Data Warehouse Fundamentals for Storage Professionals – What You Need To Know

EMC Proven Professional Knowledge Sharing 2011

Bruce Yellin
Advisory Technology Consultant
EMC Corporation
Bruce.Yellin@EMC.Com

Table of Contents

Introduction

It might have started with a specific question, such as “Can we sell more beer to men when they buy diapers?”, “How do we increase cell phone customer loyalty?”, or “What do we need to do to sell more Chevys?”. Maybe the question focused on expense controls such as “Could credit card fraud be prevented when gas is bought in a customer’s hometown of Miami, Florida followed by the purchase of airline tickets at a Fairbanks, Alaska ticket counter minutes later?”

Businesses, scrambling to stay afloat in this difficult economy, also ask general questions. Their annual reports include vision statements with phrases such as "the need to grow market share and revenue" and “improve profitability and invest in the future”. Regardless of type, senior managers look to their data warehouse for help in answering these tough business questions.

Data warehouses, sometimes referred to as a Decision Support System (DSS), are found in almost every organization, including the government, medical, Internet, phone, financial, and retail industries. Data flows into a DSS from internal transactional databases and external sources including cell phones, GPS devices, TV set-top boxes, ATMs, credit card usage, RFID merchandise chips, medical images, the U.S. Census Bureau, and a host of others. Google Analytics even supplies detailed data on the effectiveness of corporate advertisements by examining a customer’s web activity as they browse online catalogs and make purchases.

This concept of turning data into actionable information started about 30 years ago and is critical to companies like eBay, Facebook, Walmart, Bank of America, and Dell. Small data warehouses may use less than a terabyte of disk storage and large ones can exceed a petabyte. With over 9PB of DSS content, eBay analyzes customer buying patterns and believes that “knowledge is power”. Walmart adds a billion rows of data hourly to their system[1]. Decision support ranks among the highest of business priorities[2], and if your organization has a data warehouse, it has probably doubled in size over the last few years as data sources are added and new uses arise.

Regardless of your IT role, when your system contains historical <u>and</u> detailed data for every transaction, the amount of storage required can be intimidating. eBay’s data warehouse, even with the largest drives available today, probably needs over 6,000 disks just to contain it, never mind back it up, maintain it, or grow it by 50 percent a year.

The critical nature of a DSS and its size means you need to know more about “the nature of the beast”. If some of the terminology is new to you, rest assured that these pages will demystify data warehousing. It will turn you from a novice to an informed member of an active warehouse team, allowing you to leverage your current storage expertise and newly acquired insight. This paper discusses data warehousing and data marts, reviews design concepts, steps to feed it, systems that feed off of it, backup, maintenance, virtualization, and the cloud, NAS versus SAN, reliability, availability, scalability, performance, and disaster recovery/business continuity.

Data Warehouse Background

What Is a Data Warehouse?

Most of your company’s dealings are handled by online ***transaction*** processing (OLTP) applications that store information in Oracle, Sybase, SQL Server, DB2, Informix, or other brands of databases. OLTP is tactical in nature. These transactions are your organizations’ lifeblood resulting in business “deals” that could include airline reservations, online purchases, ATM withdrawals, health record updates, or even grocery point-of-sales.

Yet when it comes to finding ways of improving the business, increasing profitability, retaining customers, and other noble tasks, these transactional systems fall short. OLTP systems are great for detailing every minute aspect of the transaction, but can not determine optimal traffic patterns in a retail store, proactively help the design of advertising and marketing campaigns to find new, highly profitable customers, or increase the market basket of existing loyal shoppers. That is where online ***analytical*** processing (OLAP) and data warehouses shine.

While it’s true that transactional and analytical systems both use databases, including data tables, indexes, and keys, it is how the data is organized and how the data is processed that defines them. The difference begins with the design of the database schema and tables, whether two or more distinct data sources are being merged, and generally whether the data can be processed in rows or columns. You do not necessarily need to purchase new software, but you will likely need to organize it differently.

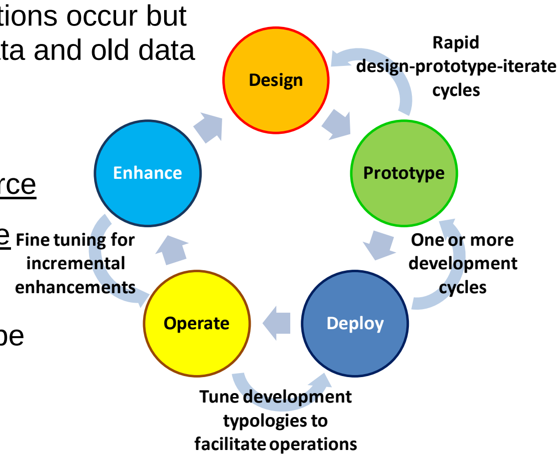
The data processed by transactional systems is usually row oriented, similar in nature to the old punch card or a row in Excel. A row design allows new data to be efficiently **INSERT**ed, existing data **UPDATE**d, and old data **DELETE**d. OLTP data is usually accessed or viewed with a single dimension (e.g. credit card number) whereas OLAP data must be viewable in multiple dimensions (e.g. geography, time, and product). Transactional systems avoid duplicating data while a DSS leverages duplication to increase query speed. OLTP has current data and transactional details. OLAP is historical in nature and has summarized data.

In terms of data organization, transactional systems are typically designed using entity-relationship diagrams (ERD) or object-oriented models, such as products and prices, invoices and amounts, or clients and names. When the business asks different types of questions, the data needs to be structured differently to answer them optimally, for instance, “What is the gross margin for all the cars sold by model in China and North America?” This type of question needs to span product types and map them to sales regions to calculate margins. Analytical systems easily answer these types of questions because of how the data is organized. A good example of this is a star schema design–more on this in the next section.

Over the last few decades, data warehousing has had many definitions, but the classic explanation comes from the “father” of the data warehouse, Bill Inmon. He says it is an integrated, subject-oriented, time-variant, nonvolatile database that supports decision making:

- **Subject-oriented**–focus is on attributes of the company (customers, sales and products) and not applications so data relating to the same event links together.
- **Integrated**–various data feeds are brought together into a common, consistent format with other data–i.e. a single date or currency format.
- **Time-variant**–data accuracy and validity is based on a point in time and could represent snapshots of the business. This allows reports to show changes over time.
- **Non-volatile**–data is typically not updated as transactions occur but regularly in batches. New data is added to existing data and old data remains untouched.

To simplify this, a warehouse is a set of processes and source data feeds collected into a database for the primary purpose of helping a business analyze data to make decisions. This illustration shows that the path between design and prototype



can be iterative, and the same holds true for the transition to deployment, operations and enhancement.

A warehouse phrase you will hear is ***denormalized,*** meaning redundant data can be stored to improve database READ performance. Conversely, OLTP databases are ***normalized*** to remove duplication and save space. A warehouse might store a customer's name in multiple tables, such as an "order" and "shipping_info" table while OLTP keeps it in just a customer_table.

While transactional databases are "hot rods" in terms of WRITE speed and high processing rates, data warehouses are "family cars" in terms of READ speed, providing a platform for data marts and support analytics that make sense of data, usually bulk loaded after the transaction occurs. A DSS allows data relationships to be discovered by searching for patterns across parts of an organization. Their purposes are also different–a clerk needs to record the sale of three widgets for $234.56 plus tax, while the manager needs to analyze sales patterns over time.

Comparison	Transactional Database (OLTP)	DSS (OLAP)
Purpose	Day to day operational processing	Decision Support, analytic
Usage	Data capture and reporting	Inference
Users	Clerk or IT professional	Knowledge worker
Transactions	Typically simple and short	Typically complex queries
# Records	Usually few to tens	Can be into the millions
Optimization	Designed for high "write" rates	Designed for high "read" rates
Record types	Raw, detailed transactions	May be messaged, summarized
Authentication	Validate data before record commit	Validation occurs during ETL
Timelines	Data is to the second as transaction occurs	Data updated every hour, day, week or month
Data attributes	Current, flat, relational, compartmentalized	Historical, static, summarized, multidimensional, consolidated
Processing	Repetitive	Ad-hoc, unstructured
Sizing	Designed for operational efficiency. Big is 100's of GB to a few TB	Data warehouse fed by many sources. Can be >100TB
Table joins	Many	Rarely, some
Duplicate data	No (Normalized)	Yes (Denormalized)
Derived data	Rare	Usual
Indexes	Few	Many

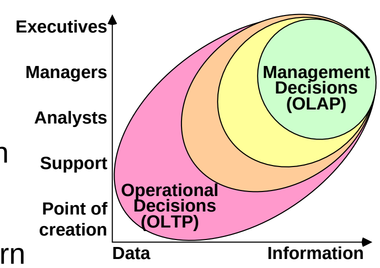
OLTP systems interact with dozens to thousands of end-users, performance expectations are high, and redo logs account for a great deal of I/O. Typical databases can expect a 10 to 20 percent daily change rate and cached storage frames can help performance, especially when the data is focused on a small number of volumes. Records are small and random READs are the norm.

DSS on the other hand usually involves complex queries and reporting for a handful of users. Users are not expecting millisecond response time since the design focuses on high throughput. With massively parallel processing (MPP) systems, queries can be divided into parallel tasks to independently access logical volumes. Typical DSS READs are sequential and involve a large amount of data, especially with full table scans. Warehouses are often updated in batch and redo logs play a small role when doing a WRITE. The change rate for a large DSS is typically well under 5 percent a day.

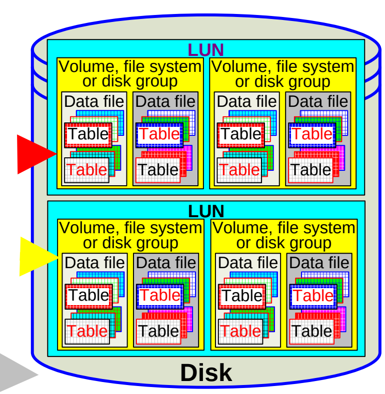
Modest analytical requirements may work well with a transactional database design, but as DSS value increases and the queries become more complex, the design becomes a significant issue. If you ran a multi-dimensional query with table joins (discussed in the schema section) on an OLTP system, it could slow to a crawl and impact transactional speed. A DSS brings disparate data together while transactional systems are narrow in scope. It is analogous to driving a screw into a wall with a hammer–it works, but if you have a lot of screws, you want to use a power screwdriver.

DSS value comes from being able to answer business questions like:

- Which customers give us our highest and lowest margin sales?
- What promotions can we run to increase revenue?
- What is the revenue and margin impact when we offer product XYZ to our customers?
- How can we prevent our loyal customers from "jumping ship" to our competition?
- What is the market basket profile of banking customers?
- Will sales volume increase or decrease if we raise prices 3 percent?
- What real-time credit card profiles can detect fraud before the purchase completes?



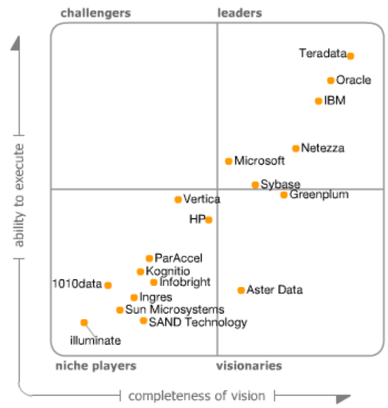
In the end, the significance we place on data and information often depends on your job function. While data tends to be tactical, turn raw data into information and it becomes strategic. OLAP helps turn OLTP transactional data into actionable management decisions.



From a storage standpoint, let's review the general relationship between a database and its storage. As shown, the basis of the database is the table. Tables are grouped together by function and placed into a volume, file system or disk group based on the architecture. Volumes are put in a Logical Unit (LUN), which are assigned to a disk drive. The drive gets RAID protection and possibly striped across RAID groups for better performance.

Gartner's "magic quadrant" of DSS thought leaders[3] include Teradata, Oracle, and IBM, all of whom have products going back over 20 years. Netezza was recently acquired by IBM and

EMC purchased Greenplum®. Others like Infobright, Kognitio, ParAccel, Veritica, and Aster Data Systems were founded about six years ago.



DSS vendors may use MPP techniques to divide the work such as a full table scan across multiple CPUs, memory, and storage to speed up operations. Others use general purpose designs, hardware appliances, columnar technology[4], or combinations of features.

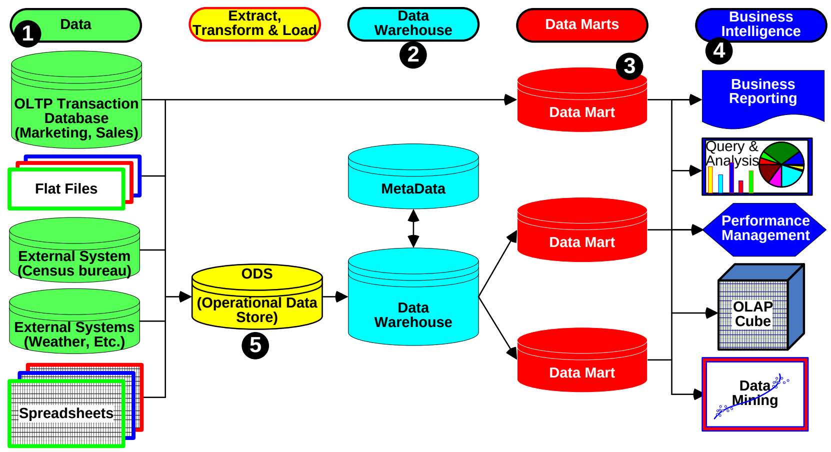
An appliance is a self-contained device that may combine all aspects of the data warehouse (extract, transform and load (ETL), the warehouse database, data marts, and so on.) or just offload a portion of the task such as the warehouse itself. Either way, appliances do not signal the end of system administration. There are still configuration issues, storage provisioning, backup and restore, part replacement, and a host of other storage administration challenges.

Data Mart Defined

Given the warehouse is a single repository of all business data, a data mart (tiny warehouse) is a <u>special purpose</u>, <u>specific</u>, <u>subject oriented</u>, or <u>departmental</u> subset of information built from the warehouse, independent sources, or OLTP systems (see Inmon vs. Kimball below). For example, a personnel department's data mart may be fed by the main warehouse, yet accessible company-wide via customized reports and dashboards. While the same data likely exists in the warehouse, you might not want thousands of users running queries against it. Since it acts on a subset of data and performs perhaps predefined functions, its performance requirements could be radically different than the full warehouse. Marts often have a *many:one* relationship to a warehouse.

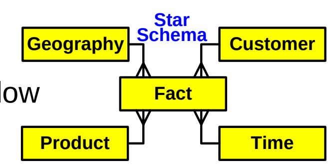
Comparison	Data Warehouse	Data Mart
Scope	- Application independent - Enterprise or Central - Planned	- Application specific - Departmentally decentralized - Evolved of planned
Data	- History, detail, summarized - Some denormalization	- Some history, detail, summary - High denormalization
Subjects	- Multiple	- Single
Source	- Many internal and external sources	- Few internal and external sources
Other	- Flexible - Data oriented - Long life - One complex structure	- Rigid - Project oriented - Limited life - Many simple structures, may become complex

In this data flow, source systems❶ are transformed instantly or in batch cycles, and then loaded into the warehouse❷. Specific data is put into data marts❸, and analyzed❹. Some environments use Operational Data Stores (ODS)❺ as the focal point for real-time transformation prior to warehouse integration.



Schemas and Data Models

All databases, including those used in a data warehouse, employ a formal “blueprint language” called a schema that lays out data tables, fields in the tables, and how those fields and tables interact. A schema needs to look familiar to the warehouse user, so simplicity is important.



Most data warehouses use “star” schemas where dimension tables surround a central fact table and are shaped like a star. The design below is a “fact” table and four “dimension” tables – geography, customer, product, and time. A fact table contains numeric facts (*measurements*) such as quantity and currency, and pointers (*foreign keys*) to dimension tables. Fact tables can have billions of rows of information, all accessible by key dimensions such as employee_code, product_code, and so on. A dimension table contains descriptions of the facts. When a *join* is needed to respond to a

Geography Dimension	employee_code	emp_name	city_code	city	state_code	state
	101	Bruce	25	Miami	24	Florida
	123	Sally	34	Chicago	12	Illinois
	134	Jim	34	San Diego	41	California
	201	Alice	15	Houston	33	Texas

Customer Dimension	customer_code	cust_name	age_code	age	sex_code	city_code	city
	3325	Jim King	M	43	M	25	Miami
	9654	Art Fern	Y	23	M	211	Abilene
	54112	Amy Wit	O	62	F	9	Austin
	71243	Pam Mun	M	41	F	1034	Dayton

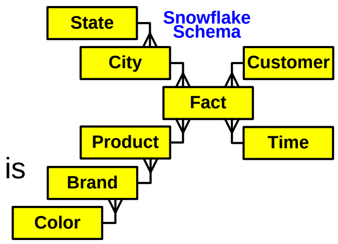
Fact Table	employee_code	product_code	customer_code	day_code	units_sold	revenue
	101	602	71243	34	7	1284
	123	600	54112	51	8	8192
	134	610	9654	200	5	475
	201	608	3325	120	3	1099

Product Dimension	product_code	prod_desc	brand	color_code
	600	Sweater	Chaps	4
	602	Jeans	Levis	7
	608	Shirt	Arrow	21
	610	Belt	Buxton	3

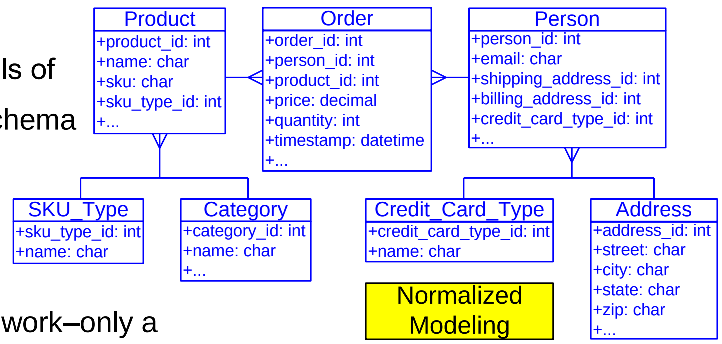
Time Dimension	day_code	date	day_of_week	month_num	month_short_name	quarter	year
	34	02032011	Thursday	2	Feb	1	2011
	51	02202011	Sunday	2	Feb	1	2011
	120	04302011	Saturday	4	Apr	2	2011
	201	07192011	Tuesday	7	Jul	3	2011

query, it means a fact and dimension tables are “glued” together using the fact table’s *primary key* and the foreign key of one or more dimension tables. *Indexes* can speed up data access and are usually associated with a table. A *view* is a preset common query against tables in a schema.

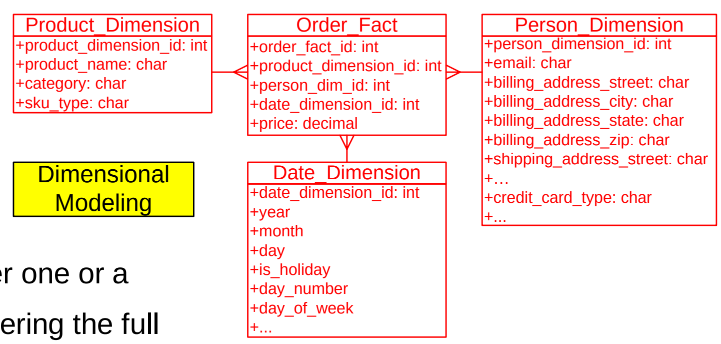
Snowflakes, fact constellation schemas, hybrids, and optimal snowflakes can also be used to represent the data organization. Fact constellation schemas are used when multiple fact tables share dimension tables. Knowing which schema to use in your environment is a topic best left for your data warehouse architect.



Now that you know a little more about a schema, let's go back to the fundamentals of OLPT versus a DSS. This is an OLTP schema of a hypothetical transactional database and employs a "normalized" model in 3rd normal form[5]. Normalization is similar to how data deduplication backup systems work–only a single copy of data is maintained and subsequent repetitions are pointers to the original. So when a new product SKU is created, only a single row is updated–everything else is a point to the sku_type_id. These schemas are design to handle thousand of real-time transactions a second because there is little overhead in creating or updating rows of new information.



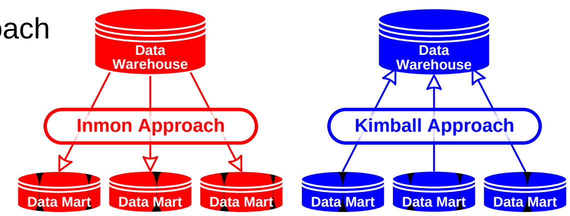
Compare this to the data warehouse's star schema–it is like the OLTP schema, only flattened and organized a bit differently. This "dimensional" design is optimized for queries rather than data updates. Dimensional models have either one or a few fact tables, with dimension tables offering the full description of what is in the fact table. That is why a data warehouse is verbose (*denormalized*).



Data Warehouse Design – Top Down or Bottom Up?

Before an organization starts a DSS project, a likely initial question is: which should be constructed first, the data warehouse or the data mart? In short, it depends on whether your architect believes the teachings of Bill Inmon or the diametrically opposed views of Ralph Kimball. Each camp has strong views that often resemble a religious war. Regrettably, this is a very complex issue well beyond the scope of this paper.

Briefly, Mr. Inmon's top-down data warehouse approach stores the lowest level of normalized data in the warehouse with data marts holding departmental or specific views of data–a grand-plan approach.



Mr. Kimball advocates a bottom-up design where data and analysis tools are assembled for specific needs such as the sales department. Eventually, data marts are integrated into a central warehouse as they collect data from within a company. This chart summarizes the pros and cons of each approach.

Top-down Design		Bottom-up Design	
Pros	Cons	Pros	Cons
Corporate wide endeavor	Longer project	Fast, easy design broken into small groups	Marts are often narrow in scope
Designed from start rather than evolved	High risk, highly complex	Results evident quicker	Duplicate data betweens marts
Single data repository	Large teams	Lower risk	Independent data views can contradict or be irreconcilable
Well defined	Expensive	Can prioritize which marts come first	Favors unique tools and look-and--feel
Easier to enhance and enrich	ROI from results can take a while	Gradual ramp up for BI team	Disparate teams, duplicate equipment

A DSS can also be a hybrid combining aspects of top-down and bottom-up, or federated (uniting multiple warehouses, marts, and other systems). Regardless of the approach, multiple data sources are brought together building as complete a "picture" of the enterprise as is realistically possible. As data is gathered, it is "cleaned", which leads to a critical aspect of DSS – ETL.

Extract, Transformation, and Loading (ETL)

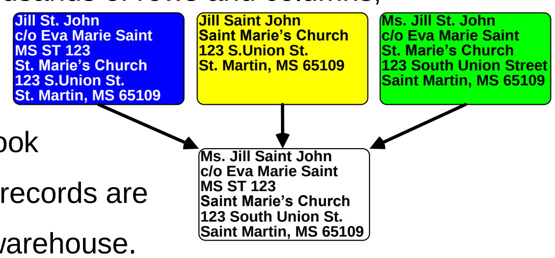
If all of your organization's data was standardized, it would be easy to merge the data together. However, workers keep data in spreadsheets, MS-Word, PDF documents, plain text, MS-Access databases, and others. Without guidance and enforcement, data can take on different meanings, formats, and representation. ETL is a resource-intensive, on-going effort to keep new data coming into the DSS in a consistent manner. This process collects data from various sources, integrates it with other operational data using business rules, and adds it to the database so it reflects a "single version of the truth"–i.e., data values with unique definitions.

Avenue	Center	Circle	Expressway	Street
Av	Cen	Cir	Exp	St
Ave	Cent	Circ	Expr	Str
Aven	Center	Circl	Express	Street
Avenu	Centr	Circle	Expressway	Strt
Avenue	Centre	Crcl	Expw	
Ave	Cntr	Crcle		
Avnue	Ctr			

Achieving a common data view when merging sources can be a daunting task. Just look at the problem that names and addresses cause the U.S. Post Office. When optically scanning postal mail, they recognize seven spellings for Avenue[6]. The problem is further complicated when key words and abbreviations must be determined contextually. For example, a brute force transformation of this fictional address leads to a lot of confusion–in context, is “MS” a salutation, mail-stop, or state abbreviation? Is “ST” part of a person’s middle or last name, part of a church name, a town name, or a street name?

Ms. Jill **St.** John	Salutation or **State**, name or **Street**
c/o Eva Marie **Saint**	Name or **Street**
MS ST 123	Mail-stop or **State** or **Street**
St. Marie’s Church	Church name or **Street**
123 S.Union **St**.	**Street**
St. Martin, **MS**	Name of a town or **Street, State of Mississippi**

When you have dozens of data sources, each with thousands of rows and columns, transforming it requires a lot of compute power. DSS commonly uses name and address reconciliation as data is merged from sources or tables. For example, look at the complexity of determining if these “Jill St. John” records are the same, or even which one should be stored in the warehouse.



If you are part of a multi-national organization, you have likely experienced the confusion caused by various currency and date formats. While currency is straight forward, the effort to standardize dates continues. Depending on where the data comes from, you can see dates with dots, hyphens, leading zeroes, and abbreviations for the “mm” such as “AUG” or August. Here are some formats used by just three nations and what February 2, 2010 looks like. While 2022010 is February 2, 2010 no matter how you interpret it, 3022010 is either February 3, 2010 or March 2, 2010. Before data enters the warehouse and reports and analysis are done, this confusion must be cleared up.

America mm/dd/yyyy	United Kingdom dd/mm/yyyy	China yyyy/mm/dd
2/2/2010	2/2/2010	2010/02/02
2.2.2010	2.2.2010	2010.02.02
2022010	2022010	20100202
2/2/10	2/2/10	10/2/02
Feb 2, 2010	2 Feb, 2010	二零一零年二月二号

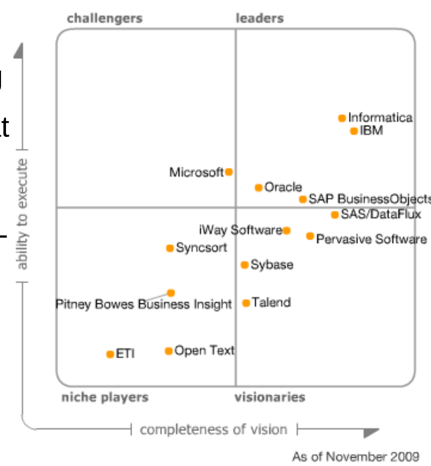
Data classification is an ETL process that converts values to defined ranges. In this sample of 20 names and ages, the goal is to group data by age ranges and sex. The age range is obvious, but determining male and female is not. Names prefixed by “Mr.”, “Ms.”, “Mrs.”, or “Miss” are clear-cut, but those on the

Line	Classification Table		Age Range				Sex		
	Name	Age	20-30	30-40	40-50	>50	Male	Female	Unk.
1	Ms. Sherri Melton	31		1				1	
2	Karen Puckett	40			1			1	
3	Mrs. Elsie Hamilton	31		1				1	
4	Sandy Raynor	41			1				1
5	Mr. Neal Lawrence	61				1	1		
6	Mr. Eric Steele	25	1				1		
7	Franklin Vick	65				1	1		
8	Miss Marcia Walsh	47			1			1	
9	Mr. Dwight Monroe	52				1	1		
10	Mr. Neal Middleton	47			1		1		
11	Ms. Gretchen Goldstein	34		1				1	
12	Mrs. Shelley Weeks	28	1					1	
13	Mr. Douglas Ross	26	1				1		
14	Katherine Bender	56				1		1	
15	Paul Woods	56				1	1		
16	Ms. Patricia Mangum	36		1				1	
17	Mr. Louis Rosenthal	47			1		1		
18	Harvey Underwood	33		1			1		
19	Miss Shirley Merritt	65				1		1	
20	Don Sharpe	44			1		1		
	Totals		**3**	**5**	**6**	**6**	10	9	1

highlighted purple lines lack prefixes and must be checked against a list of male and female first names. Names like “Sandy” on line 4 are unisex names (male or female), so an “Unknown” category is needed to insure warehouse integrity.

The general ETL steps for getting clean data into your warehouse are:

1. **Extract** data from numerous flat files, electronic feeds, and databases sources. The data may be in different formats.
2. **Transform** the extracted data into the agreed format using business rules. This step can employ data filters, sorting, data joins, aggregation, data cleansing, calculation, validation, and other processes.
3. **Load** the transformed data to update the warehouse or data marts, either in real time or less frequently with batch processing.



Processed data needs testing before warehouse integration to avoid contamination. Think of the ETL complexity of combining 100 Oracle tables with parts of 50 Sybase tables, dozens of flat files and spreadsheets where the first and last names need reordering, and converting birthdates to a common format. ETL can account for more than 60 percent of the DSS I/O activity. When low performance sub-systems are used, ETL can slow down the entire system. Gartner’s[7] list of the premier ETL vendors shows the leaders are Oracle, IBM, Informatica, and SAP.

Why You Build a Data Warehouse: Business Intelligence

Simply loading a database from disparate sources does not make it a DSS. Your organization must derive value from it through data analysis. The following examples illustrate how vast amounts of warehouse data can become actionable information to gain competitive advantages, improve efficiency, and lower costs. To do that, techniques including business queries and reports, business analysis (OLAP), data mining, scorecards, and dashboards are used.

- When working for a DSS vendor, one of our customers was a large phone company. Cell phones used to be very expensive and they wanted to attract customers who would generate a lot of revenue and have a low churn (turnover) rate. One of our observations, original or not, was that families ordering take-out pizza twice a week often implied the parents led busy lives and likely had children at home–a great cell phone profile. To prove our hypothesis, we combined Call Detail Records, local pizzeria receipts, and other public domain information. Our customer found the information very valuable.

- Catalina Marketing is an in-store behavior marketing company. When you receive their coupon at a grocery checkout, it was because their DSS analysis of your purchases showed you would likely buy an item they were promoting. Rather than send everyone a coupon, they focus direct marketing costs on the segment showing a propensity towards buying that product. They have an incentives redemption rate of 6.3 percent, or 8X greater than other methods. Their 2.5PB DSS "…raised ad awareness by 16 percent, improved ad recall by 24 percent, and drove volume lift by 35 percent"[8]. As a simple example, your shopper card may show you only buy Oreo cookies when they are on sale and had not done so in a while. Their analytics decides to print you a milk and Oreo coupon redeemable the next time you shop.

- Supermarkets could maximize profits by deducing the relationship of shoppers and cart contents. Frequent shopper card data was analyzed to learn which items were purchased together, allowing them to rearrange the store to position and promote items near each other. It is claimed that Walmart increased sales of beer and diapers by 20 percent to men on Friday nights by co-locating these items. Walmart is also known to feed weather data into their DSS giving them 48 hours to position umbrellas to the front of a store when rain was forecasted. Sales of Pop-Tarts and beer increased dramatically before a hurricane was going to hit, perhaps by people originally setting out to buy a flashlight and batteries.

Business queries and reports inform about transactions and events that already occurred. Similar to traditional OLTP systems, the warehouse or mart database is the source for ad hoc queries and standard reports (usually produced on a fixed interval and sometimes run in batches). The reports are typically graphical documents or screen displays that portray data in an un-cluttered manner, either on paper or electronically, and can include mobile devices.

Templates and easy-to-use reporting languages allow business analysts to organize information for specific audiences while hiding the underlying complexity. For customization, they may need to use the Structured Query Language (SQL). To write a SQL query, start with words such as "I want a list of customers that have never ordered from us[9]". In SQL, that might be:

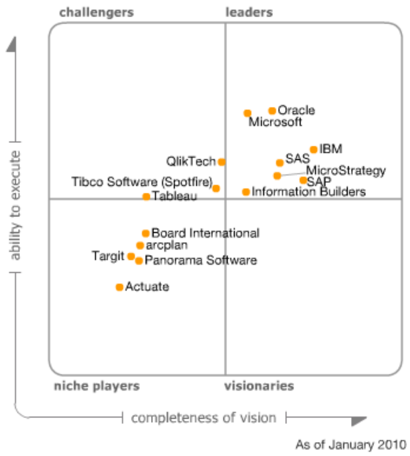
```
SELECT customers.* FROM customers LEFT JOIN orders ON customers.customer_id
= orders.customer_id WHERE orders.customer_id IS NULL
```

Or questions like "What customers have not ordered anything from us in the year 2010?"

```
SELECT customers.* FROM customers LEFT JOIN orders ON (customers.customer_id =
orders.customer_id AND year(orders.order_date) = 2010) WHERE orders.order_id IS NULL
```

Higher levels of abstraction using GUI elements, pull down menus, radio buttons and other intuitive interfaces have also become popular for business queries and reports because of

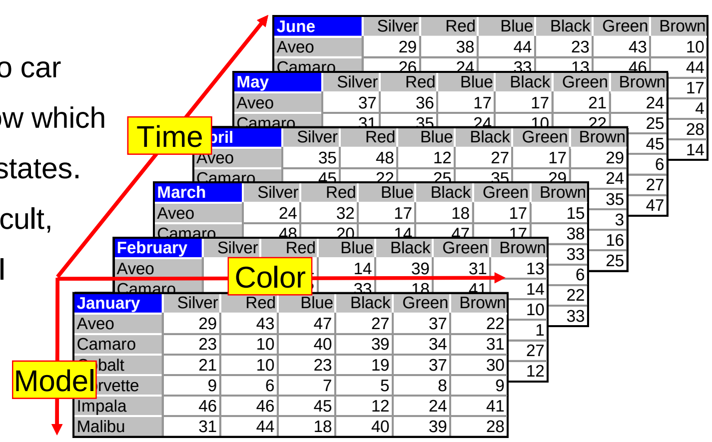
SQL's complexity. New tools allow users to create their own reports and answer business questions without having to understand the nuts and bolts of a database schema.



Business queries and reporting vendors: Oracle's Business Intelligence Suite Enterprise Edition (OBIEE), Microsoft's SQL Server Reporting Services (SSRS), SAS's Business Intelligence, Information Builders' WebFocus, SAP's BusinessObjects (Web Intelligence, Crystal Reports), MicroStrategy, and IBM's Cognos (Query Studio, Report Studio) are Gartner's thought leaders[10]. Some vendors also offer a business information (BI) tool suite.

Business analysis (OLAP) is used to address questions of what happened and why. Since the warehouse or mart contains historical information, aggregated data can be visually presented using multiple dimensions, one of which is time. For example, let's use fictitious New Jersey January-June car sales data. You can see there is a lot of information portrayed with the sales of over 10,000 vehicles; however, a great deal of data is not shown even though the warehouse was populated with detailed records of each sale (including the owner, sales date, color and options, and detailed personal information if a credit score was used for financing).

Jan-June Chevy Sales	List Price	Units	Avg.	Revenue	Silver	Red	Blue	Black	Green	Brown
Avalanche	$35,725	289	32,322	$9,340,914	60	59	36	40	48	46
Aveo	$11,965	974	12,158	$11,842,054	185	208	151	151	166	113
Camaro	$22,680	1,043	23,484	$24,493,638	194	153	169	162	189	176
Cobalt	$14,990	905	15,430	$13,963,698	121	131	162	144	177	170
Colorado	$16,985	168	17,735	$2,979,480	36	30	24	31	27	20
Corvette	$48,930	174	50,264	$8,745,849	19	24	32	38	32	29
Equinox	$22,615	1,223	23,251	$28,435,769	189	158	236	201	231	208
Express	$28,515	186	28,705	$5,339,161	33	34	23	38	23	35
Express Cargo	$24,655	146	24,766	$3,615,836	33	33	24	26	12	18
HHR	$18,720	1,074	19,261	$20,686,493	139	160	241	149	183	202
Impala	$24,290	1,114	25,193	$28,064,445	210	178	218	150	197	161
Malibu	$21,825	1,122	22,532	$25,280,530	198	201	163	215	186	159
Silverado 1500	$20,850	169	21,420	$3,619,924	34	29	20	29	31	26
Silverado 1500 Hybrid	$38,340	149	39,712	$5,917,014	28	19	22	24	33	23
Silverado 2500HD	$27,465	201	27,992	$5,626,426	44	35	32	33	28	29
Silverado 3500HD	$27,685	169	27,864	$4,709,044	25	35	26	19	20	44
Suburban	$40,635	169	41,571	$7,025,555	41	20	30	25	33	20
Tahoe	$37,280	187	37,560	$7,023,627	25	17	41	38	35	31
Tahoe Hybrid	$50,720	175	51,414	$8,997,421	33	28	14	25	37	38
Traverse	$29,224	534	29,066	$15,520,977	72	87	88	99	93	95

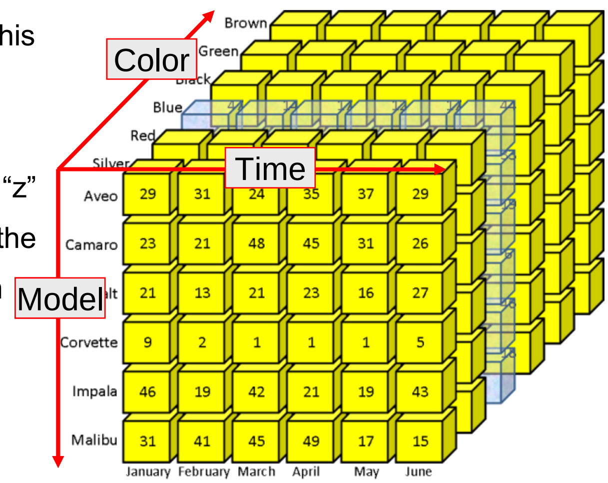


If a General Motor's manager needed insight into car manufacturing efficiency, they might want to know which color sold better (or worse) per month for all 50 states. Getting that information from a giant table is difficult, so data reduction removes the truck data as well as unnecessary fields for the query such as list price, total units sold, average price, or revenue by model. The list could have each month broken out individually with three dimensions–model ("y" axis), color ("x" axis) and time ("z" axis). This “cube” can be manipulated in many ways, and other data can easily be substituted such as a drill down into trim levels for each car, leather versus cloth seats, engine size, demographics, and so forth. Data is logically viewed as a cube but it is not physically stored as a cube in the storage system.

Users can manipulate the multidimensional data and find that April’s best selling color was “silver”, so the manager should want to make more silver cars in February and March. Likewise, “green” outsold all colors and fewer people bought “brown” cars in February even though it was May’s best selling color.

Best Selling Color by Month	Silver	Red	Blue	Black	Green	Brown
January	159	159	180	142	179	161
February	127	112	123	179	159	77
March	181	147	124	152	136	147
April	194	177	164	124	131	132
May	121	161	138	131	131	174
June	145	139	166	132	211	117
Total	927	895	895	860	947	808

To see why it’s called a cube, consider this “Rubik’s cube” view from which the data can easily be “sliced and diced”. In this case, the cube is rotated to display “color” on the “z” axis and “time” on the “x” axis, perhaps because the person wanted to look at “blue” car patterns. With a few mouse clicks, the data representation can easily be changed without any knowledge of programming or database schemas.



Alternatively, a cube slice could quickly show which model sold best each month. This table shows that in March the best selling models were the Impala and Malibu, with Malibu being the best selling model overall.

Best Selling Model by Month	Jan	Feb	Mar	Apr	May	Jun	Total
Aveo	205	139	123	168	152	187	974
Camaro	177	169	184	180	147	186	1043
Cobalt	140	104	167	202	141	151	905
Corvette	44	11	29	19	36	35	174
Impala	214	165	192	155	196	192	1114
Malibu	200	189	192	198	184	159	1122

While this data was available in detailed warehouse reports, it is hard to visualize and draw the conclusions necessary to guide the business. Cubes give users focused analysis without SQL or other programming languages. OLAP allows for cost and currency calculations, time series analysis, what-if scenarios, data mining, and other capabilities. Cubes are ***pivoted*** (change or rotate the multi-dimensional views), ***rolled-up*** (less detail), ***drilled into*** (more information), and ***sliced and diced*** (selection). While specialty tools provide data visualization, common tools such as Excel pivot tables can also be used.

Business analysis (OLAP) vendors: The most common are IBM Cognos (PowerPlay DOLAP[11], MOLAP[12], Analysis Studio, and TM1), Hyperion Essbase MOLAP/MDDB[13], Brio, SAS OLAP Server, MicroStrategy ROLAP[14], Oracle Express, SAP Business Objects, and Pentaho Analysis.

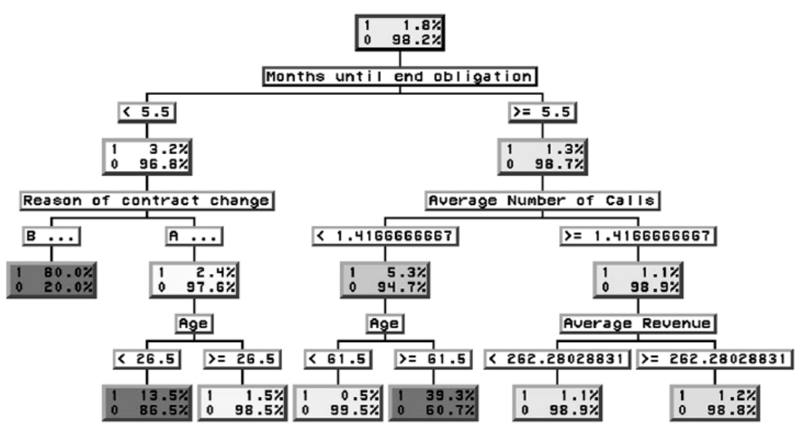
Data mining is a process that uncovers statistically significant data trends and patterns that are advantageous to an organization. Performed by statistical analysts with data mining tools expertise, they start with a general business question such as "tell me something I don't know" or a specific question as with the Walmart beer and diapers query, and search for data trends, correlations, patterns, categories, and relationships.

The basic flow that data mining analysts follow includes:

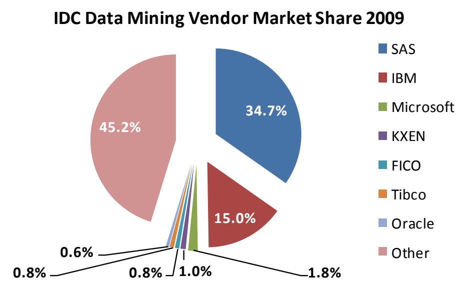
1. understand what needs to be predicted
2. which of four models should be used (clustering, classification, regression, or trending)
3. select/ETL data for the study
4. construct the model (neural network, linear regression, decision trees, and others)
5. interpret the results, possibly with the assistance of a subject matter expert (SME)
6. format the results in an easier to use form

Data mining is often associated with these types of analysis or questions (partial list):

- **Market Basket Analysis**–the concurrent relationship of items. For example, purchasing milk and cookies, shampoo and conditioner, beer, and diapers.
- **Forecasting**–understanding patterns in future data to optimize a function. For example, during what month should a car company build more silver models?
- **Web site analysis**–examining what turns prospects into customers based on web site visits. For example, changing the look-and-feel of a site can improve a user experience to generate more sales.
- **Churn Analysis**–the study of what keeps customers loyal. A supplier wants to keep churn low because of the high expense of attracting new customers. For example, this is a SAS decision tree analysis of the months until a contractual obligation

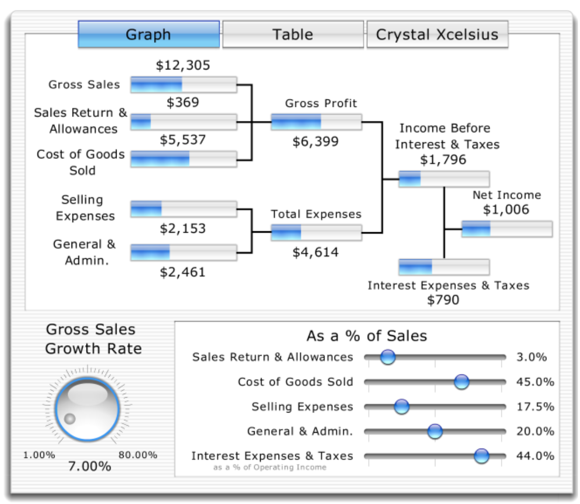


ends, contract change reasons, average number of calls, age, average revenue, and other factors. Darker areas highlight areas of business concern.



Data Mining vendors: according to a 2009 IDC study[15], while many DSS vendors offer data mining solutions, SAS and IBM are the thought leaders in this space and collectively have about a 50 percent market share.

<u>Dashboards and scorecards</u> give an instant view of specific warehouse/mart information. It is designed for repetitive use without having to write a detailed SQL query or sift through detailed reports, and does not require familiarity with the underlying complexity. Similar to gauges and "idiot lights" on a car dashboard, these readouts provide visual data "cues" for the question at hand. While they give you a feel for your business question, you might supplement it with a query, report, or a cube if you need specific data.



Output from dashboards and scorecards are also used in briefing packages since after all, "a picture is worth a thousand words". They also can form the basis for command center readouts using green-yellow-red status as well as other metaphors.

Dashboard and scorecard vendors: Many warehouse/mart and OLAP suppliers sell excellent integrated dashboard and scorecard tool suites.

Vendors such as Microsoft are constantly upgrading their BI. For example, PowerPivot is a new tool that uses Excel and SharePoint on large tables and database feeds with automatic sorting and filtering in an attempt to provide "self-service" BI. [Note: while IT professionals want their users to have the best possible experience, some believe that "self-service" may have gone too far. When users create and circulate ad-hoc reports on their own, it is difficult to maintain a single “version” of the truth. If not done properly, spreadsheet data becomes stagnant and could lead to poor decisions if not constantly refreshed by ever-changing warehouse/mart data.]

Technology to the Rescue?

There are many things that impact the success of a DSS. As with any complex architecture, a single technology cannot fix bad designs, underlying process flow issues, or data quality concerns. There could also be poor tools, old software, old hardware, aggressive deadlines, team composition, and other reasons for delay. Nonetheless, when performance issues plague a warehouse to the point of failure, especially as its size grows, technologies such as solid state disks (SSD), cache memory, additional host bus adapters (HBAs), load balancing multipath software, and in-memory tables can help it perform like new.

The warehouse profile is typified by multiple users that run canned reports and issue ad hoc queries, as well as ETL processing. When you monitor the storage system, you will see a continuous, yet random I/O pattern that is predominately comprised of “READ” activity. This randomness may require you to use techniques such as thin provisioning to leverage the maximum number of physical drives and the large number of input-output operations per second (IOPS) they offer. Auto-tiering (e.g. EMC sub-LUN FAST) might also help by promoting frequently accessed data to the highest performing disks. In some cases, cache can help even out I/O storms system wide and provide better query performance; the greater the cache-assist, the better the performance.

Should backup and recovery become a problem, perhaps due to the volume of data or the time it takes to backup, review the upcoming section on “Data Warehouse Backups”. Today’s data deduplication equipment can make backup tasks faster and improve restore capabilities. Adding tape drives, especially newer LTO-5 drives, can also provide a tech answer for size and time.

Scalability is another issue, both in terms of network size and storage capacity. The latest storage area networks (SANs) offer 8Gb of Fibre Channel (FC) bandwidth. Fibre Channel over Ethernet (FCoE) is the newest networking technology, combining the world of FC with 10Gb Ethernet networks. From a storage standpoint, when you use star and snowflake schemas, be prepared for large fact tables–all the dimensions, aggregation tables, indexes, and other design elements can occupy more space than the data itself. The good news is that a combination of tiering, thin provisioning, larger capacity SATA drives, and SSDs can now deliver petabytes of usable capacity at very attractive price points. Compression and archiving of old data can keep your BI environment space in check and even help with backup issues.

If BI capital expenses (CAPEX) are getting out of hand, then solutions that run the gamut from open source DSS vendors, to grid solutions and even cloud warehouses can be used. Infobright has a robust offering based on Oracle's open source MySQL, and others offer PostgreSQL and Ingres platforms. High end open source solutions such as EMC's Greenplum using a modified PostgreSQL can give your organization an extremely advanced, highly scalable, cloud-ready MPP solution at a fraction of the cost of a Teradata solution. Open source ETL tools such as Pentaho (a fairly robust transformation library), Talend (Perl and Java), and Octopus (Java-based ETL) also keep your costs down.

Don't confuse "open source" with "free" because there are still support costs, staff training, and much more to a warehouse. A major DSS cost is labor-intensive maintenance as they are continually being fed new data and periodically reorganized.

A data warehouse is an ecosystem of dependent software, hardware, processes, and people. While technology alone can add functionality or fix issues, merely swapping old technology for new may not turn things around with multifaceted issues. Change can also make the situation worse. For example, your team may be skilled with an ETL tool, but the tool may be inadequate. Yet if a new, superior tool has a steep learning curve, it may not produce the desired result if the team was already under time pressure.

RASP - Reliability, Availability, Scalability, and Performance

RASP is a warehouse architectural imperative. While the warehouse does not run real-time tactical business transactions, it nevertheless serves an important strategic function. DSS storage has to be **reliable** and dependable, especially with hundreds of terabytes of capacity. A recent Google study showed the mean time to failure (MTTF), mean time to repair (MTTR), and average failure rate (AFR)[16] for hundreds of drives is a critical design factor. They found a 10 percent AFR for highly utilized 3-month old drives, 2 percent after 2 years and 4.6 percent after 5. Multiply this by your drive count and it is easy to determine the importance of having RAID protection and hot sparing.

Reliable systems start with reliable storage. When a RAID disk fails or has too many retries, a hot spare (when available) is activated to non-disruptively assume the data load of the bad drive, thus keeping MTTR to a minimum. RAID 1 uses mirrored disks, so the spare duplicates the data from the surviving drive, while RAID 5 and 6 employ group parity computations to copy data to the spare. RAID 5 has one parity drive per group and tolerates a single disk failure without data loss. RAID 6, common with large 1-2TB SATA disks since the time to rebuild them can easily take over a day, uses dual parity drive to cope with two simultaneous failures.

Single points of failure are also part of RASP, so you want a storage frame with redundant components. For example:

- Are there dual power circuits feeding the system?
- Should a power supply fail, will others carry the load until a non-disruptive repair is made?
- How about redundant cooling fans and battery backup systems?

Pay attention to network connectivity–for example, with FC SANs, it is best practice to use dual fabrics, load balancing/failover software, two or more HBAs in each server, redundant ports on the storage frame, and so on.

From an **availability** standpoint, servers and storage do not process transactions (i.e. make money) during support windows; nonetheless software and hardware upgrades, maintenance, cleanup, and data reorganizations are necessary to keep them running optimally. A good design should include servers can fail-over to other servers, SAN fabrics can be taken down one at a time, storage frames can copy their data or keep it in sync with another remote frame, and so forth.

Scalability is another key RASP DSS concept. Unprecedented data growth[17] means your environment needs to support hundreds if not thousands of drives. Years ago, a large system had a few hundred drives. Today, large systems can have a thousand or more drives. Port counts for network connectivity or connection speeds also has to grow or adapt as time goes on. For example, 4Gb FC technology first became popular in late 2005/early 2006. By 2008, 8Gb FC started to take hold, and forecasts show 16Gb FC could be available this year. In the meantime, FCoE is gaining momentum and could become popular this year as well. In short, since tomorrow's technology will be different than today's, plan for scalability and flexibility.

Scalability and performance also go together. Until recently, the fastest FC and SAS drives rotated at 15,000 RPM, and were used for logs and table spaces. 7,200 RPM SATA drives were generally used for slower applications or archives. As a rule of thumb, faster rotation meant the drive offered more IOPS and better performance.

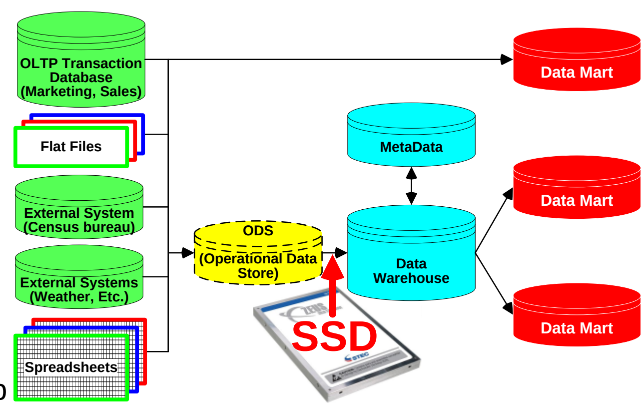
FAQ – Can large capacity disks be used in a DSS?
Answer – The tradeoff is capacity versus performance. A 10TB DSS could use five 2TB SATA disks if performance is not an issue (ignoring usable space and parity). However, ETL, queries, and reporting are resource intensive.

That's where IOPS come in. SATA is ideally rated at 80 IOPS, so five SATA disks = 400 IOPS (very low). Blended SSD, 10K/15K RPM FC/SAS, and SATA designs are better.

Use FC or SAS disks for higher performance. A 300MB 15K FC disk has 180-200 IOPS. It takes 34x300GB disks to equal the capacity of five 2TB disks, but they yield 6,800 IOPS. The 34 FC/SAS disks cost much more than five 2TB disks, and that is why blended strategies are used.

Design for the worst case – a full table scan or ETL. A scan running against a single 2TB SATA disk is limited to the performance of that disk. Multiple faster disks striped together can increase table scan speeds by 10X or 100X.

SSD changed the "rules of the game" when they became affordable and reliable. Without moving parts, an SSD has the performance of 30 mechanical drives. A blended storage profile with SSD saves floor space and lowers power/cooling expense. When you profile your DSS I/O, some areas might need the fastest drives available, other parts need good performance, and the rest can easily reside on slower disks. For example, a vast amount of data is added to the warehouse during the ETL phase. To help absorb the I/O burst, SSD can receive the load and "gently" push it out to mechanical disks over a longer duration. This can speed up warehouse loading 5-10 fold. When selecting a DSS storage frame, pick one that accommodates SSD should your performance profile demand it.



SSD has significantly greater throughput and IOPS than a rotating disk, but also costs more, so a tiered drive blend has become popular (as this chart shows). SSD are rated at almost 40,000 IOPS, but IOPS depend on many factors, so your mileage may vary. The DSS IOPS are conservatively listed as 3,927 because of its READ intensive large block I/O profile (10 percent of specification). Leverage this information along with costs to build your storage design. A rule of thumb is to service 75 percent of the activity with SSD and FC/SAS storage. Infrequently accessed data tables and archived data go on dense SATA drives.

Drive Performance Chart	SSD	FC	SAS	SATA
Spindle Speed	n/a	10K-15K	10K-15K	5.4K-7.2K
Internal transfer rate Mb/s	3500	725-2,225	725-2,225	500*-1,287
Small-block IOPS	39,273	120-200	120-200	50-80
DSS IOPS (* estimate)	3,927	180-200	180-200	80
Bandwidth	✓✓✓	✓✓	✓✓	✓
Capacity	✓	✓✓✓	✓✓✓	✓✓✓✓✓
Cost	$$$$	$$$	$$	$

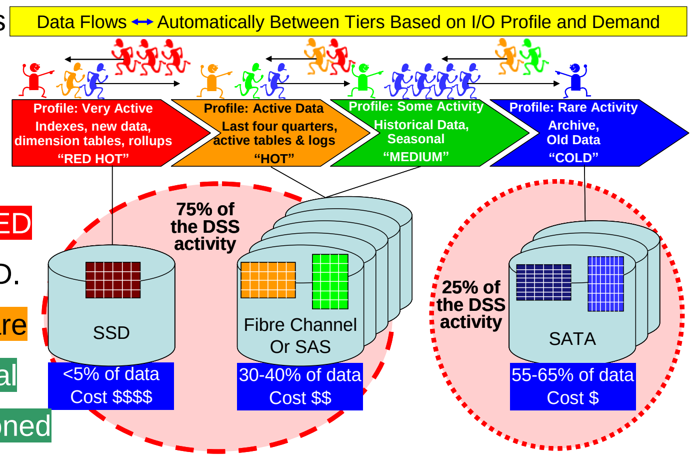
Performance is also a function of RAID protection. Some RAID 1 systems use a READ service policy to minimizes seek times by instructing the disk whose head is closest to a READ request to respond. Some disk controllers allow both RAID 1 drives to service READ requests effectively doubling READ transfer rates. RAID 1 is useful for high I/O activity (log files) and for fast READ response. RAID 5 and 6 are great for table spaces, with RAID 6 being a little slower for WRITE activity than RAID 5 because of a higher write penalty[18]–an issue for ETL activity.

When your warehouse slows down, your first line of defense is your DSS supplier's analysis tools such as Oracle's STATSPACK. Third-party companies, like Precise, have analysis tools that examine performance from the query to the storage layer, pinpoint issues, and offer solutions to the problems. These tools help balance warehouse performance during ETL, queries, business cycles, and so forth.

If your environment has ad-hoc queries, tuning it for dynamic I/O workloads is a challenge, even with the best of tools. For example, when you set up an optimized pattern for Monday, it might not work well on Tuesday. Patterns can also even change seasonally. Storage administrators should look for design assistance from their warehouse or storage provider. For instance, EMC, HDS, and other vendors have white papers to help you set up your particular storage frame for various brands of warehousing[19] with details on optimal stripe size and other guidelines.

Tuning can also be difficult because of the burst of WRITE activity within the warehouse when indexes are built, sums are performed, and data is sorted. Warehouse databases are also reorganized and as they get larger, they take more resources to accomplish. As a result, DSS architects are recommending a dynamic approach that avoids manual tuning, as depicted in the drive mix diagram below. The system finds its own harmony at your price point with auto-tiering and auto-tuning. For example, EMC's latest sub-LUN FAST moves small pieces of a LUN to the proper storage SSD, FC/SAS, or SATA tier based on I/O usage patterns without application downtime. Other examples include IBM's "Easy Tier" and Hitachi's "Tiered Storage Manager".

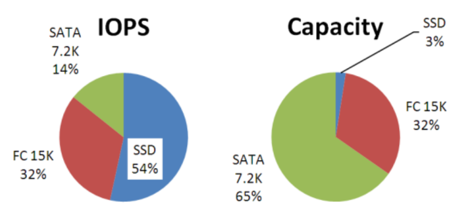
DSS data placement is not that different from an OLTP system, but its sheer size and complexity means it takes more effort. There are two basic data placement approaches–static purposeful and "spread it wide". With the static design, you separate indexes, logs, data tables, binaries, and so forth. The spread it wide concept is typified by thin provisioning where many drives are brought together in a pool and data is distributed in thin slices over all the drives to maximize parallel I/O without regard to disk contention. These two approaches can usually be combined on storage frames such that indexes can be static and data tables thinly provisioned.



Finding the right mix of DSS drive speeds is tricky since faster disks mean higher costs. This **drive mix diagram** shows that static placement works well when active indexes, new data, dimension tables, and rollups (RED HOT) are on your fastest disks such as SSD. Recent data, active tables, and other logs are put on the 15K FC/SAS tier (HOT). Historical and seasonal data goes on the thin provisioned 10K disk tier (MEDIUM), and archive or old data put on SATA drives (COLD).

Type of drive	SSD 400GB	FC 15K 450GB	SATA 7.2K 2TB	Total
Number of drives	8	95	42	145
RAID Protection	5 (7+1)	5 (4+1)	6 (12+2)	
Usable capacity TB	3	32	65	100
Allocation %	3%	32%	65%	100%
IOPS	31,418	19,000	8,400	58,818

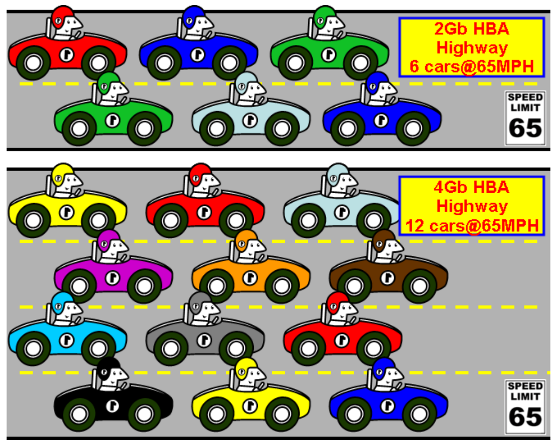
To illustrate this, assume a 100TB DSS has three basic tiers–8xSSD for 3 percent of the workload, 95xFC 15K RPM drives for 32 percent of the capacity, and 42xSATA for 65TB. With this mix, 54 percent of the tiered performance is from SSD while 65% of the usable capacity is on SATA. This lowers CAPEX and operational expenses (OPEX) while offering high performance. Without SSD, hundreds of rotating disks are needed to provide the same performance.



As previously noted, ETL WRITE performance is very important since large amounts of data are moved into the DSS in a limited amount of time. Cache systems allow the updates to be processed faster since they land in storage memory rather than on physical drives. As cache fills up, it independently stores the data to disk. "Intelligent" cache can also help sequential READs by having drives read-ahead and prefetch into cache what is likely the next few data blocks. Cache is small compared to the DSS size but can achieve a very high success rate–i.e. I/O requests are serviced in memory and not disk. Table scans are cache's "enemy"–a cache "miss" means the data is retrieved from a slower mechanical disk. Some systems have upgradeable cache. For example, EMC's FAST Cache uses up to 2TB of SSD as secondary READ-WRITE cache and NetApp's PAM II cards can improve READ performance.

Compression, offered by many databases and storage frames, also improves performance. With rotating disks, 2:1 compression means twice as much data is retrieved per I/O Operation than if it was uncompressed. Compression overhead is measured in <u>micro</u>seconds while disk I/O is in <u>milli</u>seconds.

Performance between the server and storage is also critical, and a good design starts with the proper number of HBAs. We know that two HBAs and dual fabrics are the minimum to balance the I/O load to the storage frame, provide an alternate path if one fails, and help serviceability. Each HBA represents an operating system kernel thread, and more threads mean more I/O. It is not uncommon to see multi-core servers with more than two HBAs. Reducing the number of kernel threads can lead to a performance issue, especially when systems are consolidated and a larger number of slower HBAs are replaced by fewer, faster ones.



Confusion can also arise when HBAs, SAN speed, and throughput are discussed. In short, 4Gb or 8Gb HBAs transmits data at the <u>same speed</u> as 2Gb HBAs, but they are capable of 2-4 times the throughput. For example, picture a two-lane highway where cars kept to a 65 MPH speed limit. If it had 4 or 8 lanes and the speed stayed the same, the cars would not travel any faster, but there could be 2–4 times the number or cars on the road.

DSS vendors' design guides, reference architectures, sample configurations, and services help with server sizing and storage layouts. For example, Greenplum's CLARiiON® building block concept[20] begins with 4 servers and 71 drives allowing growth from 21TB to 5PB of capacity. Another example is Oracle's Automatic Storage Management (ASM), which like a storage frame's thin provisioning, spreads database files across all the disks to maximize performance. Storage vendors also document best practices leveraging multiple back-end disk controllers and multi-path HBA software.

RASP goes beyond specifications–it requires planning and attention. Creating a performance baseline can help determine when things are getting slower and head off end-user complaints.

Baselines require measuring and reporting tools that encompass the entire warehouse as well as alerting tools to trigger SNMP messages when metrics exceed the baseline. Whether the DSS is in your data center or remote, plan to “expect the unexpected”. Its size can put a strain on systems and procedures, and when things get larger, they tend to take more time to fix.

Data Warehouse Backups

A corporation makes money with its OLTP systems while its DSS improves the business. So while OLTP is mission-critical, a data warehouse is a close second. Both systems need to be backed up, but the characteristics of each can lead to different backup approaches.

With OLPT’s financial and legal criticality, the minimum rule of thumb is to use incremental daily and full weekly backups. However, this approach faces time pressure with DSS because of their large size and other upkeep issues. For example, a full backup of 250TB of usable DSS data, even with the latest LTO-5 tape drives and 8Gb FC links, takes over 16 hours with 16 tape drives. Target-based disk systems like the Data Domain® DD880 would take more than 14 hours. These backup approaches are expensive, and in the real world, could take even longer.

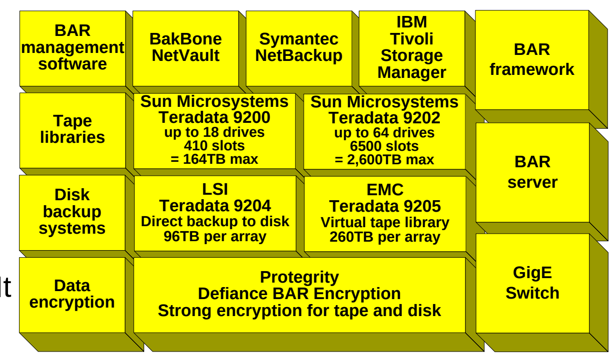
Source disk cloning is viable but expensive. Our 250TB data warehouse needs 168x2TB SATA drives for a full clone copy. It is hard to tell how long it takes to make a clone copy of that size, but it should be substantially less than 16 hours since the data has not changed much since the last clone copy. Once the clone is done however, it still takes 16-32 hours to back it up for an off-site copy. So in all likelihood, you might need <u>two</u> full clone sets–one for the active clone and another to backup yesterday’s data image. Snapshots at perhaps 10 percent of the source disk would take far fewer drives, but the backup process could put a big strain on warehouse performance due to its dependency on the source system when full backups are done.

At worst, a DSS can likely be reprocessed/reloaded from source systems since most of it is READ-only historical data; that being said, a full restore could take days or weeks (your <u>**R**</u>ecovery <u>**T**</u>ime <u>**O**</u>bjective or RTO). Try backing up static data only once. For example, a table of 50 state abbreviations will not change, or could easily be recreated. Some believe monthly full backups and daily/weekly incremental backups works best since the data change rate is low.

If you planned to back up the warehouse while it is running, remember that READ/WRITE access to database tables requires locking, so you need to be careful about ETL operations or active queries during backups. There are approaches that can skip locked data tables and back them up at a later time, but the overhead of trying to run a complex OLAP query against a system that is also being backed up can make queries run significantly slower.

For full backups, a database is shut down or put in hot backup READ-only mode until completion. Incremental backups target modified or new tables. Systems like MS-SQL have backup and restore APIs, allowing them to stay online while maintaining consistency and permitting specific tables or files to be backed up or restored (versus the full image).

Each DSS vendor has specific best practice recommendations for backing up and restoration. For example, Sybase IQ's "Very Large Database Management Option" enables the logical data partitioning by grouping data into READ-only, READ/WRITE, offline, or online categories that focuses and shortens backup and restore cycles. Oracle promotes Recovery Manager (RMAN), a free utility that backs up and recovers their databases. RMAN tracks the backup, maintains the recovery catalog, and holds scripts and commands for database backup/recovery tasks. It tracks what needs to be backed up, determines what is needed for a recovery, and helps repair database corruption. Teradata's Backup, Archive and Restore (BAR) framework encompasses 3rd party products to speed up backup and restore. Archiving older data to another device like an EMC Centera® has become popular because the data does not fully partake in the backup stream, yet is always available on demand.



The disk structure of a warehouse is in many ways like any other large database. For example, an Oracle structure consists of datafiles, multiple redo logs, and control files with data stored in tablespaces consisting of multiple datafiles. Redo logs track changes made to datafiles so they can be backed out or reapplied. Redo logs are stored as archive redo logs because they are critical to Oracle's RMAN strategy. Control files track the database structure, checkpoints, and metadata for RMAN. Failure to back up control files can make restoration difficult or impossible.

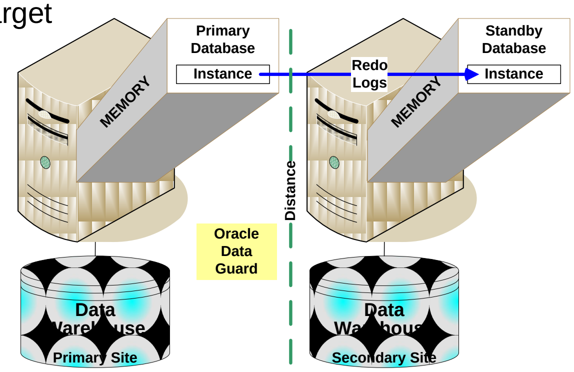
Data Warehouse Disaster Recovery/Business Continuity

Years ago, a data warehouse was not critical to the business and could be reloaded from source systems. Today, these systems are critical, large, and a disaster could mean weeks or months before your organization can continue making strategic decisions. If your DSS is critical, it is time to think beyond normal backup/recovery operations. A data warehouse needs the same high-availability insurance provided by transactional disaster recovery/business continuity methodologies. As with other key systems, don't wait for a disaster to start your planning.

Depending on budget, DSS size, criticality, the nature of the disaster, and so forth, protection could be as simple as taking backup tapes offsite. The next level of "insurance" could include real-time data replication to a disaster recovery hot-site, or if you send tapes to a service provider such as Iron Mountain, they can also sell you a service to restore the tapes at their site.

For real-time transmission, vendors like Data Domain can replace tape libraries with disk-based units that deduplicate and compress backup data to a fraction of its original size (remember, it is mostly a denormalized READ-only environment). They can transmit the reduced data to another Data Domain frame over a small network. The data stays there at rest just like "Iron Mountain" tapes, or could continually be restored to another DSS storage system for fast recovery in the event of a disaster.

The next highest level of protection comes from active replication of changed data to a duplicate environment. Most storage frames have remote mirroring to allow the data in your production system to be transmitted to a geographically remote target system. Your organization would be up and running without the delay of reloading backup tapes.



Regardless of the approach, plans need testing. Unfortunately, many organizations *schedule* disaster recovery tests in advance. Real disasters and disruptions, like snow storms, floods, chemical spills, fires, earthquakes, terrorist attacks, transit strikes, blackouts, and so on, rarely give enough advanced notice. It is not merely a disruption to your data center, for if workers are unable to get to work, then you've got a big problem. Protection requires real planning.

This process should begin with your DSS or storage provider. Teradata has a “Disaster Recovery Solution” to reduce downtime and Oracle’s Data Guard uses server-based, in-memory log replication. EMC’s Symmetrix® Remote Data Facility (SRDF®) and Hitachi’s TrueCopy replicate data written from one storage array to another. Greenplum leverages Data Domain replication or storage replication. When discussing disaster recovery with your vendor, review the importance of RTO, Recovery Point Objective (RPO), ease of use, and cost.

NAS versus SAN – Is It That Simple?

The question of building a DSS with NAS or a SAN comes up often, and it depends on a number of factors, some obvious, some not. For example, Teradata is not supported on NetApp, so in that case it is not even an issue of NAS versus SAN.

Sometimes the application dictates the protocol, and some suppliers have a preference for one approach versus the other. Microsoft’s SQL Server (article 304261) specifies a SAN or local disk and not a NAS system. IBM’s InfoSphere install guide requires that DB2 transaction logs and data be placed on separate spindles and LUNs, RAID 10 (RAID 1 plus striping) for transaction logs and RAID 10 or RAID 5 for data LUNs, so that may rule out some NAS solutions.

On the other hand, Oracle supports NAS. Just keep in mind the performance and scalability limitations, and check with your DSS vendor before you buy NAS storage.

The basic difference between NAS and SAN data warehousing is performance. While both OLTP and OLAP use databases which are NAS-capable, warehousing generates a lot of large I/O transfers (high throughput) and a large number of IOPS. For example, while Oracle supports NAS, it might favor SAN for a particular workload because of overhead concerns, especially if the NAS system implements FC as a file system, and not as a true “block” or LUN.

From a support perspective, most companies build warehouses with a SAN. Where NAS and in particular NFS generally come into play is with small warehouses or with the reporting process.

With the discussion of NAS versus SAN, the notion of iSCSI and even FCoE comes up. iSCSI can certainly work with a warehouse, but keep in mind that iSCSI is generally slower than FC and can lead to higher server overhead. FCoE supports IP, iSCSI, and FC, so it will also work. It may come down to vendor support, staff expertise, risk, cost, and a handful of other issues.

If you still want NAS, use a 10Gb Ethernet network because your DSS might need high throughput. In theory, 10Gb Ethernet is 10X the performance of 1Gb Ethernet and slightly faster than 8Gb FC. Remember that blocking factors and other real world issues come into play, so it is hard to predict utilization rates–i.e., some network designs only achieve 40 percent to 70 percent efficiency.

Data Warehousing, VMware and the Cloud – the Next Wave?

The notion of virtualization seems straightforward, and some of the vendors promoting virtualized warehousing include Teradata, Veritca, Greenplum, and SAP. While DSS virtualization may not be your CIO's top priority, a VMware warehouse gives your organization some interesting benefits including scalability, simple development environment, easy transition from development to Q/A to production, painless training systems, and HA (high availability).

When your organization is ready, it can take advantage of cloud computing to move to a web-based, pay-as-you-go business model for servers, services, and applications. Off-premises public clouds promise massive computing power and speed for a fraction of internal costs. Vendors who offer virtualization and cloud solutions include:

- Teradata supports both the private/public cloud paradigm with solutions like Teradata Express built for the Amazon Elastic Compute Cloud (EC2) environment. This can be very useful for developing, testing, and evaluating Teradata solutions without a large CAPEX.
- Vertica offers a high-performance warehouse running as a virtual machine as well as an MPP design on standard Intel/AMD servers using Red Hat Linux. Their design supports VMware and Amazon's EC2, so it operates in the private and public cloud.
- Greenplum's Enterprise Data Cloud platform and Chorus product leverage VMware and cloud computing to provide self-service sandboxes and databases on demand. They also offer MPP software and appliances that speed up ETL and queries.

There are still issues to be addressed with virtualization, clouds, and data warehousing. For instance, it is difficult to guarantee performance when memory space and data paths are shared with other virtual applications. Security and service level agreements are critical public cloud issues since cloud security is a work in progress and it may be difficult to negotiate a meaningful service level agreement (SLA). As such, organizations that find cloud benefits appealing are opting for the security and availability of a private cloud in a traditional data center.

Conclusion

When economic times are tough, organizations are focused on making do with less. At the same time, they are trying even harder to increase revenues, grow market share, improve profitability, increase productivity, and reduce expenses. The field of data warehousing is not new, so in all likelihood one already exists where you work.

We know that “knowledge is power”, so the more you know about the ever changing world of data warehousing the more you can contribute to the DSS team and its success. Decision support is a complex process, and that is why those groups need your help. The next time the topic is mentioned, ask yourself if you can apply your skills and background to the data warehouse effort – everyone will be glad you did!

Footnotes

[1] http://www.computerworld.com.au/article/208773/idc_report_data_creation_outstrips_storage_first_time/
[2] http://www.gartner.com/it/page.jsp?id=1283413
[3] http://www.businessintelligence.info/docs/estudios/Gartner-Magic-Quadrant-for-Datawarehouse-Systems-2010.pdf
[4] Columnar technology allows for data to be organized and processed in columns rather than rows. It is useful when aggregating a large number of similar items.
[5] http://en.wikipedia.org/wiki/Third_normal_form
[6] http://www.usps.com/ncsc/lookups/usps_abbreviations.html
[7] http://www.gartner.com/technology/media-products/reprints/informatica/volume4/article2/article2.html
[8] http://www.catalinamarketing.com/company/FAQs.html
[9] http://www.paragoncorporation.com/ArticleDetail.aspx?ArticleID=27
[10] http://www.gartner.com/technology/media-products/reprints/oracle/article121/article121.html
[11] DOLAP (Desktop OLAP) – multidimensional datasets can be stored on the desktop
[12] MOLAP (Multidimensional OLAP) - MOLAP is 'classic' OLAP and requires the pre-computation and storage of information in the cube. Pre-computing the data produces faster queries, but uses more disk space. It is faster than ROLAP.
[13] MDDB (Multidimensional Database) – in contrast to a RDBMS which stores data in rows and columns, an MDDB adds additional dimensions to make it more practical for cubes. You would not use an MDDB for OLTP.
[14] ROLAP (Relational OLAP) - Computes results without pre-computation. Useful for infrequent queries or with very large data sources. Slower than MOLAP, but needs less disk space.
[15] http://www.sas.com/news/analysts/IDC-BITools09VendorShares.pdf
[16] http://static.googleusercontent.com/external_content/untrusted_dlcp/labs.google.com/en/us/papers/disk_failures.pdf
[17] http://www.emc.com/collateral/demos/microsites/idc-digital-universe/iview.htm
[18] http://en.wikipedia.org/wiki/RAID
[19] Examples of vendor white papers for setting up Oracle data warehouse environments
http://www.emc.com/collateral/hardware/white-papers/h5548-deploying-clariion-dss-workloads-wp.pdf
http://www.emc.com/collateral/hardware/white-papers/h4005-symmetrix-dmx4-oracle-wp.pdf
http://www.emc.com/collateral/hardware/white-papers/h6015-oracle-data-warehouse-sizing-dmx-4-dell-wp.pdf
http://www.hds.com/assets/pdf/best-practices-library-deploying-oracle-11g-rac-with-hds.pdf
http://www.hds.com/assets/pdf/deploying-oracle-database-11gr2-with-asm-and-hdps-best-practices-guide.pdf
[20] http://www.emc.com/collateral/hardware/white-papers/h7015-greenplum-clariion-build-block-wp.pdf