University of Tromsø

Department of computer science

INF-2700: Database system concept

Assignment 3
DBMS elements

Camilla Stormoen, cst042

November 16, 2015

1 Content

2 Introduction

This report describes the implementation of some extended features of the DBMS with join operations. The natural join is be implemented using a nested-loop join and a block nested-loop join.

3 Technical Background

3.1 Natural join

Natural join is a binary operation where R and S are relations, $\mathbf{R} \bowtie \mathbf{S}$. The result of the natural join is a set of all combinations of tuples in R and S that are equal on their common attribute name. [1].

A result of an natural join should be:

TBL1

Id	Name
1	Frida
2	Lars
4	Ola
5	Heidi

TBL2

Id	Name2
1	Siv
2	Stein
3	Ola
5	Kari

TBL3

Id	Name	Name2
1	Frida	Siv
2	Lars	Stein
5	Heidi	Kari

4 Design

4.1 Nested-loop join

The nested-loop join consist of a pair of nested for-loops. We have two relations, where the first relation is called the outer relation and the second relation is called the inner relation.

When receiving a join-operation, a search through both the outer- and inner relation are performed. The field name to both relations determine whether a join can be executed or not. If the condition is satisfied, a new schema is made from the two relations. The new schema, or relation, is the destination schema where the result from the loop-join is being stored.

When the new schema is made, the actual join takes place. New records are made for every relation, also the new one. When looping through the records pages, a check can be done to find the records value. If the records value match, the value can be appended to the new destination record.

It is important to remember to set and update the pages position when looping through the records, and also set the table position the beginning every time.

4.2 Block nested-loop join

The block nested-loop join starts the same as a nested-loop join. However, here is every block of the inner relation paired with every block of the outer relation. In this function, a loop iteration over the records pages are done for both the outer- and inner relation. But, since this is a block nested-loop, two loops more are necessary. A loop through the records in one block to the last record in the block for outer relation and also one for the inner relation. *Figure 1* and *Figure 2* shows how the iteration works for the block nested-loop.

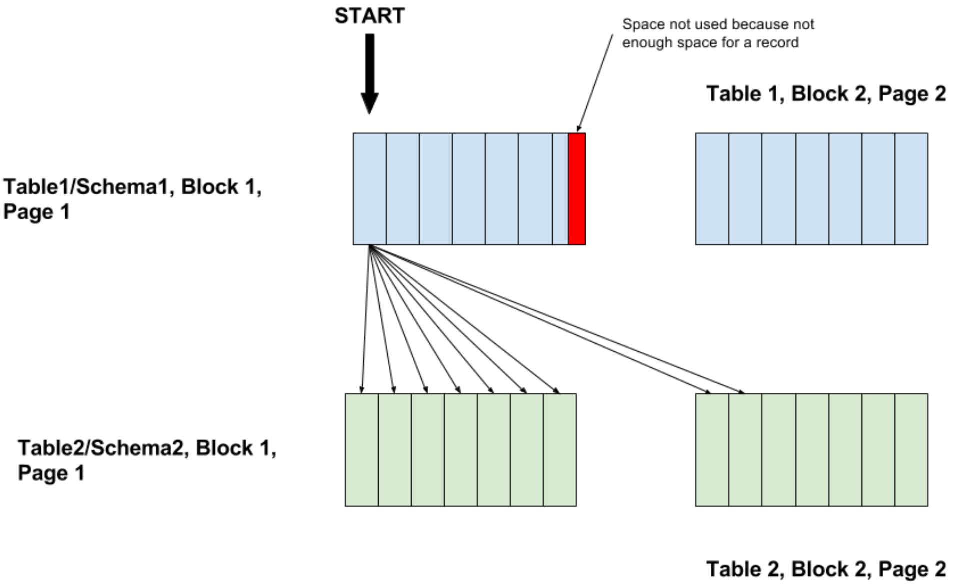
START
Space not used because not enough space for a record
Table 1, Block 2, Page 2
Table1/Schema1, Block 1, Page 1
Table2/Schema2, Block 1, Page 1
Table 2, Block 2, Page 2

Figure 1 *Show how the block nested-loop join works. Search through outer relations schema block against inner relation schema block. When searched through all records in block 1, block 1 will never be search in again since all the matches/mismatches have been found in that block.*

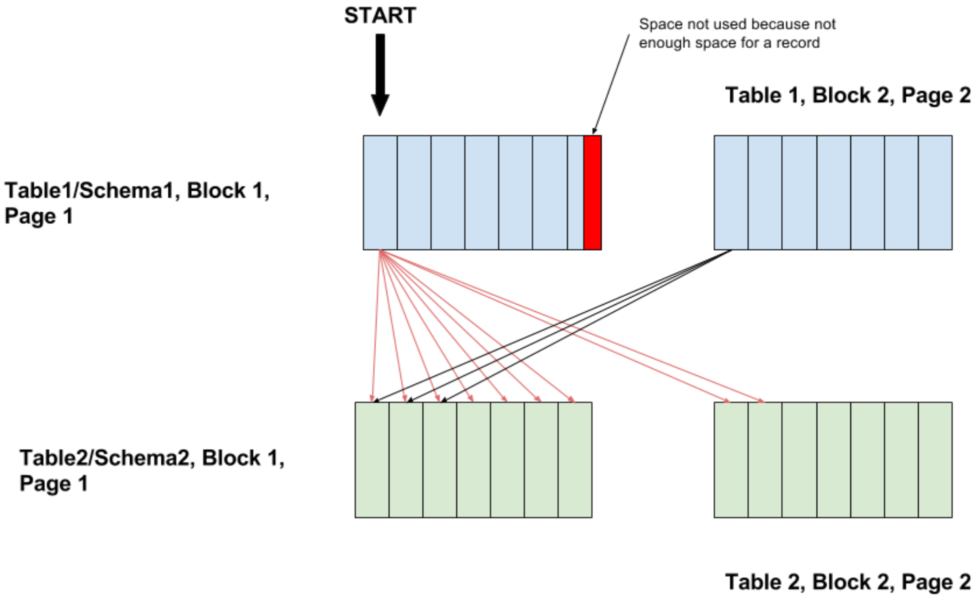
START
Space not used because not enough space for a record
Table 1, Block 2, Page 2
Table1/Schema1, Block 1, Page 1
Table2/Schema2, Block 1, Page 1
Table 2, Block 2, Page 2

Figure 2 *Show how the iteration works. When searching through block 1 and found matches, the iteration goes to next block to find matches.*

One block is 512 bytes and the records in the blocks can be of various sizes. A calculation of how many records there is in one block should be done because a block may not hold every records or there's no space in the block to contain a record, so there will be "free" space in a block (the red block in *Figure 2*). If this calculation is not made, the program may try to read something it should read and the program will exit or an error will occur.

5 Implementation

The implementation is written in C language and executed using the Linux terminal on the ifilab computers.

The performance evaluation of this system is used by the given profiling capability in the base program.

6 Discussion

6.1 Nested-loop join vs block-nested loop join

A nested-loop join is expensive since it examines every pair of tuples in the two relations, as we can see in *Figure 3.* The nested-loop join would work fine in a small set of records, maximum 200-400 records before numbers of disk seek, read, writes and IO goes high because there is no more space in memory, so the program will needs to go to disk.

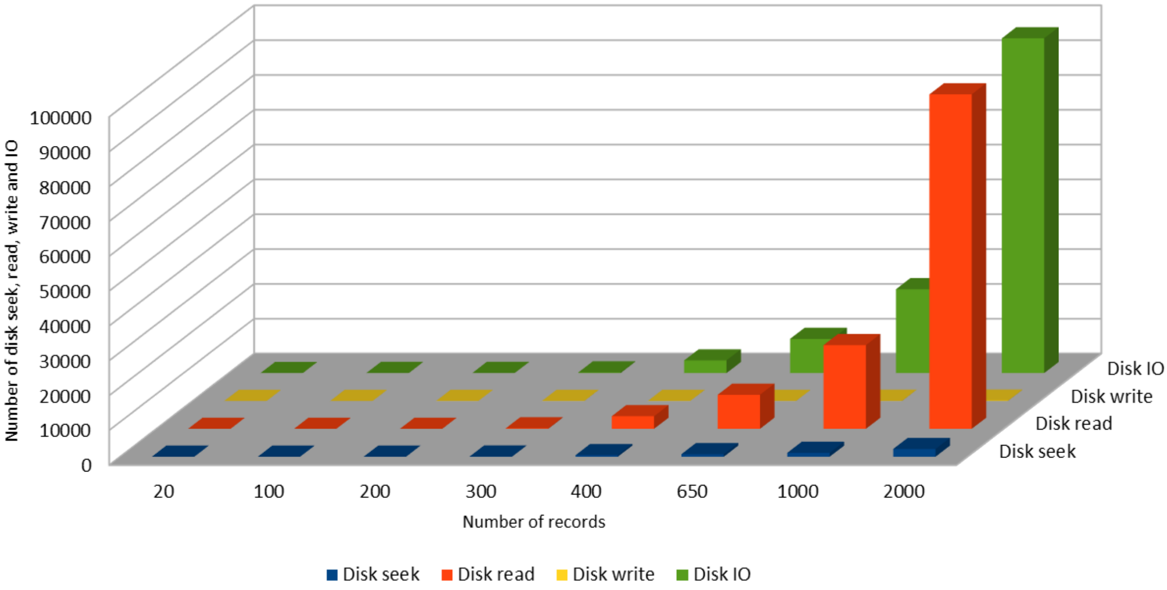

Figure 3 *Shows number of records and how many disk seek, read, writes and IO the program use in a nested-loop join.*

A block nested-loop joins worst case is if each block can in the inner relation is read only once for each block in the outer relation, instead of just once for each tuple in outer relation. However, the best case is if the inner relation fits in memory, then need only 2 seeks. Because if the outer relation have blocks and inner relation fits in memory, the only times checking with disk is when the outer relation have a block in disk. But when the program is finished searching through the block, the block will never be searched in again, in contrast of a nested-loop join which will always check every block every time.

7 Conclusion

This report has described the implementation of some extended features of the DBMS with join operations. One feature is the nested-loop join and the other is the block-nested loop join. The conclusion of which join-operation that is the best, is the block nested-loop join because it saves a lot of disk read, writes and seek. However, the nested-loop join is effective when search through a small number of records.

8 References

[1] Wikipedia contributors, «Join (SQL),» Wikipedia, The Free Encyclopedia., 10 November 2015. [Internett]. Available: https://en.wikipedia.org/w/index.php?title=Join_(SQL)&oldid=690000927. [Funnet 14 November 2015].