NMIT : Course DAT501 in 2012

Unit 03 : Denormalisation

Denormalisation

In the previous unit we looked at the process of normalisation, which is essentially developing a database that stores all the data in the system in a clear and logical manner. In particular, a normalised system should end up in a place where :

- Each table has a single theme (e.g. customer, product, etc)
- Each single piece of data is in a single appropriate place.

A normalised system makes data updating relatively easy, because each piece of data is in only one place, but data updating is not the only function of a system. There are a number of needs that take the system design in other directions. The major ones are :

- Controlling access to the data.
- Verifying that the system is working correctly.
- Providing effective performance.

These are all practical real-world issues. Dealing with them often involves the deliberate and controlled use of structures that are not normalised. The introduction of these features is referred to as denormalisation.

Denormalisation to Provide Access Control

Imagine a system that has a table of personnel data keyed on Personnel ID. It is likely that this data will have the need for several access controls, including :

- Only some users will be able to update particular fields.
- Only some users will be able to see particular fields.

In an ideal world, all this data could be stored in a single table with access controls enforced by the database management software. In practice, achieving these access controls with adequate security often means maintaining several tables in different user areas with different access permissions. These tables would be related one-to-one, which is a relatively unusual type of relationship except in this sort of circumstance.

Denormalisation to Verify System Operation

A purely normalised database design contains no redundant data. Each piece of information is stored in only one place. Such a system has no way to detect faults in operation.

Imagine a payroll system containing information about all employees, past and present. If we want to know how many employees have salaries over $35000, we scan the records and count the number that fit the criteria. If one of the records is dropped by the database management system because of a fault, how would you know? Verifying correct system operation usually involves adding redundant data (e.g. record counts, hash totals of fields, etc) in some appropriate control record and scanning the database periodically (e.g. once per day) to verify that the data in the control record matches the expected value.

Denormalisation to Improve Performance

The third and most complex category of denormalisation is the set of changes that may be needed to provide effective performance. A normalised design is usually very efficient for updating data because the data is only held in one place, but it is often very inefficient at retrieving data; this is because retrieval often involves accessing a lot of related data, and this involves going to many different places in the database.

The starting point of any work to improve performance should be to use the simplest techniques (e.g. creation of multiple indexes on tables); these will often have a very beneficial effect on performance, and are quick and easy to implement. If this does not provide effective system performance, there are many types of situation where denormalisation can provide very significant improvements to performance; these include :

- Maintaining pre-joined tables. In relational systems, table joins are obviously very common, and are usually optimised within the database management system. Even so, they usually take much longer than a single table access. By keeping pre-joined tables (e.g. employee+department), performance will be improved.
- Maintaining redundant data. By keeping the daily balance for each bank account in addition to the individual transactions, accessing this data will be much faster. It can also provide part of the data needed to verify correct system operation.
- Mirrored data. This improves performance by sharing the transaction load across more than one database. This can be more than one copy of the database on entirely different machines, or simply more than one copy on the same machine. If more than one copy can be updated, they will need to be synchronised.
- Splitting tables horizontally. If the data in a table consists of several clearly different sets of records (e.g. the room bookings for three different campuses), the overall size of data that needs to be held and processed can be reduced by holding them in different databases, possibly on different systems.
- Splitting tables vertically. If many users only require a small subset of the available data in a table, access speed to the subset data can often be improved by splitting the table into different sets of columns, related one-to-one.
- Holding repeated groups in one field. This may involve making a careful judgement of the maximum number of repeats allowed, and balancing the extra storage and access cost against the cost of a table join.
- Holding derived data, such as pivot tables, data warehouses, data marts, and the data cubes used by Online Analytical Processing (OLAP).

There are costs involved in producing a denormalised design. These include :

- The extra system development time to create and maintain the code concerned.
- The possibility that maintaining denormalised data will slow down updates.
- The extra disk storage space used.

If the performance of the system is acceptable with a normalised design, it is often easier to continue with the normalised design and accept the occasional report that takes 2 minutes, even if it could be produced with a denormalised design in 3 seconds. Like most issues to do with performance, it is a balanced judgement involving assessment of costs and benefits. We will look at performance issues in more detail later.

It is worth noting that most real systems are based on some form of denormalised design. All denormalisation of a database design should be carefully documented so that everyone concerned is aware of the effects on the system and the reasons that it was carried out.