Is it wise to store total values into the SQL database? [duplicate]

Asked 2 years, 7 months ago Active 2 years, 7 months ago Viewed 881 times

1

This question already has answers here:

Best practise to store total values or to sum up at runtime (2 answers)

Closed 2 years ago.

I wanted to ask if it is a good idea to store total values into SQL database. By total values I mean stuff like **total cost** of an order, or the **total quantity**.

As an example, I'll have a table called **orders** which will have information like when it was created, who made the order and etc. **Orders** will relate to the table **order_items** which will hold the *id* of the item in the order, the quantity of the item in the order and the price at which it was sold (not the item`s original price).

In this situation, should I calculate the total cost of the order once and store it in the **orders** table, or should I calculate it every time I retrieve the order with the corresponding items?

sql database design-patterns store

asked Nov 5 '17 at 16:11

Vladislav Achramovic

79 ● 2 ● 10

1 In general, I'd say it's not wise. It's denormalization. It forces every use-case adding/removing or updating an order item to update the total, and to do it in a way that is safe even if two concurrent requests add or remove items at the same time. But of course, there are always tradeoffs, and sometimes it can make things much simpler or faster. Especially if you're often interested only in the total,and there are many, many order items per order. – JB Nizet Nov 5 '17 at 16:16

2 Answers

Active | Oldest | Votes

There are pros and cons to each choice. If you choose to store the computed totals, you'll have to update them every time you update

3

the details they are dependent on. To do that reliably you'll need to enforce it close to the database, in a trigger for example, rather than relying on on business logic to get it right all the time.

On the other hand if you don't store the computed value you'll have to compute it every time you want access to it. A view will make that trivial.

The decision to store vs recompute comes down to how frequently does the data change vs how often is it accessed. If it changes infrequently, then storing the value may be okay, but if it changes frequently, then it's better to recompute it each time you access it. Similarly if the data is accessed infrequently, then recomputing the values is good while if you access it frequently, then precomputing may be the better option.

Depending on the database you are using, you may be able to get the best of both worlds, by creating a materialized view.

answered Nov 5 '17 at 16:24

Sentinel

5,838 ● 1 ● 13 ● 21

2

For the particular case of an order, the total amount is traditionally stored with the order. The order items then contain the prices of each item in the order.

Why? Simple: certain amounts are available only on the order-level. For instance:

- Shipping charges
- Taxes (the taxes on the item levels may not add up correctly to the order level)
- Order-level discounts (such as discounts based on the overall order size or bundle discounts)

In addition, orders are usually put into the database once, with no or very rare changes on the items (the calculations may not be made until the order is finalized).

Other types of entities might have other considerations. This addresses the specific issue of whether totals should be stored for orders.

answered Nov 5 '17 at 16:29

Gordon Linoff

976k ● 41 ● 416 ● 533