



DMIF, University of Udine

---

# Data Management for Big Data

## *Data Warehousing*

Andrea Brunello

andrea.brunello@uniud.it

April 2022



- 1 Introduction
- 2 Data Warehousing Fundamental Concepts
- 3 Data Warehouse General Architecture
- 4 The Multidimensional Model
- 5 Operations over Multidimensional Data

# Introduction



Nowadays, most of large and medium size organizations are using information systems to implement their business processes.

As time goes by, these organizations produce a lot of (heterogeneous) data related to their business, but often these data are **not integrated**, being stored within one or more platforms.

Thus, they are hardly used for decision-making processes, though they could be a valuable aiding resource.

A **central repository** is needed; nevertheless, traditional databases are not designed to review, manage and store historical/strategic information, but deal with ever changing operational data, to support “daily transactions”.



# What is Data Warehousing?

Data warehousing is a technique for **collecting and managing data** from different sources to provide meaningful business insights.

It is a blend of components and processes which allows the strategic use of data:

- Electronic storage of a large amount of information which is designed for query and analysis instead of transaction processing
- The ultimate goal is that of transforming data into information and making it available to users in a timely manner to make a difference



# Why Data Warehousing?

A normalized, relational database for an inventory system has many tables related to each other through foreign keys.

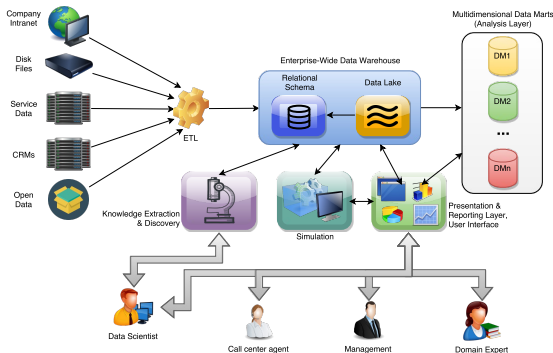
A report on monthly sales information may include many joined conditions.

This can quickly slow down the response time of the query and report, especially with millions of records involved.

A data warehouse provides a new design which reduces the response time and thus helps to enhance the performance of queries for reports and analytics.

A data warehouse is typically the central component of a *Decision Support System*, i.e., an (descriptive / predictive / prescriptive) information system that supports business or organizational decision-making activities.

E.g., providing monitoring tools, graphs, reports, simulations.



# Data Warehousing Fundamental Concepts



According to William Inmon, a data warehouse is a *subject-oriented, integrated, consistent, non-volatile, and time-variant collection of data* in support of management's decisions.

The analyst job in the data warehouse environment is easier than in the legacy environments:

- single integrated source of data
- data is easily (and rapidly) accessible

The data warehouse is at the heart of the *decision support system* (DSS) operation.



The data warehouse focuses on enterprise-specific *concepts*, as defined in the high-level corporate data model. Subject areas may include:

- Customer
- Product
- Order
- Claim
- Account

Conversely, operational databases hang on enterprise-specific *applications*, meaning that data in them is typically organized by business processes, around the workflows of the company.



Data is fed from multiple, disparate sources into the data warehouse.

As the data is fed, it is converted, reformatted, resequenced, summarized, and so forth (ETL – Extract, Transform, Load).

Data is entered into the data warehouse in such a way that the many inconsistencies at the operational level are resolved.

Consistency applies to all application design issues, such as naming conventions, key structure, measurement of attributes, and physical characteristics of data.



After the data is inserted in the warehouse it is neither changed nor removed.

The only exceptions happen when false data is inserted or the capacity of the data warehouse is exceeded and archiving becomes necessary.

This means that data warehouses can be essentially viewed as read-only databases.

When subsequent changes occur, a new snapshot record is written. In doing so, a historical record of data is kept in the data warehouse.



Time variance implies that the warehouse stores data representative as it existed at many points in time in the past.

A time horizon is the length of time data is represented in an environment; a 5-to-10-year time horizon is normal for a data warehouse.

While operational databases contain current-value data, data warehouses contain sophisticated series of snapshots, each snapshot taken at a specific moment in time.



# OLTP: On-Line Transaction Processing

OLTP queries are typical of operational, daily systems.

Such queries generally read or write a small number of tuples, executing transactions over detailed data.

A typical OLTP transaction in a banking environment may be the transfer of money from one account to another.

Always enforcing data consistency and handling concurrency aspects is essential for this kind of applications, because otherwise money may for example get lost or doubled.

“On-line” means that the analyst should obtain a response in almost real time.



# OLAP: On-Line Analytical Processing

On the contrary, the type of query generally executed in data warehouses is OLAP.

In OLAP applications the typical user is not interested in detailed data, but usually in aggregating data over large sets.

E.g., calculate the average amount of money that customers under the age of 20 withdrew from ATMs in a certain region.

OLAP data originates from data found at the operational level, but it is then summarized and shaped by the requirements of the management (*multidimensional data model*).

This typically requires complex and time consuming transactions to pre-process data.

OLAP queries do not change data warehouse content.



# What is a Data Mart?

A data mart is focused on a single functional area of an organization and contains a subset of data stored in a Data Warehouse.

A data mart is a condensed version of Data Warehouse and is designed for use by a specific department, unit or set of users in an organization. E.g., Marketing, Sales, HR or finance. It is often controlled by a single department in an organization.

Data marts are smaller in size and are more flexible compared to a Data Warehouse.





# What is a Data Lake?

A Data Lake is a storage repository that can store large amounts of structured, semi-structured, and unstructured data.

- It is a place where to store every type of data in its native format with no fixed limits on size or type
- It allows to access data before the ETL process, thus it retains all data coming from the sources
- Data is only transformed when the user is about to use it (*schema on read*, vs. *schema on write* in the data warehouse)
- Storing information in a data lake is relatively inexpensive with respect to storing them in a data warehouse



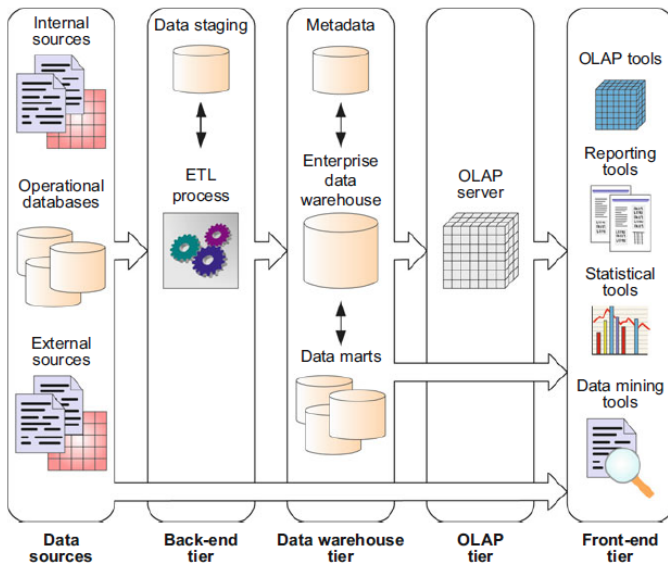
A Data Lake is not a substitute for a Data Warehouse.

For instance, a Data Warehouse guarantees quick answers for interactive queries thanks to the schema on write approach.

If not properly managed, a data Lake can grow without control and become useless (data swamp).

# Data Warehouse General Architecture

# Data Warehouse Architecture Schema





A modern general data warehouse architecture typically consists of several tiers:

- *The back-end tier* includes extraction, transformation, and loading (ETL) tools and a data staging area
- *The data warehouse tier* is composed of an enterprise data warehouse and/or several data marts and a metadata repository (e.g., schema definitions, data lineage)
- *The OLAP tier* is composed of an OLAP server, which provides a multidimensional view of the data
- *The front-end tier* is used for data analysis and visualization. It contains client tools such as OLAP tools, reporting tools, statistical tools, and data mining tools



# Extract, Transform, Load

The ETL process is in charge of loading data into the data warehouse, in bulk or as regular updates:

- *extract*: data is gathered from multiple, heterogeneous sources
- *transform*: data cleansing (errors and inconsistencies removal), integration (data reconciliation, e.g., formats) and aggregation (to the level of granularity of the data warehouse)
- *load*: regularly feed the data warehouse the new data

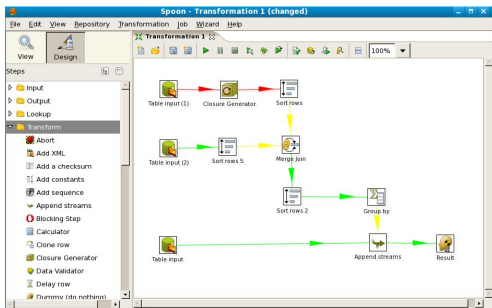
Such operations are typically performed within a *data staging area*, i.e., an intermediate database.

Always remember: *garbage in = garbage out*

*QuerySurge* is built specifically to automate the testing of Data Warehouses & Big Data.

*MarkLogic* is a NoSQL data warehousing solution that includes a fully-fledged data integration and data management solution.

*Pentaho Data Integration / Talend Open Studio* support the creation of complex ETL workflows.



# The Multidimensional Model





# The Multidimensional Model

The distinctive features of OLAP applications suggest the adoption of a multidimensional representation of data, since running analytical queries against traditionally stored information would result in complex query specification and long response times.

The multidimensional model relies on the concepts of *fact*, *measure*, and *dimension*, and makes use of two kinds of tables: *fact tables* and *dimension tables*.

As we shall see, the key idea here is that of pre-aggregating some of the data.



In a data warehouse context, a *fact* is the part of your data that indicates a specific event or transaction that has happened, like the sale of a product, or receiving a shipment.

A fact is composed of multiple numerical *measures*, that describe it.

As an example, a fact may be receiving an order for some shoes, detailed by the measures 'price' and 'quantity'.



*Dimensions* provide a way to **categorize/index facts**, e.g., considering spatial or temporal aspects.

The primary functions of dimensions are threefold: to provide filtering, grouping and labelling to facts.

Typically, dimensions are organized internally into one or more hierarchies. For instance, the dimension *date* may have the hierarchy:

- Days (grouped into) Months (grouped into) Years



Going back to our previous example, the order may be detailed by the following 2 *measures* and 3 *dimension attributes*:

- total amount US\$ 750
- quantity purchased is 10
- received yesterday at 2 pm
- served by our store in New York
- placed by customer #XAZ19

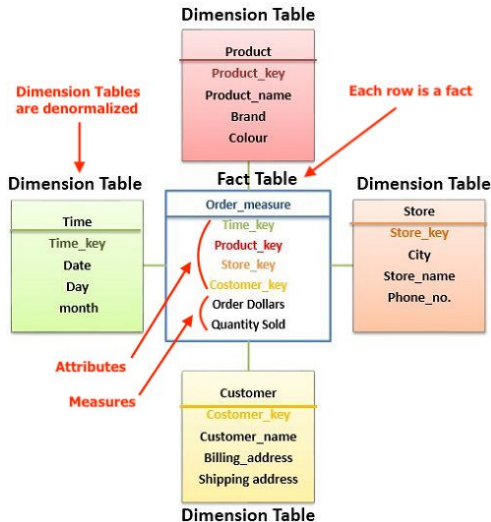
## Fact Table:

- A fact table is a central table in a dimensional model
- It contains facts/measures and foreign keys to dimension tables

## Dimension Table:

- A dimension table contains the dimensions of a fact
- There is no limit on the number of dimensions
- The dimension can also contain one or more hierarchical relationships

# Example (Star Schema)



# Operations over Multidimensional Data



# The Multidimensional Model

In the multidimensional model, data is represented in an  $n$ -dimensional space, usually called a data cube or a hypercube.

A data cube is defined by dimensions (cube edges) and facts (cube cells):

- Dimensions are perspectives used to analyze the data (their hierarchies represent the granularity/level of detail)
- Facts have related numeric values, called measures

Data cubes can be sparse: there may not be a cell value for each combination of dimensions.





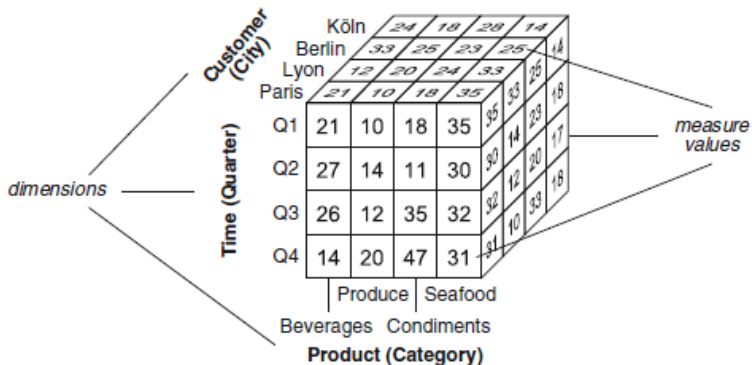
# 2-dimensional Data in a Spreadsheet

Bi-dimensional pivot table, considering:

- measure 'Amount'
- dimensions 'Place' and 'Product'
- facts are the amount of products sold in each country

	A	B	C	D	E	F	G	H	I	J
1	Category	(All) ▼								
2										
3	Sum of Amount	Column ▼								
4	Row Labels ▼	Apple	Banana	Beans	Broccoli	Carrots	Mango	Orange	Grand Total	
5	Australia	20634	52721	14433	17953	8106	9186	8680	131713	
6	Canada	24867	33775		12407		3767	19929	94745	
7	France	80193	36094	680	5341	9104	7388	2256	141056	
8	Germany	9082	39686	29905	37197	21636	8775	8887	155168	
9	New Zealand	10332	40050		4390			12010	66782	
10	United Kingdom	17534	42908	5100	38436	41815	5600	21744	173137	
11	United States	28615	95061	7163	26715	56284	22363	30932	267133	
12	Grand Total	191257	340295	57281	142439	136945	57079	104438	1029734	
13										

# 3-dimensional OLAP Cube Example





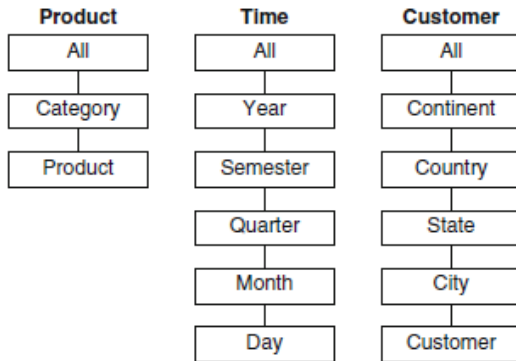
To extract strategic knowledge from a cube, it is necessary to view its data at several levels of detail.

Dimension hierarchies allow one to define a sequence of mappings relating lower-level, detailed concepts to higher-level, more general concepts.

As an example, dates of purchase could be aggregated into coarser grained levels of detail, such as months, or years.



# Hierarchy Example





# Aggregation of Measures

Each measure in a cube is associated with an aggregation function that combines several measure values into a single one.

Aggregation of measures takes place when one changes the level of detail at which data in a cube are visualized.

This is performed by traversing the hierarchies of the dimensions (e.g., from monthly to yearly sales).

Pay attention to the aggregation functions that you apply to each measure while navigating the hierarchies (e.g., mind the difference between *sum* and *count*).

The four types of analytical operations performed on OLAP cubes are:

- Roll-up
- Drill-down
- Pivot (rotate)
- Slice and dice

It involves summarizing the data along a chosen dimension (e.g., sum), navigating from a finer level of detail (down) to a coarser one (up).

Time (Quarter)	Customer (City)	Product (Category)			
		Produce		Seafood	
		Beverages	Condiments	Beverages	Condiments
		Product (Category)		Product (Category)	
		Product (Category)		Product (Category)	
Q1	Köln	21	10	18	35
Q1	Berlin	33	25	23	25
Q1	Lyon	12	20	24	33
Q1	Paris	21	10	18	35
Q2	Köln	27	14	11	30
Q2	Berlin	30	14	23	17
Q2	Lyon	26	12	35	32
Q2	Paris	31	10	33	18
Q3	Köln	14	20	47	31
Q3	Berlin	33	30	42	68
Q3	Lyon	33	30	42	68
Q3	Paris	33	30	42	68
Q4	Köln	33	30	42	68
Q4	Berlin	33	30	42	68
Q4	Lyon	33	30	42	68
Q4	Paris	33	30	42	68

(a) Original

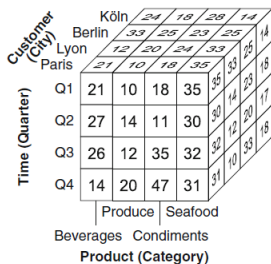
Time (Quarter)	Customer (Country)	Product (Category)			
		Produce		Seafood	
		Beverages	Condiments	Beverages	Condiments
		Product (Category)		Product (Category)	
		Product (Category)		Product (Category)	
Q1	Germany	33	30	42	68
Q1	France	33	30	42	68
Q2	Germany	33	30	42	68
Q2	France	33	30	42	68
Q3	Germany	33	30	42	68
Q3	France	33	30	42	68
Q4	Germany	33	30	42	68
Q4	France	33	30	42	68

(b) Roll-up to the Country level

# Drill-down

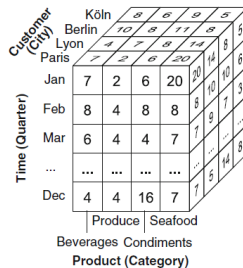
It allows the user to navigate among levels of data, ranging from the most summarized (up) to the most detailed (down), along a given hierarchy.

Looking at the detail beneath a summary number may be useful, especially where the summary number is surprising.



Time (Quarter)	Customer (City)	Product (Category)			
		Produce	Beverages	Condiments	Seafood
Q1	Köln	24	18	28	14
	Berlin	33	25	23	25
	Lyon	12	20	24	33
	Paris	21	10	18	35
Q2	Köln	35	33	14	23
	Berlin	30	14	20	17
	Lyon	27	14	11	30
	Paris	26	12	35	32
Q3	Köln	31	10	33	18
	Berlin	14	20	47	31
	Lyon	21	10	18	35
	Paris	27	14	11	30
Q4	Köln	26	12	35	32
	Berlin	31	10	33	18
	Lyon	21	10	18	35
	Paris	27	14	11	30

(c) Original

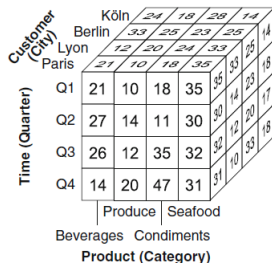


Time (Quarter)	Customer (City)	Product (Category)			
		Produce	Beverages	Condiments	Seafood
Jan	Köln	8	6	9	5
	Berlin	10	8	11	8
	Lyon	4	7	8	14
	Paris	7	2	6	20
Feb	Köln	20	14	8	6
	Berlin	8	10	10	3
	Lyon	6	4	4	7
	Paris	7	9	7	...
Mar	Köln	...	...	...	...
	Berlin	...	...	...	...
	Lyon	...	...	...	...
	Paris	...	...	...	...
Dec	Köln	4	4	16	7
	Berlin	7	5	14	8
	Lyon	...	...	...	...
	Paris	...	...	...	...

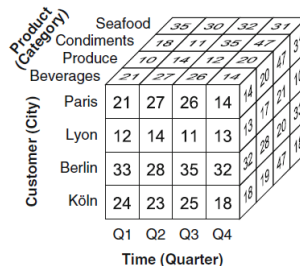
(d) Drill-down to the Month level



This operation allows an analyst to rotate the cube in space to see its various faces.



(e) Original



(f) Pivot to show Customer vs. Time

It is the act of picking a rectangular subset of a cube by choosing a single value for one of its dimensions, obtaining a new cube with one fewer dimension.

Time (Quarter)	Customer (City)	Köln				Berlin				Lyon				Paris			
		24				18				28				14			
		33				25				23				25			
		12				20				24				33			
		21				10				18				35			
Q1	21	10	18	35	35	30	14	23	16								
Q2	27	14	11	30	30	12	20	17	15								
Q3	26	12	35	32	32	10	33	15									
Q4	14	20	47	31	31												
		Produce				Seafood											
		Beverages				Condiments											
		Product (Category)															

(g) Original

f

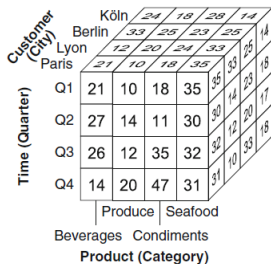
Time (Quarter)

Q1	21	10	18	35
Q2	27	14	11	30
Q3	26	12	35	32
Q4	14	20	47	31

Produce Seafood  
Beverages Condiments  
Product (Category)

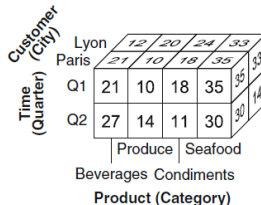
(h) Slice on City='Paris'

The dice operation is a generalization of slice. More than one dimension can be tested, with multiple conditions.



		Customer (City)				
		Köln	Berlin	Lyon	Paris	
Time (Quarter)	Q1	21	10	18	35	35
	Q2	27	14	11	30	30
	Q3	26	12	35	32	32
	Q4	14	20	47	31	31
		Produce		Seafood		
		Beverages		Condiments		
		Product (Category)				

(i) Original



		Customer (City)			
		Lyon	Paris		
Time (Quarter)	Q1	21	10	18	35
	Q2	27	14	11	30
		Produce		Seafood	
		Beverages		Condiments	
		Product (Category)			

(j) Dice on City='Paris' or 'Lyon' and Quarter='Q1' or 'Q2'



# Implementing OLAP

- Extensive usage of pre-computed aggregation tables
- Precomputing all the possible aggregations is space and time expensive
- $2^n$  GROUP BY combinations for  $n$  dimensions
- It is better to precompute only some aggregated functions and derive the others exploiting the previous ones
- For instance sums wrt (item\_name; color) can be obtained from (item\_name; color; size)
- However, this is not possible in some cases (e.g., median)



- Talend, open source, [www.talend.com](http://www.talend.com) (ETL)
- Kettle Pentaho, o.s., [kettle.pentaho.com](http://kettle.pentaho.com) (ETL)
- Pentaho Business Analytics, [www.pentaho.com](http://www.pentaho.com)
- Weka, o.s., [www.cs.waikato.ac.nz/ml/weka](http://www.cs.waikato.ac.nz/ml/weka) (Data mining)
- RapidMiner, o.s., [www.rapidminer.com](http://www.rapidminer.com) (ETL, data mining)
- Mondrian, o.s., [mondrian.pentaho.com](http://mondrian.pentaho.com) (OLAP)
- Palo, o.s., [sourceforge.net/projects/palo](http://sourceforge.net/projects/palo) (MOLAP)
- jPivot, o.s., [jpivot.sourceforge.net](http://jpivot.sourceforge.net) (client per Mondrian)
- Jasper, o.s., [jasperforge.org](http://jasperforge.org) (ETL, OLAP, . . . )
- Wabit, o.s., [code.google.com/p/wabit](http://code.google.com/p/wabit) (OLAP)



## Software for DW (continued)

- Teradata, [www.teradata.com](http://www.teradata.com) (row/column store)
- Greenplum, [www.greenplum.com](http://www.greenplum.com) (row/column store)
- HP Vertica, [vertica.com](http://vertica.com) (column store)
- Aster Data, [www.asterdata.com](http://www.asterdata.com) (DBMS, map/reduce, R)
- Oracle BI, Essbase, . . . , [oracle.com](http://oracle.com) (vari prodotti)
- Sybase IQ, [www.sybase.com/products](http://www.sybase.com/products) (column store)
- IBM DB2, [www.ibm.com/software/data/db2](http://www.ibm.com/software/data/db2)
- IBM Netezza, [www.netezza.com](http://www.netezza.com)
- IBM Cognos, [www.ibm.com/software/analytics/](http://www.ibm.com/software/analytics/)
- HP Vertica, [vertica.com](http://vertica.com) (column store)
- LucidDB, o.s., [www.luciddb.org](http://www.luciddb.org) (column store)
- MonetDB, o.s., [www.monetdb.org](http://www.monetdb.org) (column store)
- SciDB, o.s., [www.scidb.org](http://www.scidb.org)



A. Vaisman, E. Zimányi *Data Warehouse Systems - Design and Implementation*, 2014

A. Silberschatz, H. F. Korth, S. Sudarshan *Database system concepts*, 7th Edition, 2020

W. I. Immon *Building the Data Warehouse*, 4th Edition, 2005