Join algorithms in Database

Difficulty Level : Easy • Last Updated : 24 Dec, 2021

There are two algorithms to compute natural join and conditional join of two relations in database: Nested loop join, and Block nested loop join.

To understand these algorithms we will assume there are two relations, relation R and relation S. Relation R has T_R tuples and occupies B_R blocks. Relation S has T_S tuples and occupies B_S blocks. We will also assume relation R is the outer relation and S is the inner relation.

Nested Loop Join

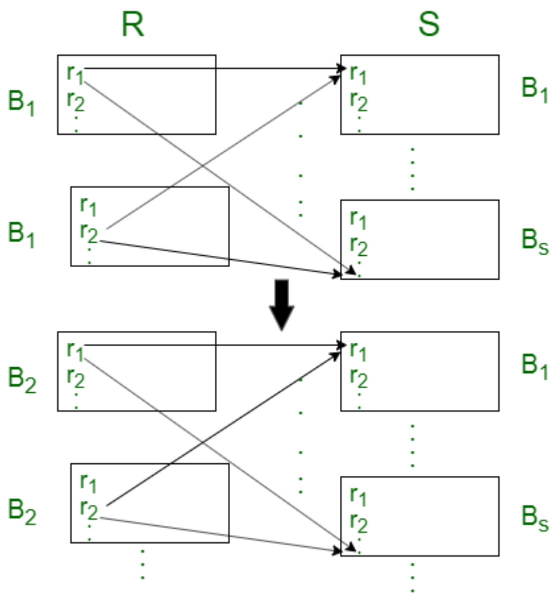
In the nested loop join algorithm, for each tuple in outer relation, we have to compare it with all the tuples in the inner relation then only the next tuple of outer relation is considered. All pairs of tuples which satisfy the condition are added in the result of the join.

```
  for each tuple $t_s$ in $T_s$ do
   compare ($t_R$, $t_s$) if they satisfies the condition
   add them in the result of the join
  end
 end
```

This algorithm is called nested join because it consists of nested for loops.

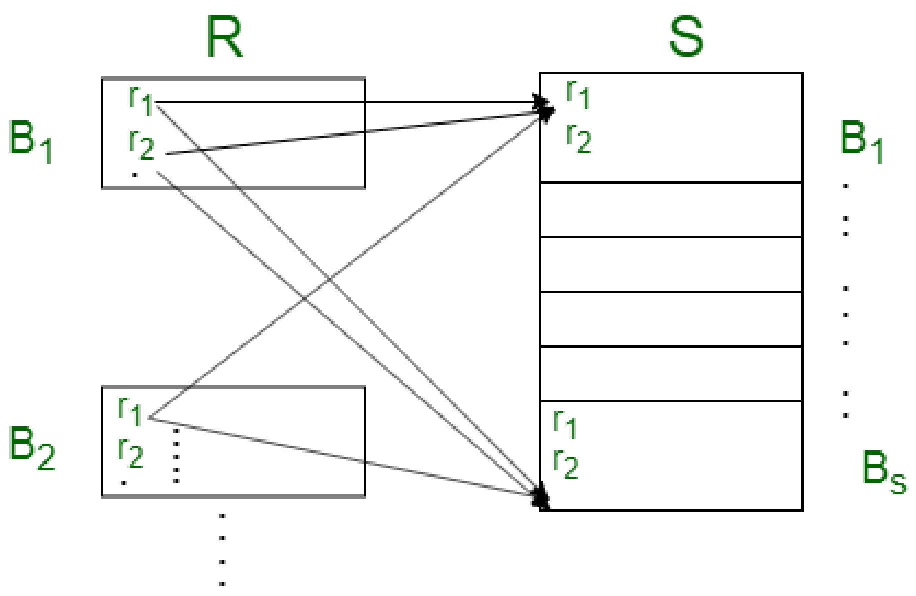
Let’s see some cases to understand the performance of this algorithm,

Case-1: Assume only two blocks of main memory are available to store blocks from R and S relation.



So, the total block transfers needed = $T_R * B_S + B_R$

Case-2: Assume one relation fits entirely in the main memory and there is at least space for one extra block.



In this case, the blocks of relation S (that is, the inner relation) are only transferred once and kept in the main memory and the blocks of relation R are transferred sequentially. So, all the blocks of both the relation are transferred only once.
So, the total block transfers needed = $B_R + B_S$

The relation with a lesser number of blocks should be the outer relation to minimizing the total number of blocks access required in the main memory to complete the join. That is, min(B_R, B_S)+1 is the minimum number of blocks in the main memory required to join two relations so that no block is transferred more than once.

In nested loop join, more access cost is required to join relations if the main memory space allocated for join is very limited.

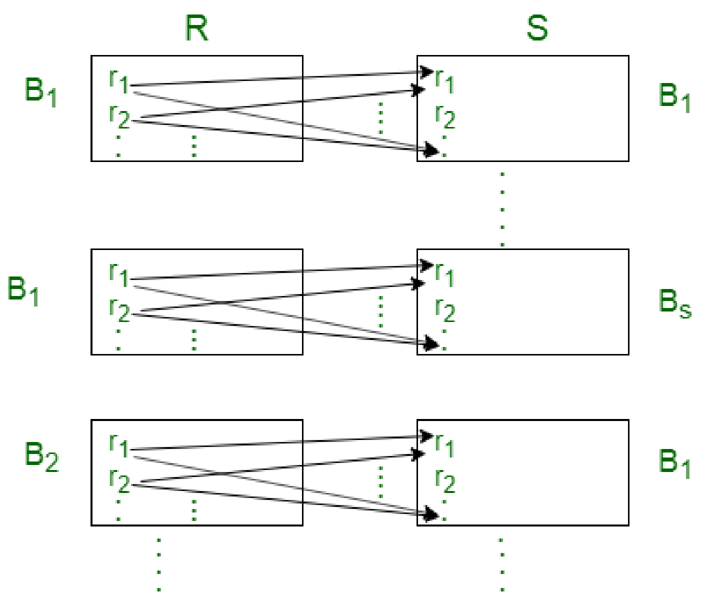
Block Nested Loop Join:

In block nested loop join, for a block of outer relation, all the tuples in that block are

```
    for each tuple $t_R$ in $T_R$ do
     for each tuple $t_S$ in $T_S$ do
      compare ($t_R$, $t_S$) if they satisfies the condition
      add them in the result of the join
     end
    end
  end
end
```

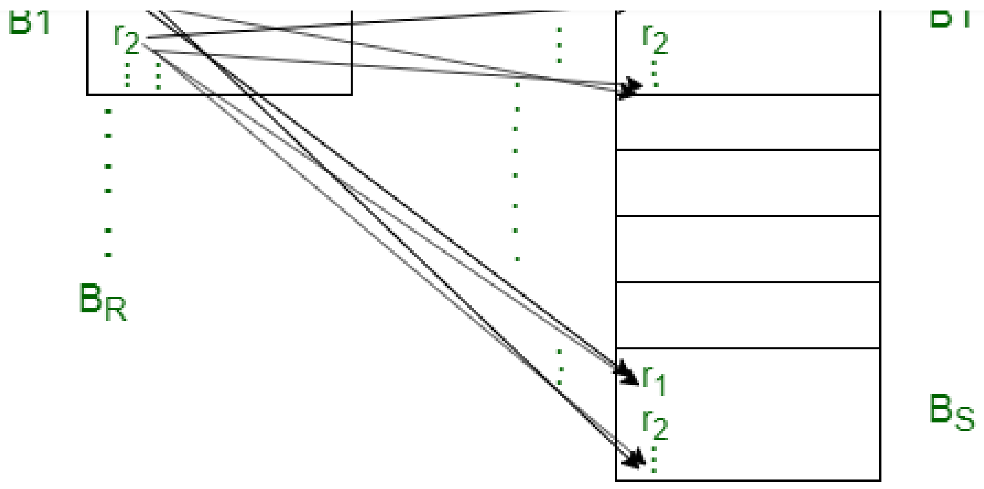
Let's look at some similar cases as nested loop join,

Case-1: Assume only two blocks of main memory are available to store blocks from R and S relation.



For each block of relation R, we have to transfer all blocks of relation S and each block of relation R should be transferred only once.
So, the total block transfers needed = $B_R + B_R * B_S$

Case-2: Assume one relation fits entirely in the main memory and there is at least space for one extra block.



In this case, total block transfers needed are similar to nested loop join.
Block nested loop join algorithm reduces the access cost compared to nested loop join if the main memory space allocated for join is limited.

Related GATE questions:

- Gate IT 2005 | Question 84
- Gate IT 2005 | Question 85

Related Articles

1. SQL | Join (Cartesian Join & Self Join)
2. Inner Join vs Outer Join
3. Difference between Inner Join and Outer Join in SQL
4. Difference between Natural join and Inner Join in SQL
5. Difference between Natural join and Cross join in SQL
6. Full join and Inner join in MS SQL Server
7. Left join and Right join in MS SQL Server

Start Your Coding Journey Now!

10. Difference between Hash Join and Sort Merge Join

Like 1

Previous Next

Article Contributed By :

SreejitBose
@SreejitBose

Vote for difficulty

Current difficulty : Easy

Easy Normal Medium Hard Expert

Improved By :	devanidarshak01
Article Tags :	DBMS, GATE CS
Practice Tags :	DBMS

Improve Article Report Issue

Start Your Coding Journey Now!

Company

About Us

Careers

In Media

Contact Us

Privacy Policy

Copyright Policy

Advertise with us

Learn

DSA

Algorithms

Data Structures

SDE Cheat Sheet

Machine learning

CS Subjects

Video Tutorials

Courses

News

Top News

Technology

Work & Career

Business

Finance

Lifestyle

Knowledge

Languages

Python

Java

CPP

Golang

C#

SQL

Kotlin

Web Development

Web Tutorials

Django Tutorial

HTML

JavaScript

Bootstrap

Contribute

Write an Article

Improve an Article

Pick Topics to Write

Write Interview Experience

Internships

Start Your Coding Journey Now!