COMP231

Join Algorithm

Prepared by Raymond Wong

Presented by Raymond Wong

Example

- Consider the following SQL query.
  - E.g., student (sid, sname, age)
    take (sid, cid)

```
select sname
from student S, take T
where S.sid = T.sid
```

Student: 500 pages, 80 tuples/page, 50 bytes/tuple
Take: 1000 pages, 100 tuples/page, 40 bytes/tuple

- 200 courses in table Take
- 1, 2, …, 40 in attribute Age of table Student

Example

- Assuming the following table sizes:
  - Student: 500 pages, 80 tuples/page, 50 bytes/tuple
  - Take: 1000 pages, 100 tuples/page, 40 bytes/tuple
- Assume that there are 200 courses in table Take

Outline

- Simple-Nested Loop Join
- Block-Nested Loop Join
- Sort-Merge Join
- Index-Nested Loop Join
- Hash Join

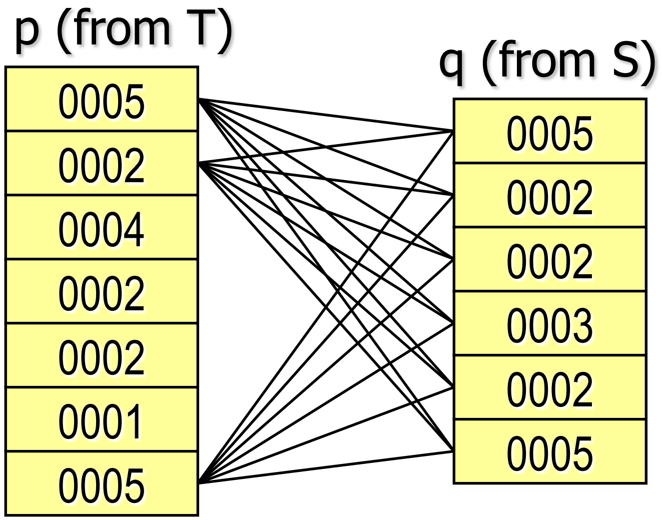
Student: 500 pages, 80 tuples/page, 50 bytes/tuple
Take: 1000 pages, 100 tuples/page, 40 bytes/tuple

- 200 courses in table Take
- 1, 2, …, 40 in attribute Age of table Student

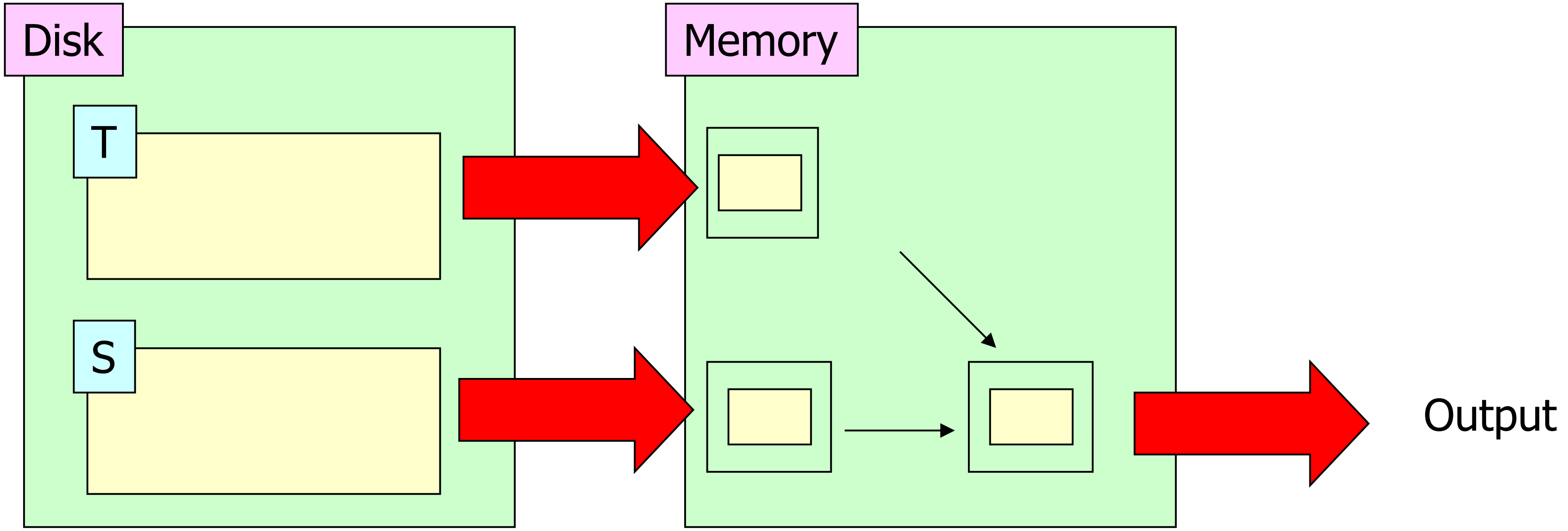
Simple-Nested Loop Join

```
For each page p of T                              (T is called outer relation)
        for each page q of S                      (S is called inner relation)
                for each tuple $t \in p$ and each tuple $s \in q$
                        such that t.sid = s.sid
                        output <t, s> to the output
```

```
For each tuple r in page p
    For each tuple s in page q,
        If r.sid = s.sid then
            output the tuple joined from
            tuple r and tuple s
```



Simple-Nested Loop Join



Student: 500 pages, 80 tuples/page, 50 bytes/tuple
Take: 1000 pages, 100 tuples/page, 40 bytes/tuple

- 200 courses in table Take
- 1, 2, …, 40 in attribute Age of table Student

Simple-Nested Loop Join

```
For each page p of T                          (T is called outer relation)
        for each page q of S                  (S is called inner relation)
            for each tuple $t \in p$ and each tuple $s \in q$
                such that t.sid = s.sid
                    output <t, s> to the output
```

We need 3 pages for buffer

1 page for p (from T)

1 page for q (from S)

1 page for the output

Student: 500 pages, 80 tuples/page, 50 bytes/tuple
Take: 1000 pages, 100 tuples/page, 40 bytes/tuple

- 200 courses in table Take
- 1, 2, …, 40 in attribute Age of table Student

Simple-Nested Loop Join

```
For each page p of T                          (T is called outer relation)
        for each page q of S                  (S is called inner relation)
            for each tuple $t \in p$ and each tuple $s \in q$
                such that t.sid = s.sid
                output <t, s> to the output
```

Cost of Reading T = 1000 pages

The total number of times that S is read = 1000

Cost of Reading S (with multiple times) = 1000*500 = 500000 pages

Total Cost = 1000 + 500000 = 501000 pages

Outline

- Simple-Nested Loop Join
- Block-Nested Loop Join
- Sort-Merge Join
- Index-Nested Loop Join
- Hash Join

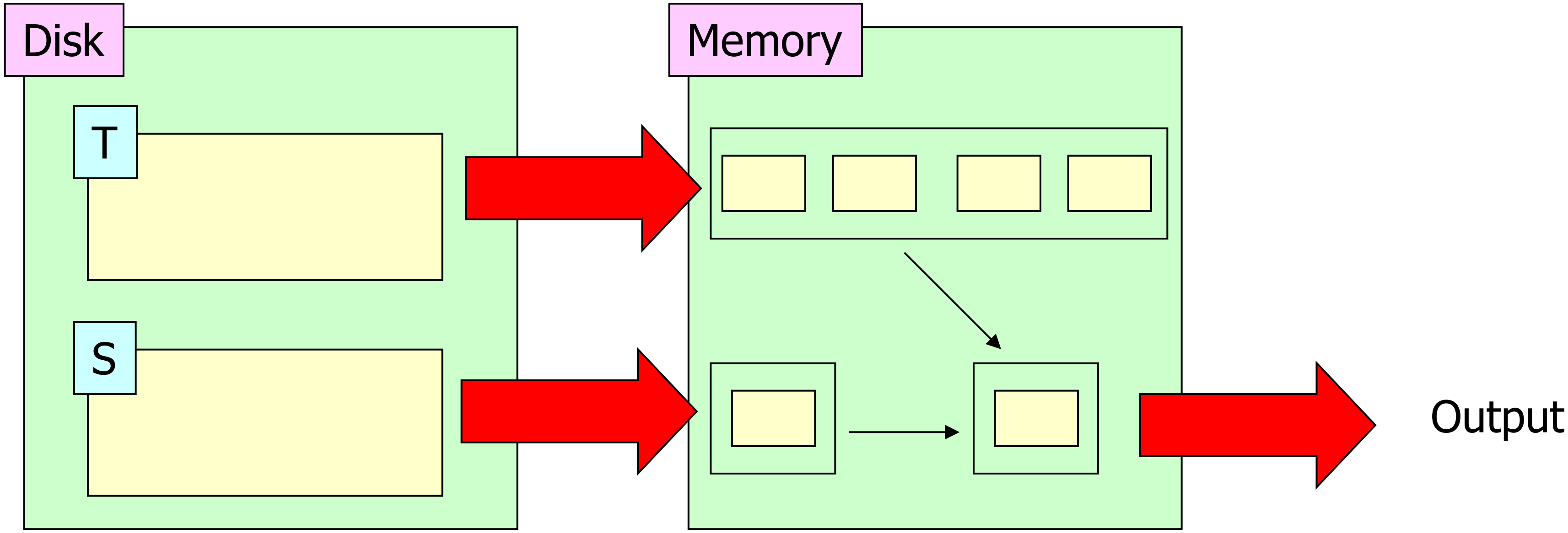
Student: 500 pages, 80 tuples/page, 50 bytes/tuple
Take: 1000 pages, 100 tuples/page, 40 bytes/tuple

- 200 courses in table Take
- 1, 2, …, 40 in attribute Age of table Student

Block-Nested Loop Join

```
For each block P of T                          (T is called outer relation)
        for each page q of S                   (S is called inner relation)
              for each tuple t ∈ P and each tuple s ∈ q
                        such that r.sid = s.sid
                   output <t, s> to the output
```

Block-Nested Loop Join



Student: 500 pages, 80 tuples/page, 50 bytes/tuple
Take: 1000 pages, 100 tuples/page, 40 bytes/tuple

- 200 courses in table Take
- 1, 2, …, 40 in attribute Age of table Student

Block-Nested Loop Join

Suppose that B = 6 pages

```
For each block P of T                          (T is called *outer relation*)
            for each page q of S               (S is called *inner relation*)
                  for each tuple $t \in P$ and each tuple $s \in q$
                        such that t.sid = s.sid
                      output <t, s> to the output
```

Suppose we have B pages in memory (or B buffer pages)

B-2 pages for P (from T)

1 page for q (from S)

1 page for the output

Student: 500 pages, 80 tuples/page, 50 bytes/tuple
Take: 1000 pages, 100 tuples/page, 40 bytes/tuple

- 200 courses in table Take
- 1, 2, …, 40 in attribute Age of table Student

Block-Nested Loop Join

Suppose that B = 6 pages

```
For each block P of T                       (T is called *outer relation*)
        for each page q of S                (S is called *inner relation*)
            for each tuple t ∈ P and each tuple s ∈ q
                such that t.sid = s.sid
                    output <t, s> to the output
```

Cost of Reading T = 1000 pages

The total number of times that S is read = $\lceil 1000/(6-2) \rceil = 250$

Cost of Reading S (with multiple times) = 250*500 = 125000 pages

Total Cost = 1000 + 125000 = 126000 pages

Student: 500 pages, 80 tuples/page, 50 bytes/tuple
Take: 1000 pages, 100 tuples/page, 40 bytes/tuple

- 200 courses in table Take
- 1, 2, …, 40 in attribute Age of table Student

Block-Nested Loop Join

Suppose that B = 6 pages

- T is the outer relation
- S is the inner relation
- How about the following?
  - T is the inner relation
  - S is the outer relation

Student: 500 pages, 80 tuples/page, 50 bytes/tuple
Take: 1000 pages, 100 tuples/page, 40 bytes/tuple

- 200 courses in table Take
- 1, 2, …, 40 in attribute Age of table Student

Block-Nested Loop

Suppose that B = 6 pages

```
For each block P of S                         (S is called *outer relation*)
          for each page q of T                (T is called *inner relation*)
                for each tuple s ∈ P and each tuple t ∈ q
                        such that t.sid = s.sid
                      output <t, s> to the output
```

Cost of Reading S = 500 pages

The total number of times that T is read = $\lceil 500/(6-2) \rceil$ = 125

Cost of Reading T (with multiple times) = 125*1000 = 125000 pages

Total Cost = 500 + 125000 = 125500 pages

Block-Nested Loop Join

- Execution 1
  - T is the outer relation
  - S is the inner relation
  - Cost = 126000 pages
- Execution 2
  - S is the inner relation
  - T is the outer relation
  - Cost = 125500 pages
- Which one is better?

Outline

- Simple-Nested Loop Join
- Block-Nested Loop Join
- Sort-Merge Join
- Index-Nested Loop Join
- Hash Join

Sort-Merge Join

1. Sort table T according to sid
2. Sort table S according to sid
3. Merge table T and table S

T
0004
0005
0001
0002
0002
0005
0002

S
0003
0002
0005
0002
0005
0002

Sort-Merge Join

1. Sort table T according to sid
2. Sort table S according to sid
3. Merge table T and table S

T
0001
0002
0002
0002
0004
0005
0005

S
0003
0002
0005
0002
0005
0002

Sort-Merge Join

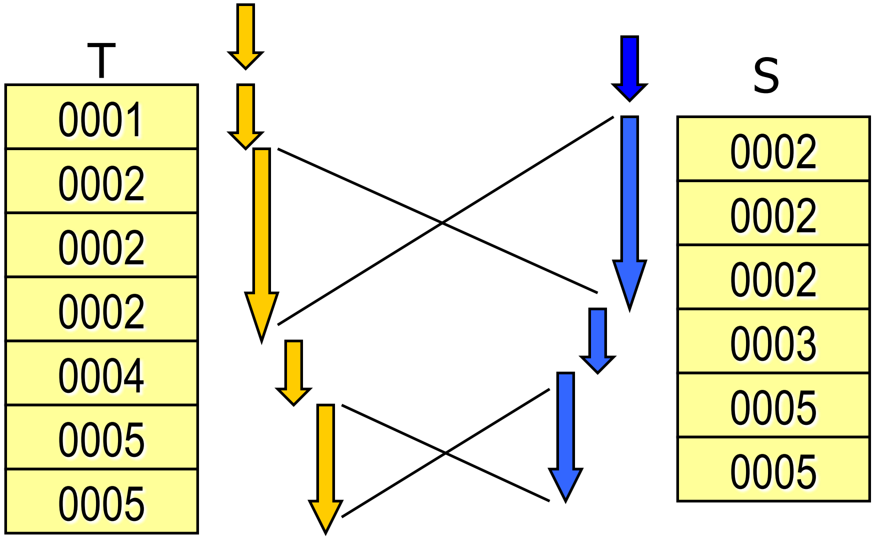
1. Sort table T according to sid
2. Sort table S according to sid
3. Merge table T and table S

T
0001
0002
0002
0002
0004
0005
0005

S
0002
0002
0002
0003
0005
0005

Sort-Merge Join

1. Sort table T according to sid
2. Sort table S according to sid
3. Merge table T and table S



T
0001
0002
0002
0002
0004
0005
0005

S
0002
0002
0002
0003
0005
0005

Sort-Merge Join

1. Sort table T according to sid
2. Sort table S according to sid
3. Merge table T and table S

Cost of Sorting = 2N * (# of passes)

where no. of passes= $1 + \lceil \log_{B-1} \lceil N / B \rceil \rceil$

Student: 500 pages, 80 tuples/page, 50 bytes/tuple
Take: 1000 pages, 100 tuples/page, 40 bytes/tuple

- 200 courses in table Take
- 1, 2, …, 40 in attribute Age of table Student

Suppose that B = 6 pages

Sort-Merge Join

1. Sort table T according to sid
2. Sort table S according to sid
3. Merge table T and table S

Cost of Sorting T = 2*1000*($1+\lceil \log_{6-1}\lceil 1000/6 \rceil \rceil$)
= 10000

Cost of Sorting S = 2*500*($1+\lceil \log_{6-1}\lceil 500/6 \rceil \rceil$)
= 4000

Cost of Merging = Cost of Reading T + Cost of Reading S
= 1000 + 500
= 1500

Total Cost = 10000 + 4000 + 1500 = 15500 pages

Outline

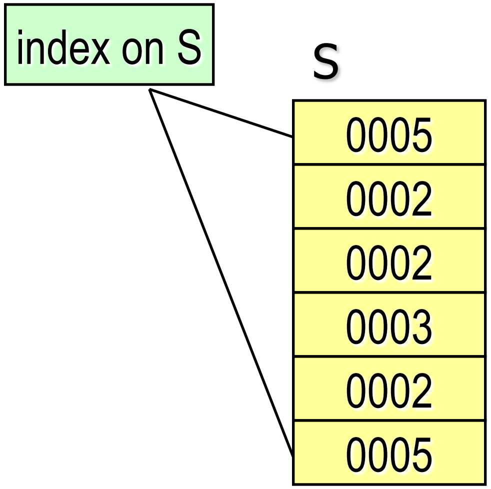
- Simple-Nested Loop Join
- Block-Nested Loop Join
- Sort-Merge Join
- Index-Nested Loop Join
- Hash Join

Student: 500 pages, 80 tuples/page, 50 bytes/tuple
Take: 1000 pages, 100 tuples/page, 40 bytes/tuple

- 200 courses in table Take
- 1, 2, …, 40 in attribute Age of table Student

Index-Nested Loop Join

- Assume that there is an index built on attribute sid for table Student
- E.g., If we search for the tuples with sid = 0005



index on S

S

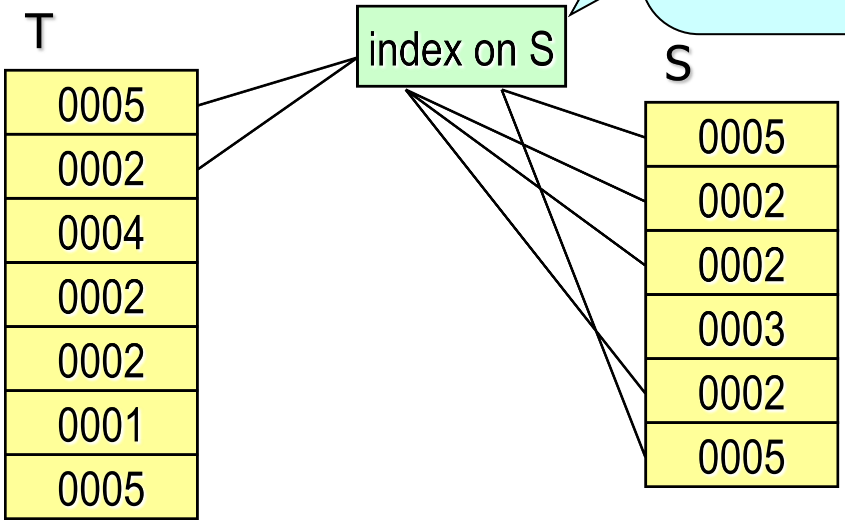
0005
0002
0002
0003
0002
0005

Index-Nested Loop Join

```
For each tuple t of T
    ID ← t.sid
    use the index to obtain a tuple s (or tuples) in S
        with s.sid = ID
    for each such s in S
        output <t, s> to the output
```

Assume that we have a hash index on sid for table S

The cost of accessing a hash index is 1.2 pages



Student: 500 pages, 80 tuples/page, 50 bytes/tuple
Take: 1000 pages, 100 tuples/page, 40 bytes/tuple

- 200 courses in table Take
- 1, 2, …, 40 in attribute Age of table Student

Index-Nested Loop Join

Suppose that B = 6 pages

```
For each tuple t of T
      ID ← t.sid
      use the index to obtain a tuple s (or tuples) in S
            with s.sid = ID
      for each such s in S
            output <t, s> to the output
```

Cost of Reading T = 1000 pages

The total number of tuples in T =1000*100=100000

Why "+1"?

Cost of Reading S (with multiple times) =100000*(1.2 + 1) = 220000 pages

Total Cost = 1000 + 220000 = 221000 pages

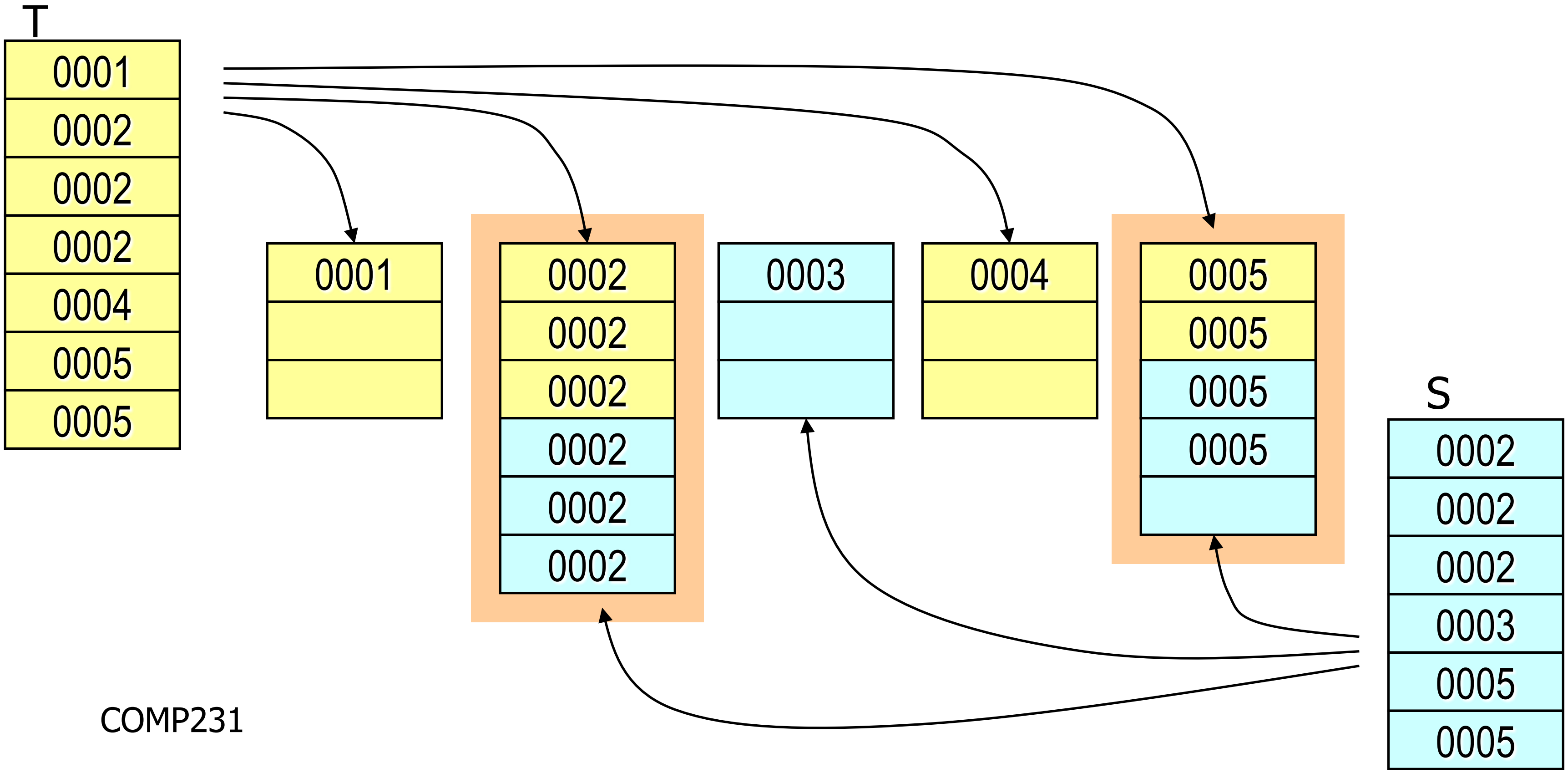
Outline

- Simple-Nested Loop Join
- Block-Nested Loop Join
- Sort-Merge Join
- Index-Nested Loop Join
- Hash Join

Hash Join

1. Hash all tuples in T based on sid
2. Hash all tuples in S based on sid
3. For each hash slot,
    1. Join all tuples t from T and all tuples s from S where t.sid = s.sid
    2. Output all these tuples

Hash Join


T
0001
0002
0002
0002
0004
0005
0005
0001
0002
0002
0002
0002
0002
0002
0003
0004
0005
0005
0005
0005
S
0002
0002
0002
0003
0005
0005