Unleash the power of self-service reporting on your Microsoft Dynamics AX 2012 deployment

- Six considerations needed for optimal results

1 INTRODUCTION

This whitepaper highlights some of the common issues encountered when utilizing self-service reporting tools on a Dynamics AX system. Common tools in this category include (but are not limited to):

- **Qlick (formerly Qlick View)**
- **Tableau**
- **Alteryx**
- **SiSense**
- **Targit**
- **Panorama**
- **Dundas**
- **Altair**
- **Microsoft PowerBI**
- **Microsoft PowerPivot**
- **Microsoft Analysis Services (SSAS) Tabular cubes**
- **Microsoft Datazen**

The subject matter is somewhat technical. However, the content – and language used - is suitable for all employees, including BI consultants and ERP consultants, as well as controllers, line-managers and executives, who wish to gain a deeper understanding of the topic. The paper also serves as a primer for anyone unfamiliar with either Dynamics AX or business intelligence (BI) processes.

The white paper does not address the overall business benefits of implementing Business Intelligence solutions or self-service data visualization. It is assumed that the reader is already aware of the benefits of such tools.

1.1 THE ERP DATA CHALLENGE (TABLES IN ERP VS. TABLES REQUIRED FOR REPORTING)

The ERP solutions available to companies and organisations often rely on complex and ever-growing data models. It is therefore natural that business managers long for tools to help them visualize their data in charts and dashboards. Aggregating and visualizing data generates clarity and insights that the ERP systems themselves seldom provide. Unfortunately, the complexity of the ERP data models makes it hard to pick the right data to analyse and visualize. The entire data model is optimized for other tasks, such as transactional processing (inserts/updates), and saving storage space (normalization). As a result, data needs to be transformed and simplified before it can be properly analysed and visualized.

Visualizing data is a lot easier than preparing it. In the case of Dynamics AX (both in 2012 R2/3 and the new Dynamics AX version released in 2016), there are between 3.800 and 9.000 data entities to choose from. Things that sound simple, such as a list of products, may in fact be based on more than 20 tables with intricate links and rules that define their relationships and behaviour. The complexity extends to the ERP dimensions (Financial, Inventory, etc.) that are arguably a cornerstone of ERP transactional data.

It is rarely sufficient to simply link tables together with a few key columns. Special attributes often mean that some data must be ignored (filtered) and database relationships are rarely defined in the database itself. Financial transactions is a good example. There may be opening and closing records that need to be ignored or filtered during reporting, in order to correctly aggregate (sum) the values. The vast majority of this complexity is handled by the application itself – often an application server – which means that to extract the data and analyse it, the rules and relationships need to be replicated during the export process.

The challenge, then, is to transform complex ERP data into something more easily understood and used by managers and reporting experts alike, while still adhering to the internal logic, rules and relationships of the ERP system.

1.2 GETTING DATA OUT OF AX 2012

There are numerous ways to extract data from Dynamics AX, depending on the scope and complexity. Each has their pros and cons:

Extract option	Pros	Cons
OData feed	• Easy to read in Excel and other tools • Adheres to security settings in the application.	• Large technical (protocol) overhead • Poor performance compared to other options • Read-only in AX 2012 • Limited CRUD support in New AX • Only pre-defined (based on AOT queries), though it can be customized.
Office ad-in (Excel)	• Data complexity is reduced • Adheres to security settings in the application • Respects business logic, also on inserts/updates.	• Only pre-defined (based on AOT queries), though it can be customized • Not feasible for large data quantities.
Copy/Paste data out of AX	• Easy for end-users.	• Not feasible for large data quantities • Requires AX and Excel on the same machine; works poorly or not at all in Citrix and Terminal Server environments • By default only visible columns in the list can be extracted • Not possible in new Dynamics AX.
Webservice API	• All database relationships and logic is handled by the Application server • Every scrap of data can be accessed, updated or inserted.	• Built for developers and system integrators • Requires expert knowledge.
AX standard OLAP cubes	• Relationships and logic is implemented in the cubes • Supports real BI analytics options and methods, such as MDX • Includes Time dimension.	• Hard to customize beyond basic additions of simple data • Many analysis tools simply “flatten” the model and export the data, making the cubes obsolete.
Direct SQL data access	• Unrestricted access to every scrap of data • Performs extremely well on large quantities of data.	• No relationships or business logic defined • No security restrictions apply • No clear documentation • Requires expert knowledge.

Even though there are many approaches, none of them are really optimal for self-service reporting. Companies and organisations need a new option that simplifies and prepares data for reporting in a way that combines performance and data scalability with adhering to business logic and security settings.

Traditionally, a Business Intelligence stack with a Data Warehouse has been that option. If properly implemented it has most (or all) of the benefits needed by the business. The issue is price!

Even with high volumes of data, it is perfectly possible to achieve real-time reporting and eliminate latency. The question is only if it is worth the cost. And the answer is, more often than not, a clear “no”.

1.3 EXTRACTING DATA VS. DIRECT ON SOURCE

Regardless of approach, the first thing to decide is whether to extract the ERP data to a separate database and then apply reporting- and analysis tools. The alternative is to let users read data directly from the source database using self-service tools. There are good reasons for choosing either approach.

- **Extract to a Staging area or Data Warehouse:**
  - Low load on ERP database.
  - Easier to enrich or combine with data from other sources.
  - Data can be prepared and structured for reporting, which is easier to consume by report authors.
- **Direct on source:**
  - Low latency.
  - Data is not prepared and structured for reporting, making it much harder for report builders to consume, but everything is there.

In reality, you will likely be using a combination of both (at least from a technical perspective). Some reports must be based on live data. This applies to things like invoices, sales orders, work orders and packing slips that must be current and compiled with zero-latency data. For the scope of this whitepaper, it is assumed that such reports are well covered already by your ERP system.

This leaves reporting scenarios that are less sensitive to latency and which require both aggregation and new user-defined combinations of data. The distinction is one of the things that separates “reporting” from “analysis”. In a BI analysis sense, a latency of a few hours or perhaps even a full day is often unimportant.

2 WHY DATA MODELS MATTER

If you were to ask a person who works with ERP or business reporting if understanding the data model is important, the answer would be a resounding YES. Still, poor understanding of or misconceptions about the data model are the most common reasons why reports are complex to build or contain grievous errors.

The complexity of the data model is often at the heart of the problem. It is an issue that is often ignored by self-service BI tools and ETL/DWA tools. In reality, no ERP database has data stored in the perfect streamlined design that demos often show. ERP databases are complex. There are technically sound reasons for the complexity, but they are all related to transactional processing. There is no reason to maintain this state of affairs when carrying out data analysis.

To make data available to report builders in an easily understandable form and also helps minimize the risk for errors and miscalculation, it has to be prepared and optimized for reporting. The best approach is to extract the data rather than accessing it directly on the source.

Loading can be handled automatically by the self-service tool once it is installed and directed to the database with the extracted data, and very little needs to be done to make the data available for reporting. The hard part is extracting and transforming the data into something easily consumable.

2.1 THE PROMISE OF SELF-SERVICE ANALYSIS

It is an interesting time for data analytics. More data has been produced and stored in the past two years than in the previous three decades combined. As a result, reporting complexity has exploded. Fortunately, new technologies and products have been able to mitigate most of this challenge.

One of the largest revolutions is self-service analysis. Gartner and IDC reports clearly underline how self-service analysis (or, as some prefer, “self-service data visualization”) is a multi-billion dollar business which promises massive change. In addition, it is here to stay.

Business agility and decision-making have never depended more on data analysis than today. At the same time, analysing data requires increasing levels of business understanding. Letting more people – from the business side rather than IT – interact with the data is much more than just a good idea. It is crucial for any modern business.

> *The shift in the BI and analytics market and the corresponding opportunity that it has created for new and innovative approaches to BI has drawn considerable attention from a diverse range of vendors. The list spans from large technology players — both those new to the space as well as longtime players trying to reinvent themselves to regain relevance — to startups backed by enormous amounts of venture capital from private equity firms.*
>
> *A crowded market with many new entrants, rapid evolution and constant innovation creates a difficult environment for vendors to differentiate their offerings from the competition. However, these market conditions also create an opportunity for buyers to be at the leading edge of new technology innovations in BI and analytics and to invest in new products that are better suited for Mode 2 of a bimodal delivery model than their predecessors.*
>
> *- Gartner Group, Magic Quadrant for Business Intelligence and Analytics Platforms, Feb. 2016*

Self-service analysis software promises to deliver agile and easy access to your business data. Self-service solution providers also promise that their products can read data directly from the source database. They even come with training in the form of webcasts, videos or live presentations and promise to make it easy to stich your data together and make flashy dashboards in just a few hours.

These promises are built on a false premise. The data models used by self-service providers to illustrate how great their product are od not match the reality of 2016. The data and data models might match a CRM system

from 2007, but not a system like AX 2012 with its thousands of tables and complex relationships. Even a trained and dedicated Qlick or Tableau consultant will struggle mightily to get correct results out of an AX database, if they have no prior knowledge of its concepts and design.

Invoice lines is a good example of what can go wrong. Here is what the self-service software sales person will do:

- Select the CustInvoice table. This is the Invoice header
- Select the CustInvoiceLines table. This is the Invoice line
- Link the two tables using the abc -> xyz columns
- Produce a report that shows the customer name from the Invoice header, and the sum of invoice line amount
- Reduce the customer field to a top-10 and make a nice chart to support the data
- Include a gross profit calculation as well, perhaps in a geo-map layout with bubbles in different colours and sizes for cities across the globe.

There are many problems with this approach – likely including errors in the data used. Most of the issues will not surface until the report and its data is used in a real business setting.

For example, consider the following:

- The real relationship between Invoice lines and the header is actually much more complex. In reality, the relationship is based on four different keys, plus Partition and Company, for a total of six keys. It is almost certain that a report constructed used the method above will contain false results
- Free text invoices are not included, leading to more potentially false results
- Simply including data from the customer name column can create problems. This column is just for printing and record keeping. It would likely be better to build a list of customers from the Customer table, and link it to the invoices. Otherwise, you will not be able to differentiate between billing customer, actual customer, etc. Linking to the Customer entity in AX means linking and going through approx. seven tables, as these are stored in the Global Address table and not in an easily accessible customer list
- Some self-service products do not support relationships based on multiple keys. If vendors try to read directly from AX, it will end with wrong results.

There are also other problems that are complex to solve in self-service tools:

- AX separates data for separate business entities with the “company” column. For data to be correct, the company relationship must be established for each table. AX supports siloing of multiple business entities in “partitions”, which can lead to a lot of clutter in the data model
- Financial dimensions, which are arguably a corner stone of ERP, are not present. However, they are not just a simple column or link you can add to your reports
- Some data in AX has validity date ranges. In other cases, the application is logging prior versions of all records. In such cases, you will need to carefully filter the data to exclude historic records (which are not current and produce wrong aggregations). You also need to decide if you want only the last valid record, or a record for each valid period. Simply adding a table with these properties to a report is all but guaranteed to produce erroneous results.

These issues do not equal that that self-service reporting solutions are not valuable tools. Putting the report-authoring capabilities in the hands of the business instead of a small handful of experts is definitely a boon for businesses. However, vendors keen to sell their products will at time sweep issues like those mentioned above under the rug.

This is a shame, as the self-service tools definitely have a lot of value, even when taking the real level of back-end complexity of data into consideration.

2.2 WRONG RESULTS WORSE THAN NO RESULT

The biggest issue when making reports and dashboards is that the issues mentioned in the previous section can lead to erroneous results that lead to bad decisions. They also stop businesses looking for the correct results, because they believe that already have them.

Getting wrong results is more likely than most people realize. Companies should be wary of software vendors that claim selecting and exporting data from AX (or indeed any modern ERP system) is “easy” - especially if a user or report author has to pick the right tables by him/herself. Here are a few examples of what can go wrong:

- Selecting the wrong table when near-identical tables contain data that looks correct to the untrained eye. In AX, this could be picking Customer Transaction records or Invoice header records for turnover reports instead of Invoice Lines
- Not adding multiple related tables together into a single data set. In AX, this could be merging free-text invoices with normal invoice lines, or combining different Project posting lines to get the complete set of project transactions (these are in three separate tables)
- Not joining related tables correctly. This could lead to getting more records than you should (sometimes called “exploding” data sets or a Cartesian product) or less data than expected
- Not handling data correctly when it has start/end dates. A common example is the Worker table
- Not realising when you need to configure a slowly changing dimension (AX favours Type 2 style configuration, but your analysis software may work/need a different approach) when you actually need historic records in your report.

Tools need to be able to read and parse ERP metadata (like AX AOT can, for example) to do this correctly. Letting report authors link ERP tables is generally not a good idea if they are untrained in handling SQL data.

3 SIX THINGS YOU NEED TO CONSIDER

So far, this whitepaper has identified many potential problems. This does not mean that the task of getting data out of AX is an impossible one. Far from it. Like many other IT-related matters, the maxim *"it's easy when you know how" applies*.

Carefully addressing the following six items will set you well on the way to a stellar performance with your data analysis on Dynamics AX.

- Read your data source correctly
- Simplify relationships in data
- Optimize your data model for reporting
- Push your data to the cloud / Utilize the cloud
- Keep your (vendor) independency
- Automate the process

3.1 READ YOUR DATA SOURCE CORRECTLY

As previously mentioned, it is easy to simply connect to an ERP database and sum up some figures. However, it is also the wrong approach. The complexity of the data model means this approach will often result in grievous errors.

To reduce / eliminate errors in your reporting output it is vital to read the data model correctly. You need *domain knowledge* about the system you are connecting to. In the case of Dynamics AX, most companies use one or more of the following options for ensuring correct data for reports:

- Having access to a highly skilled technical consultant or developer
- Looking up the data model and its various relationships in the application (AOT)
- Using an ETL / DWA tool that can read and parse the actual data mode (AOT metadata)
- Finding the appropriate information in technical documentation from the vendor (for AX, this would be Microsoft)
- Doing trial-and-error tests on the data itself, figuring things out on the go

The latter option sounds like the least attractive one, because making heads or tails of thousands of ERP tables is cumbersome. It is, however, also a widely used approach. Even large companies (Energy, Telco, and Insurance) use this approach. Logic dictates that this means that inaccurate business reports are not a rare occurrence.

For the data professional, the optimal solution is access a skilled technical AX consultant or developer. An alternative is using an ETL/DWA tool that reads and parses the actual data model (as opposed to just letting you select tables manually linking them). In you do not have access to such resources; you need to obtain the necessary knowledge yourself.

A software vendor like Microsoft provides various documentation, but the data model of their ERP databases is not something you will find in a single document. It is spread over many different technical documents. Things get even more complicated if you want to access data from customizations you have made to your ERP system. While your system integrator may have made some documentation relating to this, it will often not include technical details about how the tables in the SQL database are linked, or which fields contain what.

The morale of the story is that no matter the approach you take simply connecting a flashy piece of software directly to your large ERP system is all but guaranteed to be the wrong solution. If you do not have access to a tool that is capable of reading the technical definitions of the ERP data model, including your customizations, you will need to find/hire a specialist. This person or team will in turn need access to the appropriate documentation, or risk running endless trial-and-error runs until the results in your new report can be validated through other means/sources.

3.2 SIMPLIFY RELATIONSHIPS IN DATA

One thing that makes life a lot easier for report authors is consolidating relationship keys.

Technically speaking, consolidating relationship keys means reducing primary keys to a single column, and doing the same with foreign keys on all related tables. This ensures that when report authors are linking tables in their analysis tool, they will not need to define complex relationships with multi-column ("composite") keys. Instead, tables can be linked with just a single key column.

In fact, some of the major visualization tools on the market do not directly support multi-column keys, in which case making a consolidated key would be required anyway.

3.3 OPTIMIZE YOUR DATA FOR REPORTING

Gathering data directly from the main ERP database is seldom the optimal approach. It should be made easier to 'consume'. This means simplifying the data model and making everything more accessible to report authors. The concept of self-service means we have not yet identified specific reporting requirements. In reality, this means extracting a lot of data from different areas of the ERP database that we "suspect" will be needed by analysts.

There are many ways data can be optimized, but the primary ones are:

- Denormalizing tables to simplify the model
- Split tables into facts / dimensions to increase transparency and usability
- Rename tables so they follow a logical name tree

3.3.1 DENORMALIZATION

Denormalizing tables means consolidating columns from multiple tables into a single table. An example is making a list of products or a list of customers. The information we need is spread over 5-7 tables in Dynamics AX, but for reporting we would like that reduced to a single "Product" or "Customer" table.

The easiest approach is to start with the most detailed piece of information (usually also the table with the largest number of records) and then add columns from related tables. For invoice data, this would mean starting with invoice lines and then adding columns from the invoice header.

3.3.2 SPLIT TABLES INTO FACTS / DIMENSIONS

Splitting tables into facts and dimensions is an old BI concept. The idea is to divide the data into measurements and context to better understand it. Measurements are mostly numeric and are defined as facts. You can think of facts as something very concrete that occurred at a specific point in time. Facts are surrounded by different kinds of context that is broader and more verbose. The context defines circumstances related to the fact which can be divided into logical groups or lumps which we call Dimensions. If you consider invoice lines a fact, the line amount is the measurement and the context surrounding each invoice line, like Customer, Product, Time, Sales person, etc. are different Dimensions.

Dividing data into facts and dimensions makes everything more obvious and transparent for report builders. It also makes calculations and visualizations easier. While some tools can automatically present a single table as different measures and dimensions, properly preparing and building true dimensions is likely to yield better results.

Invoice data shows what can otherwise go wrong. Consider a customer that changes name or moves to a different address. In AX, this would result in Invoice headers with both the old and new name and address of the customer. Using the customer name column on Invoice headers when counting unique customers gives a wrong result because the old name and new name are seen as two different customers. If you have a single customer dimension, you would only have the most recent name, and only one occurrence of each customer.

Using Dimensions also simplifies your data model because attributes on dimension attributes only need to exist one time in your reporting data model. Your Customer dimensions can include details like the customer's name, address, phone number, zip code, etc. so that you do not need to denormalize this information onto every fact related to customers. In other words, building a Customer dimension eliminates the need to put customer data on Invoices, Sales orders, Shipping notes, ledger transactions, etc.

3.3.3 RENAMING TABLES

One of the old virtues in BI is putting well-defined and proper labels on *everything*. Most traditional OLAP cube servers, including Microsoft Analysis Services, have a Label layer where facts, measures, dimensions and attributes can be given nice and logical names that differ from the data model. This capability is often not available in self-service tools. The second best solution is to rename entities so their purpose and contents become clearer.

If you skip this, your reporting authors will see "DirPartyTable" instead of "Customers", "EcoResProduct" instead of "Product" and "HcmWorker" instead of "Employee". They can usually rename these in the final reports and dashboards but they will have to do that for each and every report they produce.

3.4 STAY AGILE IN YOUR APPROACH

The nature of self-service reporting means you are unlikely to have a well-defined list of requirements when you start building your analysis solution. This can be a challenge, and is also a good reason to pick an agile development approach over a classic waterfall model. In fact, a self-service analysis project could be considered the best fit for an agile development process precisely because the business requirements are largely absent at the beginning, and are often expanded and solidified gradually.

There are many benefits to an agile development process. Specifically applied to self-service data analysis the following are worth considering:

3.4.1 SPEED TO MARKET

Skip the lengthy requirement process and proceed straight to using the strengths of the self-service tools; which is discovering things you did not know about your company to begin with. It enables you to discover answers to important questions nobody has been able to formulate, and lets you share them with your organization faster than before.

3.4.2 QUALITY

In agile development, one of the key principles is that testing is integrated throughout the life cycle. This means you will regularly be inspecting the results you produce, resulting in earlier discovery of quality issues.

3.4.3 VISIBILITY

The whole process is highly collaborative and encourages active user involvement to a much higher degree than traditional projects. This makes your progress and results more visible to stakeholders.

3.4.4 RISK MITIGATION

Small incremental releases means it is much easier to respond to changes in requirements or issues with data quality. Taking the necessary decision at the earliest possible opportunity means saving valuable time and resources.

3.4.5 FLEXIBILITY / AGILITY

In traditional project management, you write up-front specs and work hard to avoid changes (or keep them to a minimum). Agile development is different. Change is accepted, and in fact expected. This means you can respond much faster to changing priorities as well as making your priorities (backlog) much more visible.

3.4.6 COST / CONTROL

The combination of fixed timescale (iterations) and evolving requirements means it is easier to control cost and risk. The scope of the product is the variable rather than cost and resources.

3.5 PUSH YOUR DATA TO THE CLOUD

Depending on how you plan to analyse or share your data, it might make sense to migrate (copy) it to a secure database in the Cloud. This opens up a number of possibilities like accessing it from cloud-based tools like Microsoft PowerBI or sharing the data with different branches.

The obvious choice for Dynamics AX data is a Microsoft SQL Azure database. It is easy to configure, cheap to maintain and highly scalable. And best of all: traffic between Azure SQL and PowerBI is free – because it is all in the same infrastructure.

Many tools, including Alteryx and Tableau, can also connect directly to Azure databases, which makes it an easy way to share data across technological or geographical boundaries.

3.6 KEEP YOUR (VENDOR) INDEPENDENCY

Because the market for data analysis tools is in constant flux, it makes sense to keep your vendor independence as much as possible. Extracting your data from AX and preparing it for reporting in a generic way that is not tied to a single software product reduces your company's risk when selecting an analysis tool. It ensures that you do not need to do that part of your work twice if you decide to replace one self-service tool with another. It also allows you to more easily try cutting-edge tools in a bid to stay ahead of your competitors.

3.7 AUTOMATE THE PROCESS

To ensure the stability of your solution, you need to automate the data transfer. It is usually not hard to do because virtually all tools support scheduling of batch jobs. Still, it is something that needs a bit of planning and perhaps monitoring so you know if things are working as expected.

While it can be tempting to update the data as often as possible, there are a few caveats to consider before setting the scheduler to run every 15 minutes.

- Bad things can happen if a new batch update starts before the previous one has finished
- It taxes the ERP database server
- The business might not need it
- Some reporting solutions are "offline" while data is refreshed. Evaluate the capabilities of the software you work with before updating the data too frequently.

This is not to say that you should not update four times every hour if you can. It just means there are reasons why that might not be as straight forwards as it sounds.

4 OTHER CONSIDERATIONS

This whitepaper has not mentioned a few complex topics because they are either fully automated or completely unsupported by most self-service tools.

4.1 DELTA LOADING

The concept of extracting or loading only data that has changed "since last time you read it" is often called a *delta load*. As a concept "delta", or Δ, is the mathematical concept of "change" between two figures (as well as being the fourth letter of the Greek alphabet). In reporting this means determining which data was created, altered or delete since your last extract.

Generally speaking, delta-loading is only needed if the amount of data you handle is in the millions of records. Anything lower than that can easily be handled by contemporary self-service tools in under 60 seconds. Since re-reading data is simpler to both implement and manage, you should avoid delta loads unless you have specific reasons not to.

If you think (or know) you need delta-loading, you should contact a dedicated BI specialist or look for an ETL tool that automates this process.

4.2 TRACKING CHANGES

Sometimes noticing the fact there has been some kind of unspecified change is not enough. Knowing specifically *what* has changed could be important. It sounds easy when your manager asks for "a report of deleted customers" or "list of employees that have changed name in the last two years", but in terms of data, it can quickly get messy.

Keeping track of data that has been deleted in your source database requires planning and care – because without retaining a copy of previously extracted data you have no way of knowing what was deleted from the source.

Likewise, monitoring and reporting on changes to existing records means generating extra data lines to represent the changes, or keeping copies of your older extracts.

If you need to make such reports, consider getting in touch with a BI specialist, or be sure your self-service analysis tool supports such change tracking. It is not something you should start reporting on without understanding the challenges involved.

4.3 LOW-LATENCY DATA TRANSFERS

As a BI specialist I have often come across customers desire to have "real time data" in their reports and dashboards. While the wish is completely understandable, it is worth noting that very few business areas actually need real-time data.

The few common exceptions are sensory data, machine monitoring and IoT devices, but there are (yet) still very few companies actually dealing with that kind of data in relation to Business Intelligence.

Often the concept of *"real-enough time"* is sufficient because advanced business analysis does not depend on the one hundred transactions made during the last hour.

In most businesses, the average turnover for top-ten customers/products/projects is unlikely to change significantly with half a day's worth of data. And truth be told, if that weren't the case then managers would know about it without reporting because the cause would be a major deal just landing, or a major account gone bust.

That is not a compelling reason not to report in real-time, though. If the technical implementation of both methods could be achieved with the same amount of work, we would not even consider batch processing. The real reason is cost-benefit-analysis (CBA). Handling real-time data requires more planning, better servers, better network infrastructure, and more expensive software because real-time processing is an “Enterprise” feature.

If you believe your business needs real-time analysis, contact a BI specialist and do a CBA to determine if the cost is worth it.

4.4 BECOMING YOUR OWN BI SPECIALIST

If contacting a BI specialist is out of the question be sure to drop by your favourite bookstore and buy a copy of “The Data warehouse Toolkit” by Kimball Group. The book, currently in its third edition, is generally considered The Holy Bible for SQL data analysts. Even though modern BI software has made some parts of the Kimball method obsolete it is still highly relevant.

You can use it as a reference rather than reading it from beginning to end. And let us be honest: the parts of that book that are no longer relevant to you are still highly relevant to the developers of the BI software you buy…

http://www.kimballgroup.com/data-warehouse-business-intelligence-resources/books/data-warehouse-dw-toolkit/

5 CONTACT INFORMATION

SCALES A/S
Automatikvej 1
DK-3200 Søborg
DENMARK

www.scales-group.com

Jakob Bjerg-Heise, Principal BI

Mobile: +45 5358 8495
jakob.bjerg-heise@scales-group.com

SCALES is a business and technology service company. SCALES provides business consulting, systems integration and solution implementation based on the Microsoft platform and covers ERP and Infrastructure.

More information is available at www.SCALES-group.com