

## Re: Was: what does "serialization" mean?

**Source:** <http://coding.derkeiler.com/Archive/General/comp.programming/2004-07/0907.html>

---

**From:** Edward G. Nilges (*spinoza1111\_at\_yahoo.com*)

**Date:** 07/12/04

Date: 11 Jul 2004 17:05:28 -0700

Nick Landsberg <SPAMhukolauTRAP@SPAMworldnetTRAP.att.net> wrote in message news:<hnKHc.78319\$OB3.20323@bgtncs05-news.ops.worldnet.att.net>...

> *Edward G. Nilges wrote:*

>

>> *Nick Landsberg <SPAMhukolauTRAP@SPAMworldnetTRAP.att.net> wrote in message news:<BmBHc.222121\$Gx4.16309@bgtncs04-news.ops.worldnet.att.net>...*

>>

>

> *[ MUCH MORE SNIPPAGE SO AS NOT TO REFLOW*

> *THE SAME FIELD ]*

>

>>

>> *I apologize for what was the appearance of brusqueness. The point I*

>> *was trying to make was that hardware speeds are only part of the*

>> *story.*

>>

>

> *OK ... I won't snarl at you if you don't snarl at me :)*

>

> *[ MORE SNIPPAGE ]*

>

>>>

>>> *In addition, the sales and inventory database has*

>>> *to support such queries as:*

>>>

>>> *Report the sales trends week by week*

>>> *for the last 6 months on the top and bottom*

>>> *200 selling products ordered by product category,*

>>> *total sales, geographic region and store.*

>>> *(No, I'm not going to write the SQL for this.)*

>>

>

> *I should have said "ad-hoc" queries in the above*

> *since the customers are usually loathe to define*

> *the specific queries up front.*

One of the problem here is that arguable business psychological logic, in which it is said that we don't know what the queries will

comp.programming: Re: Was: what does "serialization" mean?

be...when, in fact, logic with existential and universal quantifiers  
in fact defines the range of sensible queries.

We really should, I think, disabuse ourselves of the very idea that  
men like Donald Trump can be trusted to define the limits of  
sensibility...even where their definition is a refusal based on  
nothing more than their ignorance.

The worst case condition seems to be a universal claim of the form  
 $(x)[a(x)]$  (for all  $x$ ,  $A$  is true of  $x$ ), or its negation  $\sim(x)[a(x)] \equiv$   
 $(\exists x)[\sim a(x)]$ .

This may not be the worst case:  $a(x)$  may conceal another universal  
quantifier for a search of higher order than  $O(n)$ : but its sensible  
response (below) ALSO address higher order queries.

But the response to the worst case need not be ooooo we need a fast  
(and like wow totally incompatible) server. It COULD be sensible  
software design that indexes the result of the worst case query so  
that the NEXT time the query is run the query is fast, a warning "hey  
bozo, this query is going to take a long time. Why don't you step  
outside for a smoke?" and a progress report including ETA.

The Don Trump style claim is "my guys, the clowns I hired to make my  
data base, don't know squat" when in fact the Donald clone doesn't  
know how to write sensible questions in ordinary English.

Perhaps the response to this should be better edumocation for CEOs  
including an end to grade inflation at prestige universities  
(Princeton is working on this) and an end to free passes for frat boys  
and members of the Texas Air National guard.

>  
> >  
> >  
> > *Let me try extempore without having the time to check syntax:*  
> > *corrections welcome:*  
> >  
> > *SELECT FIRST 6 MONTHLY\_TOTAL\_SALES FROM WHATEVER ORDER BY*  
> > *(PRODUCT\_CATEGORY, TOTAL\_SALES, REGION, STORE)*  
> >  
> > *SELECT LAST 6 MONTHLY\_TOTAL\_SALES FROM WHATEVER ORDER BY*  
> > *(PRODUCT\_CATEGORY, TOTAL\_SALES, REGION, STORE)*  
> >  
> > *WHATEVER would probably be a separate query to summarize total sales.*  
>  
> *I will revisit WHATEVER a little later. Bear with me.*  
>  
> *[ SNIP ]*  
>  
> >

Re: Was: what does "serialization" mean?

## comp.programming: Re: Was: what does "serialization" mean?

- > > *But a second irony is that around SQL a number of urban legends have*
- > > *grown up in communities of scribes who have in fact been assigned the*
- > > *task of using SQL.*
- > >
- > > *One is the privileging of "table" over "relation" when in fact the*
- > > *concept of "relation" is more general and more powerful than "table",*
- > > *subject to the caveat that your queries should not result in the*
- > > *continual creation and re-creation, of relations.*
- >
- > *I agree with the comment about "relations" vs. "tables"*
- > *however, I am ambivalent about the last part of that*
- > *paragraph, the caveat. There are two conflicting*
- > *forces at work in any database. The need for*
- > *data integrity and the need for speed. (To oversimplify*
- > *it drastically.) Those theorists who are in the camp of*
- > *data integrity would claim that no permanent relations*
- > *should exist which can be derived from other relations.*
- > *There are those who will go so far as to claim that*
- > *no summary tables should exist permanently because they*
- > *can always be derived from the base tables and thus are*
- > *replicating data. Those who are solely concerned with speed of*
- > *access create other anomalies within the database*
- > *where the consistency and integrity of the data*
- > *is compromised. For any given database, there is*
- > *a "happy medium" which is struck by a series of*
- > *design tradeoffs.*

Creating an index, I agree, does pose a statistical risk even if you do it right, for the code to create it is an exposure. Some bozo (like oneself) may change it wrong at a future date.

In fact, it contributes to reliability to view the maintainers, the operators of any system as objects without a psychology (or membership in society) who are subject less to human law than statistical mechanics such that it is not possible to say they won't screw up.

Thus replication IS a risk.

But the risk in a high-end proprietary server is you then become dependent upon the willingness of the vendor (or some clown who buys the vendor out) not to gouge you for continued support.

Microsoft solutions, of course, pose the same risk. However, note that Microsoft's very market leadership (restricted I admit to small businesses and American larger firms) means that a larger customer basis exists to keep it honest.

Whereas some whizbang vendor of a powerful server with a small customer base may elect to gouge each customer in the short term. His behavior is less predictable and more suspect to the quantum effects of individual greed.

Re: Was: what does "serialization" mean?

comp.programming: Re: Was: what does "serialization" mean?

Of course, there's always that loose cannon Ballmer. But again it seems there are simply more bodies available at Microsoft to pile on Ballmer and give him a wedgie if he gets random.

>  
> *So, getting back to your WHATEVER above. The first  
> camp would fight you tooth and nail, while the  
> second camp would accept it while possibly ignoring  
> its side-effects. If this were a "canned query"  
> I would personally go with the summary table  
> approach, but in that world, not all queries  
> are canned and one cannot tell ahead of time  
> what query some corporate droid might dream up.*

>  
Under the laws of logic we have no duty to support a query such as "identify all the data base indexes that do not index themselves, and make an index of same".

> >  
> >  
> >> *And yes, that's a real query, and I was told  
> >> third hand, that the results were used, in part,  
> >> to establish price and inventory levels of  
> >> cold medications during a flu epidemic somewhere  
> >> in the midwest.*  
> >>  
> >> *In order to satisfy such a query in most  
> >> database systems (including the ones you mention),  
> >> one must visit a vast majority of the records in  
> >> the sales and inventory database. Thus, the response  
> >> time to such a query is dominated by disk access  
> >> times and data transfer rates between the disk  
> >> and the CPU's.*  
> >>  
> >  
> > *...Only if the system is poorly designed. The above query can be  
> > optimized into straight line code as long as a stored procedure exists  
> > for "whatever".*  
>  
> *See my comments about "whatever" above.*  
>  
> >  
> > *The whole purpose of effective data base design is to AVOID having to  
> > visit a vast majority of records in cases like the above.*  
> >  
> > *I won't say "what part of 'indexing' don't you understand" because you  
> > understand indexing.*  
> >  
> > *Let's assume WalMart doesn't have a single record for total sales but  
> > instead a single table of all sales, completely normalized, at one  
> > item for each distinct sale of each distinct item (the actual sale, of*

Re: Was: what does "serialization" mean?

comp.programming: Re: Was: what does "serialization" mean?

- > > *multiple products as the shopper makes, we hope, a series of impulsive*
- > > *purchases of dreck she does not need, is broken up into distinct*
- > > *items).*
- > >
- > > *If the designers don't put aggregate information including total sales*
- > > *somewhere as the sales come in, they should be WalMart greeters and*
- > > *not designers, but even if they don't, the total sales has only to be*
- > > *developed once.*
- >
- > *This is the tradeoff I would make with most DBMS systems today*
- > *if the queries were known up front.*
- > *Note that this would require the application (or a trigger)*
- > *to update the aggregate as part of the same atomic operation*
- > *as entering the sale. Yet another tradeoff between speed*
- > *up update vs. speed of reports.*
- >
- > >
- > > *The single query in most cases therefore won't visit all records in*
- > > *any halfway decent system.*
- >
- > *The Teradata "dinosaur" actually was able to do a decent*
- > *job on ad-hoc queries because of specialized hardware and*
- > *software without having to rely on aggregate information*
- > *being updated as the individual records came in. And, for*
- > *ad-hoc queries it did visit (almost) all the data.*
- >
- > *[SNIP, it's getting late]*
- >
- > *NPL.*