

Expand

Design Challenges in Global Data Warehousing

David Cox, Head of Delivery, Saksoft Europe
Added on Mar 05, 2010



The data warehouse is now a familiar feature of the technical landscape in many organisations. As data warehousing as a discipline and the technology to support it has matured, ever more ambitious requirements have been tackled.

In recent years there has been increasing demand for data warehouses capable of operating on a global scale. Such implementations collect and consolidate data from multiple geographies and provide reports and analyses back to information consumers across the world.

Although there are several architectural approaches to these global systems all of them have to cope with design issues which tend not to crop up in most local data warehouses. This article explores these issues and suggests ways in which they may be resolved. In particular this article will focus on handling multiple time zones, currencies and languages as well as possible approaches to maintaining a continuous 24/7 update and publishing strategy.

Multiple Time Zones

The vast majority of data warehouses operate on an "overnight update" basis. Data is collected at some point after the end of the working day and processed ready for reporting and analysis the following morning. This tends to work well because the data warehouse is either in "update" mode or "query" mode but not both at the same time. This means update performance is not affected by query processing and vice versa. However for an organisation operating in multiple time zones the notion of "overnight" may not be so straightforward.

Consider a company with operations in eastern Australia, the UK and the west coast of the USA. Figure 1 shows when data is collected, processed and made available for each time zone.

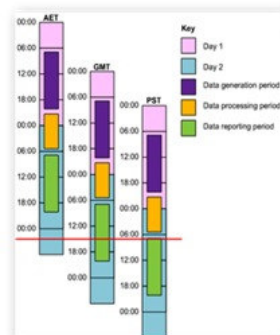


Figure 1: Multiple Time Zones Example

Day 1 starts first in Australian Eastern Time. Data is generated (via normal operational activity) during the working day and processed overnight ready for processing at around 07:00 AET on Day 2.

For the UK and the USA, the same pattern occurs but Day 1 starts later in each case. It is not until around 07:00 of Day 2 in Pacific Time that all the data for Day 1 operations becomes available for reporting. By this time a whole further working day has taken place in eastern Australia. From a global perspective some of the data in the warehouse will be nearly 2 days old...so much for overnight updates!

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#)

[Data Availability](#) , [Data Modeling](#) , [Data Warehousing](#) , [Deployment & Management](#)



Expand

Design Challenges in Global Data Warehousing

David Cox, Head of Delivery, Saksoft Europe
Added on Mar 05, 2010



In this situation the update strategy must be agreed with the users. There is a choice between updating global reports with data as it arrives from each geography and accept that the reports will change as new data is added or waiting until all the data for a given day is available before publishing reports.

Even if the latter strategy is chosen, it may become impractical to preserve the neat split between "update" and "query" modes. In our example, if we wait until all the data from around the world is available the earliest opportunity to update the warehouse will be overnight in Pacific Time. This is partly concurrent with the working day in both AET and GMT time zones when, presumably, some users will be querying previously loaded data.

Concurrent Updates and Queries

Assuming that query activity is always quite high somewhere in the world, methods have to be found to efficiently update the warehouse without unduly affecting query performance. There is no "magic bullet" for this issue; the right solution will depend on the characteristics of the data, the reporting environment and how much brute force in the form of hardware your budget will stretch to.

In some cases, depending on the various time zones involved and data volumes, it may be possible to find a good time to update the warehouse when query activity around the world is relatively quiet and when the business as a whole is prepared to accept no access or downgraded query performance. Be aware though that this window of opportunity may be very small and you cannot necessarily rely on all the incoming data arriving on time.

Updates to the data warehouse during working hours will inevitably conflict to a greater or lesser extent with query performance and some traditional loading techniques may become untenable. For example, it is common practice in many implementations to drop indexes prior to performing a bulk data load and then re-build them to support query performance. This technique is likely to cause performance problems for users if they attempt to run queries without the indexes being in place.

1 2 3 4 5 6

[Data Availability](#) , [Data Modeling](#) , [Data Warehousing](#) , [Deployment & Management](#)



Expand

Design Challenges in Global Data Warehousing

David Cox, Head of Delivery, Saksoft Europe
Added on Mar 05, 2010



Assuming updates to the database are unavoidable while it is being queried, it is generally better to update "little and often". In practice this usually means loading data as soon as it arrives (which may be almost continuously) even if you choose not to make such data available in published reports. With this approach all users tend to suffer about the same impact on query performance, rather than a specific group suffering disproportionately because updates always take place during the same processing window. It is also recommended that users have some way of knowing what data is available at the point when a report is run so they can correctly interpret query results.

Whilst your selected database platform may offer some features to help ease the conflict, the infrastructure on which the warehouse sits must be up to the job of simultaneously supporting reasonable query and update performance.

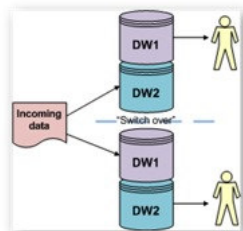


Figure 2: "Double" Data Warehouse

At the other end of the scale to the above approach is the "double data warehouse environment" in which two copies of the data warehouse are maintained. Whilst DW1 is being queried, DW2 is being updated. At a given point, updates to DW2 are suspended, the reporting tools switch over to DW2, automatically picking up latest data as needed. Meanwhile DW1 is updated. The two copies continually "leapfrog" each other in terms of being most up to date.

Clearly such a solution requires additional resources and some careful management to ensure the "switch over" does not disturb the end user experience.

1 2 3 4 5 6

[Data Availability](#) , [Data Modeling](#) , [Data Warehousing](#) , [Deployment & Management](#)



Expand

Design Challenges in Global Data Warehousing

David Cox, Head of Delivery, Saksoft Europe
Added on Mar 05, 2010



Multiple Currencies

Financial data sourced from local systems is typically provided in one or more local currencies. Global reports clearly need to represent such data in some standard way to allow meaningful analysis across currency boundaries.

In some cases data may already have been converted to a standard global currency at source and all the data warehouse has to do is include the converted data. More typically the data must be converted as part of the warehouse load.

The main issues that arise are: what exchange rate should be used and how should information about currencies and exchange rates be organised for reporting? This is not always as straightforward as it may appear. Let's take the example of some data that has been provided in the form of sterling and needs to be converted to dollars as this happens to be the global standard currency. One option would be to convert the data as part of the load and include it in a Fact table as shown in Figure 3.

Local Currency	Exchange Rate	Standard Currency
£12.56	1.628	\$20.45
£39.14	1.628	\$63.74
£100.00	1.593	\$159.30
£25.25	1.588	\$40.10

Figure 3: Currency Conversion in Fact Table

Whilst this approach is simple it suffers from a couple of disadvantages. Firstly the Exchange Rate is not an additive fact and arguably it belongs in a dimension table. Secondly, users may require the name of the local currency to be reported, so there would need to be a Currency dimension anyway. This could be a simple list of currencies but we may choose to include the exchange rate in the same dimension. However, we will need to agree with the business what exchange rate to use for each currency. Almost always, this rate is variable over time.

A common requirement is for a monthly exchange rate to be based on the average daily exchange rate throughout the month. This can work well but it introduces the problem of deciding what exchange rate to use whilst the current month is in progress, since clearly the average for the month can't be calculated until the month is over.

1 2 3 4 5 6

[Data Availability](#) , [Data Modeling](#) , [Data Warehousing](#) , [Deployment & Management](#)



Expand

Design Challenges in Global Data Warehousing

David Cox, Head of Delivery, Saksoft Europe
 Added on Mar 05, 2010



One option is to do away with monthly exchange rates altogether and alter the grain of the Currency dimension to be, say, daily. Whilst this obviously means the dimension has many more rows, it is still manageable. It may not suit the business however if historical data is normally expressed based on monthly average exchange rates. In this case you may need to include a slightly more sophisticated approach. For example the Currency dimension could contain both daily and monthly level exchange rates (see Figure 5). Data for the current month uses the daily level rows. At the end of the month, a new row is created (or an existing one amended) at month level showing the average exchange rate. The surrogate keys on the Fact rows that currently point to daily level dimension rows then need to be modified to point to the new month level row. Note that although this sounds like the Currency dimension includes mixed grains, expressing the period covered by a particular exchange rate as a date/time range avoids this. With this approach, any reports based on the daily level exchange rates would show a different result once the month had closed.

Currency Key	Currency Name	From Date	To Date	Exchange Rate
3674	Euro	1/1/10 00:00	1/1/10 23:59	1.587
3675	Euro	2/1/10 00:00	2/1/10 23:59	1.576
3676	Euro	3/1/10 00:00	3/1/10 23:59	1.459
3677	Euro	1/12/09 00:00	31/12/09 23:59	1.541
3678	Euro	1/11/09 00:00	30/11/09 23:59	1.404
3679	Euro	1/10/09 00:00	31/10/09 23:59	1.399

Figure 5: Currency Dimension with daily and monthly exchange rates

Multiple Languages

Global data warehouses often serve user communities which do not share the same language. This leads to the issue of how data should be reported. The data we are talking about here is really dimensional data, since in most designs the fact tables will consist largely of surrogate keys and numerical columns.

Some organisations standardise on a particular language (nearly always English) and all reports are expressed in this language even if the reporting tools themselves are localised versions. Typically with this approach standard dimensional data is managed centrally, rather than relying on local dimensional data to be supplied in the standard language or somehow translating supplied data before it is published in reports.

1 2 3 4 5 6

[Data Availability](#) , [Data Modeling](#) , [Data Warehousing](#) , [Deployment & Management](#)



Expand

Design Challenges in Global Data Warehousing

David Cox, Head of Delivery, Saksoft Europe
Added on Mar 05, 2010



This approach is simple but may not be acceptable to the business, particularly where there are large numbers of report consumers who do not speak the "standard" language. In these circumstances, dimensional data may need to be held in multiple languages with users able to select which language they prefer to report in. The key point with such an approach is that the choice of language should not affect the results; the same report run in two different languages should produce exactly the same figures.

The simplest and most reliable method of achieving this is to include specific columns in dimension tables for each language to be supported. For example a Calendar dimension table normally includes a column to store the name of the month. This column should be repeated for each supported language (See Figure 6). Reports which show the name of the month can be expressed in any of these languages without affecting the query results.

Date	Month English	Month French	Month Spanish
29/01/2010	January	Janvier	Enero
30/01/2010	January	Janvier	Enero
1/02/2010	February	Fevrier	Febrero
2/02/2010	February	Fevrier	Febrero

Figure 6: Date Dimension with Alternative Month Columns

Some dimensions will clearly be much more complex than this in terms of maintaining multiple languages. For example a large, complicated product hierarchy which uses many columns just for one language can result in a very large dimension table. However, once you have taken the decision to support multiple languages, this information has to be stored somewhere anyway and it is better to keep it logically as simple as possible.

With this approach some rows may not apply to all supported languages (perhaps, for example, because the product is not sold in all geographies). The columns that represent these languages should still be populated with a value which means "Not Applicable" but expressed in the language that is normally used in those columns.

Conclusion

The Data Warehouse is now capable of supporting analytical requirements on a global scale. Many organisations tend to approach the design of such systems from the perspective of their experience with local implementations. Key issues often arise that are not necessarily apparent at the local level. It is hoped that by highlighting the most common of these issues – time zones, currencies and languages - this article may prompt organisations to consider the issues in the context of their own designs and avoid having to re-engineer during development.

David Cox is the head of delivery in Europe for Saksoft, a specialist information management solutions, services and consultancy company . He has been implementing successful data warehouse solutions since 1990 across multiple verticals.