Normalization Vs Denormalization

by Sudheer Sharma - Saturday, November 29, 2008

http://dwhnotes.com/data-warehousing/normalization-vs-denormalization

What do you think of this post?

Awesome (1) Interesting (0) Useful (0)

[amazon-carrousel height=”200? width=”600?]27ad2ec5-117c-4ba7-a05a-f35bfd2df0ff[/amazon-carrousel]

When most of us design our databases we tend to think of them as related (and un-related) tables that contain data. It is common to normalize our tables in order to create a cleaner, more manageable, and intuative Database design. Normalized means separate tables via foreign key relationship, there are however things that we should consider before we normalize our system:

Historical Data:

One of the most important is historical data. When we normalize our system we open the door for historically inaccuracy. What’s that mean? It means that we can change the data in the related table, causing us to lose the accuracy of the data thats related to it.

What do you mean “lose the accuracy”? Lets look at an example: We have a Member table which holds a foreign key referrence to the States table. If we change the data in States table, which is referrenced by the Member table, we have no way of knowing what that the Member record originally contained (like change Republic of Congo to Republic of China).

So why should we care, isn’t this the whole reason why we normalize our database? Yes and No. If we know that we will never have the need to keep an audit trail or report on the data then surely normalizing it is perfectly suitable. However we all know that software requirment change, and there may come a time when the Marketing Department wants to know the history of how its members data has changed over time, normalization would destroy this historical record (unless we some how dig it out of a backup, which would be a major pain in the rear), denormalized does not.

Reporting:

If you think that you will ever have to do any kind of Data Mining or Reporting on your data then avoiding over-normalization is important. Reporting on, or performing calculations on, large amounts of data requires that you denormalize your database. Anticipating this during the design phases can save you a lot of time and grief later on.

Check out this article : http://www.theukwebdesigncompany.com/articles/denormalization-normalization.php

What do you think of this post?

Awesome (1) Interesting (0) Useful (0)

PDF generated by Kalin's PDF Creation Station