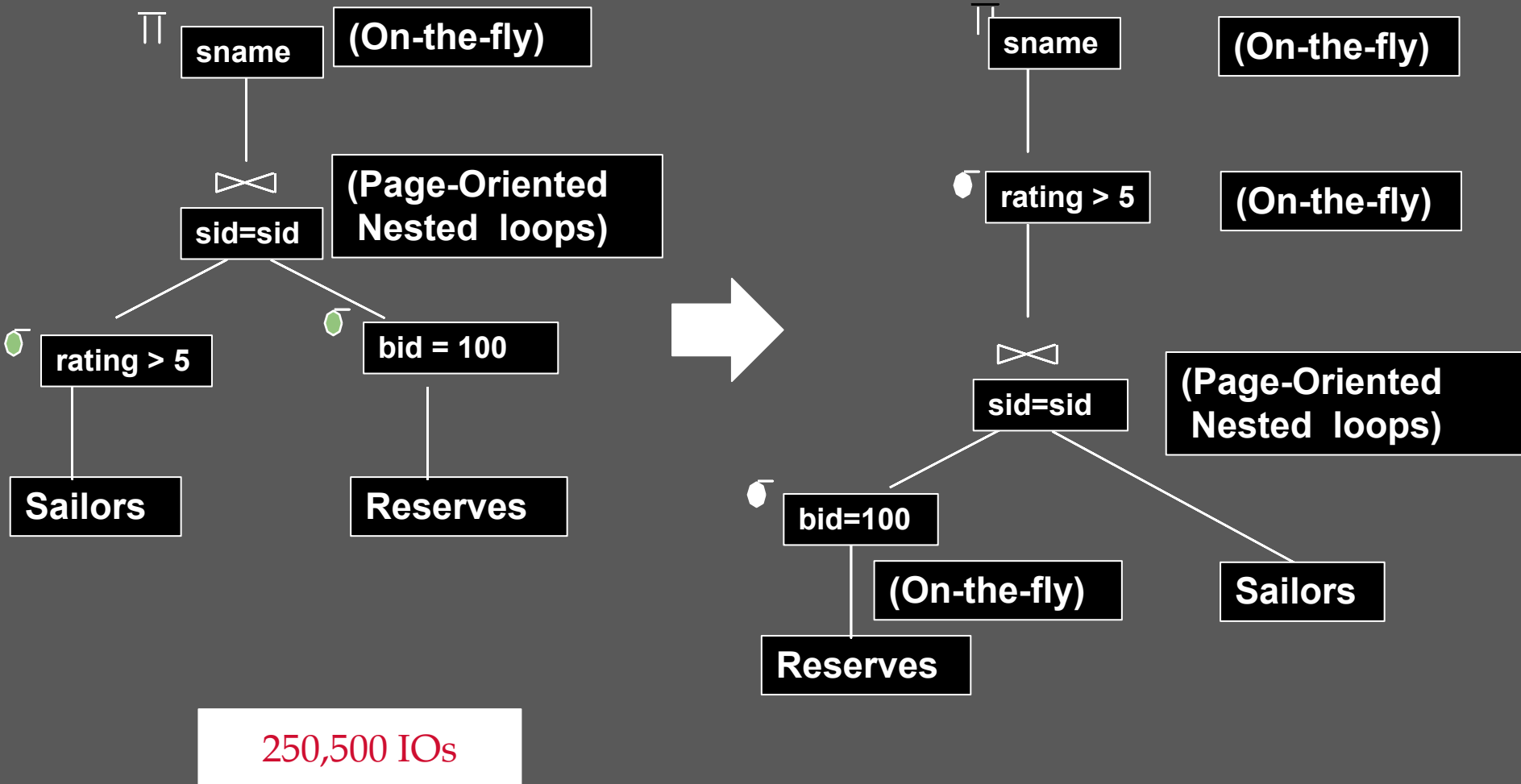
Planos alternativos

- R=1000 pg
 - R=100 Tuplas/pg
- S=500 pg
 - S=80 Tuplas/pg
- Custo:

Buffer: 5

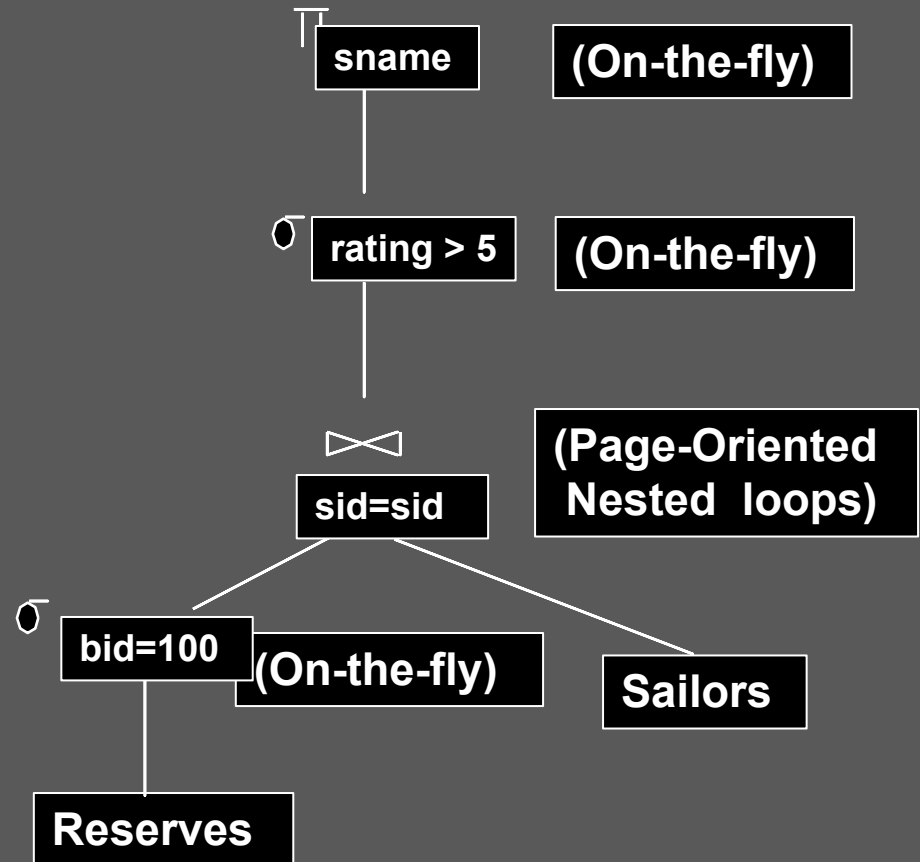


Planos alternativos

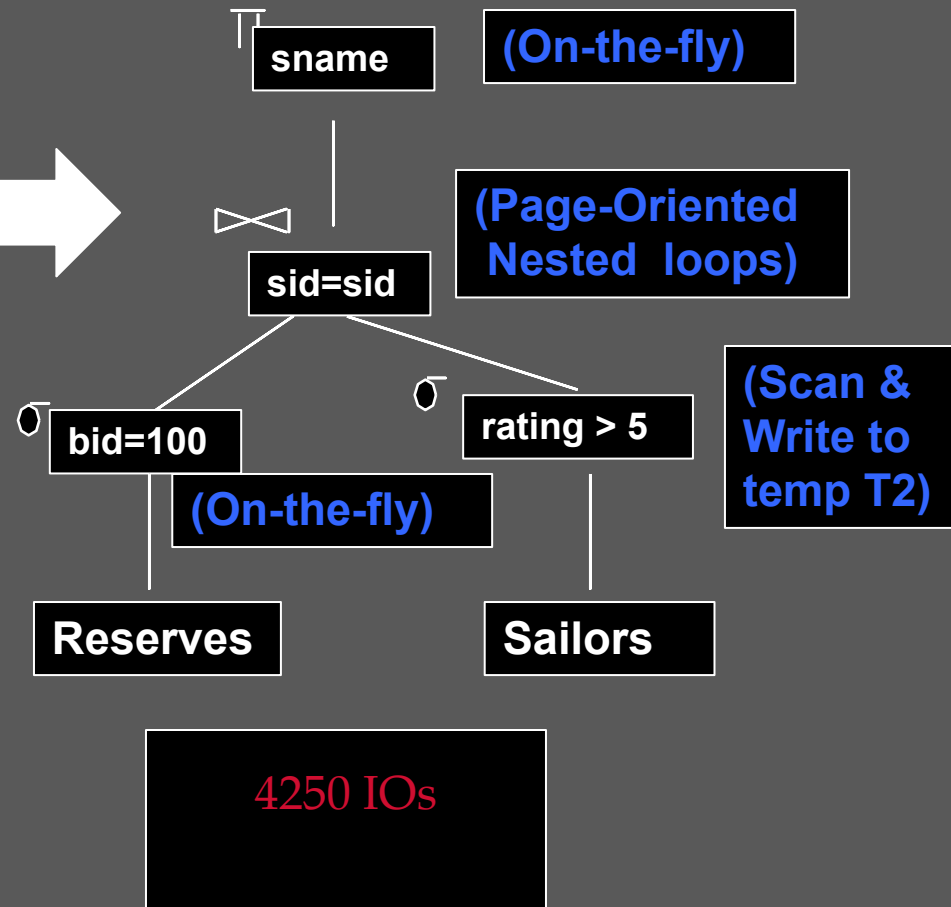
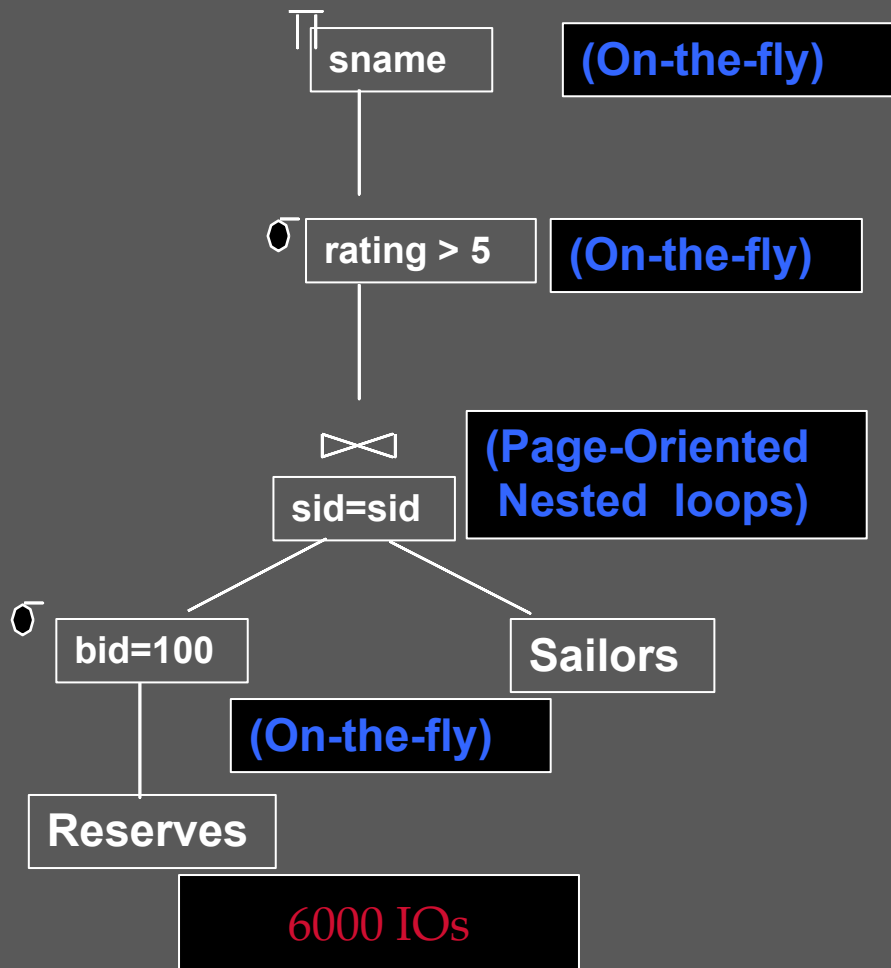


Planos alternativos

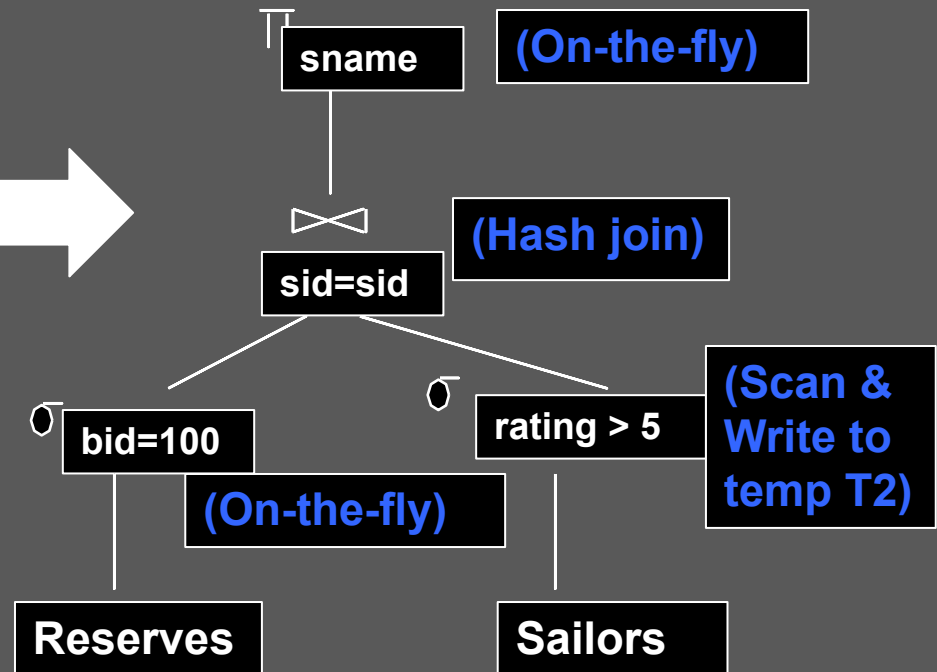
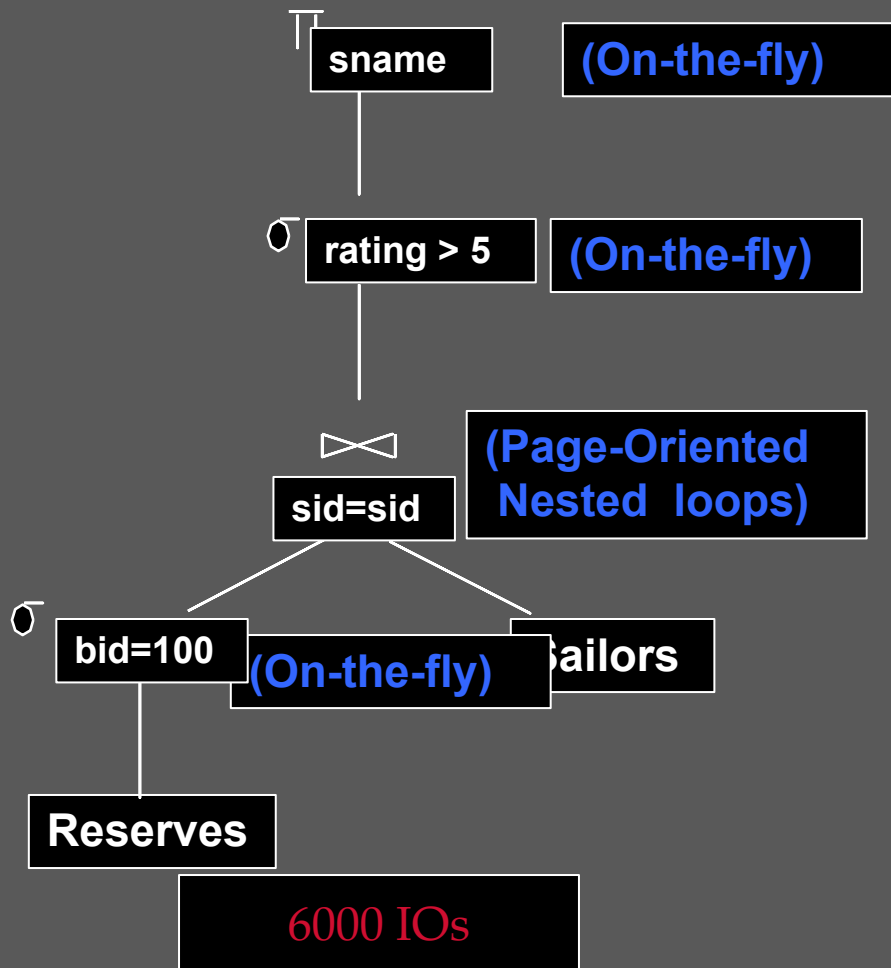
- R=1000 pg
 - R=100 Tuplas/pg
- S=500 pg
 - S=80 Tuplas/pg
- Custo:



Planos alternativos



Planos alternativos



$$1000 + 500 + 250 + 3 \cdot (10 + 250) = 1750 + 780 = 2530$$

Alterando o exemplo

Sailors (id: integer, *sname: string*, *rating: integer*, *age: real*)

Reserves (id: integer, bid: integer, day: dates, *rname: string*)

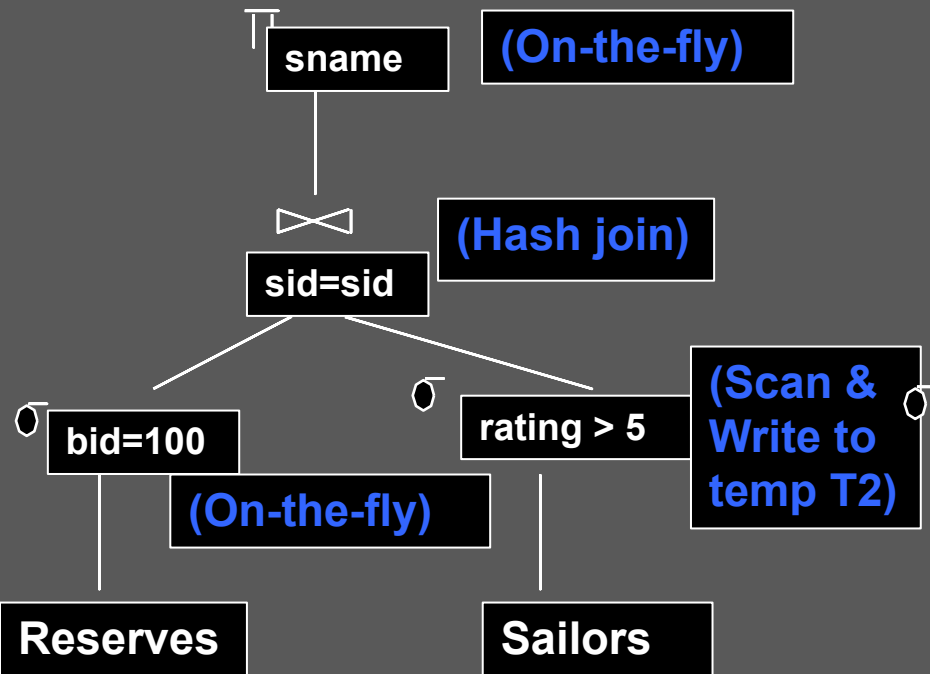
Reserves:

- Cada tupla com 40 bytes, 100 tuplas por página , 1000 páginas.
- Assumir que existe 100 barcos diferentes, **distribuídos uniformemente**;
- create index idx on Reserves(bid) USING btree WITH (FILLFACTOR=50);
- Supor que o índice está em memória;

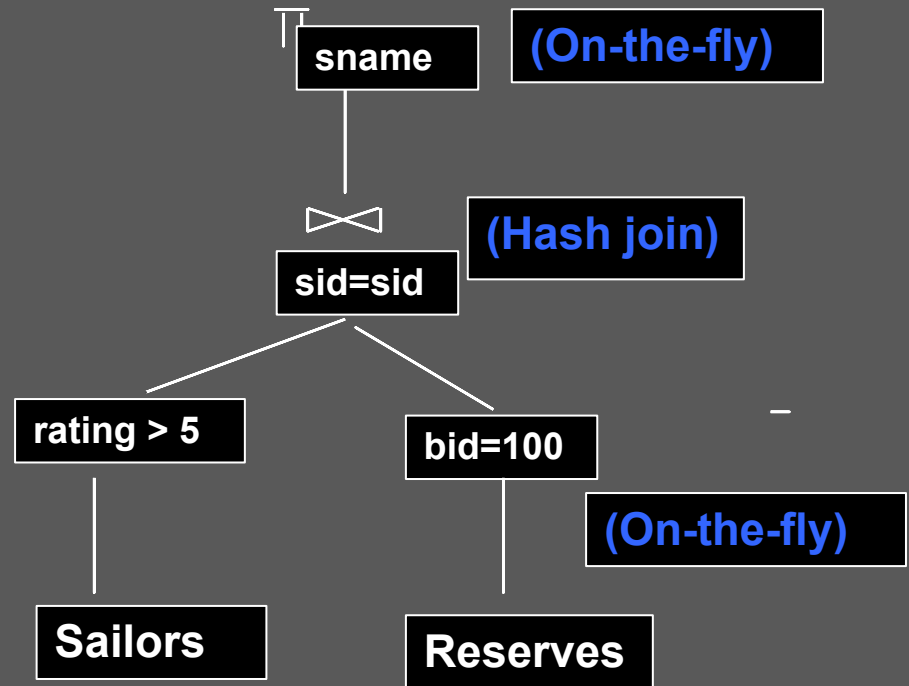
Sailors:

- Cada tupla com 50 bytes, 80 tuplas por página, 500 páginas;
- Assumir que existem 10 diferentes ratings, **distribuídos uniformemente**;

Planos alternativos



$$1000 + 500 + 250 + 3 \cdot (10 + 250) = 1750 + 780 = 2530$$



Alterando o exemplo

Sailors (*id*: integer, *sname*: string, rating: integer, *age*: real)

Reserves (*id*: integer, *bid*: integer, *day*: dates, *rname*: string)

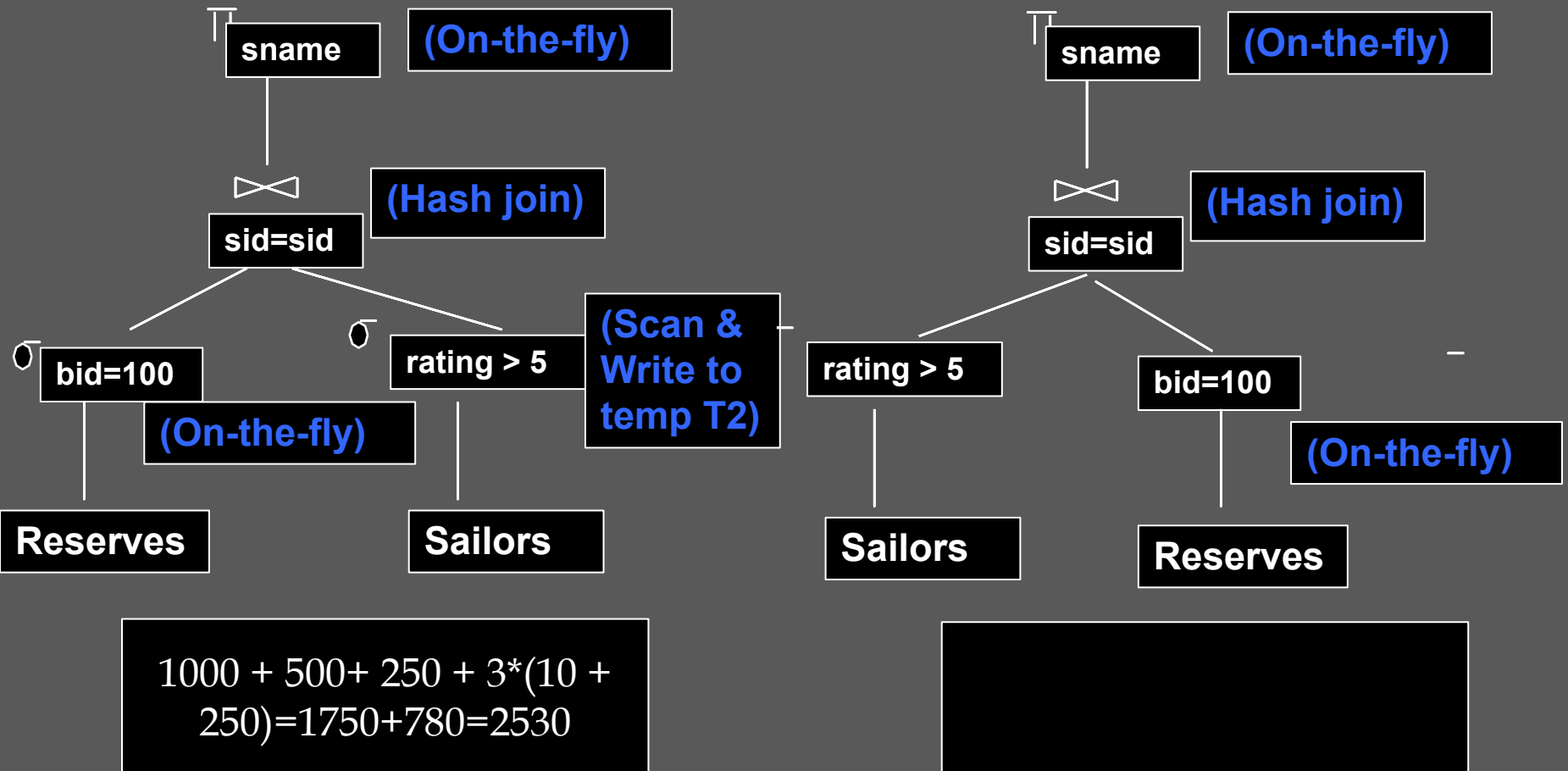
Reserves:

- Cada tupla com 40 bytes, 100 tuplas por página , 1000 páginas.
- Assumir que existe 100 barcos diferentes, **distribuídos uniformemente**;
- create index idx on Reserves(bid) USING btree WITH (FILLFACTOR=50);
- cluster Reservas using idx;

Sailors:

- Cada tupla com 50 bytes, 80 tuplas por página, 500 páginas;
- Assumir que existem 10 diferentes ratings, **distribuídos uniformemente**;

Planos alternativos



O que é preciso para melhorar uma consulta ?

1- Plano de consulta:

- Baseado na equivalência relacional

2- Estimativa de custo:

- Fórmulas
- Estimativa de tamanho, baseado no catálogo e na Seletividade

3- Algoritmo de busca:

- Busca no plano de consulta com base nos custos

Plano de Consulta

- Equivalências na álgebra relacional

- Seleções

- $\sigma_{c_1 \wedge c_2 \wedge \dots \wedge c_n}(R) \equiv \sigma_{c_1}(\sigma_{c_2}(\dots(\sigma_{c_n}(R)\dots)))$
- $\sigma_{c_1}(\sigma_{c_2}(R)) \equiv \sigma_{c_2}(\sigma_{c_1}(R))$

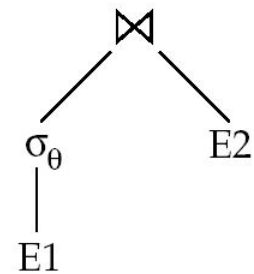
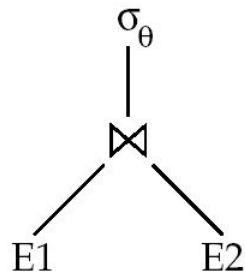
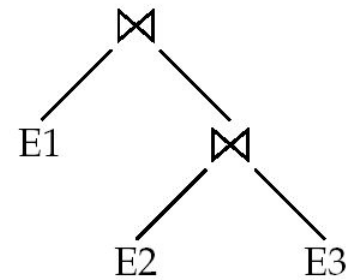
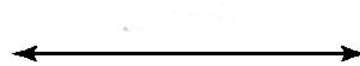
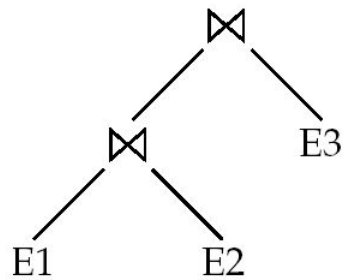
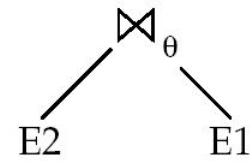
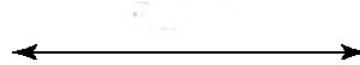
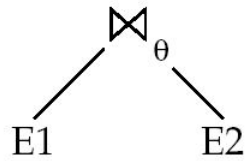
- Projeção

- $\pi_{a_1}(R) \equiv \pi_{a_1}(\pi_{a_2}(\dots(\pi_{a_n}(R)\dots)))$

- Produtos cartesianos e junções

- $R \times S \equiv S \times R$
- $R \bowtie S \equiv S \bowtie R$
- $R \times (S \times T) \equiv (R \times S) \times T$
- $R \bowtie (S \bowtie T) \equiv (R \bowtie S) \bowtie T$

Plano de Consulta



O Que Faz Um Plano Ser Bom

- O planejador escolhe um plano (entre vários) baseado no seu custo estimado
- Assume: o I/O domina o custo de uma consulta (assim, seleciona o plano que requer menos I/O)
 - I/O randômico é mais caro que I/O sequencial nos hardwares modernos
- O I/O é estimado tentando prever o tamanho dos resultados intermediários, usando as estatísticas do SGBD
 - Isso é uma ciência imperfeita (para não dizer algo pior)

Princípios Gerais de Otimização

- O custo de um nó é em função de sua entrada: o número de linhas produzidas pelos nós filhos e a distribuição de seus valores:
 - Reorganizar os nós pode mudar todo o custo
 - Uma escolha pobre perto das folhas pode levar a um **desastre**
 - Aplicar os predicados mais cedo para reduzir o número de tuplas em resultados intermediários
- Mantenha em mente a ordenação: uma entrada ordenada pode ter planos mais baratos
- Planejar os joins de forma eficiente é fundamental

Algoritmo Planejador (PG)

- Conceitualmente, três fases:
 - Enumerar todos os planos disponíveis
 - Avaliar o custo de cada plano
 - Escolher o mais barato
- Naturalmente, executar essas fases não seria muito eficiente
- Ideia básica: encontre bons planos para uma consulta simplificada com n joins ($n=2$)
 - Encontre bons planos para $n+1$ joins, junte os planos (Repita o passo anterior)

Cálculo de Custos de Planos de Execução

Exercício 1 :
Calcule o custo deste plano

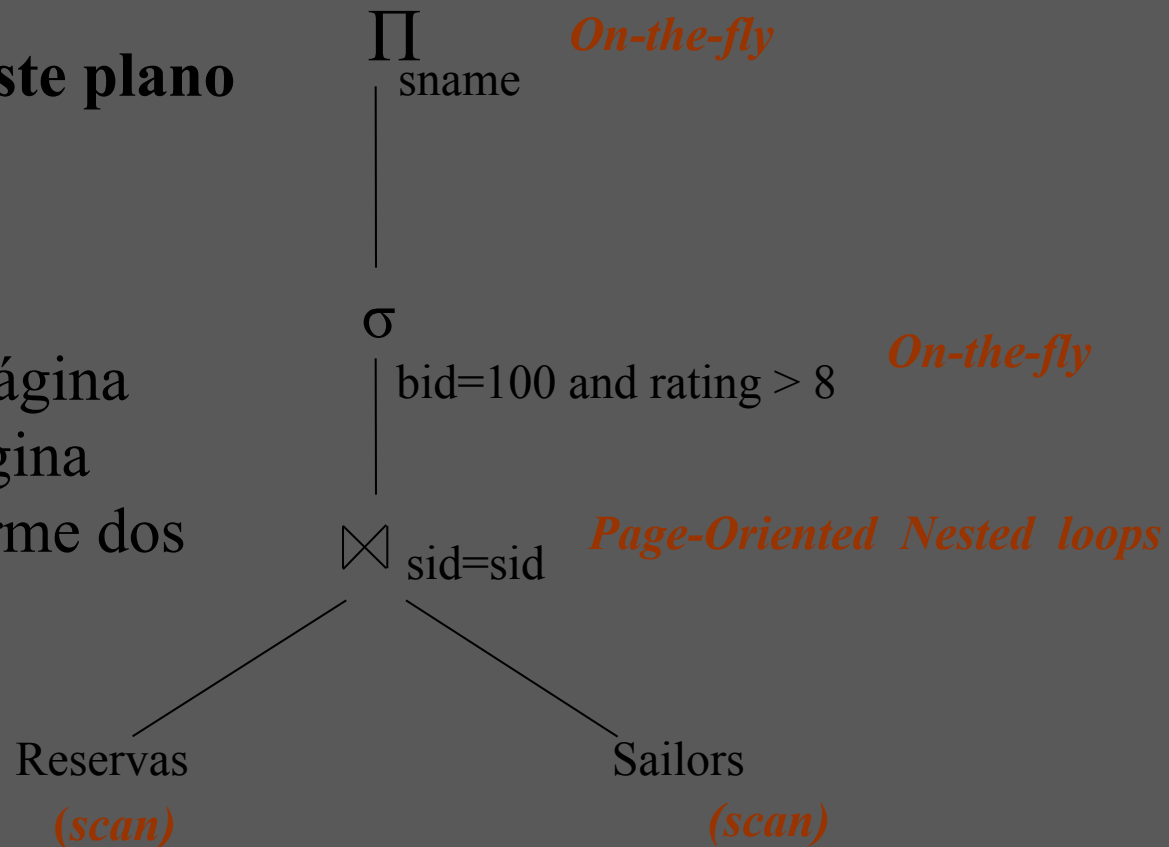
R : 1000 páginas

S : 500 páginas

R: 100 tuplas por página

S: 80 tuplas por página

Distribuição Uniforme dos dados



Cálculo de Custos de Planos de Execução

Exercício 2:

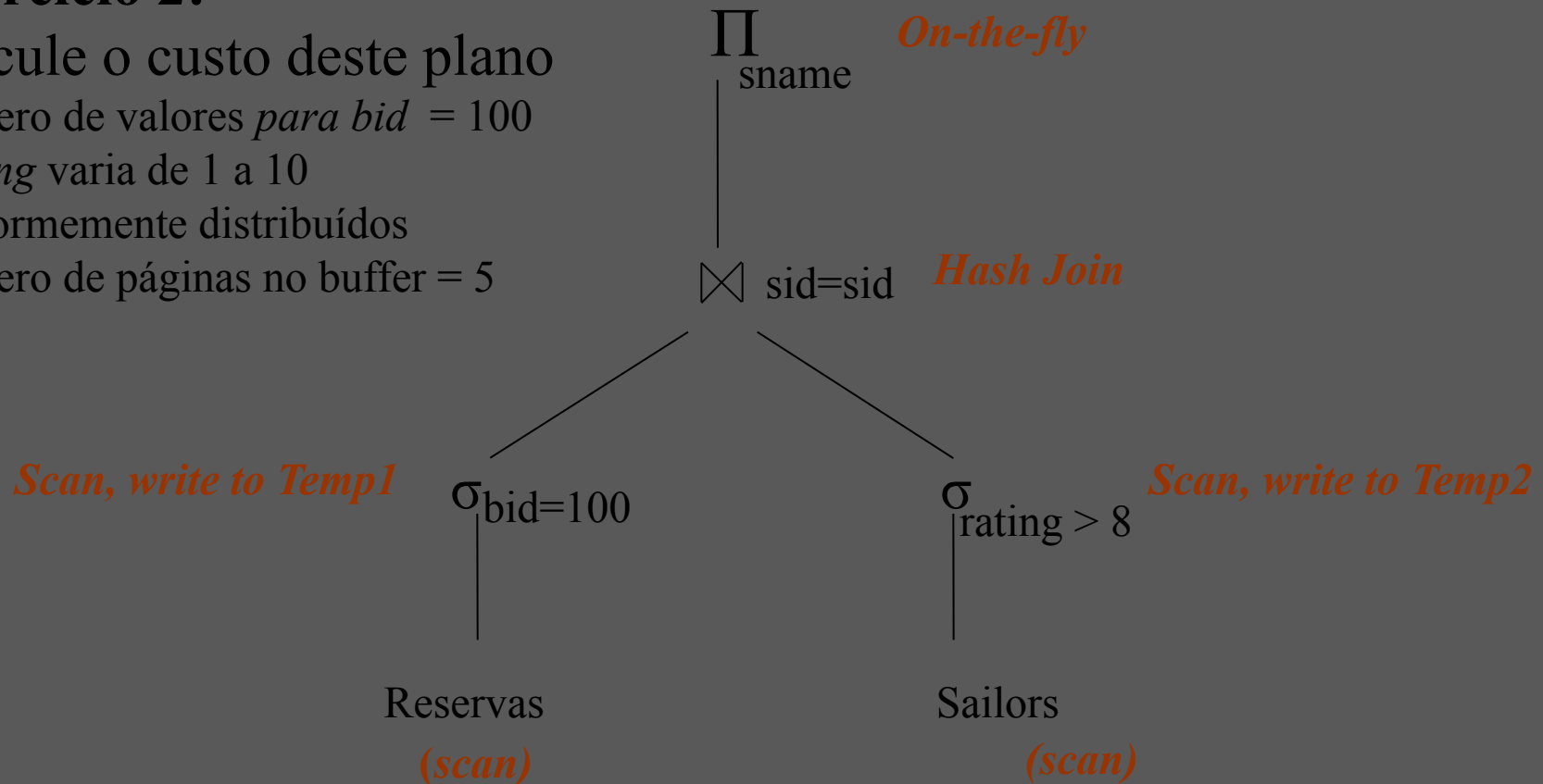
Calcule o custo deste plano

Número de valores *para bid* = 100

Rating varia de 1 a 10

Uniformemente distribuídos

Número de páginas no buffer = 5



Cálculo de Custos de Planos de Execução

Exercício 3 :

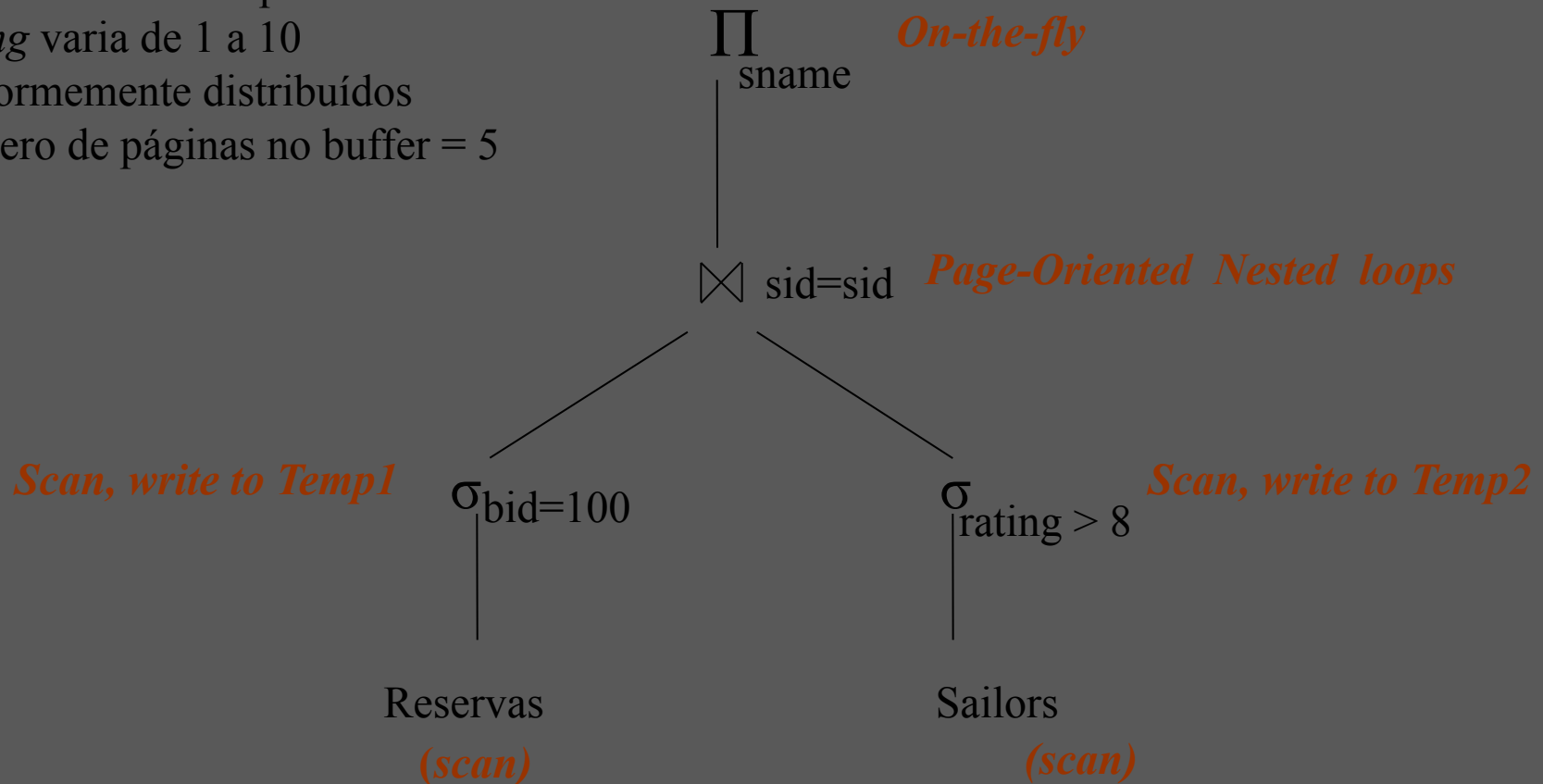
Calcule o custo deste plano

Número de valores para $bid = 100$

$Rating$ varia de 1 a 10

Uniformemente distribuídos

Número de páginas no buffer = 5



Conclusões

- Consultas são as operações mais caras do SGBD
- Encontrar a melhor forma de executar uma consulta é um problema intratável em computação
- O dicionário de dados tem um papel importante no processamento de consultas
- Índices são essenciais

Exemplo

```
CREATE TABLE a AS SELECT generate_series(1,1000000) AS i ORDER BY  
random();
```

```
CREATE INDEX a_x ON a (i);
```

```
CREATE TABLE b AS SELECT generate_series(1,1000000) AS i ORDER BY  
random();
```

```
CREATE INDEX b_x ON b (i);
```

```
DELETE FROM b WHERE i = 2;
```

```
DELETE FROM b WHERE i = 3;
```

```
ANALYZE;
```

Query

O que está em a mas nem está em b;

Resposta: 2 e 3

```
SELECT a.i FROM a EXCEPT SELECT b.i FROM b;
```

```
SELECT a.i FROM a WHERE NOT EXISTS (SELECT b.i FROM b  
WHERE a.i = b.i);
```

```
select * from a left join b on a.i=b.i where b.i is null;
```