

## Re: Computing hash values

**Source:**

<http://www.tech-archive.net/Archive/SQL-Server/microsoft.public.sqlserver.programming/2004-09/5212.html>

---

**From:** Kalen Delaney (*replies\_at\_public\_newsgroups.com*)

**Date:** 09/23/04

Date: Thu, 23 Sep 2004 07:40:44 -0700

You don't need to understand it to tune your queries.  
If you want to understand what hashing is all about, I suggest you take a look at the reference at the beginning of the thread, or use google to search for generic informaiton about hashing.

A join key is a column in one table that is matched with a column in another table, Both tables then have join keys.

It sounds like you're describing a 'join expression'.

--  
HTH

-----  
Kalen Delaney  
SQL Server MVP  
www.SolidQualityLearning.com  
"Leila" <lelas@hotpop.com> wrote in message  
news:e6846sWoEHA.3792@TK2MSFTNGP11.phx.gbl...  
> Thanks Tibor!  
> What I cannot understand is that what the meaning of "calculating hash  
> value  
> based on join key" is.  
> Because join key is only the name of two fields plus an operator between  
> them, it doesn't have any value itself (to be calculated).  
>  
> Leila  
>  
>  
> "Tibor Karaszi" <tibor\_please.no.email\_karaszi@hotmail.nomail.com> wrote  
> in  
> message news:eZi#TaWoEHA.1776@TK2MSFTNGP14.phx.gbl...  
>> For each row in the probe table, a hash value is calculated based on the  
> join key. Then SQL Server  
>> looks in the hash bucked from the build table to see if there is any  
> match. The key (no pun  
>> intended) here is that the build table is splitted up into a lot of  
> buckets, and for the other  
>> table, SQL server only have to look in a specific bucket to find if  
> there's a match.  
>>  
>> --  
>> Tibor Karaszi, SQL Server MVP  
>> <http://www.karaszi.com/sqlserver/default.asp>

## microsoft.public.sqlserver.programming: Re: Computing hash values

```
>> http://www.solidqualitylearning.com/
>>
>>
>> "Leila" <lelas@hotpop.com> wrote in message
> news:uRd8FNWoEHA.3488@TK2MSFTNGP12.phx.gbl...
>> > Does the hash table have an structure like index? If it doesn't, I
> think
>> > nested loop is inevitable for matching rows between hash table and the
> probe
>> > table.
>> >
>> >
>> >
>> > "Kalen Delaney" <replies@public_newsgroups.com> wrote in message
>> > news:eQJ6HbRoEHA.2108@TK2MSFTNGP10.phx.gbl...
>> > > A nested loop is when the inner table is processed completely for
> each
>> > row
>> > > of the outer table.
>> > >
>> > > For hash joins the inner table is read once to build the hash table,
> and
>> > > then not touched again. Then each row of the outer table leads to a
> single
>> > > access of the hash table.
>> > >
>> > > --
>> > > HTH
>> > > -----
>> > > Kalen Delaney
>> > > SQL Server MVP
>> > > www.SolidQualityLearning.com
>> > >
>> > >
>> > > "Leila" <lelas@hotpop.com> wrote in message
>> > > news:OfY1jORoEHA.3760@TK2MSFTNGP12.phx.gbl...
>> > > > Kalen,
>> > > > When the hash table is ready, will there be something like nested
> loop
>> > > to
>> > > > match rows? Because Mark described that the bottom table is
>> > > > scanned once (not in a nested loop).
>> > > > Leila
>> > > >
>> > > >
>> > > > "Kalen Delaney" <replies@public_newsgroups.com> wrote in message
>> > > > news:u26#jDRoEHA.2900@TK2MSFTNGP12.phx.gbl...
>> > > > > The 'inner' table is whichever one is chosen by the SQL Server
>> > optimizer
>> > > > to
>> > > > > build the hash table. Typically this will be the smaller one, but
> not
>> > > > > always.
>> > > > > For BOL to say the smaller of the two is the build input is a bit
> of an
>> > > > > overgeneralization.
>> > > > >
>> > > > > --
>> > > > > HTH
>> > > > > -----
>> > > > > Kalen Delaney
```

## microsoft.public.sqlserver.programming: Re: Computing hash values

```
>> > > >> SQL Server MVP
>> > > >> www.SolidQualityLearning.com
>> > > >>
>> > > >>
>> > > >> "Leila" <lelas@hotpop.com> wrote in message
>> > > >> news:eejim5QoEHA.3788@TK2MSFTNGP10.phx.gbl...
>> > > >> > Thanks Kalen!
>> > > >> > You mentioned 'the data in the inner table is organized into a
> hash
>> > > >> > table'.
>> > > >> > I read in BOL 'the smaller of the two inputs is the build
>> > > >> > input'.
>> > > >> > Are they different?
>> > > >> >
>> > > >> >
>> > > >> >
>> > > >> >
>> > > >> > "Kalen Delaney" <replies@public_newsgroups.com> wrote in message
>> > > >> > news:ODudYpQoEHA.260@TK2MSFTNGP10.phx.gbl...
>> > > >> >> The 'only' difference is a very expensive one.
>> > > >> >> If you have an index, SQL Server can take a value from the
>> > > >> >> outer
>> > table
>> > > >> >> and
>> > > >> >> use the index to find matching rows in the inner table.
>> > > >> >>
>> > > >> >> With a hash match, which is used because there IS no useful
> index,
>> > the
>> > > >> > data
>> > > >> >> in the inner table is organized into a hash table, so that SQL
>> > Server
>> > > >> >> can
>> > > >> >> find matching rows using the hash table instead of an index.
>> > > >> >> Al though the inner table is scanned only once, the process of
>> > > >> >> building
>> > > >> >> the
>> > > >> >> hash table is resource intensive, and the hash table uses a lot
> of
>> > > >> >> memory
>> > > >> >> for a big table.
>> > > >> >>
>> > > >> >> You're better off building a good index to make the nested
>> > > >> >> loops
>> > > >> >> possible.
>> > > >> >>
>> > > >> >> --
>> > > >> >> HTH
>> > > >> >> -----
>> > > >> >> Kalen Delaney
>> > > >> >> SQL Server MVP
>> > > >> >> www.SolidQualityLearning.com
>> > > >> >>
>> > > >> >>
>> > > >> >> "Leila" <lelas@hotpop.com> wrote in message
>> > > >> >> news:%23t9lNSQoEHA.2340@TK2MSFTNGP10.phx.gbl...
>> > > >> >> > Hi Kalen,
>> > > >> >> > Thanks for your suggestion.
>> > > >> >> > I'm a little confused about the difference between Hash Match
> and
>> > > >> >> > Nested
>> > > >> >> > Loops. As far as I learned from BOL, in Hash Match, the hash
```

## microsoft.public.sqlserver.programming: Re: Computing hash values

```
>> > values
>> > > are
>> > > >> > moved from the base table to a new place in memory(called
>> > > >> > hash
>> > > > table),
>> > > >> > then
>> > > >> > an operation like nested loop happens between hash table and
>> > another
>> > > >> > table.
>> > > >> > In nested loops, no value is moved from the base table,
> instead
>> > the
>> > > >> > loop
>> > > >> > begins (with no hash table in between) directly with other
> table.
>> > > >> > It seems the only difference is the existence of hash table
>> > > >> > in
>> > > > between,
>> > > >> > is
>> > > >> > that true?
>> > > >> > Thanks again,
>> > > >> > Leila
>> > > >> >
>> > > >> >
>> > > >> > "Kalen Delaney" <replies@public_newsgroups.com> wrote in
> message
>> > > >> > news:euCtqpPoEHA.3460@TK2MSFTNGP10.phx.gbl...
>> > > >> >> Hi Leila
>> > > >> >>
>> > > >> >> For your query tuning, it shouldn't matter what the actual
> hash
>> > > > values
>> > > >> > are.
>> > > >> >> If possible, you should try to build an index that will
>> > > >> >> allow
> SQL
>> > > >> > Server
>> > > >> > to
>> > > >> >> perform a different join technique than hashing.
>> > > >> >>
>> > > >> >> Microsoft does not document any details of the hash
>> > > >> >> functions
>> > they
>> > > > use
>> > > >> >> for
>> > > >> >> processing hash join operations. If you want to know more
> about
>> > > >> >> hashing
>> > > >> >> in
>> > > >> >> general, read "The Art of Computer Programming -- Volume 3:
>> > Sorting
>> > > >> >> and
>> > > >> >> Searching" by Donald Knuth.
>> > > >> >>
>> > > >> >> --
>> > > >> >> HTH
>> > > >> >> -----
>> > > >> >> Kalen Delaney
>> > > >> >> SQL Server MVP
>> > > >> >> www.SolidQualityLearning.com
>> > > >> >>
>> > > >> >>
```

