SAP HANA and Data Virtualization: Competitors or Complements?

A Technical Whitepaper

Author:
Rick F. van der Lans
Independent Business Intelligence Analyst
R20/Consultancy

September, 2012

Sponsored by

COMPOSITE
— SOFTWARE —

Note: I would like to thank the following reviewers for their comments and suggestions: Mike Ferguson of Intelligent Business Strategies, Chris Hallenbeck of SAP AG, Peter Tonies of NXP, and Jeroen Vermunt of Eclectic International Consultin. Evidently, they are not responsible for any inaccuracies in this whitepaper.

.

Table of Contents

1 Summary

Since the turn of the last century, the market of database servers has changed and evolved dramatically. Many new database servers have been introduced of which a large portion of is aimed at supporting business intelligence (BI) workloads, in other words: reports and analytics. Some use new storage formats to speed up query performance, others implement an in-memory oriented approach, and some are based on an integration of software and hardware (the so-called data warehouse appliances).

SAP HANA, introduced in 2011, is one of the most recent of those database servers. It's a powerful and feature-rich, in-memory, SQL database server. It fully exploits the memory and caching features of today's processors. In the beginning, the name HANA stood for High-Performance Analytic Appliance. Today, HANA is just a name and not an acronym anymore. HANA is SAP's answer to the ever-increasing BI reporting and analytical needs of organizations: queries have to run faster, more queries have to be executed, more users have to be supported concurrently, queries are becoming more complex, queries have to analyze larger quantities of data, and users want to analyze operational data.

Besides being a database server, HANA comes with a large set of products, including data integration tools and with that enters the realm of data integration technology. In the whitepaper this is referred to as the HANA environment. Another data integration technology that has become quite popular is *data virtualization*. Data virtualization is data integration technology that uses on-demand data integration to offer applications a unified view of a heterogeneous set of data sources.

Because the HANA environment as well as data virtualization offer data integration capabilities, some organizations ask themselves the question whether they should select HANA or data virtualization for implementing their data warehouse or data mart? Understandably, to others it feels as if these customers are comparing apples and pears. But they are not. There are enough characteristics that the two technologies have in common, that justifies such a comparison. For example, both can be used to integrate data from different data stores, both can be used to simplify the architectures of BI systems, both make operational BI a reality, and both make BI systems more agile. Still, the two are also sufficiently different. Where HANA can be summed up with the terms *performance booster* and *agility*, data virtualization can be characterized with the terms *data integration* and *agility*.

Deploying both technologies in a BI system results in one of the fastest and also one of the most agile BI architectures that can be designed with modern-day technology. Therefore, architects are recommended to study both technologies.

This whitepaper explains both technologies briefly and clarifies the commonalities and differences between these two promising technologies. Additionally, the advantages of combining the two are described. A large section is devoted to describing four different BI architectures in which HANA and data virtualization are deployed and complement each other.

2 What is SAP HANA?

HANA, which was released in the first half of 2011, is a SQL-based database server. It can be accessed by a wide range of reporting and analytical tool. Besides the support for queries, like any other database server, it supports inserts, updates, and deletes. In fact, it offers all the features one expects from a modern SQL database server.

What's special about HANA is that, from the ground up, it's designed to be an *in-memory database server*. This means that it loads the entire database into memory and processes all database operations without doing any I/O. The data loaded in memory resides in HANA's own data store, and new data entered in memory is eventually written to that data store. Keeping the data store in synch with the data in memory is handled by background processes. These processes don't interfere with the queries and transactions. The overall architecture of HANA is represented in Figure 1.

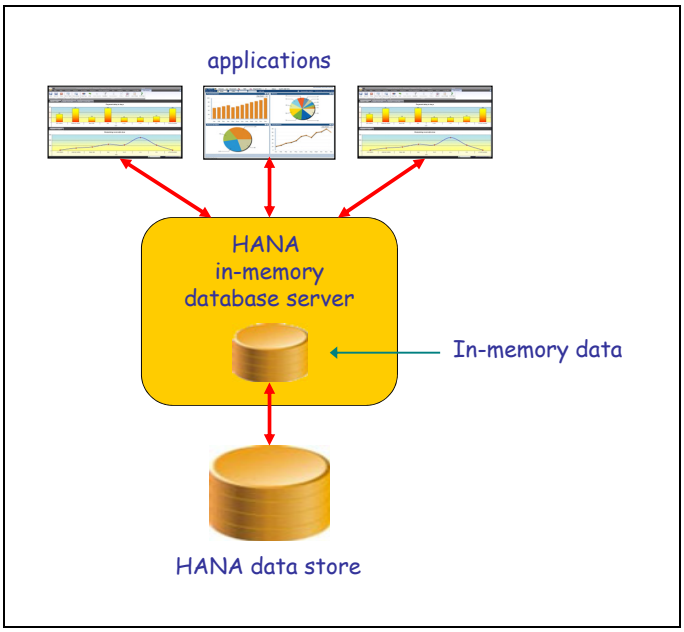


Figure 1 *The high-level architecture of the SAP HANA database server; all the data is loaded in memory and the data store is used for storing data permanently*

There are other in-memory database servers on the market, but only a few exploit the power of the on-board caches of processors of which HANA is one. Accessing data in caches can be significantly faster than accessing data kept in memory.

The HANA database server comes with a large set of additional tools, such as tools for ETL, statistics, and replication. Some are developed specifically for HANA, others are existing SAP products (sometimes slightly adapted for HANA), and there are products from third parties. To avoid misconception, in this whitepaper a distinction is made between the *HANA database server* and the *HANA environment*—when the database server is intended, the short term HANA is used.

The following aspects make HANA unique:

- The entire HANA database is loaded into memory which has the effect that all the queries are processed by retrieving data residing in memory—no I/O is involved.

- Based on the workload, DBA's can choose whether the data in tables should be stored in record-based or column-based fashion.

- To be able to load as much data in memory, SAP compresses the data. The advantages of compressing data is that it speeds up I/O when loading data into memory, in addition, compressed data requires less space in memory. SAP indicates that an average compression ratio of 6 can be achieved (evidently, this depends on the characteristics of the data).

- The size of internal memory determines how much data can be loaded. Note that HANA needs 50% of the memory as a workspace area for internal processing, such as for joins and sorts. According to John Appleby[1], determining the size of memory can be done as follows: "[Take the size of your current SQL database] without the indexes and BW dimension tables and aggregates. Divide this by 5 to get the SAP HANA database size, then double that number to include processing space. For example if you have a 5 terabyte Oracle BW database, which is 3 terabyte of used space and 2 terabyte without indexes or dimension tables and aggregates, you need 820 gigabyte of [memory]." Note that the "size" of the machine has a major impact on performance.

- HANA has a very strong view mechanism. As with almost every SQL database, classic relational views can be defined, but HANA also provides the ability to define multi-dimensional views for MDX-based applications. Because views point to tables stored in memory, accessing them is very fast even if the query is complex.

- Besides being able to run queries fast, HANA can be used for OLTP-type (online transaction processing) applications. It supports an ACID-based[2] transaction management mechanism which means that when data is committed, it's safe. HANA is not like some of the new NoSQL products that support transaction management mechanisms that are aimed at increasing scalability and not so much at safeguarding transactions.

- To load data from other data stores, the HANA environment comes with data integration tools for extracting and transforming data from other database servers and data sources. SAP delivers ETL-style and replication-style solutions for loading data into HANA. The ETL module is formed by SAP BODS product (Business Objects Data Services; formerly called BODI Business Objects Data Integrator). Replication technology is supported to replicate within milliseconds new data from operational databases to HANA. This opens the way for reporting and analytics on operational data, i.e. *operational BI*. Together, these tools make HANA a data integration platform. Data from various data sources can be extracted, transformed, aggregated, and cleansed, before it's loaded.

[1] John Appleby, The SAP BW on HANA FAQ, www.bluefinsolutions.com, May 8, 2012.
[2] http://en.wikipedia.org/wiki/ACID

- Besides all the standard features to be expected from a database server, the HANA environment also supports many other features, including R for statistical analysis, an ODATA interface, and integration with Hadoop.

- HANA is currently available on Intel-platforms only. The following vendors are offering a certified platform for HANA: Cisco, Dell, Fujitsu, Hitachi, HP, and IBM. HANA is optimized for scale-out to hundreds of nodes.

3 What is Data Virtualization?

Data Virtualization is Data Integration - Data virtualization is not database technology, but technology for integrating, transforming, and cleansing data coming from all kinds of data stores and presenting all that data as one unified view. When data virtualization is applied, an abstraction and encapsulation layer is provided that, for applications, hides most of the technical aspects of how and where data is stored; see Figure 2. Because of that layer, applications don't need to know where all the data is physically stored, how the data should be integrated, where the database servers run, what the required APIs are, which database language to use, and so on. When data virtualization is deployed, to every application it feels as if one large database is accessed.

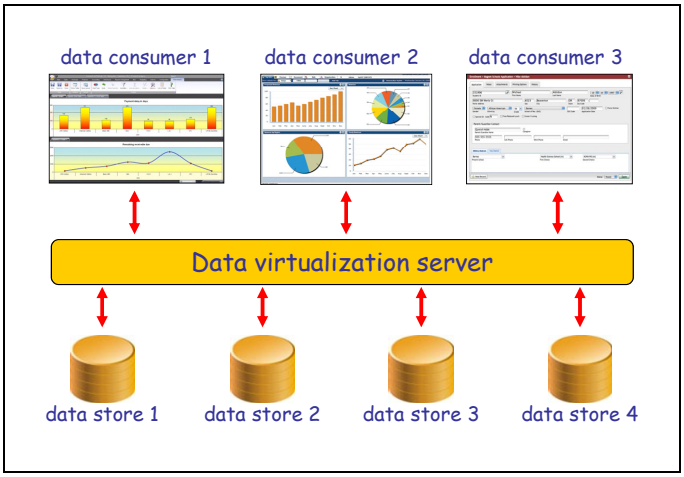


Figure 2 *When data virtualization is applied, all the data sources are presented to the data consumers as one integrated data source*

In Figure 2 the terms *data consumer* and *data store* are used. The neutral term data consumer refers to any application that retrieves, enters, or manipulates data. For example, a data consumer can be an online data entry application, a reporting application, a statistical model, an internet application, a batch application, or an RFID sensor. Likewise, the term data store is used to refer to any source of data. This data source can be anything, it can be a table in a SQL database, a simple text file, an XML document, a spreadsheet, a web service, an index sequential file, an HTML page, and so on.

The definition of data virtualization:

> *Data virtualization is the technology that offers data consumers a unified, abstracted, and encapsulated view for querying and manipulating data stored in a heterogeneous set of data stores.*

So, even if the data stores in Figure 2 use different storage formats and technologies, to the data consumers, the data virtualization layer presents them all as one integrated set of data. In addition, different languages can be used to access the same data.

With other data integration technologies, such as ETL, ELT, and data replication, the result of data integration, cleansing, and transformation is stored before it can be used for reporting. Data virtualization on the other hand, integrates, cleanses, and transforms data *on-demand*. In other words, when the data is retrieved, only then is data processed. The result is not stored, but passed to the reporting application.

Technically, when reports access data through a data virtualization server, they see tables (note that data virtualization servers can present the same data as other concepts, such as XML documents). In other words, for a reporting tool there is not much difference between accessing a database server or a data virtualization server. However, the data accessed in a data store is really stored, whereas the data in a data virtualization server isn't. The data is retrieved from the real data stores when the report (data consumer) asks for it. Data virtualization *means on-demand data transformation*, *on-demand data integration*, and *on-demand data cleansing*. The term used for tables defined in data virtualization servers is therefore *virtual table*.

Query Performance and Caching - An important instrument of data virtualization servers to improve query performance is *caching* of the virtual tables. For each virtual table a cache can be defined that holds the result of all the data integration work. Queries accessing a cached virtual table are no longer accessing a data source, but the data in the virtual table's cache; see Figure 3. If no cache is available, all data integration, transformation, and cleansing operations are executed real-time, which can slow down the performance of particular queries. In addition, every time when the query is executed the same operations are executed.

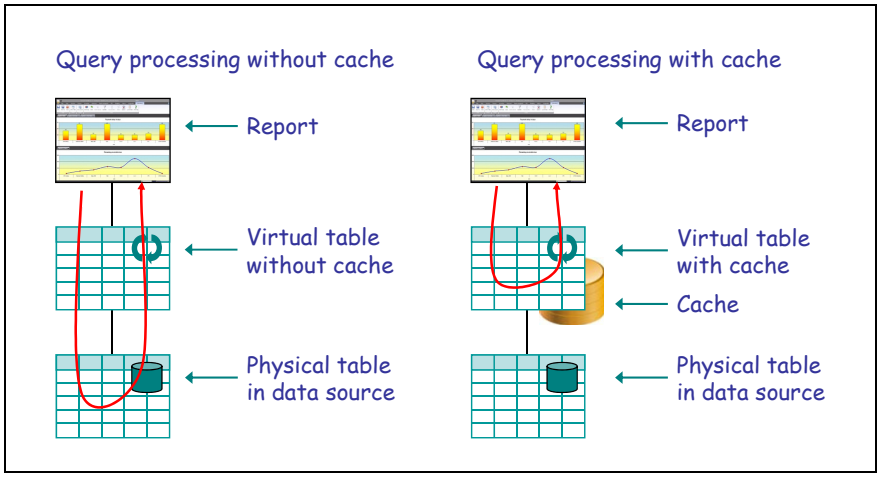


Figure 3 *When a cache is defined for a virtual table, the physical tables are not accessed*

The word cache may lead people to think that cached data is stored in memory, but for most data virtualization servers that's not the case, instead they store the caches in files or in the tables of a database. In both cases the data is stored on disk.

Improving query performance is an important reason for developing a cache, but caching can be used for other reasons as well, including load optimization, consistent reporting, increasing data source availability, complex transformations, and data security aspects.

From Data Federation to Advanced Data Virtualization - In the beginning, data virtualization products were primarily run-time products, used for making it easy for applications to access multiple data sources; in other words, they primarily offered *data federation* technology. The last few years these products have been extended to support more activities of the entire system development life cycle, such as data modeling, data profiling, data governance, data quality enforcement, lineage analysis, and collaborative development; see Figure 4. This was the main reason to change the name of this category of products from data federation servers to data virtualization servers, and nowadays the term *advanced data virtualization server* is used more and more often.

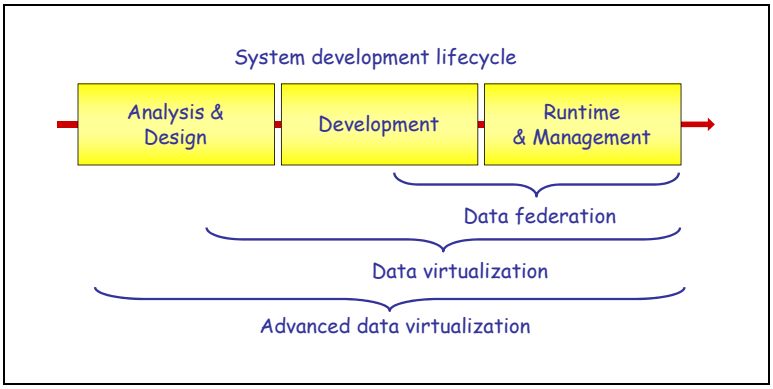


Figure 4 *Data virtualization servers have been extended to support the entire system development life cycle—they evolved from straightforward data federation servers to advanced data virtualization servers*

Note: For more information on data virtualization we refer to the book *Data Virtualization for Business Intelligence Systems* [3].

4 Differences and Commonalities

From the above two sections can be derived that the two technologies have several commonalities but also noticeable differences. In this section these are divides in four groups.

Data Integration and Data Quality

- **Data Integration:** Data Virtualization and the HANA environment both offer extensive data integration features. The HANA environment offers additional replication or ETL tools for loading and integrating data from various data sources. For data virtualization, data integration is not an additional feature, but data virtualization *is* data integration. Understandably, it supports data integration for a wider set of data sources, ranging from simple Excel files via web services and HTML websites, to SQL database servers and NoSQL platforms. The key difference between the two is that with data virtualization no need exists to store the result of data integration before it can be used, whereas with HANA the result is preloaded into memory (and stored in the HANA data store). However, after the data has been loaded, HANA doesn't need to re-integrate data for each query which speeds up query processing.

- **Data Federation:** Never will all the data of an organization be stored in HANA. There will always be data stored in other data sources. If reports need to combine data stored in

[3] Rick F. van der Lans, Data Virtualization for Business Intelligence Systems, Morgan Kaufmann Publishers, 2012.

HANA with data stored in other data sources, the reporting tools will have to do their own data federation. Combining data from different data sources, including HANA, is a core feature of data virtualization servers.

- **Data Quality:** Data virtualization servers support data cleansing operations that can be invoked when integrating data. HANA itself doesn't do any data cleansing, however, SAP's ETL tool BODS does support data cleansing. General comment: For some of the architectures presented in the next section, the recommendation for both technologies is to do data cleansing as much as possible *upstream*: as close as possible to the source.

Performance and Architectural

- **Virtual Tables:** With both technologies virtual tables can be developed (called views in HANA and in Composite's data virtualization server). By defining virtual tables, the right table structures can be developed for the reports. In both technologies the contents of the virtual tables is not stored. Their virtual content is derived when the virtual tables are accessed. When HANA views are accessed, it leads to access of data kept in memory, not to I/O. The virtual content of data virtualization tables is derived by accessing the underlying data stores (unless a cache is defined) and results in I/O.

- **Query Performance:** For a data virtualization server it's the underlying database server and network topology that predominantly determine its query performance. HANA's data is memory-resident, is stored in an optimized format, and it has full control over performance. The query performance of HANA using in-memory data is better than that of a data virtualization server using a classic I/O-based database server with no caching of the virtual tables. The performance of data virtualization can be improved when caches are defined. Even then, data is fetched from a disk-based cache. Note that a data virtualization server can use HANA as database server and thus raising its performance level to that of HANA.

- **Large Data Sets:** By leaving the data in its data source, data virtualization can operate on extremely large data sets. Even, for example, massive amounts of sensor data (big data) stored in Hadoop, can be accessed through data virtualization. Because of HANA's in-memory architecture, the amount of data it can operate on is restricted by the size of available memory. Currently, 100 terabytes is the largest certified configuration—SAP will certify larger systems on request. Note that loading 100 terabytes of data in memory requires a significant I/O operation, when only a small subset of that data may actually be necessary to satisfy the vast majority of queries.

- **Query Workload Distribution:** With data virtualization the real database workload can be distributed over multiple database servers depending on the location of data. With HANA all queries are executed by the same database server.

- **Transaction Processing:** As indicated, HANA supports high-speed transaction processing, whereas some of the data virtualization servers do not (yet) or only under certain conditions. This makes HANA more suitable for providing data to production applications. Its internal architecture also makes it possible to merge production databases and reporting databases on one HANA platform for BI reporting.

- **Software/Hardware Solution:** Data virtualization servers are a software-based solution. They can run on most classic server platforms. HANA is software and hardware. It comes with its dedicated server machine(s).

Design and Development

- **Modeling and Design:** Data virtualization servers support rich design and development environments. For example, data modeling capabilities and on-demand data profiling are supported, lineage and impact analysis is offered to understand all the relationships between the objects, and some products even support special designed, non-technical modules that business analysts can use to make collaborative development a reality. HANA, on the other hand, is primarily a run-time engine with limited support for the modeling and design stages of the system development life cycle. Some of the other products that are part of the HANA environment offer those features, such as the ETL tool BODS (for lineage) and SAP Sybase PowerDesigner (for modeling). However, these are separate products, whereas with data virtualization that functionality is built in. To summarize, HANA is an environment for developers, the HANA environment supports tools for system analysts and designers, and data virtualization server is for system analysts, designers, developers, and business analysts.

- **Report Migration:** HANA is most valuable when it replaces existing databases. But replacing databases requires a migration project. For example, the performance of a data mart can be improved by replacing it by one developed with HANA. However, this requires that the existing data mart implementation is dropped and a new one is developed. Data has to be unloaded from the old data mart and reloaded in the new data mart. The old loading programs may have to be rewritten as well, because the tool currently used may not support loading into HANA. Data virtualization does not require a migration exercise. Deploying data virtualization is a much more evolutionary approach, whereas HANA is more revolutionary.

- **Importing New Data Sources:** With data virtualization it's easy to make new data sources available to the reports. The only thing to be done, before reports can be developed on the new data, is that the data source has to be made known to the data virtualization server. With HANA, if new data has to be made available, that data has to be loaded in the database which involves more work. If the new data source is large, it may even require an upgrade of the hardware. An ongoing refresh process must also be defined, implemented, and managed to ensure the data in HANA is kept up to date, while a data virtualization solution accesses the most current data.

- **Prototyping:** Prototyping of new reports for which new data is required is easier to do with data virtualization than with HANA. With data virtualization only the virtual table that points to the new data source has to be defined. After the prototype has shown that the report can be used by the users, to speed up queries on the virtual tables, the decision can be made to create a physical data mart, to enable caching, or to deploy another optimization technique. But whatever technique is selected, the change is transparent to the reports. With HANA the data has to be loaded and stored again. This takes more time. Changing it to another solution later on is not straightforward.

- **Security:** Data stored in a HANA database can be accessed by all users who can log on to HANA. Special privileges can be assigned to users to restrict data usage, but that has to be done explicitly. Privileges are not derived automatically from the data sources. Data

virtualization supports a mechanism called pass-through—when users access a virtual table that points to some physical tables, they need to have the proper privileges to access those physical tables. When virtual tables are cached, the security mechanism changes and becomes comparable to that of HANA. Explicit privileges have to be assigned or else users can access all the data in those cached virtual tables.

Reporting

- **Operational BI:** Both HANA and data virtualization allow for reporting and analytics on operational data (also referred to as real-time reporting, online reporting, operational reporting). HANA can take live streaming data from an operational system and support reporting on that data with no need for post-processing, such as indexing and building of aggregated data. Data virtualization supports operational reporting and analytics when virtual tables point to the production databases (note that is a design decision).

- **Analyzing Semi-Structured and Unstructured Data:** Both HANA and data virtualization support extensive text search capabilities and text-analysis features.

- **Statistical Analysis:** The HANA environment includes the popular and open source statistical platform called R. Current data virtualization servers do not support statistics, unless the database servers they're accessing support them and if the data virtualization server can invoke that logic. If not, the reporting tools themselves have to execute the statistical operations.

5 Deploying SAP HANA in BI Systems

This section describes three architectures for deploying HANA in a BI system:

- HANA as performance booster
- HANA as enterprise data warehouse
- HANA for implementing operational BI

HANA as Performance Booster - The first BI architecture for using HANA is as a performance booster; see Figure 5. Here, the existing BI environment, with its enterprise data warehouse, data marts, staging area, and personal data stores, stays unchanged and data that needs to be queried fast is copied into HANA. Depending on the data source, ETL or replication technology is used for copying. As indicated in Figure 5, data from any database (data marts, data staging area, or data warehouse) can be loaded in HANA.

Note that in this architecture the costs of acquiring, running, and managing HANA are added to the existing costs of the BI system. On the other hand, being able to run queries much faster, and to run more complex queries, must have a business benefit. It's up to each organization to evaluate the advantages and disadvantages.

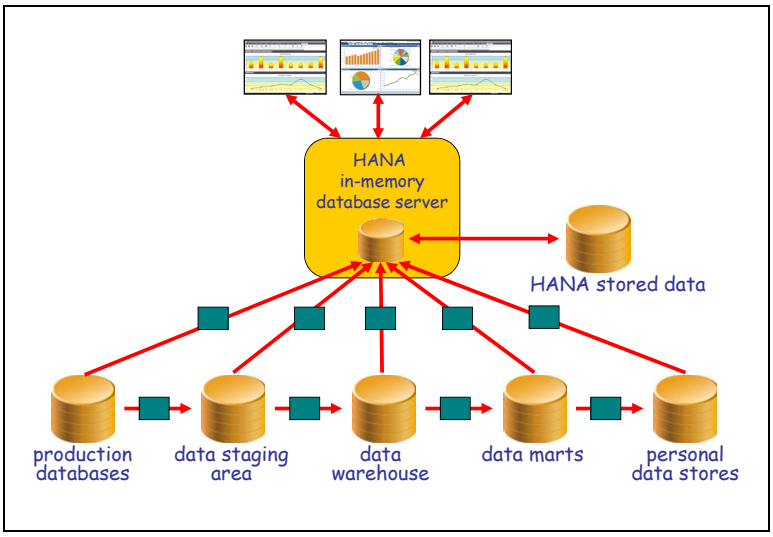
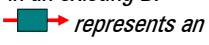


Figure 5 *HANA can be used as a performance booster in an existing BI system; the symbol*  *represents an ETL or replication process*

HANA as Enterprise Data Warehouse - In the second architecture HANA is used as a replacement for the enterprise database warehouse database plus all the derived databases, such as data marts and personal data stores; see Figure 6. In fact, all the databases whose content is derived from the enterprise data warehouse can be dropped. HANA simulates those databases using views and derives their virtual contents by accessing data in memory. These views together can be called *virtual data marts*. In this case, the reports still see the same database structures, it's just that the contents of those views are derived live from the original data. In this architecture the data warehouse resides in memory and is stored in the HANA data store.

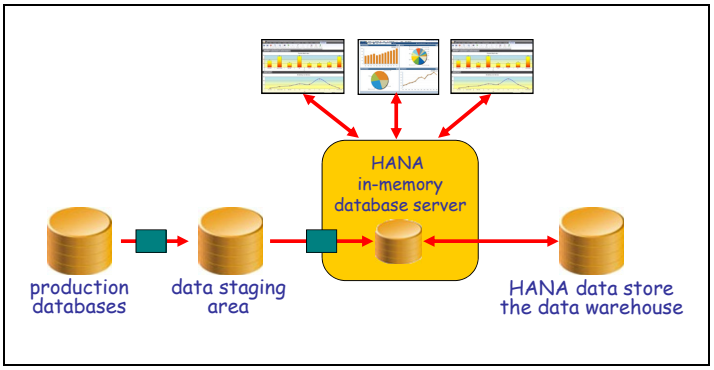


Figure 6 *HANA as a replacement for the enterprise data warehouse database and the derived databases, such as data marts and personal data stores*

This architecture is very flexible. If users want to change the virtual data marts and the required data exists in the data warehouse, the required changes are simple—no data has to be unloaded and reloaded. Only the view definitions have to be redefined. Compare this to having to change physical data marts. This may cost weeks of work. In this case, simple is more!

It's important to emphasize that in this second architecture, the data in a HANA database is not derived data, but it's the original data. The HANA database is *the* database, it's the data enterprise data warehouse.

HANA for Implementing Operational BI - In the third architecture, HANA is used as an *operational data warehouse*; see Figure 7. When new data is entered in the production databases, it's replicated to the HANA database. Because it contains up-to-date data, HANA can be used for

implementing *operational BI*. This need for reporting and analysis of operational data is growing rapidly within organizations.

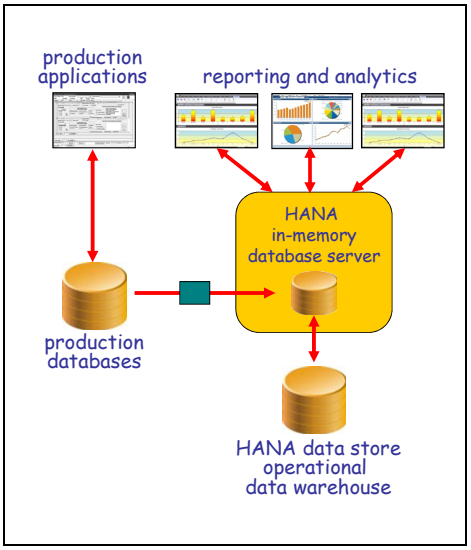


Figure 7 *HANA for implementing operational BI; new data is continuously replicated from the production databases to the HANA database*

In this architecture the operational data warehouse also keeps track of historical data. It's not just a replica of the production databases. For keeping track of history, the built-in features of HANA can be used.

Developing this architecture from scratch is not that complex, because the HANA environment comes with all the tools required to develop it, such as replication technology. However, most organizations already have a BI system in use, and merging this third architecture with the existing one, is not straightforward and probably leads to a somewhat hybrid architecture, which may come with additional costs of ownership.

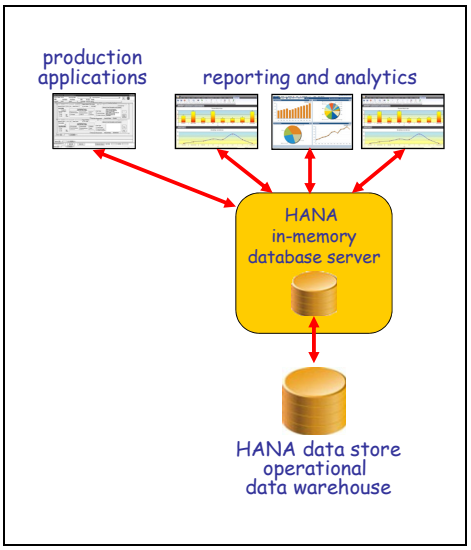


Figure 8 *Operational BI in which one database acts as production database and as data warehouse*

As indicated, HANA supports transaction processing. Therefore, it's possible to develop an architecture in which one HANA database acts as production database and as data warehouse; see Figure 8. In a way, this is an optimization of the previous architecture—it also offers operational BI. With respect to the number of components, this is the simplest architecture

possible. However, this architecture is complex to implement if the production databases already exist, and are developed using non-SAP solutions.

6 Deploying Data Virtualization in BI Systems

Figure 9 represents a BI architecture in which data virtualization is deployed. All the existing databases, such as data warehouses, and data marts, are present and the data virtualization layer provides access to all the data for the reports. Which databases are accessed is hidden for the reports by the data virtualization server. Due to this decoupling, it's easy to redirect a report from one database to another.

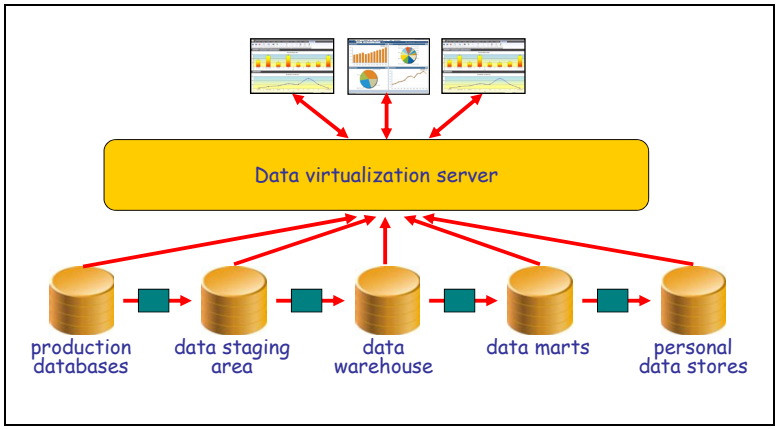


Figure 9 *The data virtualization server hides for the reports in which data stores the data is stored*

A simpler and more flexible architecture is one in which some (or maybe all) of the data marts and other derived data stores are removed; see Figure 10. In this architecture the data virtualization server simulates the data marts using virtual tables: *virtual data marts*.

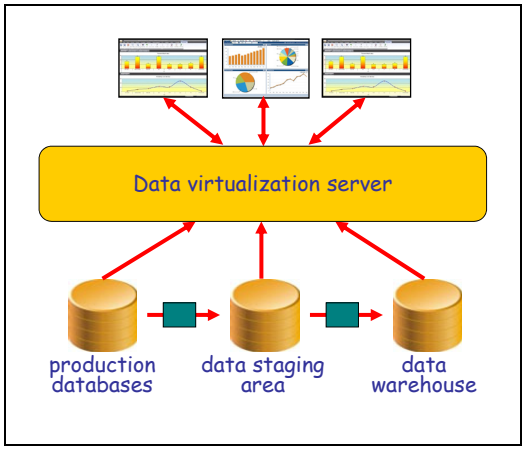


Figure 10 *The data virtualization server implements virtual data marts*

This architecture has many advantages, but the foremost one is *agility*. If the data marts have to be changed, only the specifications have to be changed. No unload and reload of data is necessary, nor do ETL scripts have to be changed. See the whitepaper *Data Virtualization for Business Intelligence Agility*[4] for the list of advantages of virtual data marts over physical data marts.

[4] Rick F. van der Lans, *Data Virtualization for Business Intelligence Agility*, www.compositesw.com, 2012.

7 Combining SAP HANA and Data Virtualization in BI Systems

Both technologies, HANA as well as data virtualization, bring value to the table. Also, both technologies have something in common; see Section 4. And when they are used together, they complement each other which leads to a powerful, fast, and agile business intelligence architecture. This section describes four architectures in which HANA and data virtualization are deployed together.

HANA as Performance Booster for Data Virtualization - In this first architecture all the data stores of the BI system are hidden by a data virtualization server and as much data as possible is loaded in a HANA database to speed up query performance; see Figure 11. Here, HANA speeds up query performance and data virtualization is responsible for data integration, development, and data governance tasks. Besides running the data warehouse and delivering high performance to the data virtualization server, HANA can be used by the data virtualization server for caching virtual tables. This architecture combines best of both worlds.

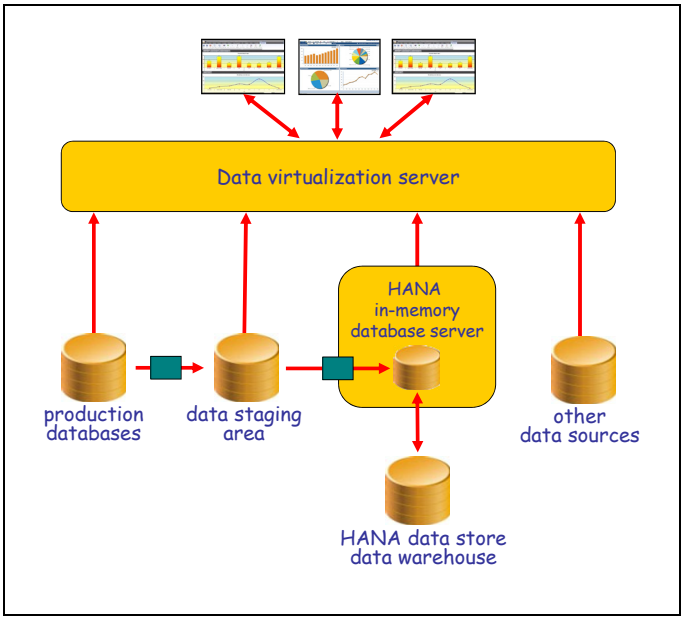


Figure 11 *In this architecture HANA is used to speed up query performance and data virtualization is responsible for data integration aspects*

In this architecture, the data virtualization server is able to access data sources from outside the data warehouse environment. It can integrate that data with the data warehouse data.

The diagram in Figure 11 doesn't show any data marts or other derived data stores. If required these can be added to the architecture. It's recommended, however, to minimize the number of databases in a data warehouse environment to keep it simple and flexible.

An advantage of this architecture is that customers can gradually migrate to it. They can start with a relatively small HANA server that stores a subset of all the data. When new data is loaded into HANA, the reports don't have to be migrated because they're accessing data via the data virtualization server. The latter hides that the accessed data is not coming from the original database anymore but from HANA.

In general, if for technical or financial reasons, data has to be migrated from HANA to another data source or vice versa, data virtualization servers hide the relocation of data. This gives DBAs

more flexibility over where to store data. They can even decide to temporarily move data into HANA for certain reports, and subsequently unload it. The reports won't notice this. All that relocating of data is transparent to them.

Data Virtualization as Data Integration Platform for SAP HANA – The HANA environment comes with several tools for integrating and loading data into a HANA database. A consequence is that all the integration specifications are managed by the HANA solution. Figure 12 shows an architecture in which all the integration of data from different data sources is not handled by HANA but by a data virtualization server. It integrates, transforms, and cleanses the data in a form required by HANA. In this case, HANA has only one data source as input. HANA can focus on supporting the reports and offering excellent query performance, whereas all the data integration aspects are centralized in the data virtualization server. The result is that both products do what they excel at.

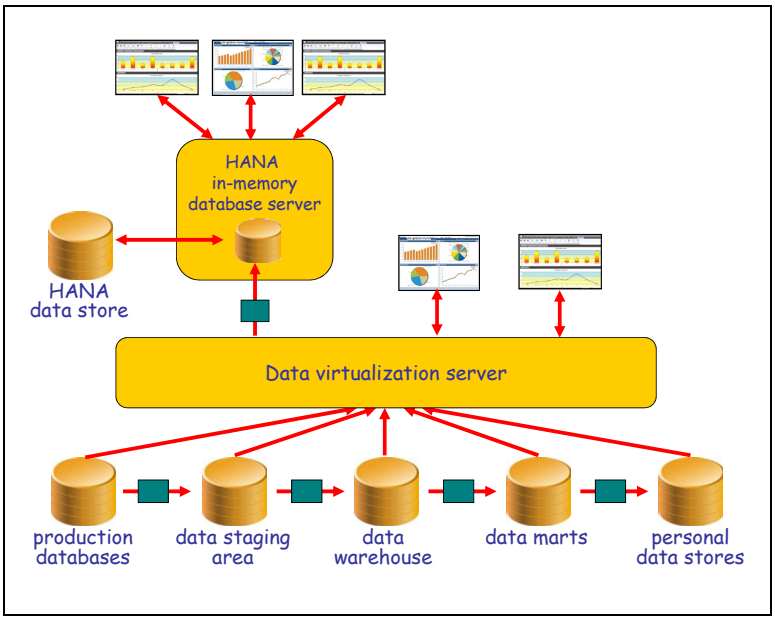


Figure 12 *In this architecture a data virtualization server is responsible for all the data integration work and HANA extracts data from it and is responsible for running all the queries*

To minimize costs, it's recommended to use the same ETL tool for loading the HANA database as the data warehouse and data marts.

An advantage of this architecture is that other applications and reports can also benefit from this data virtualization layer; see the reports in Figure 12 that access the data virtualization server directly. All the data integration specifications are shared by HANA and those other reports, improving data consistency and productivity of data integration.

SAP HANA for Caching Virtual Tables - As described in Section 3, data virtualization servers can cache virtual tables. The cached data is usually stored in files or in database tables. Access to those caches results in I/O. By making HANA the database server for storing the caches, the contents of the cached virtual tables becomes memory-resident—accessing them doesn't lead to I/O; see Figure 13. Undoubtedly, this speeds up access to the caches dramatically and thus the overall performance of the data virtualization server.

Extending an existing architecture with HANA in which data virtualization is already deployed, is seamless. No report has to be changed and no existing database has to be adapted. The only difference is that the reports run much faster.

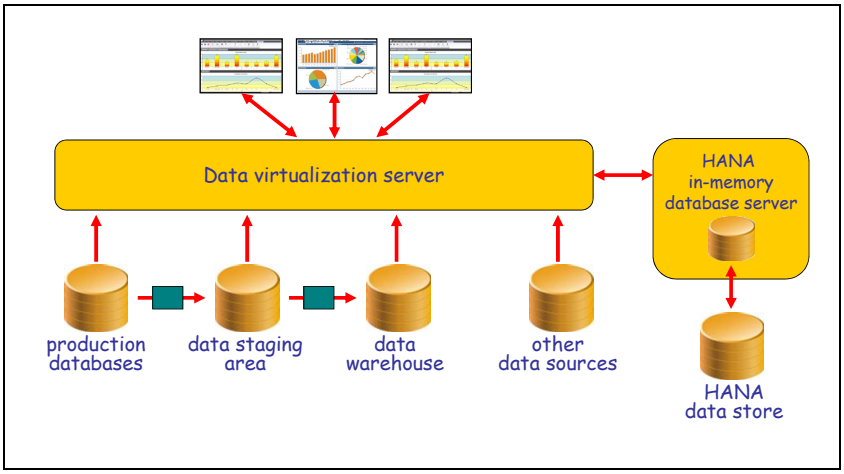


Figure 13 *In this architecture HANA is used by the data virtualization server to speed up access to the cached virtual tables*

SAP HANA and Data Virtualization Side By Side - The architecture in which the two technologies don't work together but operate side by side, is presented in Figure 14. In this architecture the two technologies process different parts of the query workload. For example, data virtualization is used for running the more temporary queries, such as the queries from a prototyping environment, the queries executed in a sandbox environment, and the queries that are only executed once or twice (sometimes referred to as disposable reports). Whereas, HANA is used for running the more traditional reports that are used frequently. In this situation, the value of HANA is that those reports are executed faster.

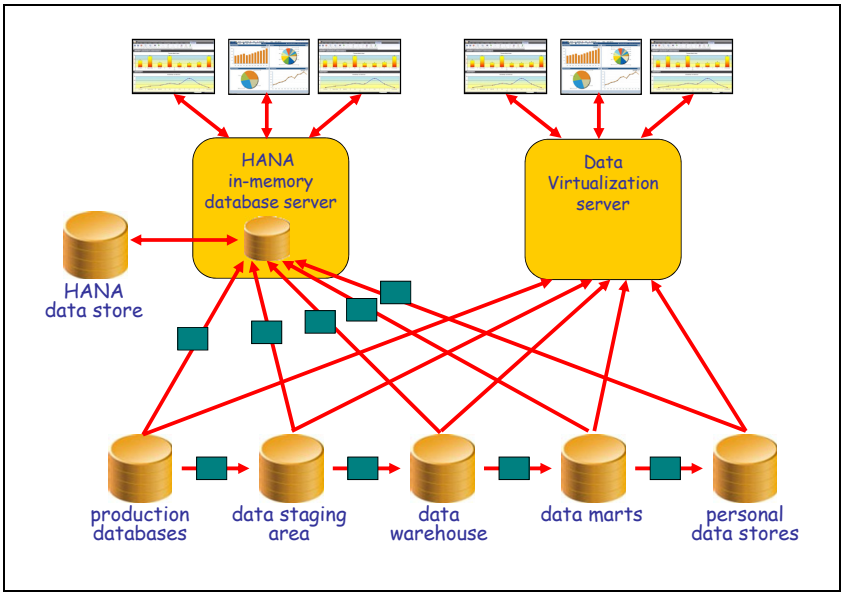


Figure 14 *In this architecture HANA and data virtualization are deployed for different reporting and analytical workloads*

The disadvantage of this architecture is that two technologies are both doing data integration work. It is hard to enforce that the data integration specifications are implemented in a consistent way and remain consistent.

8 Composite Data Virtualization Platform

A number of data virtualization servers is currently available. Composite Software's *Composite Data Virtualization Platform* is one of the more popular products and an example of an advanced data virtualization offering.

Composite Software, based in San Mateo, California, was founded in 2002. This was also the year in which the first version (1.0) of the product was released. Currently, Version 6.2 of Composite Data Virtualization Platform is being shipped. On the OEM front, the product is and has been distributed by IBM/Netezza, IBM/Cognos, Informatica, Pitney Bowes, SAS, BMC, and many more.

Besides being able to access almost any relational database server, including DB2, Microsoft SQL Server, MySQL, IBM/Netezza, and Oracle, the Composite Data Virtualization Platform makes it possible to access XML documents, MDX databases, flat files, spreadsheets, and other non-relational data stores. Composite's product 'flattens' these non-relational stores to tables. This makes it possible for an Excel spreadsheet to join data stored in an SAP InfoCube with an XML document, or for a Cognos report to combine data in an Oracle database with data from Microsoft Analysis Service, to illustrate just two examples.

Together with their other product, *Composite PerformancePlus Applications Adapters*, even modules inside applications, such as those of Oracle, Salesforce.com, and SAP, can be turned into tables that can be queried using any type of tool. In fact, any Java component can be queried as if it's a table. This allows for a report to combine data from an external source using SOAP with internal data stored in a relational database.

Reporting and analytical tools can use any of the popular API's, such as JDBC, ODBC, ADO.NET, and JMS, to access data. In addition, the Composite Data Virtualization Platform can present data as services through SOAP, REST, OData, and XQuery.

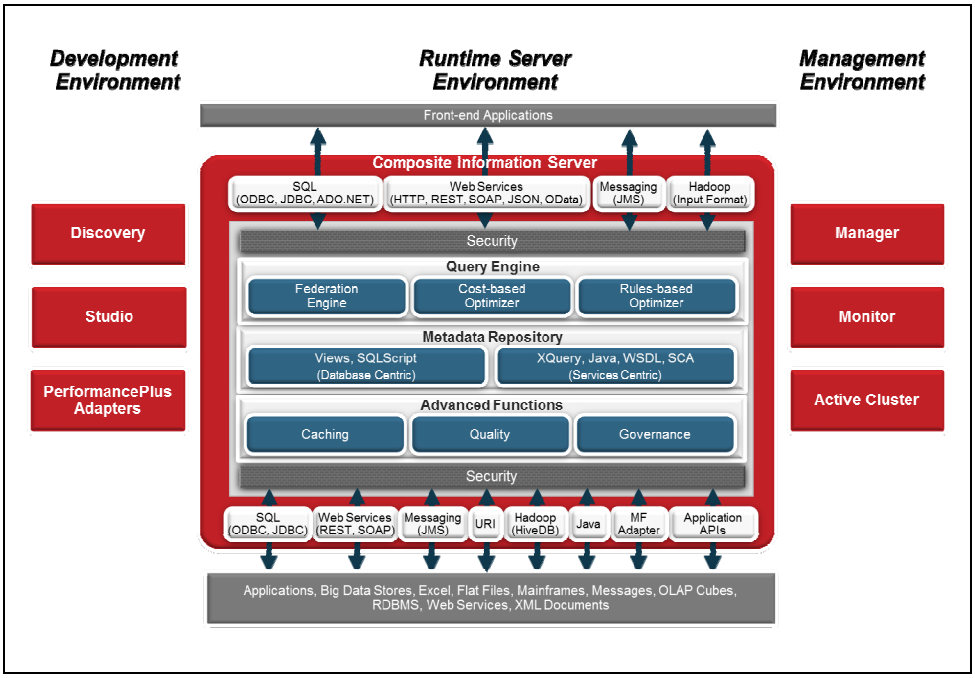


Figure 15 *A high-level overview of the architecture of Composite Data Virtualization Platform (source Composite Software)*

Figure 15 contains a diagram that shows the architecture of the Composite Data Virtualization Platform and includes most of the data stores that can be accessed and most of the API's being supported.

The Composite Data Virtualization Platform can use HANA as data source. Currently, generic JDBC and ODBC drivers are used for accessing HANA databases. Vice versa, HANA can extract data from the Composite Data Virtualization Platform using the interfaces offered by Composite and SAP BODS. The four architectures described in Section 7 can all be implemented with the Composite Data Virtualization Platform. The expectation is that the integration between the two products will become tighter as Composite adds advanced PerformancePlus Adapter features, such as bulk loading, bulk extracts, and advanced optimization algorithms, as Composite has done with IBM/Netezza, Teradata, and other popular database servers. This will result in even faster and more optimized interfaces between the two.

9 Conclusion

The SAP HANA database server is a powerful and feature-rich, in-memory, SQL database server. This database server is part of the SAP HANA environment, which includes data integration and other BI-related features. Data virtualization servers, such as the Composite Data Virtualization Platform, offer on-demand data integration capabilities, which results in more agile BI systems.

Because of the data integration capabilities offered by both, it's understandable that organizations look at the products as competitors. It's recommended to see the products as complements. Together they make it possible to develop highly scalable and agile BI systems. Considering the new BI demands, such as operational BI, self-service BI, and big data analytics, this level of agility is extremely important. BI architectures should be simplified by reducing the number of physical databases. In such a simplified architecture, HANA can act (in different ways) as the performance booster and a data virtualization server can offer the required agility level. In other words, deploying both technologies in a BI system results in one of the fastest and also one of the most agile BI architectures that can be designed with modern-day technology.

About the Author Rick F. van der Lans

Rick F. van der Lans is an independent analyst, consultant, author, and lecturer specializing in data warehousing, business intelligence, service oriented architectures, and database technology. He works for R20/Consultancy (www.r20.nl), a consultancy company he founded in 1987.

The last years he has focused on applying data virtualization in business intelligence system resulting in his new book entitled *Data Virtualization for Business Intelligence Systems*.

Rick is chairman of the annual European Data Warehouse and Business Intelligence Conference (organized in London. He writes for the eminent B-eye-Network[5] and other websites. He introduced the business intelligence architecture called the *Data Delivery Platform* in 2009 in a number of articles[6] all published at BeyeNetwork.com.

He has written several books on SQL. His popular *Introduction to SQL*[7] was the first English book on the market in 1987 devoted entirely to SQL. After more than twenty years, this book is still being sold, and has been translated in several languages, including Chinese, German, and Italian.

For more information please visit www.r20.nl, or email to rick@r20.nl. You can also get in touch with him via LinkedIn (http://www.linkedin.com/pub/rick-van-der-lans/9/207/223) and via Twitter (http://twitter.com/Rick_vanderlans).

About Composite Software

Composite Software, Inc. is the data virtualization market leader. Organizations, such as the world's largest financial services firms, pharmaceutical companies, communications providers, energy producers, and government agencies, rely on Composite data virtualization offerings to simply information access. Composite Software is privately held, with corporate headquarters in San Mateo, CA. To contact Composite, please call (650) 227-8200, visit on the Web at http://www.compositesw.com, or follow on twitter http://twitter.com/compositesw. To learn more about data virtualization visit the DV Café microsite, the Data Virtualization Channel, and the Data Virtualization Leadership Blog.

[5] See http://www.b-eye-network.com/channels/5087/articles/
[6] See http://www.b-eye-network.com/channels/5087/view/12495
[7] See http://www.amazon.com/Introduction-SQL-Mastering-Relational-Database/dp/0321305965/ref=sr_1_1?ie=UTF8&s=books&qid=1268730173&sr=8-1