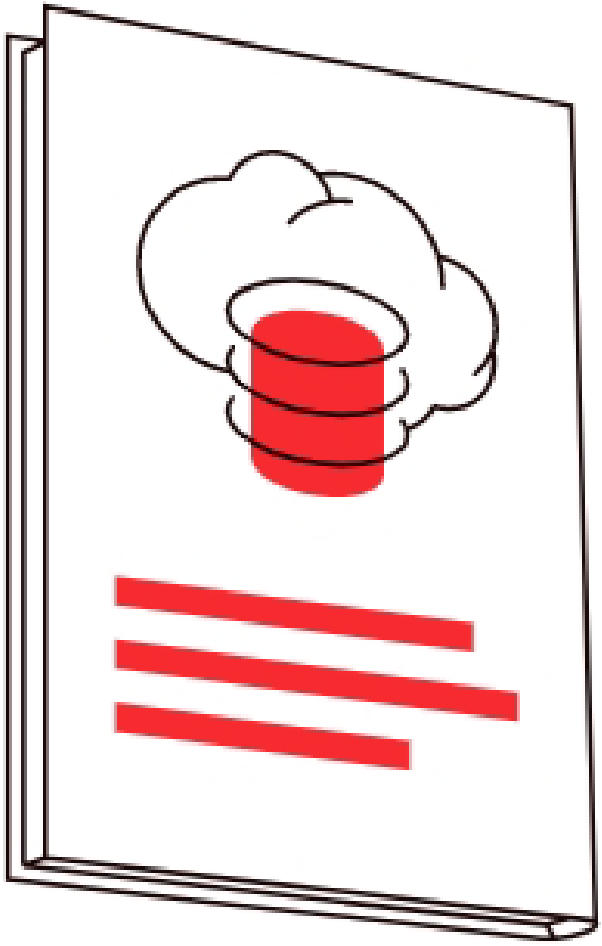


The Cloud Data Warehousing Guide

An Introduction to Cloud Data Warehousing

Cloud data warehouses help businesses store and manage data in the cloud. They represent a significant evolution in data storage, enabling flexibility, scalability, and affordability in managing increasingly large and complex data. In this whitepaper, we will use the terms "cloud data warehouse" and "data warehouse" interchangeably.

Every aspect of data management is conducted under the virtual roof of a cloud data warehouse. Most importantly, a cloud data warehouse transforms data into assets companies can use to improve their capabilities, fuel innovation, and enhance profits.

Traditional data warehouses are physical structures typically on-site. While these have served businesses well for many years, a series of challenges, including high costs and complexities with legacy hardware, have rapidly antiquated them. Cloud data warehouses are a robust and ultramodern alternative to traditional data warehouses, one that can lead to profound success for modern businesses. However, enterprises that have to adhere to special compliance or connectivity requirements still leverage on-premises solutions.

By 2026, the market value of the cloud data warehousing industry is forecast to hit $12.9 billion, a compound annual growth rate of 22.3%. While North America and Europe hold the highest market share, the fastest-growing segment in cloud data warehousing is the Asia-Pacific region, powered by the booming megamarkets of China and India.

The key factor behind these numbers is that data now drives the world, including business. Cloud data warehouses are highly scalable and provide a safe, secure environment backed up by the expertise of leading high-tech companies.

Industries that benefit the most from cloud data warehousing include manufacturing, energy and utilities, healthcare, IT, government, retail, and BFSI (banking, financial services, and insurance). Since cloud data warehousing is such a flexible solution, the use cases are diverse. But one thing is certain: cloud data warehousing is now the norm and the foundation upon which the future will be constructed.

Online Analytical Processing (OLAP)

Online analytical processing (OLAP) is a type of data processing that occurs in a data warehouse and serves different workloads and requirements.

Data consistency is optional for OLAP systems since they typically use data snapshots. OLAP systems handle large data volumes and use denormalized database designs leveraging star schema or snowflake schema. This approach increases data redundancy, improves query performance, and accelerates data-driven decision-making.

How Does Data Warehousing Work?

Data warehouses continuously collect and organize data into a dedicated comprehensive centralized repository. Data collected from various sources are systematically sorted into tables based on the data type and layout.

access accordingly.

2. An **operational data store (ODS)** is often a go-to choice for organizations with data warehouse systems failing to satisfy their reporting requirements. As ODS can be refreshed in real time, it is a popular option for storing routine activities. In a large healthcare system, real-time or near-real-time data access is crucial for patient care. Due to their batch-processing nature, traditional data warehouses can't always meet this need. This is where an ODS comes into play. The ODS can integrate data from these various sources like electronic medical records (EMR), pharmacies, laboratory systems, and radiology to present a unified and current view of the patient's data.

For instance, a doctor can access the ODS to get the most recent patient data like lab results or medication history; and a pharmacist can verify prescriptions to prevent harmful drug interactions. Thus, the ODS provides timely, integrated data to healthcare providers, improving patient care.

3. A **data mart** is designed for specific business or industry verticals. For example, they are prevalent in finance, sales, and inventory. Moreover, data marts can quickly collect data from a source.

2. An **operational data store (ODS)** is often a go-to choice for organizations with data warehouse systems failing to satisfy their reporting requirements. As ODS can be refreshed in real time, it is a popular option for storing routine activities. In a large healthcare system, real-time or near-real-time data access is crucial for patient care. Due to their batch-processing nature, traditional data warehouses can't always meet this need. This is where an ODS comes into play. The ODS can integrate data from these various sources like electronic medical records (EMR), pharmacies, laboratory systems, and radiology to present a unified and current view of the patient's data.

For instance, a doctor can access the ODS to get the most recent patient data like lab results or medication history; and a pharmacist can verify prescriptions to prevent harmful drug interactions. Thus, the ODS provides timely, integrated data to healthcare providers, improving patient care.

3. A **data mart** is designed for specific business or industry verticals. For example, they are prevalent in finance, sales, and inventory. Moreover, data marts can quickly collect data from a source.

- **Offline operational database:** Data will be copied periodically to a server from an ODS to load, process, and report. This approach is practical when data synchronization isn't a must.
- **Offline data warehouse:** Data is stored and regularly updated from the operational database and other sources to derive critical business insights.
- **Real-time data warehouse:** A real-time data warehouse is used for up-to-date insights and analysis based on the latest transactional data. All transactions in an operational database are updated in the data warehouse.
- **Integrated data warehouse:** The integrated data warehouse consolidates data into a unified view for analysis. All transactions occurring in the operational database are simultaneously updated in the data warehouse. Once updated, the data warehouse will generate transactions and forward them to the operational database.

Software tools and hardware used for storing, transforming, and analyzing data are called data warehouse appliances.

With the concepts introduced above, we can highlight three key ways in which a data warehouse can work:

1. **Basic data warehouse:** Organizations can eliminate data redundancy, which reduces the amount of data in storage. This, in turn, makes data clearer and more user-friendly. The key benefit here is that different departments from multiple sources can quickly access data directly from the warehouse.
2. **Data warehouse with staging area:** Organizations can clean data in "staging areas" before moving it to storage. This is one of the leading methods of ensuring that only relevant and valuable data is stored in the data warehouse.
3. **Data warehouse with data marts:** Organizations can enhance their data warehouse's customization level after data is processed, allowing them to streamline information to staff, teams, or departments that need it the most. This approach helps boost productivity and accelerate the decision-making pace.

LITERATURE REVIEW

Data warehousing

Definition

Business Intelligence (BI) is not a new concept. One early occurrence of the term is by H.P. Luhn (1958), who defined BI and introduced a business intelligence system as early as 1958. The need for automation and new technologies were driven much by the same reasons as now:

> *It has become evident in recent years that present communication methods are totally inadequate for future requirements. Information is now being generated and utilized at an ever-increasing rate… (Luhn, 1958, p. 314)*

> *At the same time the growth of organizations and increased specialization and divisionalization have created new barriers to the flow of information. (Luhn, 1958, p. 314)*

Luhn continues to describe in detail what can be called an early decision support system. The technology available was, according to Luhn, not at an adequately advanced level to create the system defined in the paper (Luhn, 1958). Thirty years later, Devlin & Murphy (1988) present the EBIS-architecture (E|ME|A Business Information System) employed by IBM. Devlin & Murphy (1988) proceed to demonstrate that querying transaction data and requesting information from the IS organization when needed is not a viable long-term solution. Similar issues are presented by several others (Golfarelli & Rizzi, 2009; Linstedt & Olschimke, 2015). Devlin and Murphy introduced the term Business Data Warehouse as *"The BDW is the single logical storehouse of all the information used to report on the business.". (Devlin & Murphy, 1988, p. 67)*

One of the first articles and one of the most important papers mentioning a relational model of database systems was written by Edgar F. Codd in the 1970s (Krishnan, 2013). The paper was pivotal for several reasons. One was that it was the first to introduce the idea of managing data using a relationship-based approach. Another reason was introducing the idea of abstracting the management and storage of data from the user. It

further discussed the idea of removing duplicates and minimizing redundancy (Krishnan, 2013).

According to Inmon (2005), a data warehouse is a subject-oriented, integrated, time varying, non-volatile collection of data in support of the management's decision-making process. Kimball's definition of a data warehouse is that it is a copy of transaction data specifically structured for query and analysis. (Kimball et al., 2011). Some distinguishing characteristics of a data warehouse as defined by Turban, Rainer & Potter (2007) and Inmon, Imhoff & Sousa (2002) are:

- **Organized by subject.** The data are organized by subject, e.g. by customer, vendor, product, category. The grouping supports decision support and data analysis.

- **Consistent**. All data must be coded in a consistent manner. Before loading the data, they must be transformed so that all units and data are uniform.

- **Historical**. Data are never removed from a data warehouse, and the historical data can be used for analyzing trends and for forecasting.

- **Non-volatile.** Once data have been loaded into the warehouse, no updates or changes on the data will be made.

- **Uses online analytical processing (OLAP).** Typical operational systems rely on online transaction processing (OLTP). For a point of sale system, it is imperative that all transactions are processed as soon as they occur. There is rarely a need to look back at historical sales in an OLTP-system. Therefore, the focus is on speed and efficiency. The emphasis is on data entry and data modification speed, not analytical or querying capabilities. A data warehouse is not a mission-critical system and, instead, processes a large batch of data separate from the transaction systems not to slow down daily operations (Conn, 2005).

- **Multidimensional.** A normal relational database stores data in two dimensions. A data warehouse is designed for more than two dimensions. The purpose is to be able to analyze for example sales data and group it by vendor, by time period, or

by geographic area among other attributes. The multidimensional representation of data is most often a data cube.

- **Relationship with relational databases.** The data from operational OLTP systems are loaded into the data warehouse through a process called extract, transform and load (ETL). Gathering the data from operational systems means that the OLTP-systems are not impacted during the actual analysis. It also means that the data will be current and there is no need for additional manual labor to enter the data into the data warehouse.

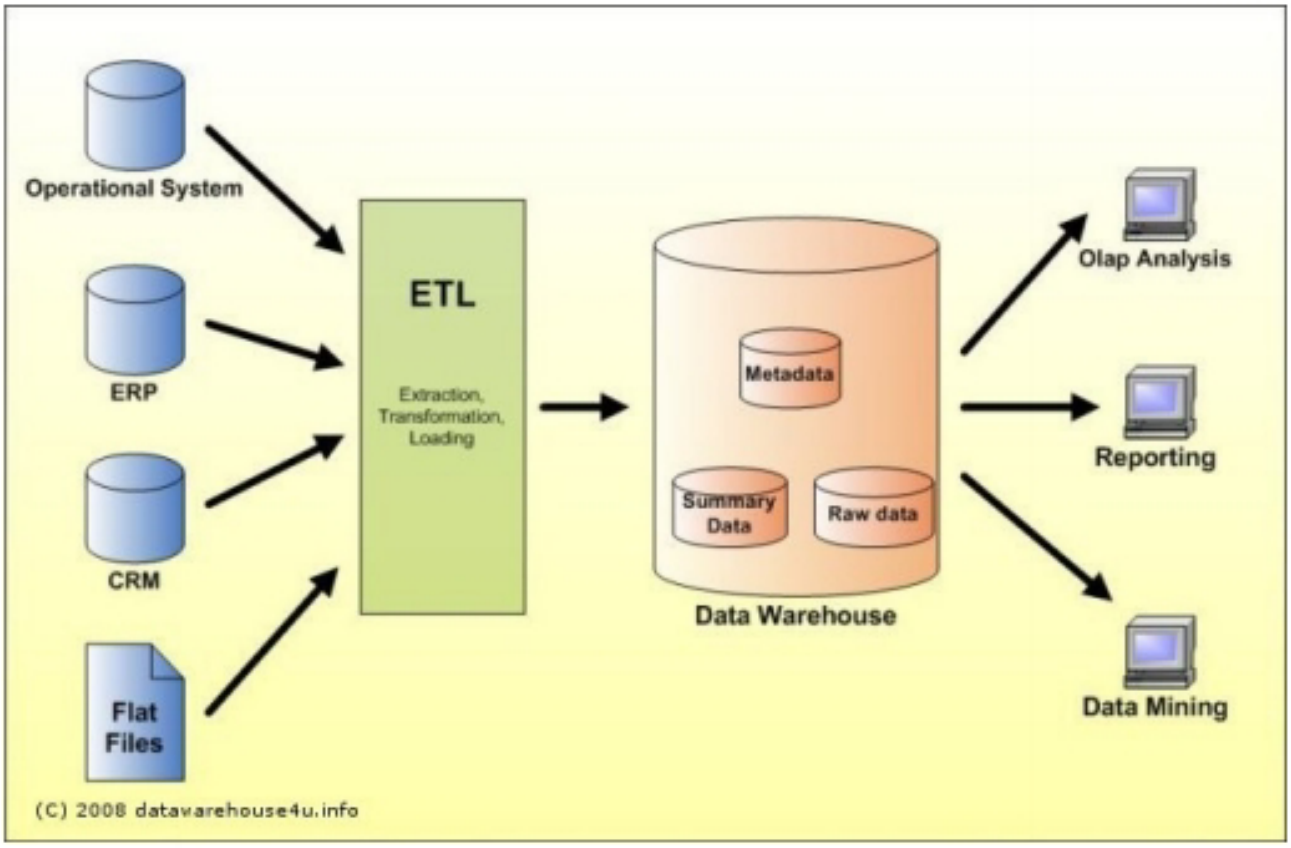


Figure 1 - Data Warehouse Architecture
(http://datawarehouse4u.info/images/data_warehouse_architecture.jpg)

As seen in Figure 1, a data warehouse system usually consists of several major parts, but is not limited to:

- **The source data**, which usually come from operational systems.
- **The staging area (ETL)**, which manages the extraction of data from source systems, transforms and cleans up the data, and then manages the inserting of the data into the DW.
- **The data warehouse databases**, where the actual data are stored, along with metadata.

- **Presentation layer,** either using data marts or directly providing the data for further use and analysis.

Kimball & Ross (2013) describe several goals for a data warehouse. The data warehouse must make information accessible, maintain consistency of information, be able to adapt to change and serve as a definitive foundation for improved decision making.

They further state that it is vital that the business accepts the data warehouse. They must see the data warehouse as the best available source of information, otherwise it might not even be used at all. This claim is supported by case studies carried out by Watson, Goodhue & Wixom (2002).

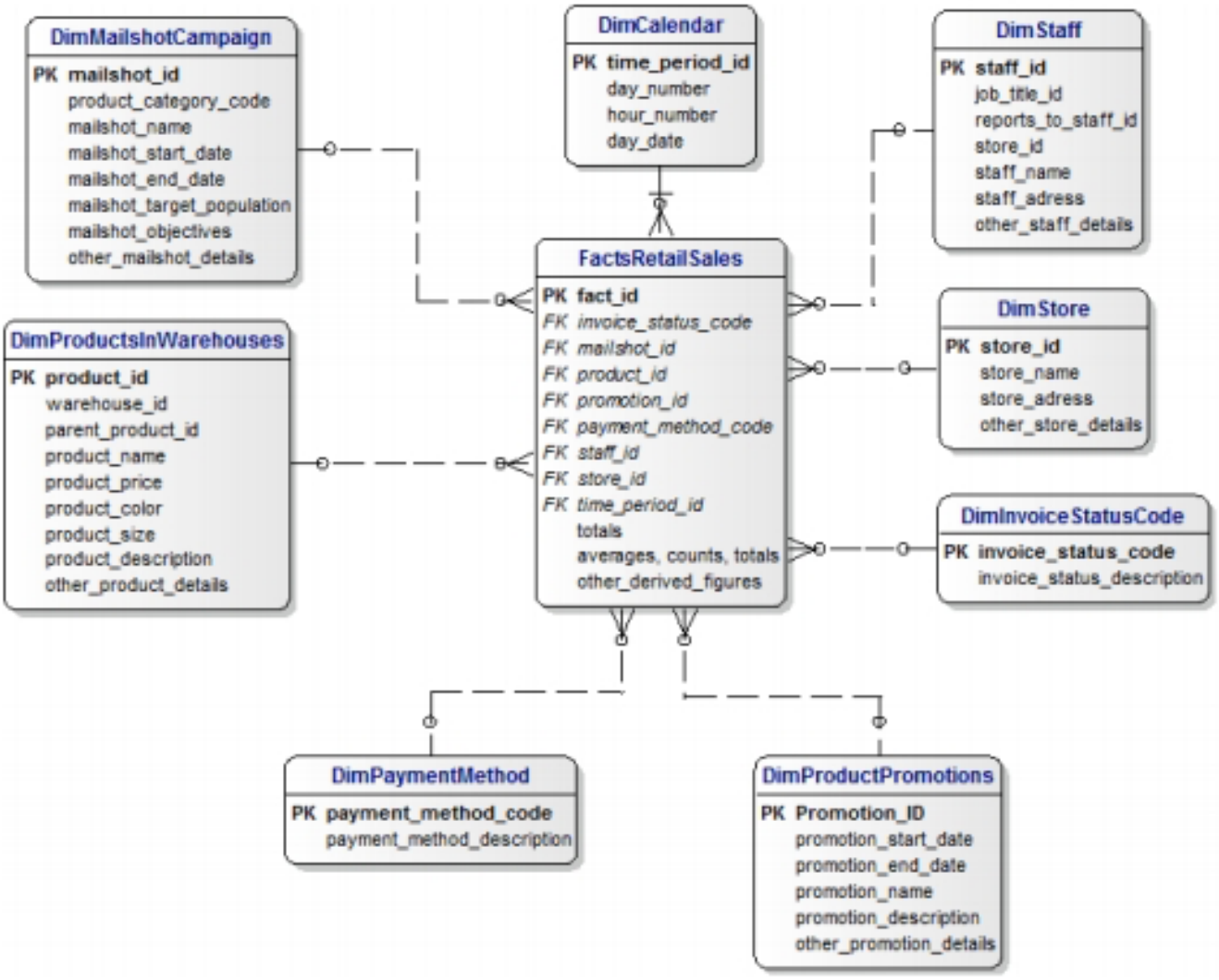
There are two different main methodologies for choosing a data warehouse architecture. There is the top-down approach of modeling a Corporate Information Factory, in third normal form by Bill Inmon (Ariyachandra & Watson, 2006; Breslin, 2004; George, 2012), and the bottom-up approach of dimensional modeling by Ralph Kimball (Ariyachandra & Watson, 2006; Breslin, 2004; George, 2012), known as the data mart bus architecture. In their survey-based study, Ariyachandra & Watson (2006) found that:

> *The bus, hub-and-spoke, and centralized architectures earned similar scores on the success metrics. This finding helps explain why these competing architectures have survived over time—they are equally successful for their intended purposes. (Ariyachandra & Watson, 2006, p. 8)*

	Inmon	Kimball
Building Data warehouse	Time Consuming	Takes lesser time
Maintenance	Easy	Difficult, often redundant and subject to revisions
Cost	High initial cost; Subsequent project development costs will be much lower	Low initial cost; Each subsequent phase will cost almost the same
Time	Longer start-up time	Shorter time for initial set-up
Skill Requirement	Specialist team	Generalist team
Data Integration requirements	Enterprise-wide	Individual business areas

Dimensional modeling

Ralph Kimball is a proponent for dimensional modeling, stating that dimensional modeling is the best way to present information as simply as possible (Mundy & Thornthwaite, 2007). The modeling will follow the business process and, as such, will be able to quickly and accurately return query results and information to users.



Dimensional modeling involves using facts and conformed dimensions to describe the data, as seen in figure 3. The facts typically contain numerical values and measurements, while the dimensions are descriptive, used for grouping and labelling. Conformed dimensions are meant to be used for all facts combined and, as such, they are normally designed first. The benefit of dimensional modeling is a fast and objective-focused modeling approach that provides quick results and is quick to implement (Kimball & Ross, 2013).

The dimensional modeling is often referred to as the bottom-up approach. Small and specific data marts are designed to suit a certain need. Data marts are implemented next to each other and then combined to become the data warehouse (Jukic, 2006; Kimball & Ross, 2013).

Corporate Information Factory

Breslin (2004) defines the Corporate Information Factory (CIF) as:

> *Inmon's architected environment consists of all information systems and their databases throughout a given organization. He calls this behemoth the Corporate Information Factory, or CIF (Inmon and Imhoff, 2002). (Breslin, 2004, p. 8)*

Instead of just considering a data warehouse alone, the entire IT-ecosystem is seen as a whole. This leads to an evolutionary lifecycle, where the data warehouse follows the design methods and technologies used to develop the operational systems (Breslin, 2004).

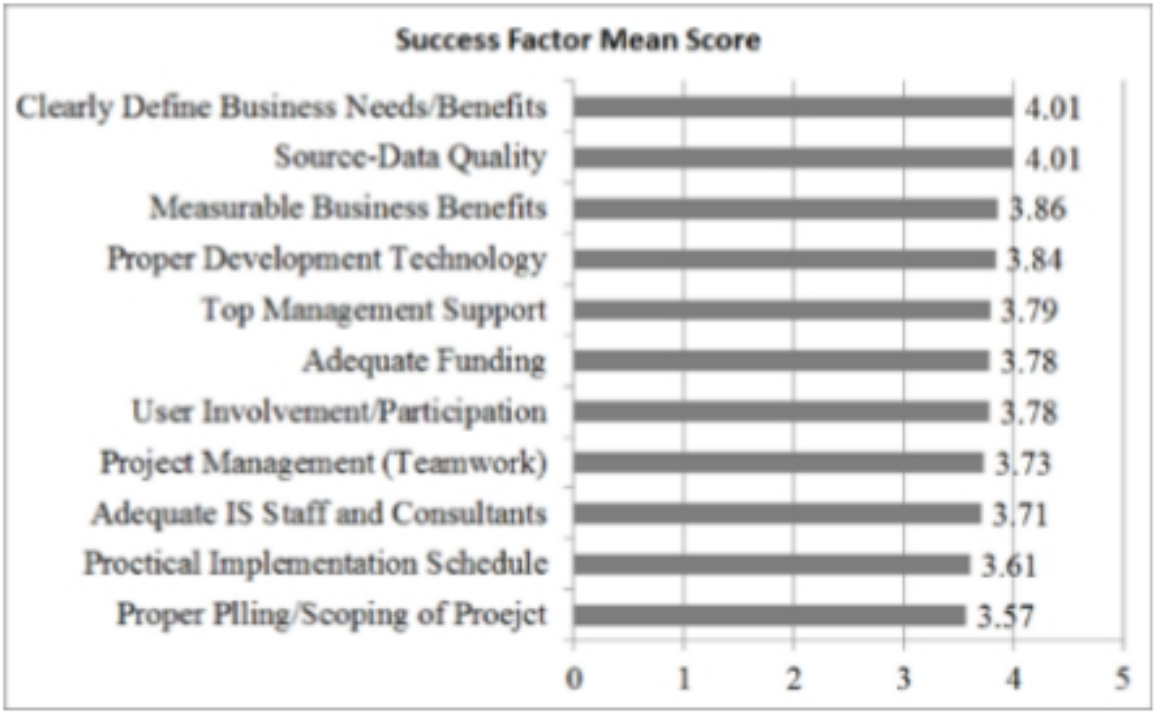
The data warehouse in a CIF was presented by Bill Inmon as an integrated database modeled using the traditional Entity-Relationship (ER) modeling, normalized to the third normal form (Inmon et al., 2002). The data warehouse will contain the most detailed version of data at the lowest granularity. Dimensionally modeled data marts are then designed and implemented with the data warehouse as source (Jukic, 2006).

Future of data warehousing

Data warehousing is one of the most important projects in the IT-industry. Data warehousing has experienced large growth since 2000 and continues to grow. Decision

support systems have evolved into more robust and advanced applications (Kimpel & Morris, 2013; Sen et al., 2012).

However, successfully implementing and finishing a data warehousing project is no easy task. An estimated 40% to 50% of data warehouse initiatives fail, most with costly consequences (Kimpel & Morris, 2013; Sen et al., 2012). Kimpel & Morris (2013) report on the success factors found through a survey answered by 98 data warehousing professionals:



Possibly the biggest and most important research topic revolves around developing and adopting data warehousing and OLAP methodologies to support analytics over big data (Cuzzocrea, Bellatreche, & Song, 2013; Šubić, Poščić, & Jakšić, 2015).

The sheer volume and complexity of big data causes major issues when trying to implement practical applications (Assunção, Calheiros, Bianchi, Netto, & Buyya, 2015). The potential size of unstructured data, the explosive number of dimensions, the technical challenges of designing an OLAP data cube for big data, and sufficient end-user performance are open and important research problems (Cuzzocrea et al., 2013). To solve this, several new models are being developed, such as Data Vault, MapReduce, Hadoop and Hive (Krishnan, 2013; Šubić et al., 2015).

> *...heterogeneity is permanent in the future of the data warehouse, and the concept remains the same as defined 40 years ago by Bill Inmon, but the physical implementation will be different from the prior generations of data warehousing. (Krishnan, 2013, p. 217)*

> *We mentioned the Data Vault and Anchor modelling methods that are designed just for the purpose of storing and processing large amounts of data in data warehouses. These models may have come too late to the market, because most of the big companies like Google, Amazon and Facebook have already switched to non relational solutions. (Šubić et al., 2015, p. 242)*

Some future research directions suggested by Cuzzocrea, Bellatreche, & Song (2013, p. 69) are:

- Methodologies for designing OLAP data cubes over big data
- Innovative solutions for computing aggregations
- High-performance architectures for implementing OLAP data cubes over big data, e.g. utilizing GPU processors.
- Optimization to MDX-language
- Semantically-rich OLAP big data cubes

Krishnan (2013) presents data virtualization as an emerging technology for next-generation data warehouses. Data virtualization would be used to create a semantic data integration architecture. The aim with data virtualization is to create a single presentation layer to the consumer. This is done through semantic transformations and by abstracting

the infrastructure. Instead of physical transformations, data are integrated through virtualizations. As the data are not moved, the database or data store can be optimized for data access, thus fully utilizing the underlying infrastructure. Data virtualization enables automated data discovery, adding new data and data sources. Contextualization can be done during discovery to reduce errors and speed up the integration (Krishnan, 2013).

Another current data warehouse research topic is real-time data warehousing coupled with real-time analytics (Kakish & Kraft, 2012). In traditional ETL-processes, the most current information is not always available. Kakish & Kraft (2012) state that the current periodic refreshes must give way to continuous updates. Instead of relying on batch runs and offline updating, one possibility for real-time ETL would be to utilize data streams of events from data stores to the data warehouse. However, this is an expensive and difficult architecture and requires further research. As they point out:

> *The importance, complexity and criticality of such an environment make real-time BI and DW a significant topic of research and practice...* (Kakish & Kraft, 2012, p. 8)

As the data sets grow larger and the need for processing power increases, new technologies will be looked at to alleviate the problem. Hadoop is an example of a distributed file system and framework for handling and analyzing large data sets (Shvachko, Kuang, Radia, & Chansler, 2010). Hadoop partitions and distributes the data and computation across vast numbers of hosts, which enables parallel processing. Application data and metadata are stored separately. Metadata are stored in NameNodes. Application data are stored in DataNodes. Application data are replicated for reliability and increased bandwidth (Shvachko et al., 2010). By default, every block is replicated at three separate nodes. The NameNodes keep track of where the blocks are, and handle the replication and writing of new data. If one of the replicating nodes does not answer within a certain time limit, then the block is replicated elsewhere and the node is considered lost. This ensures that the data are protected, without the need for local redundancy or RAID-type technologies (Shvachko et al., 2010).

> *Hadoop clusters at Yahoo! span 25 000 servers, and store 25 petabytes of application data, with the largest cluster being 3500 servers. One hundred other organizations worldwide report using Hadoop. (Shvachko et al., 2010, p. 1)*

Since 2010, Yahoo has increased its usage to 40 000 computers, reaching 455 PB of total storage, and having 4 500 nodes in the largest cluster (Miah, Hasan, & Uddin, 2015).

Hive is a petabyte scale data warehouse, built upon Hadoop and developed by Facebook (Thusoo et al., 2010). As Hadoop, Hive is also an open source project. The aim is to use Hadoop and support queries in a language that resembles SQL, called HiveQL. Instead of traditional RDBMS, Hive utilizes the distributed file system of Hadoop (Thusoo et al., 2010). Scalability is ensured by being able to add new nodes to the cluster, or upgrade existing ones (Shvachko et al., 2010; Thusoo et al., 2010). As of 2015, Facebook´s Hive data warehouse held 300 PB of data in 800 000 tables, or 900 PB considering the 3-way replication. Four PB new data are generated daily (Bronson, Lento, & Wiener, 2015).

Data Vault

There is very little reviewed and published material on the topic of Data Vaults. The creator Daniel Linstedt has written several books and white papers, but they have surprisingly few citations. It might be that the topic is too new and the model is not yet in extensive use. The following chapters will rely heavily on one of Linstedt's newest books, *Building a Scalable Data Warehouse with Data Vault 2.0.*

Data Vault modeling was invented by Daniel Linstedt in the beginning of the 1990s and released for review in 2002 (Linstedt & Olschimke, 2015). Linstedt was employed by the Department of Defense to design and build a scalable data warehouse. This was before the time of Big Data. There was a clear need for an architecture capable of accommodating large data sets, which were to be processed and loaded quickly and reliably. Scalability and iterative development were also high priorities for the project (Linstedt & Olschimke, 2015).

Architecture

The Data Vault architecture follows the commonly found three-layer architecture introduced by Inmon (2005). However, to suit the needs of an enterprise data warehouse (EDW), certain modifications must be made. Some of the modifications to the typical three-layer architecture are (Linstedt & Olschimke, 2015):

- The staging area does not store historical information and does not apply changes to the data other than ensuring the correct data type.
- The data warehouse layer is modeled according to the Data Vault modeling technique.
- Information marts that are dependent on the data warehouse layer
- Several optional vaults in the data warehouse layer, such as Metrics Vault, Business Vault and Operational Vault. These are used for optimization and for metadata purposes.

Components

The core components of the Data Vault modeling technique are called hubs, links and satellites. They are based on entities from a hub-and-spoke architecture. Hubs, links and satellites from a complex network, similar to a human brain which is a network of neurons (Linstedt & Olschimke, 2015; Williams, 2015).

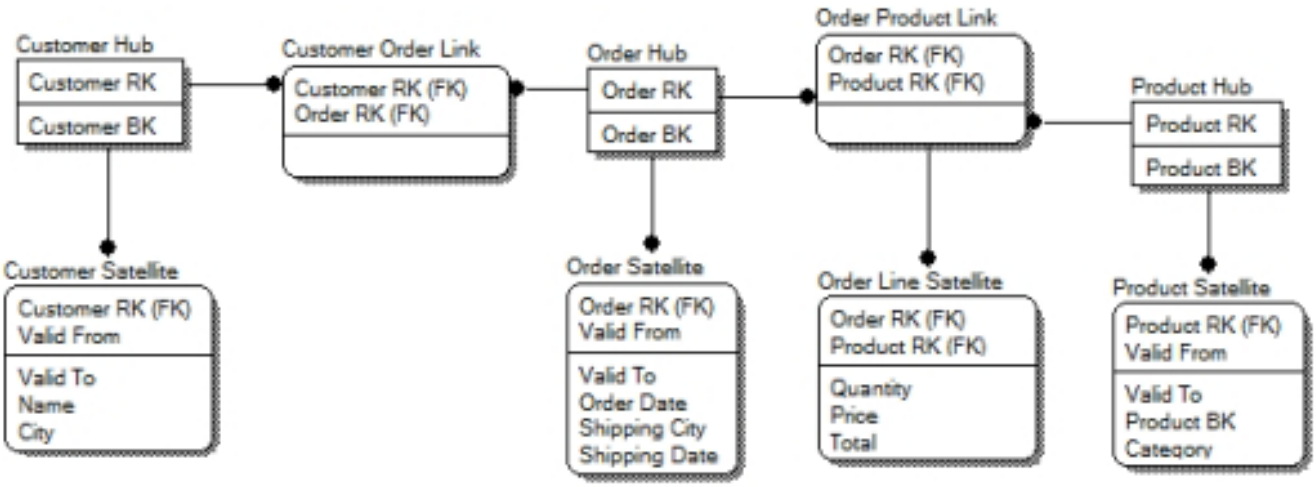


Figure 5 - A Data Vault Model (http://bukhantsov.org/2012/04/what-is-data-vault/)

Figure 5 demonstrates the hubs, links and satellites of an exemplary data vault model when modeling the entities customer, order, and product. Surrogate keys are used in the data vault structure to increase lookup performance. Business keys can sometimes be long combinations of several values, and using a surrogate key in form of a simple increasing integer will result in much faster joins. The surrogate keys of links and satellites are always foreign keys, pointing to the surrogate key of a hub (Linstedt & Olschimke, 2015).

The loading of the tables is presented in chapter 4.1.4.

Hubs

Business keys are used to identify certain business objects in operation systems. A business key might be a customer number, an invoice number or something similar, that

makes the business object uniquely identifiable. It might also be a combination of values, such as a customer number and a country code.

Hubs are lists of unique business keys which contains metadata about when the business key arrived for the first time in the data warehouse and from what source system it originates. The primary key of a hub is the surrogate key of its business key. Often, though not mandatory, the hub also keeps record of when the business key was last seen in a loading process (Linstedt & Olschimke, 2015).

2.2.2.2 Links

Links are responsible for modeling associations and transactions between business objects. A link connects related business keys and in data vault modeling, links are modeled between hubs. Having links allows changes to the structure over time. A primary key for the link will be formed by combining the two business keys being linked. This will be stored in the form of a surrogate key. A link will also contain foreign keys to surrogate keys of the hubs. Additionally, a link contains information about the load date and the source of the data. No information about validity of the data is stored, only past and present relationships between business keys (Williams, 2015).

Satellites

Satellites are where the actual descriptive data are stored. Satellites provide context to business objects and tracks changes over time (Williams, 2015). Every version of a business concept will be stored and can be tracked via the load date and load end date.

A satellite is always attached to one link or one hub. The primary key of a satellite is also a foreign key, referring to the surrogate key of the attached link or hub. The validity of a row in the satellite is tracked by a load date and a load end date. When the descriptive data of a business key changes, the previous row in the satellite for that business key will be closed, and a new row containing the new data will be created. The previous data will not be removed, as nothing should be removed from a data warehouse (Linstedt & Olschimke, 2015).

Benefits and limitations

The Data Vault model is capable of handling changes in that it enables structural changes by simply adding more objects and attributes to the model. Having a permanent staging area for source data allows loading data without delays. It also has built in inclusion of metadata, such as data source and loading time. Aspects such as these make the Data Vault model recognized as the optimal choice for DW 2.0 architecture (Jovanovic & Bojicic, 2012).

Though it has substantial benefits, the Data Vault model is not perfect. There are some limitations regarding resilience to change and the tracking of the validity of objects with respect to the real world (Bojičić et al., 2016). Handling the changes is not an issue, as it is mostly done by adding objects to the model. However, this means that the original source used for the model is no longer obtainable, due to irreversible transformation. Secondly, there is no information about the validity of the data. The data are considered valid and active, until a new load provides new metadata (Bojičić et al., 2016).

Data Vault 2.0

In his foreword for Linstedt's book *Building a Scalable Data Warehouse with Data Vault 2.0*, Bill Inmon wrote:

> *Over multiple years, he improved the Data Vault and evolved it into Data Vault 2.0. Today, this System of Business Intelligence includes not only a more sophisticated model, but an agile methodology, a reference architecture for enterprise data warehouse systems, and best practices for implementation. (Linstedt & Olschimke, 2015, p. xv)*

Data Vault 2.0 is based on Data Vault 1.0, but adds methodology and architecture to the already existing data vault modeling. One big change in the data vault model is using hash keys instead of business keys. The hashes are calculated in the staging area. This in turn allows parallel loading, as no key lookup is required between objects. Parallel loading is a substantial performance gain. Hash keys can also be generated for entire rows, which ease the comparison of whether two rows are identical or, whether a row already exists in the target table or not (Linstedt & Olschimke, 2015).

There are many various definitions of cloud computing, depending on what aspect is in focus (Abadi, 2009). The National Institute of Standards and Technology provides the following definition:

> *Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. (Mell & Grance, 2011, p. 2)*

Usually cloud computing is approached from a business point of view or a technical point of view (Baars & Kemper, 2010). The business view sees cloud computing as *"An Internet based service provision provided on a fee bases" (Baars & Kemper, 2010, p. 1530)*. The technical theme can be described as:

> *A distributed, net-based architecture where resources can be dynamically rearranged. This promises increased cost efficiency (as a result of a higher degree of resource utilization), as well as higher degrees of performance, stability, scalability, and flexibility. (Baars & Kemper, 2010, p. 1530)*

The concept of utilizing cloud computing is becoming more and more appealing. The promise of lower operational costs, higher quality and higher flexibility are one of several motivators for migrating towards cloud-based solutions (Agrawal, Das, & El Abbadi, 2011; Baars & Kemper, 2010; Marston, Li, Bandyopadhyay, Zhang, & Ghalsasi, 2011). Armbrust et al. (2010) predict that cloud computing will grow and become ever more important in the future. Developers need to take scalability into account and make sure that this scaling happens rapidly without delay. The licensing model of paying according to usage for cloud services, suits this requirement very well (Armbrust et al., 2010; Marston et al., 2011).

Privacy and security

Moving data from an on-premises solution to a cloud solution is a very problematic situation. The most radical example of potential issues is the US Patriot Act, that allows the US government to demand access to data stored on a computer or by a third-party hosting provider, in the United States. These data are to be handed over without knowledge or permission of the person/company using the service (Abadi, 2009). According to a study carried out by researchers at a Belgian University, even though US organizations are taking steps to ensure privacy concerns are resolved, there is still a need for considerable improvements (Dhont, Asinari, Poullet, Reidenberg, & Bygrave, 2004). They found that the majority of US organizations have challenges with integrating the Safe Harbor policies into their data processing, regardless of effort and best intentions (Dhont et al., 2004).

It is difficult to ensure that the cloud provider is not using the stored data or even disclosing it, intentionally or by accident. In the cloud, the responsibility for security and privacy is shared among several parties (Armbrust et al., 2010; ComPUtING, 2011).

Another factor impeding the adoption of cloud computing is regulation on a local or national level. Regulation might cover requirements for physical data audit, which becomes problematic when the data are moved to the cloud. Regulation that require SaaS providers to keep data within national or continental boundaries might negate many benefits that cloud computing offer. However, providers are working towards solutions that guarantee a specific data center at a certain geographical location is used to fulfill the requirements mentioned (Marston et al., 2011).

Because cloud computing is still in its infancy, it is reasonable to consider the technological structures immature. The regulation regarding privacy is still not updated to handle data in cloud, and the transfer of data:

> *While the legal issues facing cloud operators and cloud users stem from the fact that personal data is transferred across jurisdictional borders, applicable privacy regulation typically draws a line between data being transferred within an organisation, and data being transferred between organisations. (Svantesson & Clarke, 2010, p.392)*

RESEARCH METHODOLOGY

The thesis is divided into two parts, a theoretical part with a review of the literature in the field of data warehousing and business intelligence, and an empirical part presenting a combination of constructive research and case study research. Because the main point of the research is to implement a real product for the end-customer, an artifact will be constructed according to existing best practices, previous way of working, and scientific literature. After the actual solution is implemented, analysis on how well the artifact corresponded to the actual product will be conducted, as well as how the implementation of the product succeeded.

Constructive research

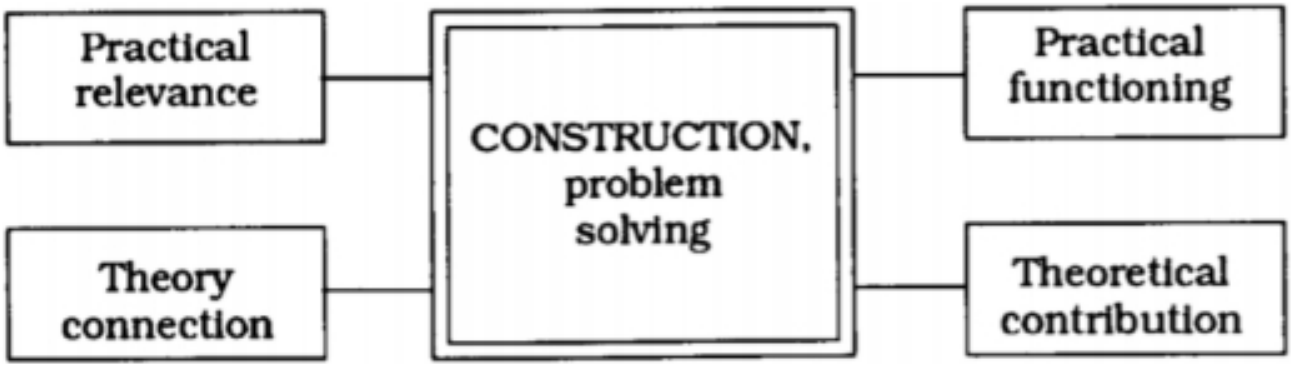


Figure 6 - Elements of Constructive Research (Kasanen, Lukka, & Siitonen, 1993, p.246)

As seen in Figure 6, an essential part of constructive research is to tie the problem and its solution to theoretical knowledge. A general understanding of the problem domain is necessary to assess the applicability of the construct in general.

Kasanen et al. (1993) respond to the critique of problem solving not generating scientific knowledge due to the uniqueness of problems, by arguing that it is *"difficult to imagine a solution to a real-world management accounting problem which would suit well the firm in question but not be suitable to other approximately similar firms."* (Kasanen et al., 1993, p.252-253)

Kasanen et al. (1993) propose that *"A significant share of Master's theses in management accounting follow the plan of problem statement, theory review, a solution of a real-world business problem, and discussion."* (Kasanen et al., 1993, p.244). Even though they target

management accounting research, a parallel can be drawn to information systems research. Kasanen et al. (1993) list some defining features of the constructive method:

- Step by step procedure, with every step defined in the framework.
- It is possible to check every step or phase in the construction.
- The procedure serves a purpose. Building the construction is a goal-oriented activity.

In practice the underlying context for this thesis is problem solving. These features align with the context of this thesis and, as such, the constructive method is a suitable method for approaching the problem.

Iivari (2003) questions the actual core of IS research. On one hand the concept of an IT artefact is central to the discipline. However, Iivari (2003) argues that information systems should form the core, not IT artifacts. Iivari (2003) further suggests that the identity of IS should be based on the view of IS as an applied science, to support IT experts in practice. These arguments support the choice of a constructive method, as the construct will be used in a real-world application.

Case study research

Benbasat, Goldstein & Mead (1987) define case research through several key characteristics. In case research, typically the phenomenon or problem is examined in its natural setting. According to Benbasat et al., *"research phenomena not supported by a strong theoretical base may be fruitfully pursued through case research."* (Benbasat, Goldstein, & Mead, 1987, p. 372) Case research is innately explorative, because the investigator builds the hypotheses as the complexity of the unit is studied. No dependent or independent variables are defined in advance (Benbasat et al., 1987).

Benbasat et al. (1987) found that the predominant theme in case studies was implementation, and the causes of success or failure. It was rare to have a clearly defined objective, and most studies were characterized as exploratory in nature.

Yin (1981) is a strong proponent of case research. He states that case studies can be done using either qualitative or quantitative evidence. The evidence may consist of *"fieldwork, archival records, verbal reports, observations, or any combination of these."* (Yin, 1981, p.58).

application to help with this, and support the loading of said data, without the need for manual labor in the staging area.

Johnsson also wants to incorporate Big Data. Big Data eventually has a structure, which means it can be modeled at some point. All it takes is to create the rules according to which the data will be handled. Johnsson points out that SmartEngine is not only a DWA-tool, it is also a tool for implementing any model. Another important feature is to support enriching of data. There needs to be a simple way to enrich source data with open data and big data.

Function and logic

Because SmartEngine is built to work on an exported ER-model of a certain format, it is reliant on Visual Paradigm. The export contains all entities, relationships, attributes and metadata, in XML-format (Extensible Markup Language). Any modeling tool could be used, but then the logic of the application would need to be tweaked to accommodate the alternate export of the model.

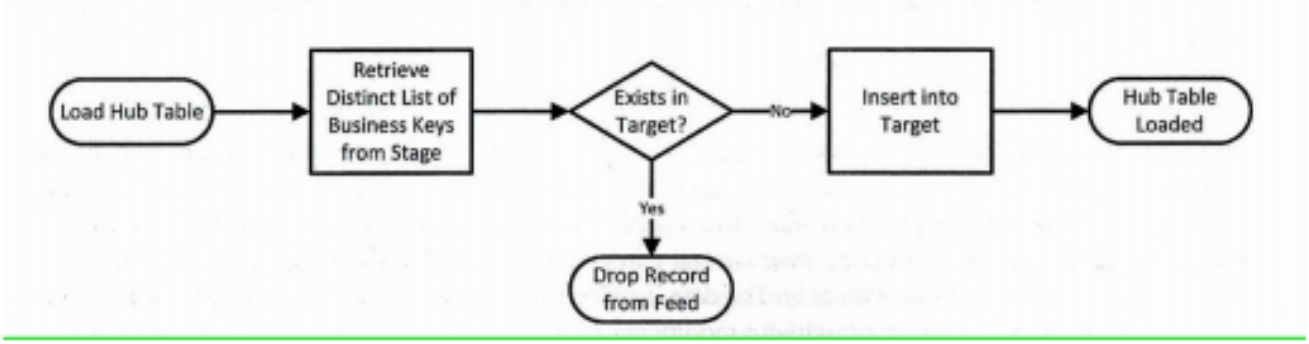
SmartEngine uses this export to create an internal object model to be used when generating the necessary SQL scripts and database objects. The rules and logic for generating the database objects will be presented in chapter 4.2.

Having a separate object model enables developers to write additional scripts for SmartEngine using C#. The object model of SmartEngine will always be standardized, regardless of what the ER-model looks like. Any changes in the source code are automatically reflected in the object model documentation. The idea is that scripts for SmartEngine are collaborative and iterated upon. Anything a developer makes could potentially be usable by the rest of the team. It allows the SmartEngine-developer to focus on updating and further improving the tool, and the DW-developers can create a script for what they need or want.

Loading the Data Vault

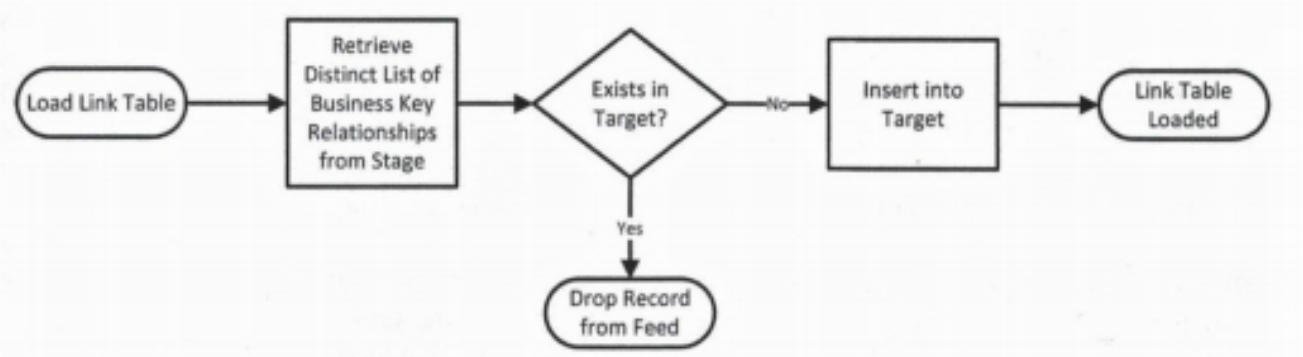
When loading a data vault, the hubs are loaded first. As seen in Figure 7, loading a hub is quite simple. All business keys must be unique in the hub and have a unique surrogate key. Hubs are not time-dependent, rows in a hub are never ended. Once a business key

has been loaded, it will remain in the hub even if it is never loaded again. All new business keys will be inserted, and the metadata for each business key present in a load batch will be updated (Linstedt & Olschimke, 2015). The structure of hubs is presented in chapter



- Template for loading a hub table (Linstedt & Olschimke, 2015, p. 434)

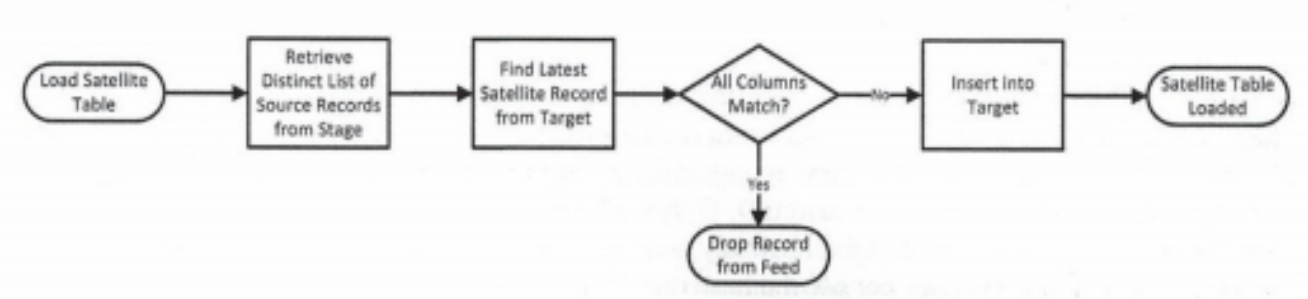
Once the hubs have been loaded, all links and satellites can be loaded in parallel. Figure 8 illustrates the steps in loading a link table. Because a link is modeled in a certain direction from one hub to another hub, there can be several rows in the link table for each source hub. Because of this, we need to maintain metadata about the validity of rows. Each row in the hub can only have one active referring row in the link. Because of this, when a new row is inserted for a business key, any rows not present in the load batch for that business key will be ended (Linstedt & Olschimke, 2015). The structure of links is examined more closely in chapter 4.2.5.



- Template for loading a link table (Linstedt & Olschimke, 2015, p. 449)

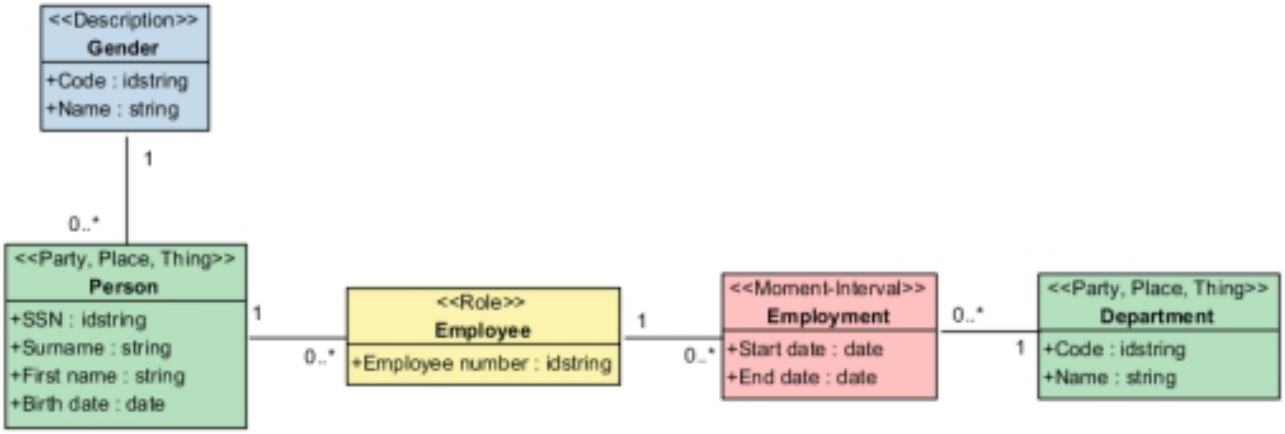
Loading the satellites involves slightly more work, as seen in Figure 9. When inserting new rows into the satellite, not only must the surrogate key of the hub be found, but every column of descriptive data must be compared with data already in the satellite table. If there is no row with exactly the data found in the staging area, it will be inserted. After insertion, all rows missing from the load batch for a loaded business key must be ended,

as with the links. For any given time, there can only be one active row in the satellite table, per business key in the hub.



Database objects generated by SmartEngine

SmartEngine is designed to create a data warehouse according to Data Vault 1.0, described in chapter 2.2. In this chapter, we will briefly examine the generated database objects and what rules they follow. To illustrate, a simple ER-model with five entities will be used to present the generated DW-objects.



FExample of five entities in an ER-model (Screenshot from Visual Paradigm, Eklund, 05.01.2018)

Figure 10 contains five entities, with direct relationships. A person always has one gender, and for any gender there can be zero or more persons. A person can be an employee, and can even be several different employees in the system, because his employee number might change if he has been absent from the company for some years. An employee can have zero or many employments, and an employment always has one employee. The employment is to a certain department, and a department can have several employments.

The entity Employment is necessary because an employee can belong to several departments over time, and departments can have several employees. This is a many-to-many relationship that cannot be modeled. This is solved by placing a time-related role entity in between.

The model is not complete and is missing many entities and several attributes, but it will be used to present the objects generated by SmartEngine and their structure in the database.

For a person, we have the attributes SSN (Social Security Number), first name, surname and birth date. For gender, we have a code and a name. For employee we have an employee number. For employment we have a start date and an end date. A more thorough examination of the ER-modeling standard the company uses, will be presented in chapter

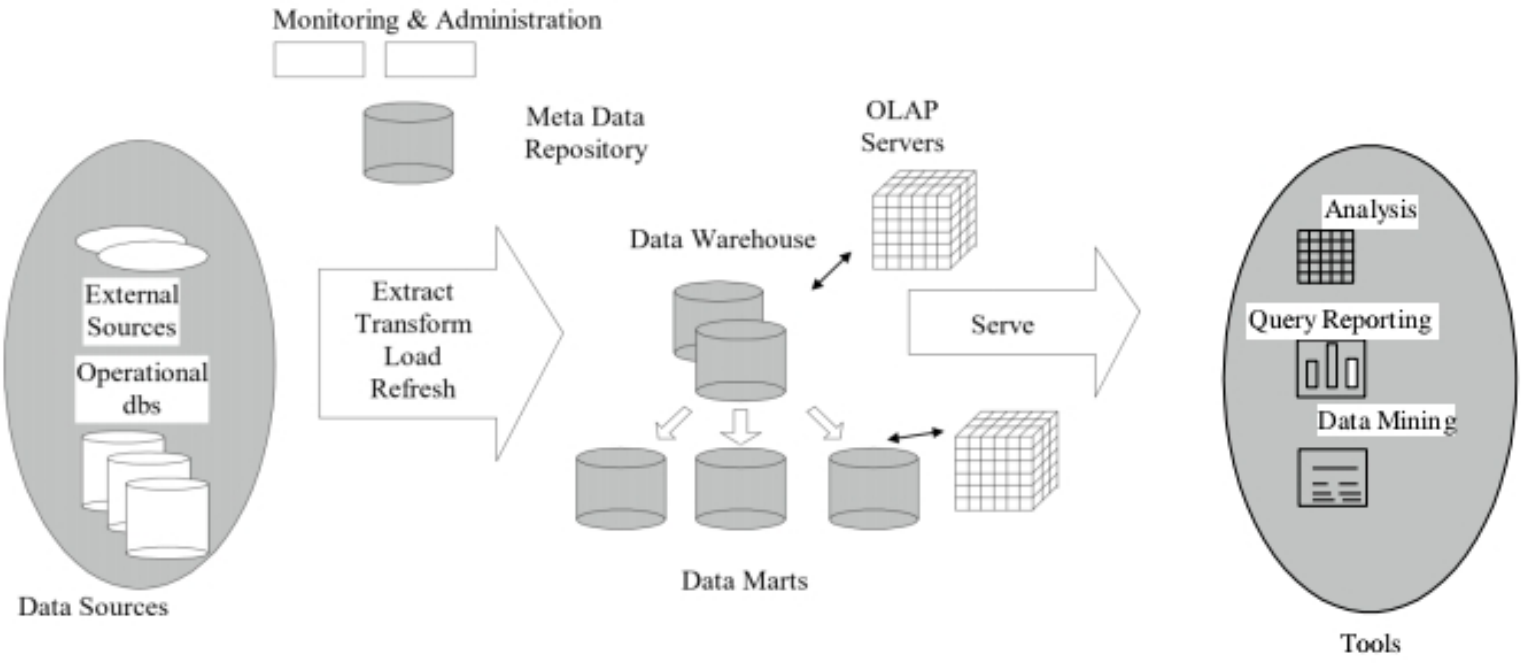
Being a Finnish company, most customers are Finnish and, therefore, the ER-models are primarily in Finnish. Internal column names will mostly be in Finnish, however some terms related to the Data Vault model theory are in English.

OUT-tables

For each entity in the model, an OUT-table is created in the staging area. The OUT-table will be the source for the loading procedures that SmartEngine generates. This implies that the source data needs to be inserted into the OUT-table, into the correct columns. Alternatively, the table can be replaced by a view with the same name, with the same structure as the generated table had.

Standard practice is to have all the columns in an OUT-table have the data type nvarchar. SmartEngine converts to the correct data type when loading the storage layer.

Data Warehousing Architecture



Two /Three – Tier Architecture

- **Warehouse database server**
  * almost always a relational DBMS rarely flat files.

- **OLAP servers**

  * **Relational OLAP** (**ROLAP**) extended relational DBMS that maps operations on multidimensional data to standard relational operations (GROUP BY operator)

  * **Multidimensional OLAP** (**MOLAP**) special purpose server that directly implement multidimensional data and operations

  * **Clients**
    - Query and reporting tools
    - Analysis tools
    - Data mining tools (e.g., trend analysis, prediction)

Warehousing Architecture

- **Enterprise Warehouse**: collects all information about subjects (customers, products, sales, assets, personnel) that span the entire enterprise

- Requires extensive business modeling

- May take years to design and build

- **Data Marts**: Departmental subsets that focus on selected subjects:

  e.g. marketing data mart: customer, sales, product

  - faster roll out, but complex integration in the long run

- **Virtual warehouse**: views over operational DBs

  - materialize some views (summaries)

  - easier to build

  - require excess capacity on operational DB servers

Operational Process

- **Data extraction**: tools, custom programs (scripts, wrappers)

  - extract data from each source

- more data and multiple sources could mean more errors in the data and harder to trace such errors

- Results in incorrect analysis

- Detecting data anomalies and rectifying them early has huge payoffs.

- Example:
  - inconsistent field lengths and orders

  - inconsistent description

  - inconsistent value assignments

  - missing entries

  - violation of integrity constraints

e.g. translate “gender” to sex”.

Warehouse Database Schema

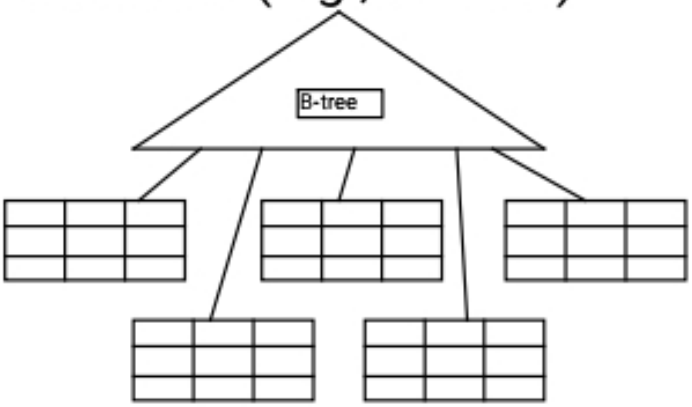
- Star schema

- A straightforward array representation has good indexing properties but very poor storage utilization when data is sparse.

- A **2-level approach** works better

  - identify one or more two dimensional array structures that are dense.

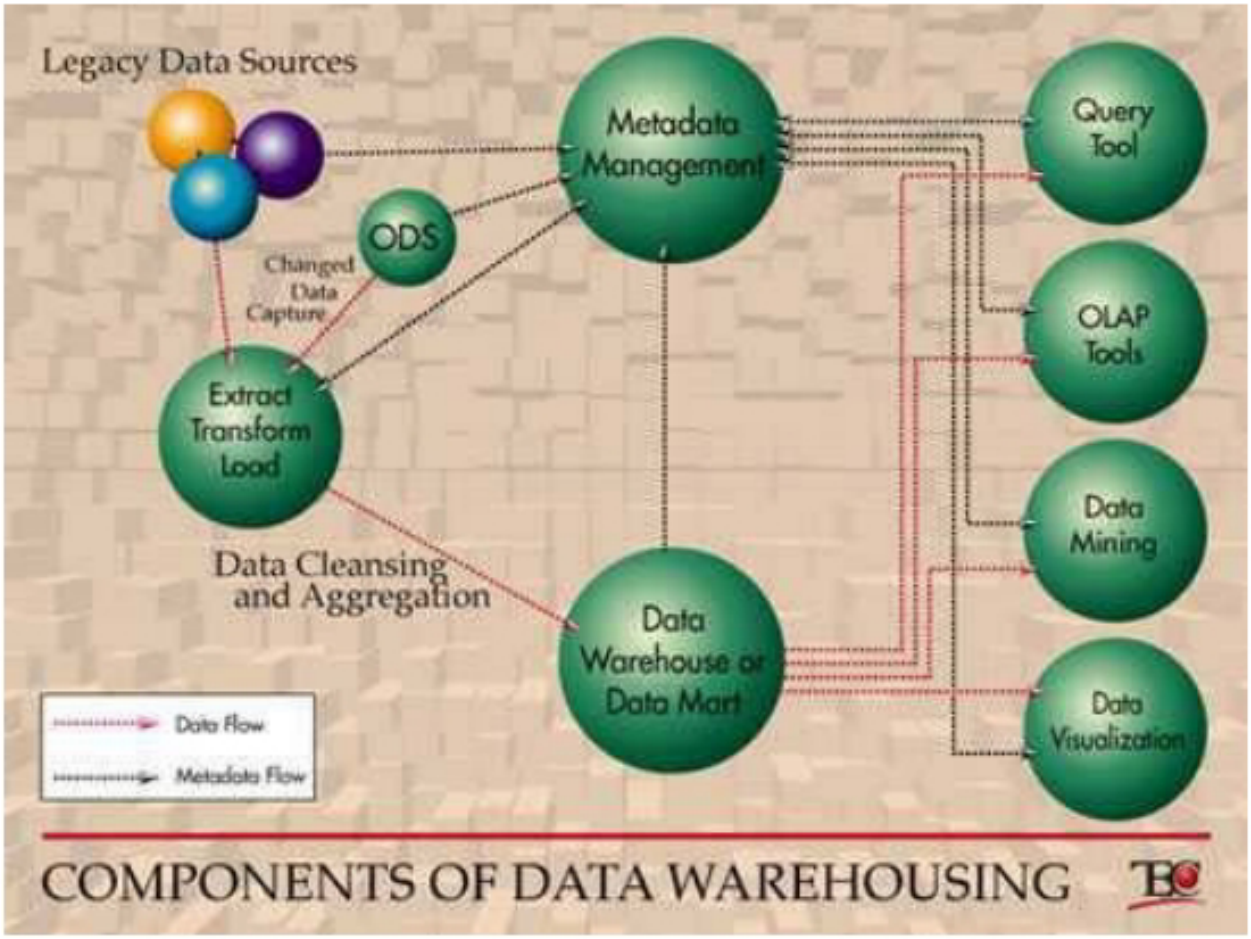
  - index to these arrays by traditional indexing structures (e.g., B+ tree)



(2 –dimensional dense arrays)

  - 2-level approach increases storage utilization without sacrificing direct addressing capabilities for “most parts”

  - **Time** is often one of the dimensions included in the array structures.



Data Warehouse

Decision Support and OLAP

- Information technology to help the knowledge worker (executive, manager) make faster and better decisions.

e.g. What were the sales volumes by region and product category for the last year?

e.g. List the top 10 best selling products of each month in 1996

- **On-line analytical processing** (**OLAP**) is an element of **decision support systems** (**DSS**)

reference: VLDB'96 tutorial notes by Chauhuri & Dayal
VLDB'97 tutorial notes by Schneider

user	Clerk, IT professional	Knowledge worker
Function	Day to day operations	Decision support
DB design	Application oriented	Subject-oriented
Data	Current, up-to-date Detailed, Flat relational Isolated	Historical Summarized Multi-dimensional Integrated, consolidated
usage	Repetitive	Ad hoc
access	Read/Write Index/hash on Prim Key	Read mostly Lots of scans
unit of work	short, simple transaction	Complex queries
#records accessed	tens	millions
#users	thousands	hundreds
DB size	100MB-GB	100GB-TB
metric	Trans throughput	Query throughput, response

Data Warehouse

- A decision support database that is maintained separately from the organization's operational databases.

- A data warehouse is

  - subject-oriented - integrated
  - time-varying
  - non-volatile

  collection of data that is used primarily in organizational decision making.

Why separate Data Warehouse?

- Special data organization, access methods, and implementation methods are needed to support multidimensional views and typical operations of OLAP.

  e.g. total sales volume of beverages for the western region last year.

- Complex OLAP queries would degrade performance for operational transactions.

- Function

  - **missing data**: DSS requires historical data, which operational DBs do not typically maintain.

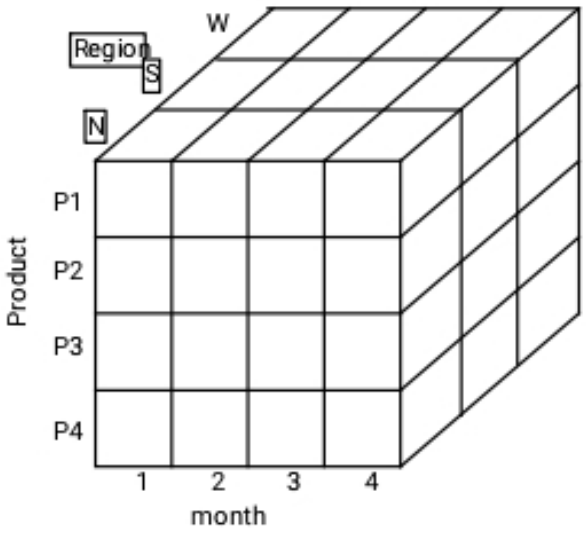
  - **data consolidation**: DSS requires consolidation of data (aggregation, summarization) from many heterogeneous sources: operational DBs, external sources.

  - **data quality**: different sources typically use inconsistent data representations, codes, and formats, which have to be reconciled.

Multidimensional Data

- Sales volumes as a function of product, time, and geography.

- Product, time, and geography are **dimension attributes** and sales volume is a **measure attribute**.



- Dimensions usually have associated with them **hierarchies** that specify aggregation levels and hence granularity of viewing data.

Operations

- **Roll up**: Summarize data
  e.g. total sales volume last year by product category by region.

- **Drill down**, **Roll down**: go from higher level summary to lower level summary or detailed data

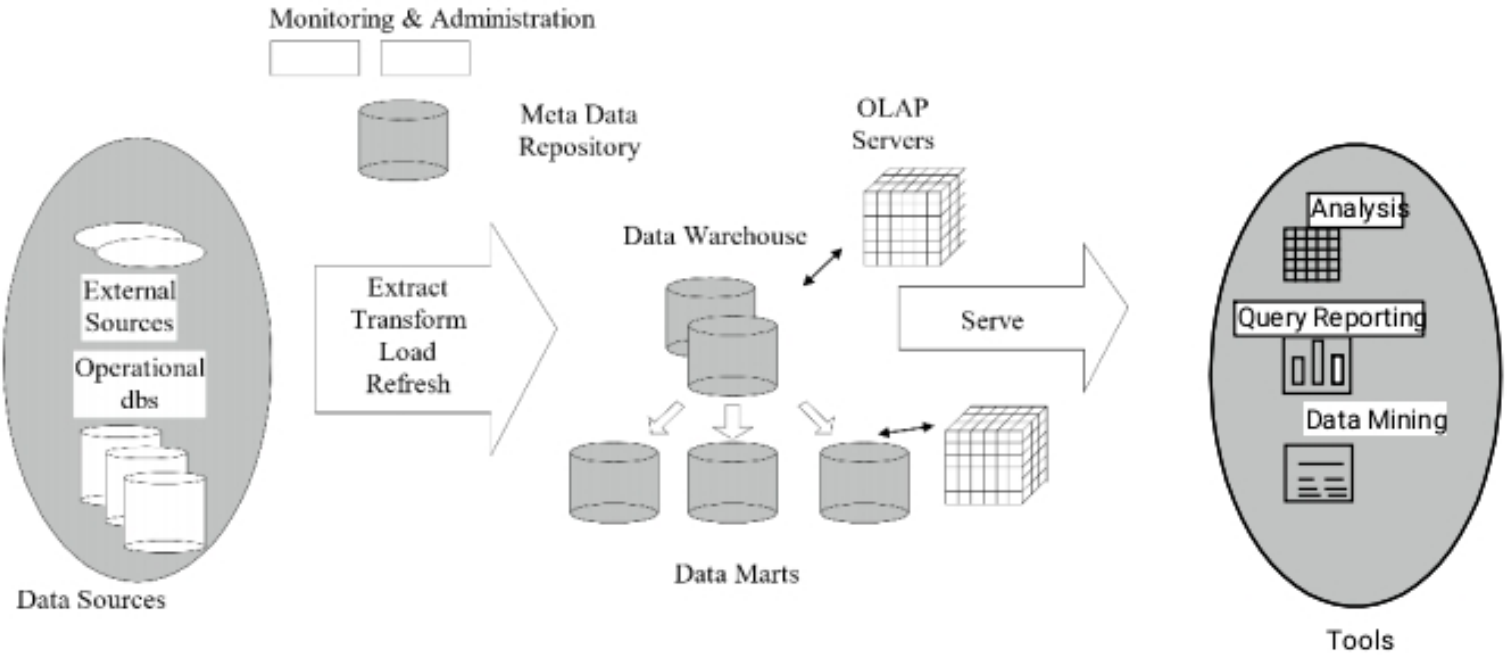
  e.g. For a particular product category, find detailed sales data for each office by date.

- **Slice** and **Dice**: select and project

  e.g. Sales of beverages in the west over the last 6 months.

- **Pivot**: rotate the **cube** to show a particular face

Data Warehousing Architecture



- extract data from each source
- cleanse transform, and integrate data from different sources

- **Data load and refresh**:
  - load data into the warehouse: load utilities
  - periodically refresh warehouse to reflect updates.
  - periodically purge data from warehouse
- **Build derived data and views**
- **Service queries**
- **Monitor the warehouse**

Data Cleaning

- Why ?
  - data warehouse contains data that is analyzed

- Requires extensive business modeling

- May take years to design and build

- **Data Marts**: Departmental subsets that focus on selected subjects:

  e.g. marketing data mart: customer, sales, product

  - faster roll out, but complex integration in the long run

- **Virtual warehouse**: views over operational DBs

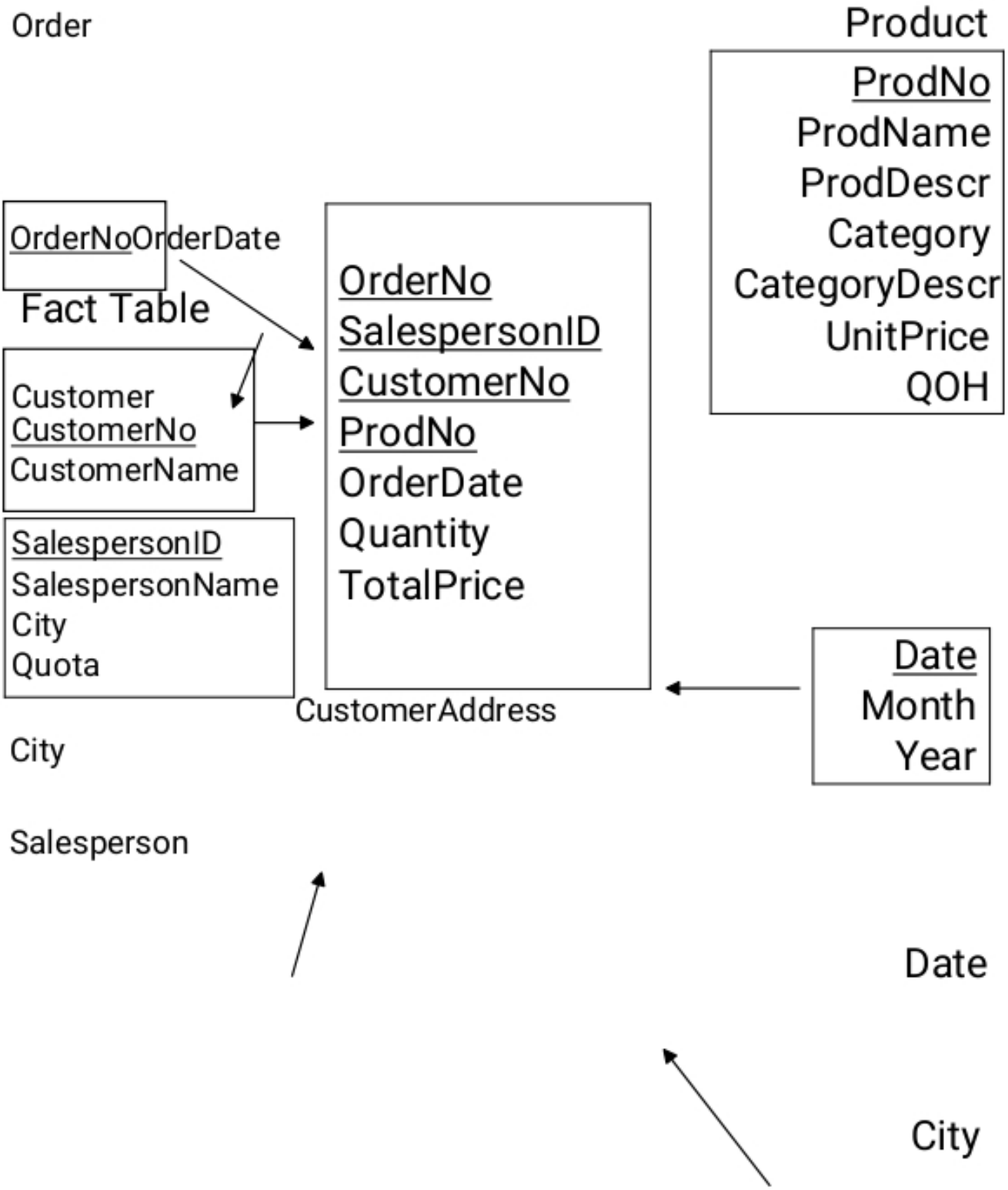
  - materialize some views (summaries)

  - easier to build

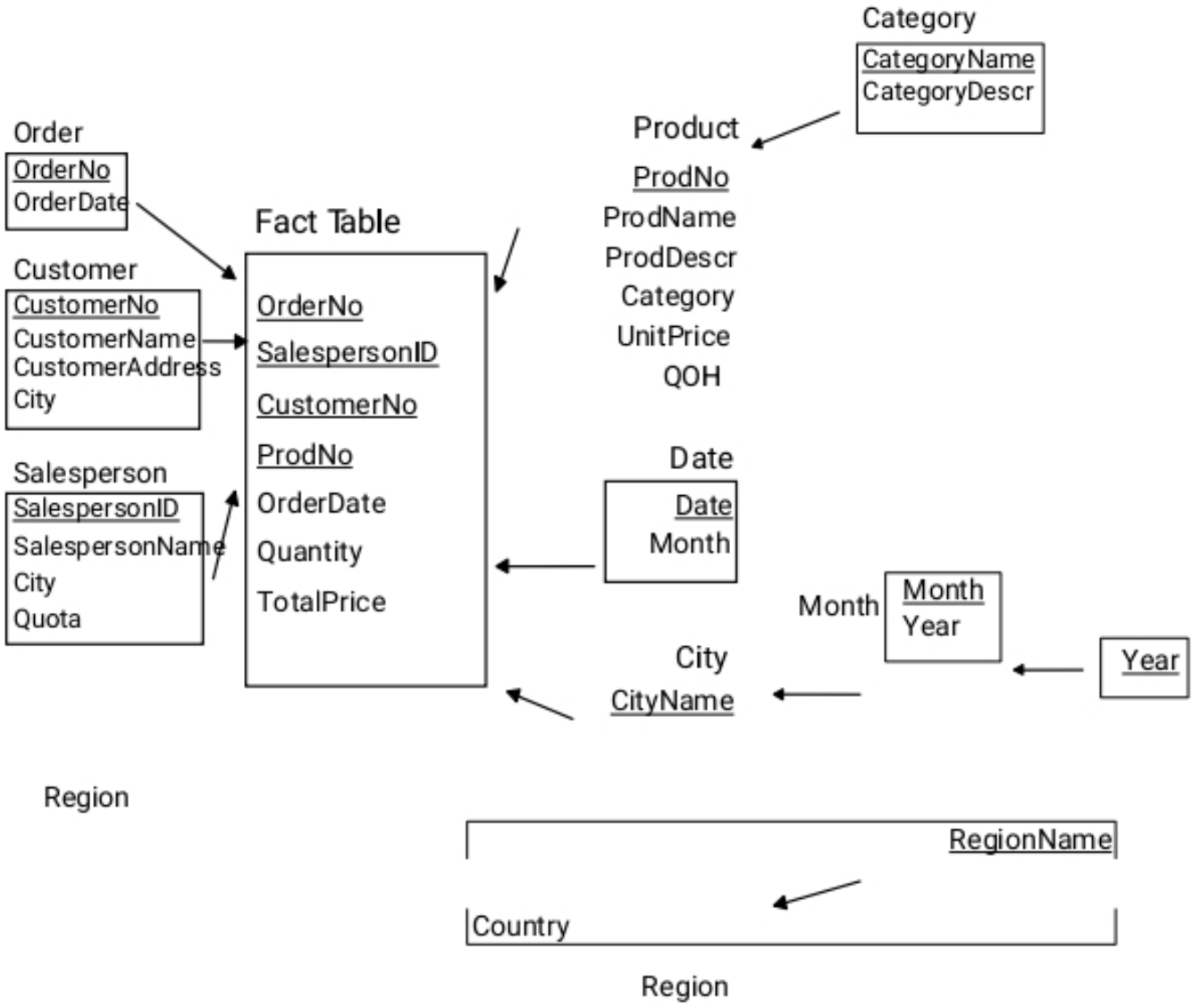
  - require excess capacity on operational DB servers **Operational Process**

- **Data extraction**: tools, custom programs (scripts, wrappers)

Star Schema


Order
Product
ProdNo
ProdName
ProdDescr
Category
CategoryDescr
UnitPrice
QOH
OrderNoOrderDate
Fact Table
OrderNo
SalespersonID
CustomerNo
ProdNo
OrderDate
Quantity
TotalPrice
Customer
CustomerNo
CustomerName
SalespersonID
SalespersonName
City
Quota
CustomerAddress
Date
Month
Year
City
Salesperson
Date
City

Snowflake Schema



- Represent **dimensional hierarchies** directly by normalizing the dimension tables
- Easy to maintain

Multidimensional OLAP (MOLAP) servers

- The storage model is an **n-dimensional array**.
- Direct addressing abilities
- Front end multidimensional queries map to servers capabilities in a straightforward way.
- Problem: handling sparse data in array representation is expensive

Product	1	2	3	4	5	6	7	8	sum
sum	30	40	20	20	30	40	10	20	210
	20	30				10			60
			20		10		10		40
P4	10			20		30		20	80
P3		10			20				30
P2									
P1									

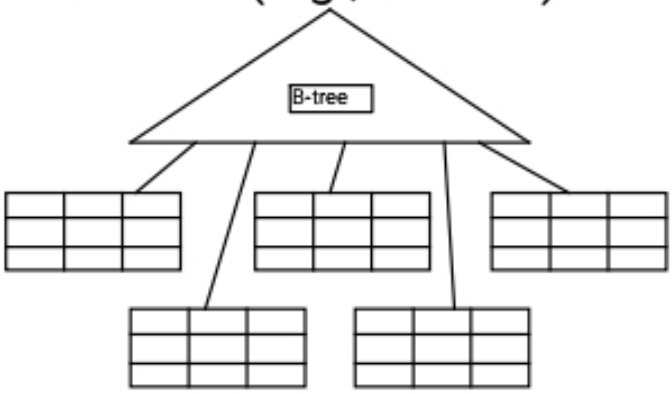
Date

- A straightforward array representation has good indexing properties but very poor storage utilization when data is sparse.

- A **2-level approach** works better

  - identify one or more two dimensional array structures that are dense.

  - index to these arrays by traditional indexing structures (e.g., B+ tree)



(2 –dimensional dense arrays)

  - 2-level approach increases storage utilization without sacrificing direct addressing capabilities for “most parts”

  - **Time** is often one of the dimensions included in the array structures.

- Data extract clean, transform, refresh

CA-Ingres Replicator	Carleton passport
Evolutionary Tech Inc. ETI-Extract	Harte-Hanks Trillium
IBM Data Joiner, Data Propagator	Oracle 7
Platinum InfoRefiner, InfoPump	Praxis OmniReplicator
Prism Warehouse Manager	Redbrick TMU
SAS Access	Software AG Sourcepoint
Sybase Replication Server	Trinzic InfoPump

- Multidimensional Database Engines

Arbor Essbase	Comshare Commander OLAP
Oracle IRI Express	SAS System

- Warehouse Data Servers

CA-Ingres	IBM DB2
Information Builders Focus	Informix
Oracle	Praxis Model 204
Redbrick	software AG ADABAS
Sybase MPP	Tandem
Terdata	

- ROLAP Servers

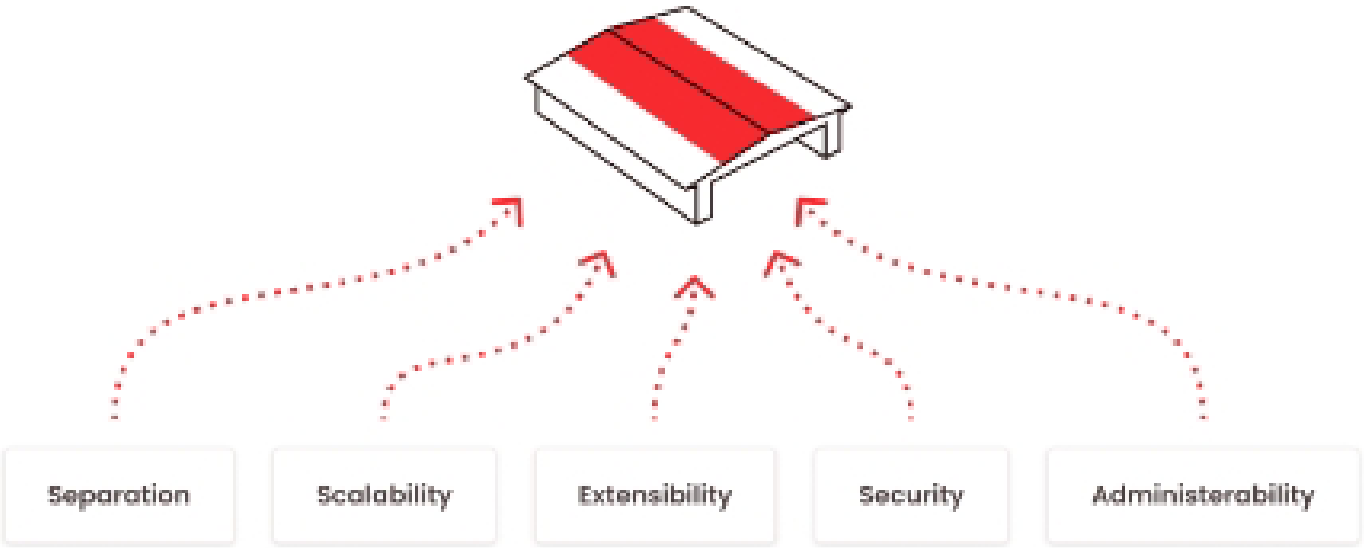
HP Intelligent Warehouse	Information Advantage Asxys
Informix Metacube	MicrosSrtategy DSS Server

- Query/Reporting Environments

Brion/Query	Business Objects
Cognos Impromptu	CA Visual Express
IBM DataGuide	Information Builders Focus Six
Informix ViewPoint	Platinum Forest & Trees
SAS Access	Software AG Esperant

- Multidimensional Analysis

Properties of data warehouse architecture



Cloud service layer

Administration

Metadata

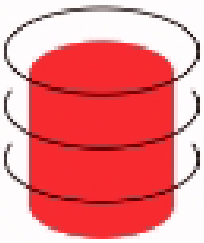
Security

Computing layer Or Query Processing layer

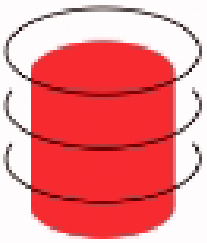
RAM

vCPU

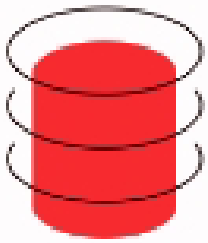
Storage layer



Database



Database



Database