# BIG Data Warehouses

*A Big Data Perspective on Data Warehouses*

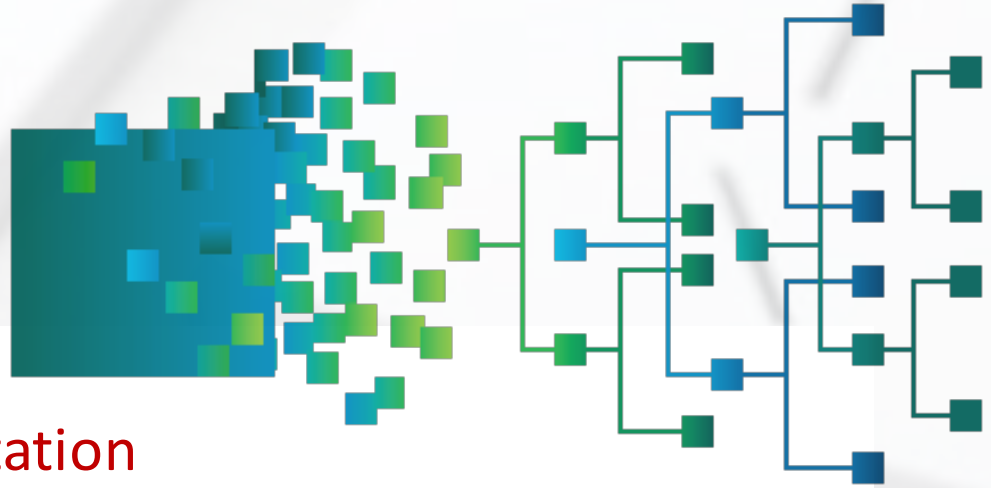
Seminario per il corso di

Sistemi di Elaborazione di Grandi Quantità di Dati

Francesca Zerbato

francesca.zerbato@univr.it

# Big Data Warehouse



Technological and implementation GAP

what enterprise has

Traditional Data Warehouse

Big Data scenario

what enterprise wants to become

hybrid data warehouse architecture

Big Data Warehouse (BDW)


# Presentation outline

1. Data Warehouse: the traditional business intelligence approach
   - Introduction to data warehousing, DFM conceptual model and ROLAP logical design

2. The arrival of Big Data: the need for scalability in DW architecture
   - Types of data in Big Data, Business Relevance and Big Data architectural requirements

3. Data Vault: an approach to enhance DWs to deal with Big Data challenges
   - Introduction to Data Vault 2.0 model and architecture

# Time(out)line

*The role of IT from passive to active*

RDBMS and relational DB model

1970

Goal: Analyze data under different perspectives to make decisions

2000

Data Vault Model: RDBMS + NoSQL

**Transactional Databases** | **Business Intelligence and DW** | **Big Data and NoSQL**

Goal: reliability, make sure no data is lost

1985

Data Warehouse and ROLAP star schema

Goal: data for the masses, everyone has access to everything

# Data ≠ Information

*An explicative (business intelligence) example*

**Problem:** Sales for lollipops have gone down in the last 6 months.

**Data:** Sales records, customer data, social network data, market analysis. Data records are grouped by time, region, customer age.

**Information:** Lollipops are bought by females older than 25 to be eaten by people younger than 10.

**Knowledge:** Mothers believe that lollipops are bad for children teeth.

**Value:** Hire a dentist to advertise lollipops.

# Data Warehouse

*Definition*

A data warehouse is a subject-oriented, integrated, time-variant and non-volatile collection of data in support of management's decision making process.

- Subject-oriented: analysis of subject areas.
- Integrated: data comes from multiple sources.
- Time-variant: historical data are collected.
- Non-volatile: no data modification/removal.

"A data warehouse is a copy of transaction data specifically structured for query and analysis" - Ralph Kimball, major DW technology contributor

# Data Warehouse

*OLTP vs OLAP systems*


On-Line Transaction Processing (OLTP): data processing system facilitating management of transaction-oriented software.

*Large number of short on-line transactions (INSERT, UPDATE, DELETE).*

*Data is detailed, not historical, highly normalized, join does not perform well.*

On-Line Analytical Processing (OLAP): data processing system enabling the analysis of multidimensional data, interactively and from multiple perspectives. DATA WAREHOUSES are designed to support OLAP operations.

*Queries are often very complex and involve aggregations.*

*Data is historical, denormalized, redundant, join is easier and faster.*

# Data Warehouse

*Architectural and use requirements*

DW use goals:

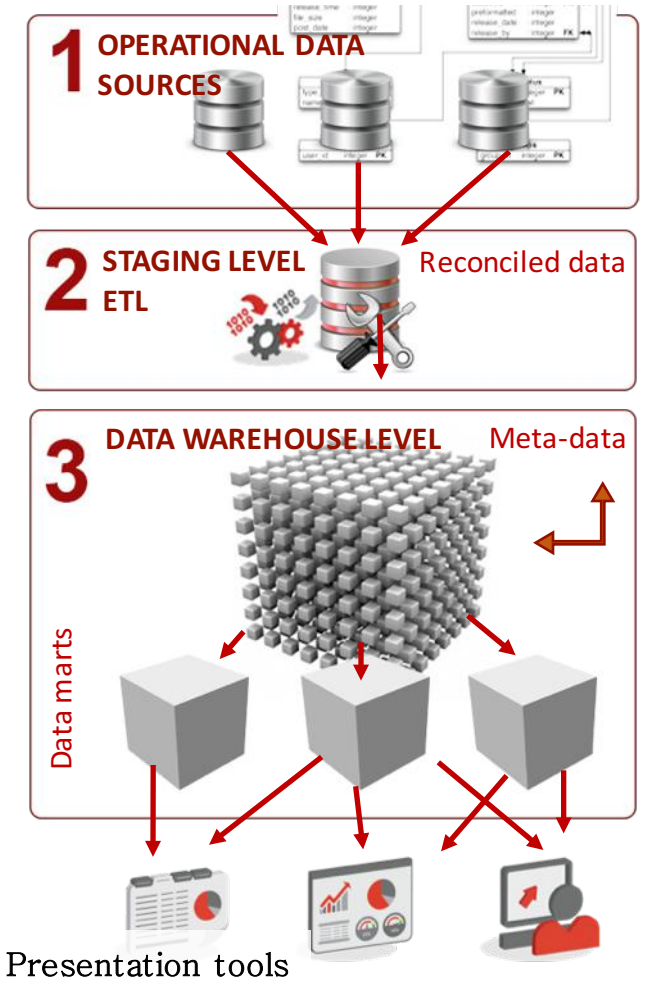
- Correctness and completeness of integrated data. Single version of the truth.
- Accessibility to users with limited knowledge of computing.
- Data is summarized/aggregated for flexible query and intuitive view.

Requirements for a DW architecture:

- Scalability: hardware and software architecture must be easily scaled.
- Extensibility: must be able to add new applications.
- Security: access control is required because of the nature of the data stored. Strategic data are memorized.

# Data Warehouse

*3-Tier Architecture*




1. Sources: operational data sources, flat files.

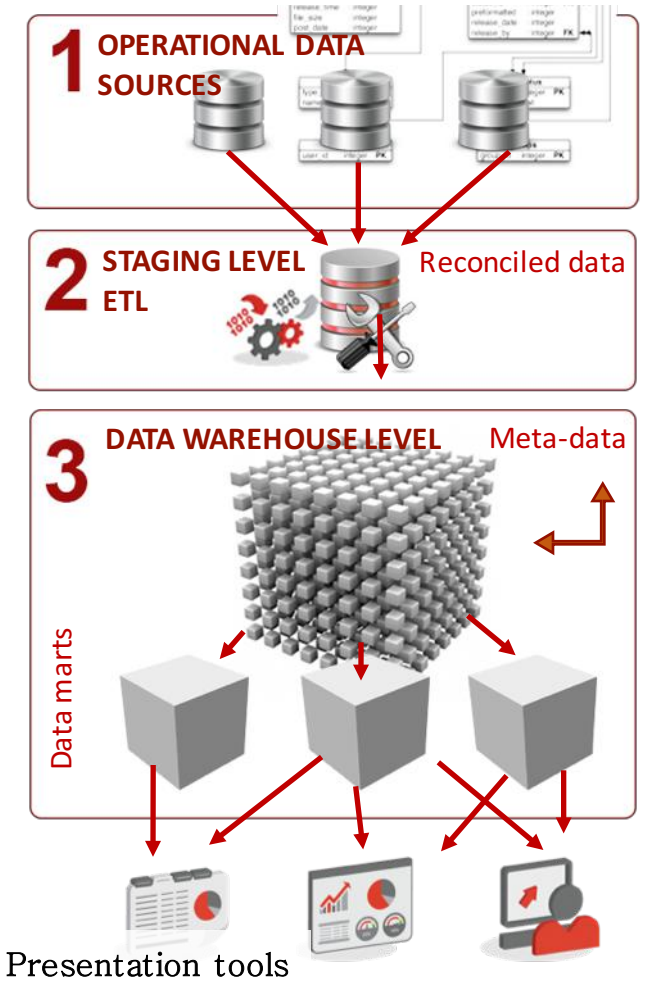
2. Reconciled Data Level: Extraction-Transformation-Loading (ETL) tools to obtain a reconciled data level before feeding the DW.

3. Data Warehouse Level: central data warehouse, data marts and meta-data repository.

Presentation (front-end): Analysis and visualization OLAP tools, data-mining tools, reporting tools, what-if analysis tools.

# Data Warehouse

*3-Tier Architecture*




1. Sources: operational data sources, flat files.

2. ction-
Tra tools to obtain a
re eding the DW.

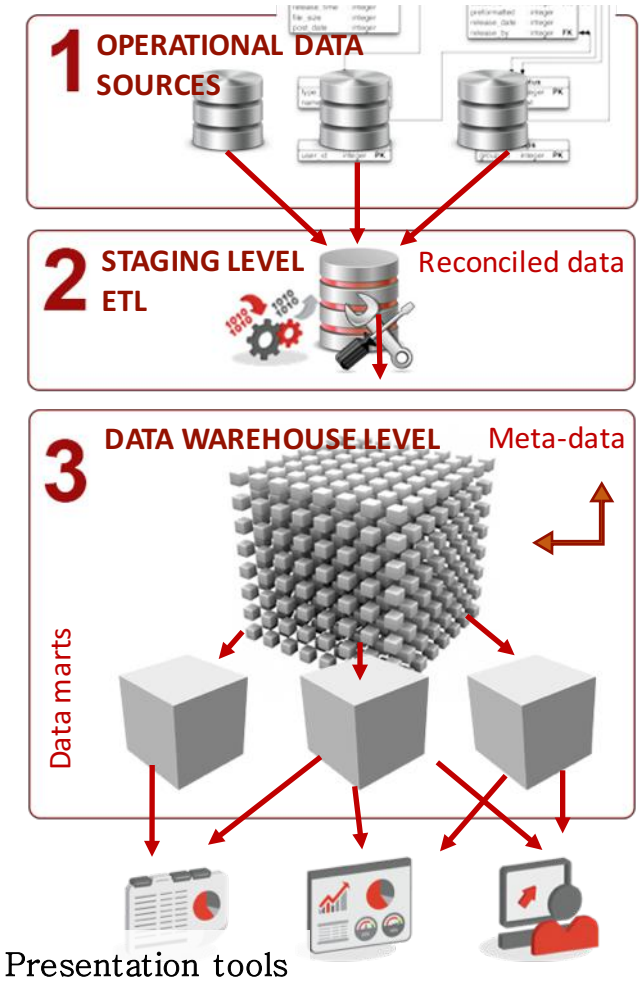
Data Marts are a subset or an aggregation of data stored in primary DW, targeted towards a particular functional area or user group.

3. Data arehouse Level: central data warehouse, data marts and meta-data repository.

Presentation (front-end): Analysis and visualization OLAP tools, data-mining tools, reporting tools, what-if analysis tools.

# Data Warehouse

*3-Tier Architecture*




1. Sources: operational dat

2.
Tra
re

Data Marts are a subset or aggregation of data stored primary DW, targeted towar particular functional area or group.

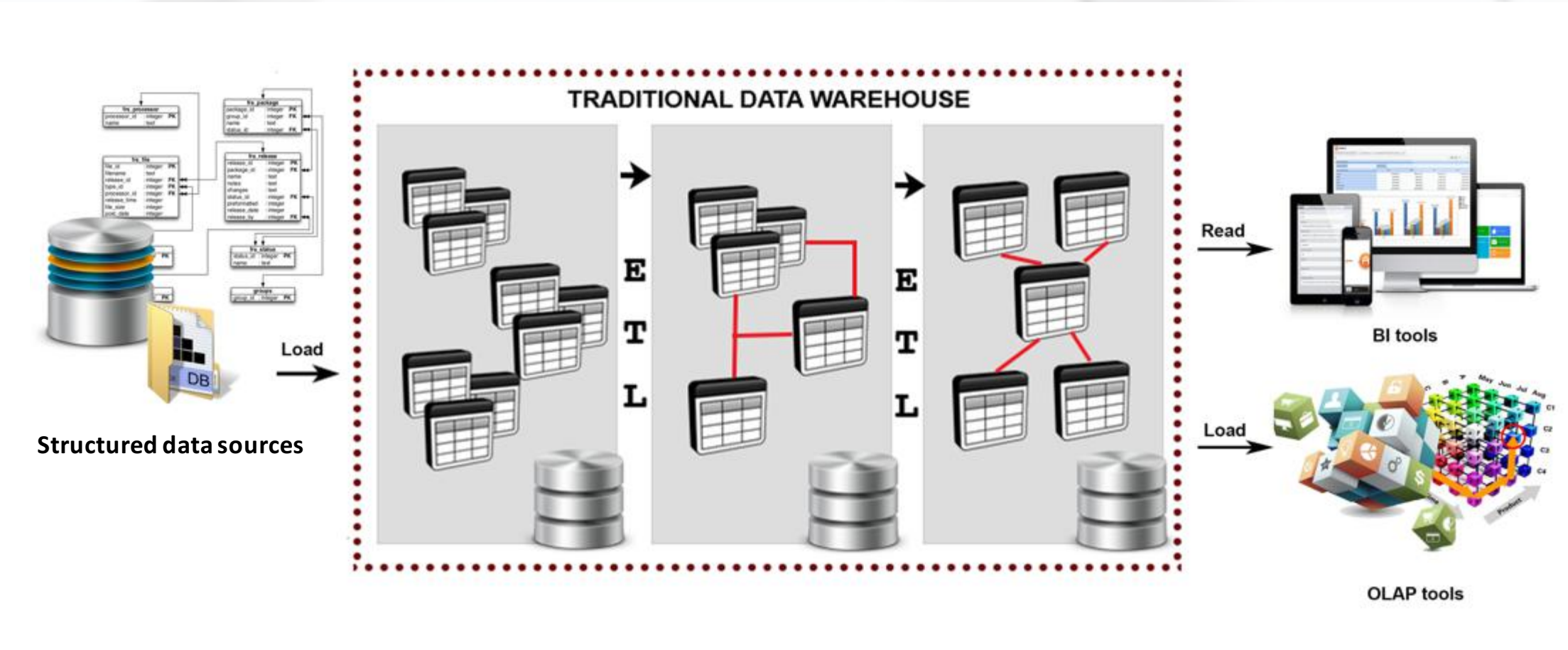
Meta-data is "data about data". Business meta-data describes semantics, business rules and constraints. Technical meta-data describes how data is stored and how it should be manipulated.

3. Data arehouse Level: central data wa use, data marts and meta-data repository.

Presentation (front-end): Analysis and visualization OLAP tools, data-mining tools, reporting tools, what-if analysis tools.

# Data Warehouse

*Architecture, another view*



# Data Warehouse

*Extraction-Transformation-Loading (ETL) tools*

ETL tools feed a single data repository, detailed, comprehensive and of high quality, which may in turn feed the DW

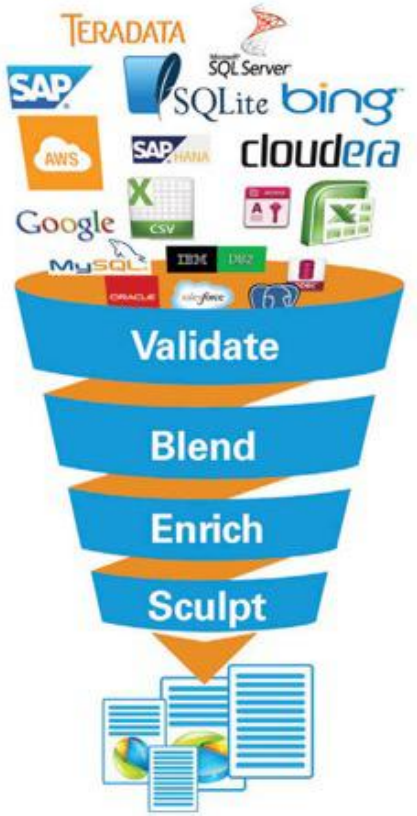
(Reconciliation process: reconciled data level.)

- Offline, carried out when DW is not in use (at night?).
- Batch processing.
- A subset of data, identified by business goals is obtained:

  GIVE A SINGLE VERSION OF THE TRUTH.

**Extraction:** data are gathered from sources.

- INTERNAL transactional systems, flat files.
- EXTERNAL sources. *ODBC, JDBC.*



# Data Warehouse

*Extraction-Transformation-Loading (ETL) tools*

**Transformation:** data is put into the warehouse format.

Business rules are used to define either presentation/visualization of data and persistence characteristics.

- **Cleaning:** removes errors, inconsistencies and converts data into a standardized format.
- **Integration:** data is reconciled, both at schema and data level.
- **Aggregation:** data is summarized according to the DW level of detail.

**Loading:** the DW is fed with cleaned and transformed data, offline.

Initial load (first DW population) or refreshment.

# Data Warehouse: data modeling

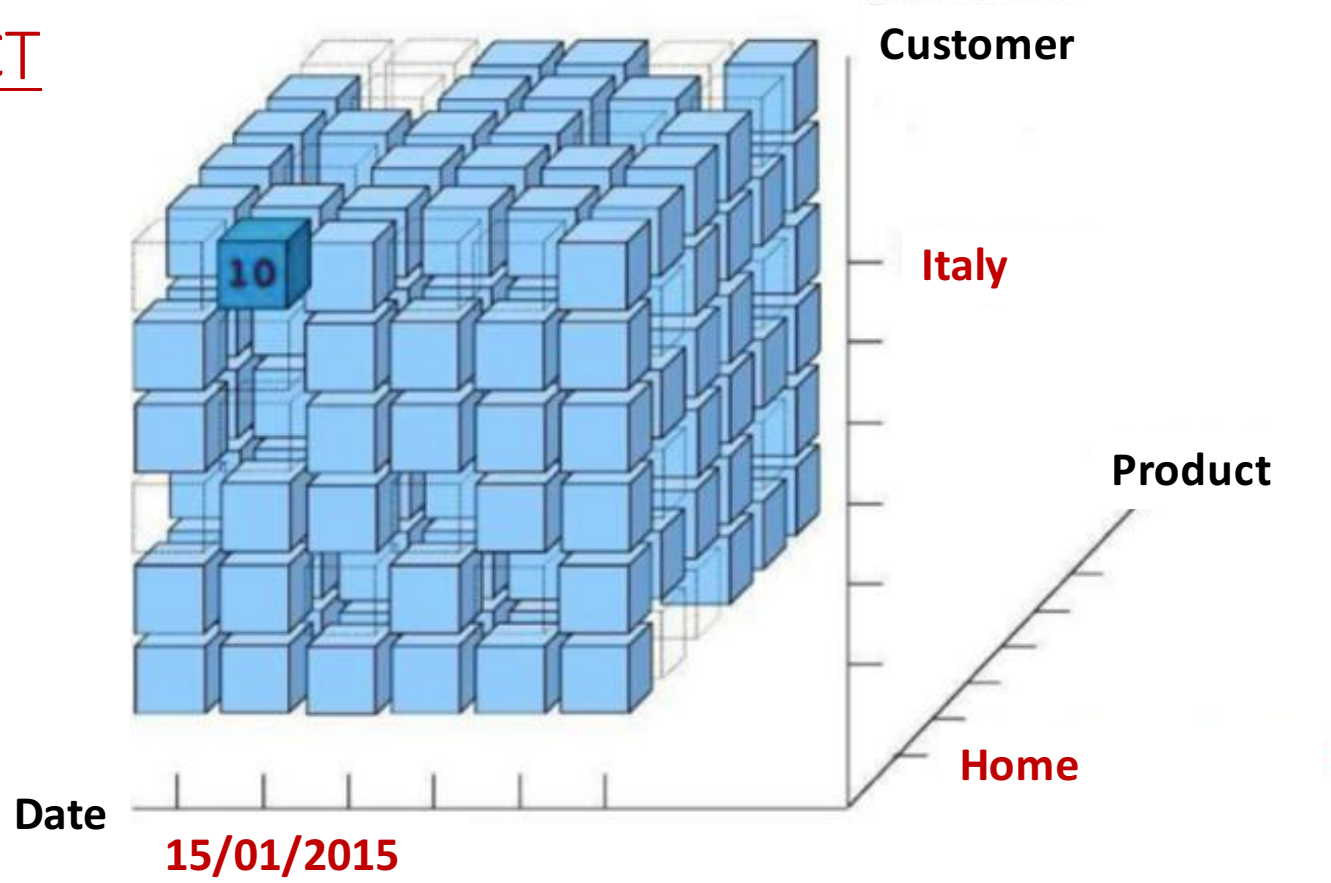
*Multidimensional Model: DFM (dimension-fact model) – DATA CUBE*

Each cell of the cube is a FACT of interest quantified by numerical measures.

Each axis represents a dimension of interest for the analysis.

Hierarchy of attributes:

- Product
  - Category (Home)
    - Sub-category (Bedroom)

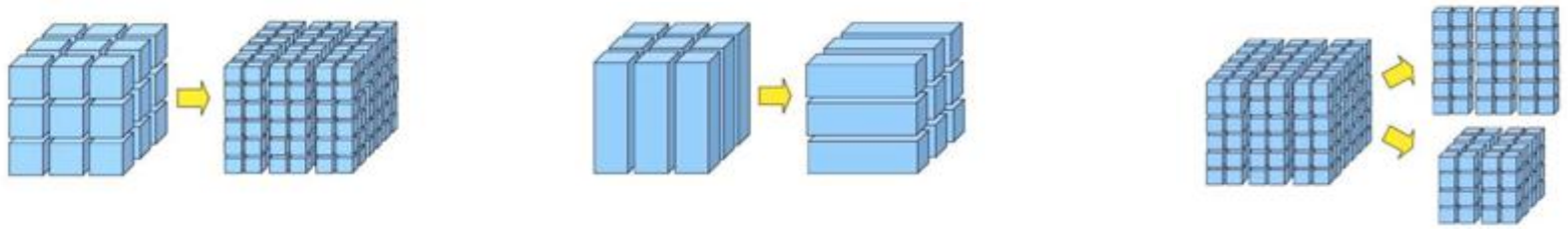


# Data Warehouse: data analysis

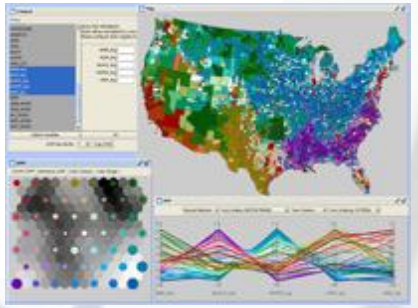
*Analyses on the data cube*

- Reporting: periodical access to structured information.

- OLAP: analysis of one or more facts of interest at different levels of detail by sequence of queries that give a multidimensional result.



- Data mining: extracting patterns from large data sets by combining methods from statistics and artificial intelligence with database management.



# Data Warehouse: logical model

*STAR Schema: relational OLAP data model*

Most Data cubes are built on the relational model.

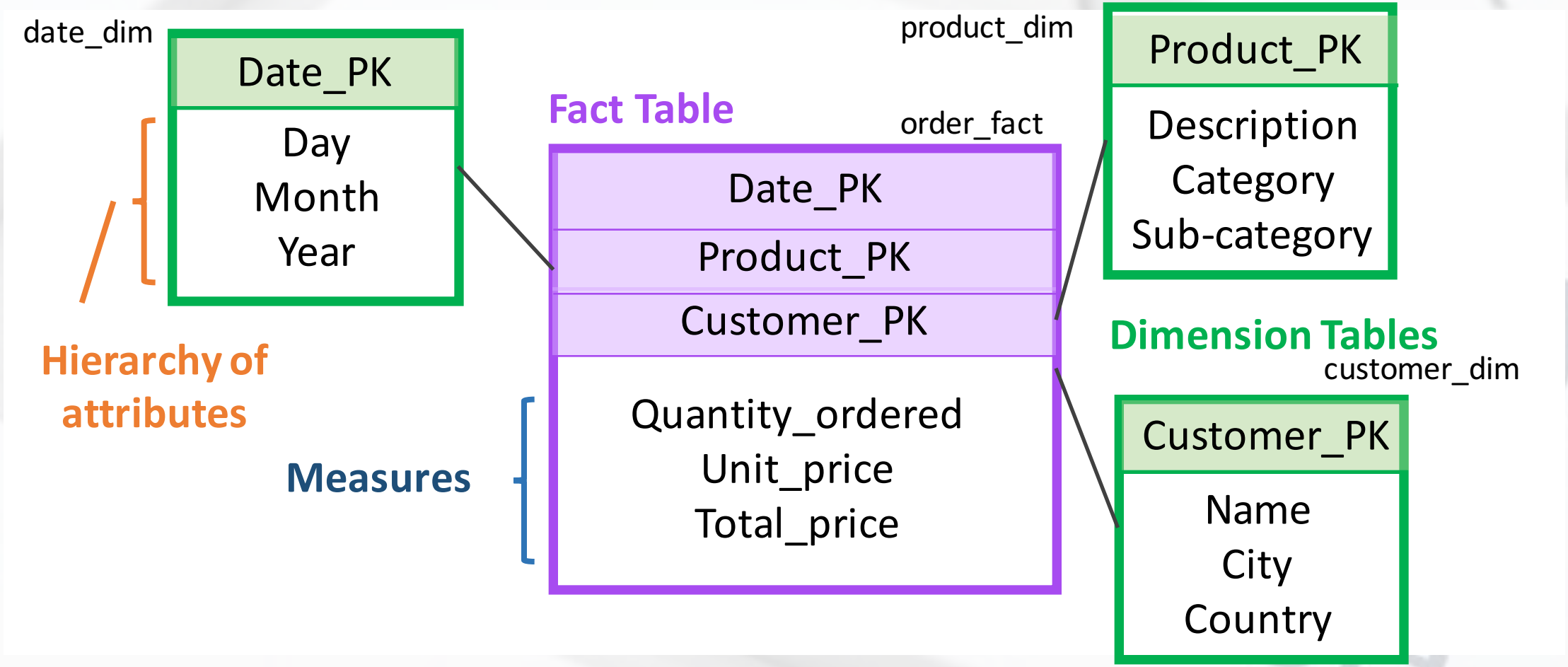
A star schema is composed by:

- A central relation FT, Fact Table, representing the fact of interest.
- A set of relations called dimension tables, each of them corresponding to one dimension of the analysis (cube axis). Every dimension table is characterized by
  - a primary key
  - a set of attributes that describe the dimensions of analysis at different levels of aggregation.

FT also contains an attribute for each measure.

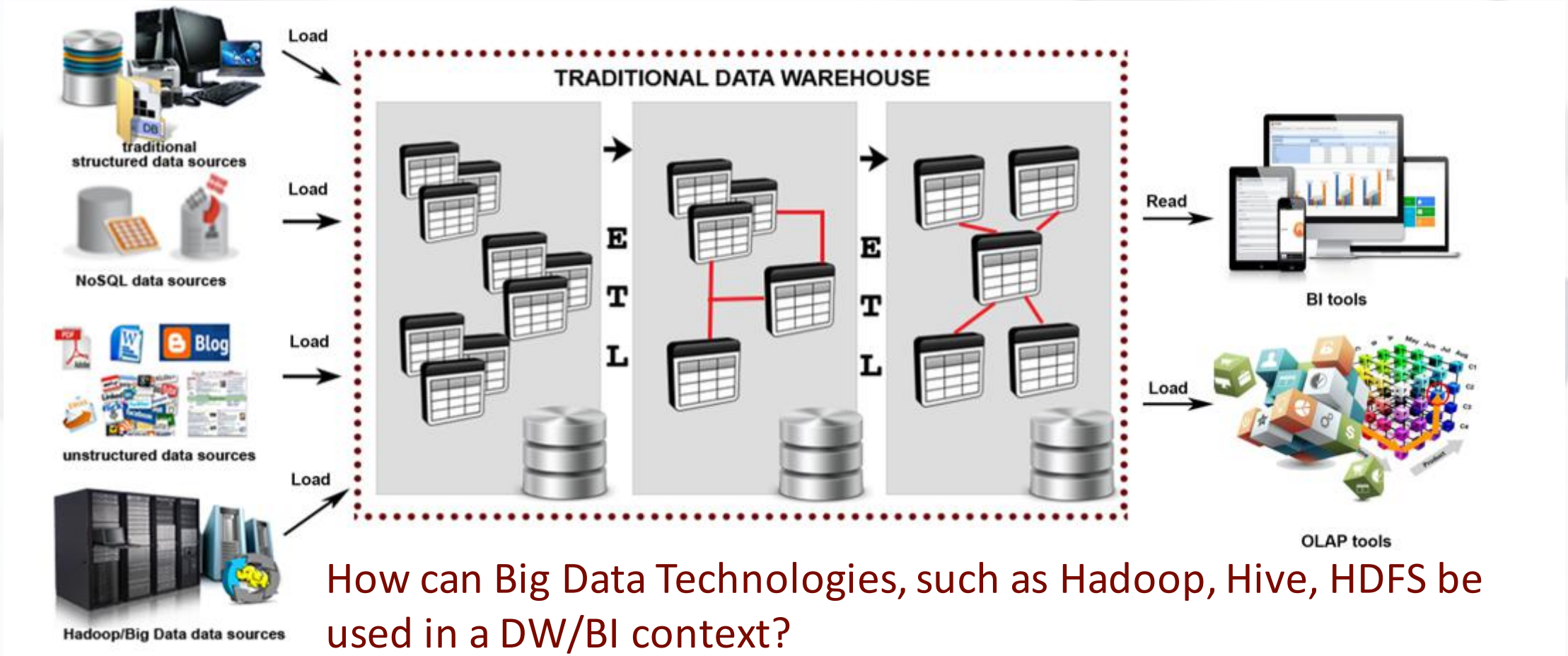
# Data Warehouse: logical model

*STAR Schema: example*



# Data Warehouse

*The arrival of Big Data: can a traditional DWH handle all this data?*




How can Big Data Technologies, such as Hadoop, Hive, HDFS be used in a DW/BI context?

# Big Data Warehouse systems

*The arrival of Big Data: can a traditional DW handle all this data?*

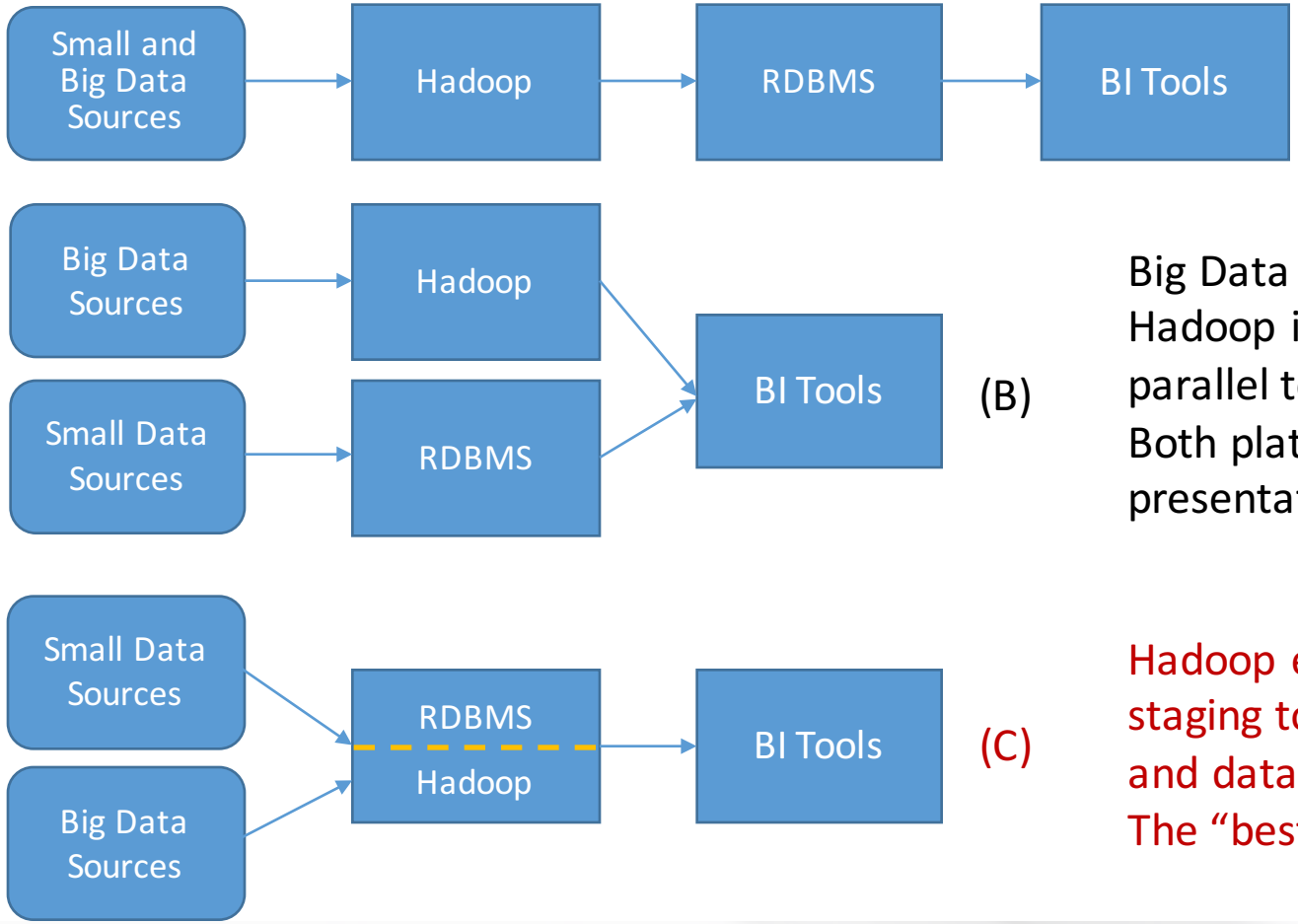
In Business Intelligence Big Data technologies can be used:

1. Standalone: with their own query/DB tools, query languages.
   - Analytical goals: Business requirements are not previously defined.


2. Complementary and supporting to enhance existing DW technologies: hybrid systems called **BIG DATA WAREHOUSES.**
   - Synergy among Big Data technologies and existing DW: wider range of data (Facts are integrated with unstructured and multidimensional data).
   - Improve scalability and reduce costs of current DW systems.

# Big Data Warehouse systems

## *Hybrid approaches*




(A) Hadoop is used only for data ingestion/staging

Big Data are kept separate from structured data: Hadoop is used as data management platform in parallel to the RDBMS.
Both platforms are used in conjunction for presentation purposes.

Hadoop enhances RDBMS as data ingestion/ staging tool, but also as data management and data presentation platform.
The "best" of both technologies is exploited.

# Big Data Warehouse systems

*Some questions*

*Before addressing Big Data Warehouses we should answer some questions..*

- BI on Big Data?
- What kind of data are found in Big Data?
- Can a traditional ETL technology handle Big Data?
- If feasible, does ETL on Big Data make sense?

# Corporate Data

*The role of Big Data in the corporation*

Corporate data is the totality of data found in a corporation.

Examples of corporate data are:

analog information, telephone records, e-mails, market research data, call center records, payments, sales, transactions, measurements, interviews, social networks..


One way of classifying the totality of corporate data is dinstinguishing between STRUCTURED and UNSTRUCTURED DATA.

# Corporate Data

*Structured data*

defined length

defined format

Structured data has a predictable and regularly occurring format.

Typically:

- it is managed by a Database Management System (DBMS).
- consists of records or files, attributes, keys and indexes.
- a fixed number of fields is defined.

Examples of structured data are those contained in a **relational DB:** a data model is clearly defined for data representation, storing, processing, accessing and querying. (ACID compliant)

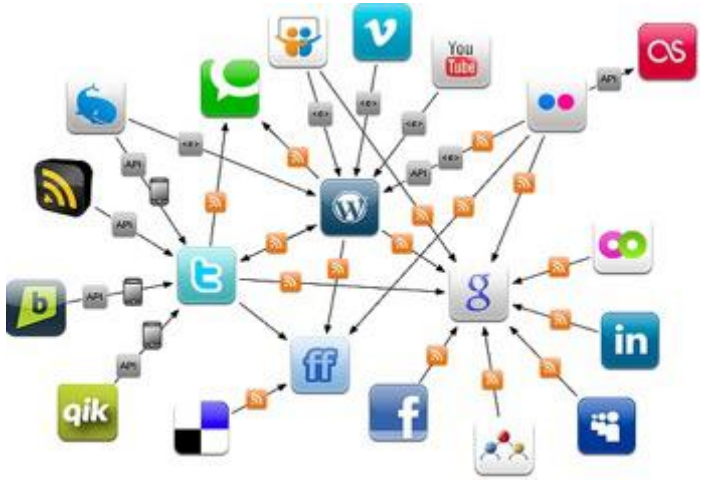
Traditional data warehouses manage structured data!

# Corporate Data

*Unstructured and semi-structured data: BIG DATA*

Unstructured data is *unpredictable*, and usually does not have an easily computer-recognizable format.

Long strings have to be searched (parsed) in order to find a unit of data!



Examples: free-text, images, videos, webpages, web server logs, …

Semi-structured data has *tags/markers* that help in discerning different data elements, but it **lacks of a strict data model**.

Examples of semi-structured data are: RSS feeds, metadata. Formats: XML, JSON, ...

# Corporate Data

## *Repetitiveness in Big Data*

Unstructured data can be divided into:

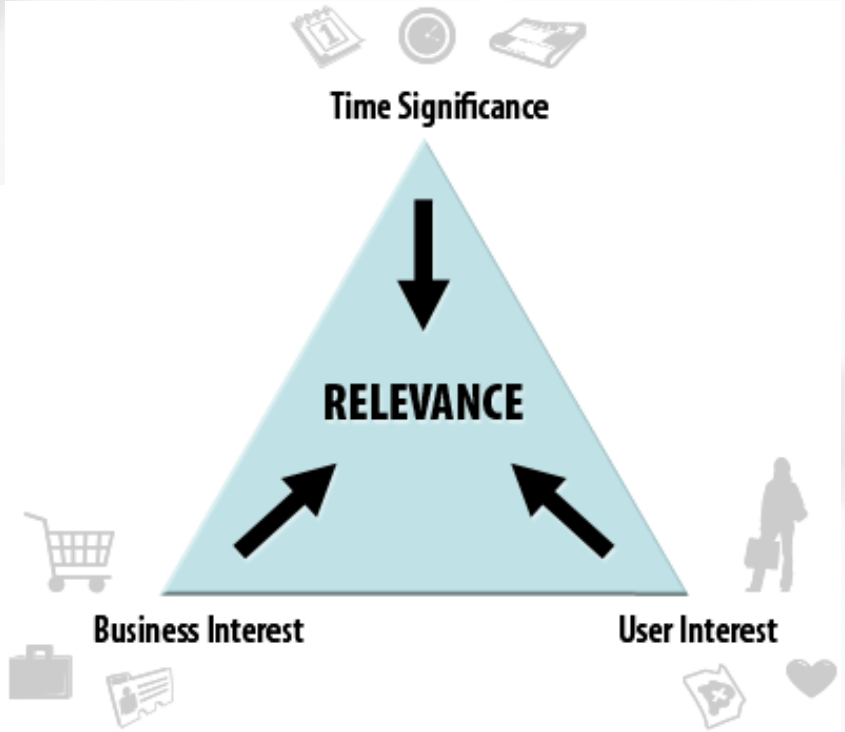
- REPETITIVE: it occurs many times, often in the same embodiment. Typically, this kind of records comes from machine interactions.
  Processing and analysis: Hadoop centric data.
  *Examples: analog processing, telephone call records.*

- NON-REPETITIVE unstructured data: records are substantially different from each other in form and content.

  Processing and analysis: NLP, Textual disambiguation: data is put into context and reformatted for standard BI analysis.
  *Examples: e-mails, healthcare records, market research, meteorological records.*

# Corporate Data

*Repetitiveness measures Business Relevance*



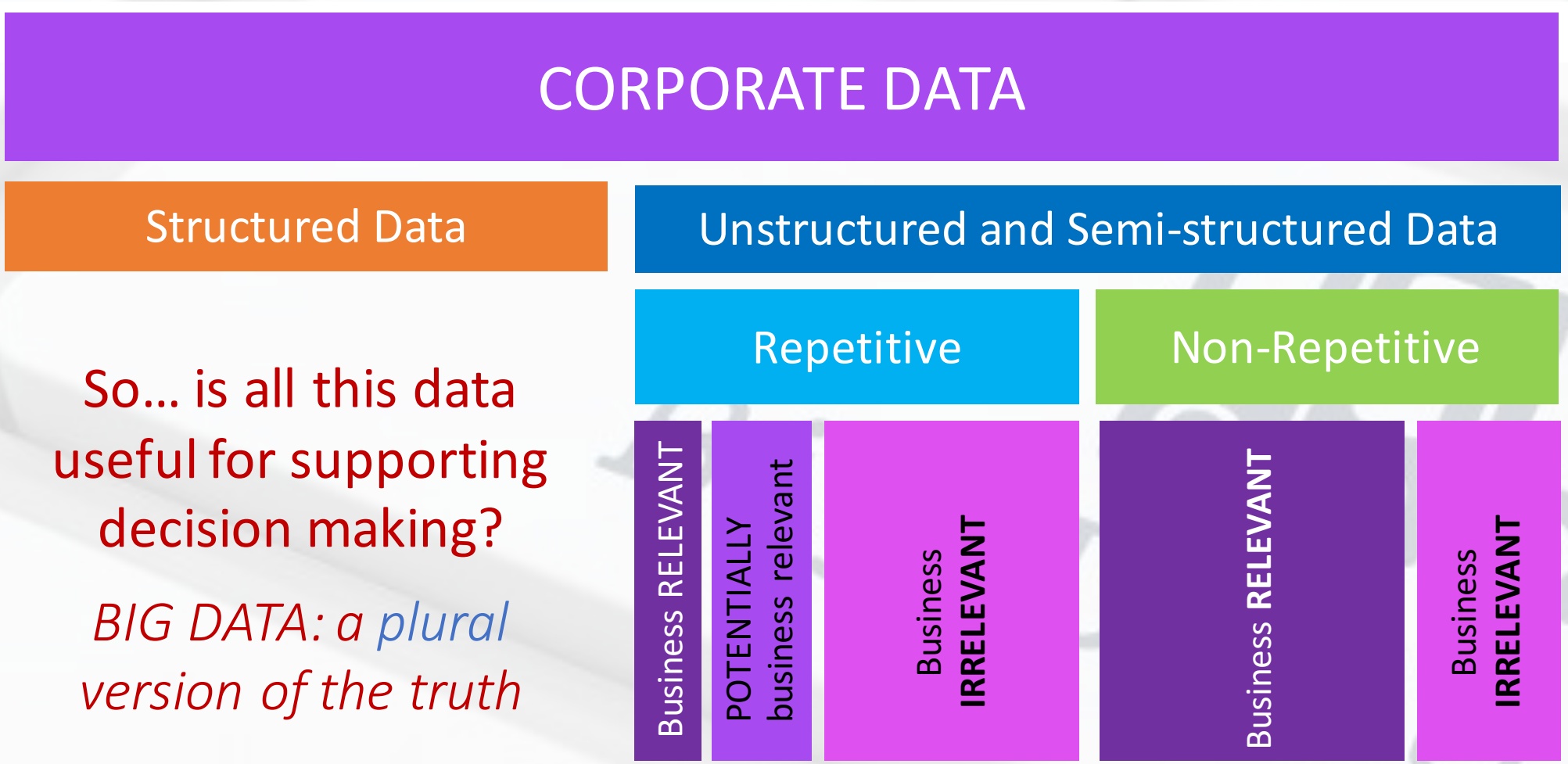

Business relevance measures the capability of data to provide information that is of interest for a specific business context.

Business relevant information is used to support decision making, solution generation and cost optimization.

REPETITIVE BIG DATA are hardly ever business relevant: *Millions of phone call records, only a few of which are relevant for governmental purposes.*

# Corporate Data

*A complete picture of corporate data*




So… is all this data useful for supporting decision making?

*BIG DATA: a plural version of the truth*

# Big Data require for DW improvements

*The need for an ecosystem to integrate Hadoop and NoSQL technologies*

Big Data require a different approach to data warehousing:

- *Volume:* Memorization and processing must be parallelized.
  - Huge workload, concurrent users and data volumes require optimization of both logical and physical design.
- ETL phase is a bottleneck and "nonsense" for Big Data: Big data goal is to gather data to be used in ways that have not been planned.
  - Discover/extract new insights in data: Exploratory approach
  - Processing is on data: lineage and meta-data are required.
  - Traditional ETL does not work well on unstructured data. Manual coding for data integration.
  - Raw data persist in the warehouse: lineage through soft business rules can be postponed according to analysis needs. Big Data call for ELT.

# Big Data require for DW improvements (II)

*The need for an ecosystem to integrate Hadoop and NoSQL technologies*

- Data complexity increases:
  - *Variety* of data requires specific processing techniques
    - Textual disambiguation, parsing, machine-generated data analysis.
  - *Velocity* of data requires almost real-time analysis capabilities:
    - Real-time data should feed directly to the DW: On-Line Transactional Processing can in part be carried out in the warehouse. Real-time data cannot undergo ETL!
  - *Veracity* of data requires strong integration and traceability.

- Analytical complexity:
  - Big data have to be in a format not foreseen by DW developers to be analyzed.

# Big Data require for DW improvements (III)

*The need for an ecosystem to integrate Hadoop and NoSQL technologies*

- Query complexity: temporal analysis and OLAP analysis on cubes are not feasible on Big Data. OLAP is optimized for relational models.
- DW Availability: addition of new data sources might compromise the availability of the overall system. It has to be carried out offline.
  - Parallelization of loading is one solution, but it must be embedded in the system.

Vertical Scaling: move to larger computers +

Horizontal Scaling:

✓ Functional scaling = organize similar data groups and spread them across DBs.

✓ Sharding = split data within the areas of functionality across multiple DBs.

# Data Vault 2.0 (DV2)

*Common Foundational Warehouse Architecture*

"The Data Vault Model is a detail oriented, historical tracking and uniquely linked set of normalized tables that support one or more functional areas of business. It is a hybrid approach encompassing the traditional star schema. The design is flexible, scalable, consistent and adaptable to the needs of the enterprise"

Goal: provide and present information, extracted from data that has been aggregated, summarized, consolidated and put into context.

# Data Vault 2.0 (DV2)

*Aspects*

1. Data model: changes to the model for performance and scalability.
   Raw data (structured + Big Data) are integrated by business keys.

2. Methodology: Scrum and Agile best practices:
two-to three-week sprint cycles with adaptations and optimizations.

3. Architecture: inclusion of NoSQL and Big Data systems for unstructured data handling and Big Data integration. Separation of business rules.

4. Implementation: Guidelines define how to implement DV2 parts.

# Data Vault 2.0 (DV2)

*Architecture*

Based on the 3-tier DW architecture: (1) staging layer, (2) enterprise data warehouse EDW layer and the (3) information delivery layer.

Additional components:

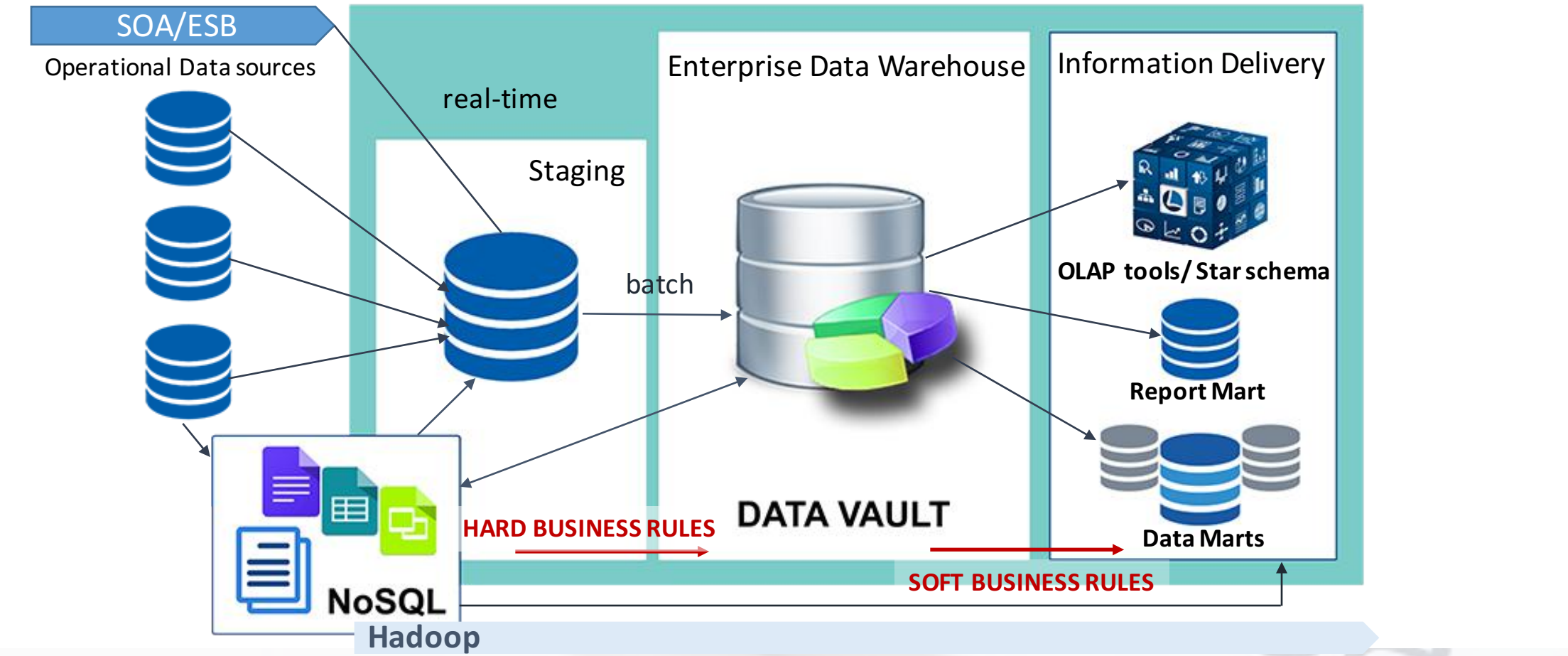
1. Hadoop or NoSQL handle Big Data (design rules on *where* and *how*)
2. Real-time information flows in/out of the EDW. Operational vault.
3. Hard and soft business rules are split.

- Data interpretation is postponed: Big data principle: data first - schema later!

Staging layer is losing importance as *raw data* persist in the EDW!

# Data Vault 2.0 (DV2)

*Architecture preview*



# DV2 Architecture

*Where do NoSQL platforms fit in DV2?*

Most common NoSQL platforms are based on Hadoop and HDFS.

- Staging: Hadoop is mostly used for data ingestion and staging for ANY DATA (structured and unstructured) that can proceed in the EDW.

- EDW: NoSQL DBs are used to store unstructured data.

- Information delivery: Hadoop is used to perform data mining.

Mining results are structured data sets that can be copied into relational database engines for ad hoc querying.

The DV2 model allows for NoSQL technologies to feed all 3 levels!

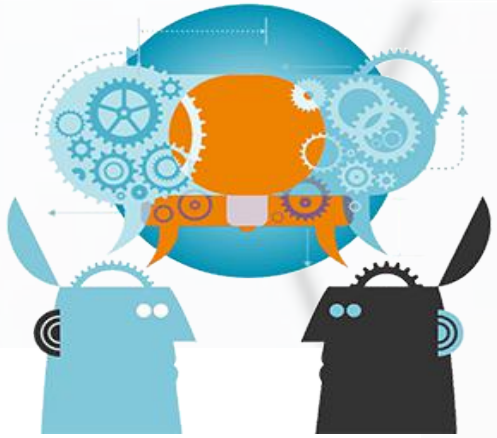
# DV2 Architecture

*How does Hadoop technology enhance DW capabilities?*

- Cheap hardware for memorization of all kinds of data.
- Local storage (preferred to Storage Area Networks).
- Allows processing directly on data and based on the kind of data:
  - Some Big Data might have a complex structure (web logs, complex sensors).
- Raw data persist in Hadoop: Transformation can be redone without the need for Extraction. History is easily maintained.
- Raw data can be re-used to add context or constraints.
- Data mining models extracted with Hadoop can be used as reliable semantic meta-data.

# DV2 Architecture

*Business Logic: Soft and Hard business rules separation*



Business rules are requirements translated into code.

In traditional DW business rules are applied before the loading phase.

DV2 IDEA: separate data interpretation (done after loading data into EDW) from data storage and alignment rules. Inside EDW raw data is preserved!

Hard rules: do not change the content of individual fields.

Examples: type alignment, split by record structure, denormalization.

---

Soft rules: change and interpret data.

Examples: standardizing name addresses, coalescing, concatenating name fields.

In traditional DW ALL business logic is applied trough ETL tools!

# DV2 Model

*Business Keys*

A Business Key identifies a key concept in business.

They have a business meaning!

They are unique and have very low propensity to change

Business keys change only when the business change!

Examples of business keys are: customer numbers, bar codes, ISBN codes, ISSN codes, E-mail addresses, credit card numbers..

Smart keys are composed of different parts which are given business meaning through position and format.

Business keys and associations are the skeleton of the Data Vault model, which is functional-oriented, not subject-oriented.

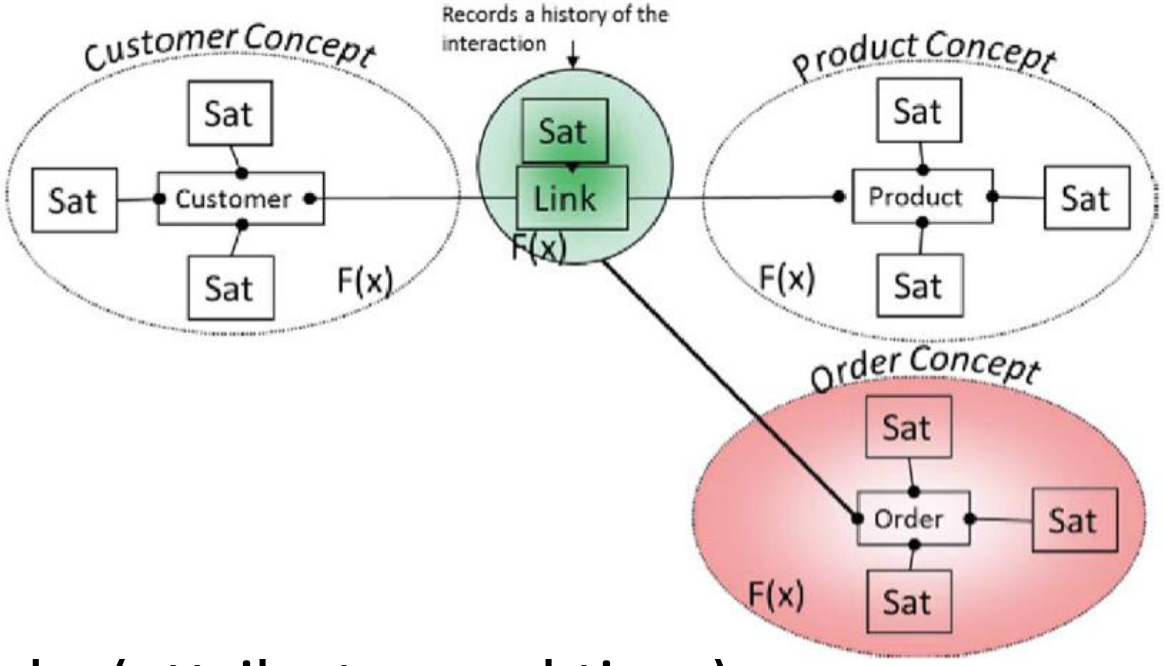
# DV2 Model

*Characteristics and Components*

DV2 Model basic entities:

- Hubs: main business concepts, represented by business keys.
- Links: relationships between hubs, thus between business keys.
- Satellites: context of hubs and links (attributes and time).

  Real data warehousing components: nonvolatile data are stored over time.

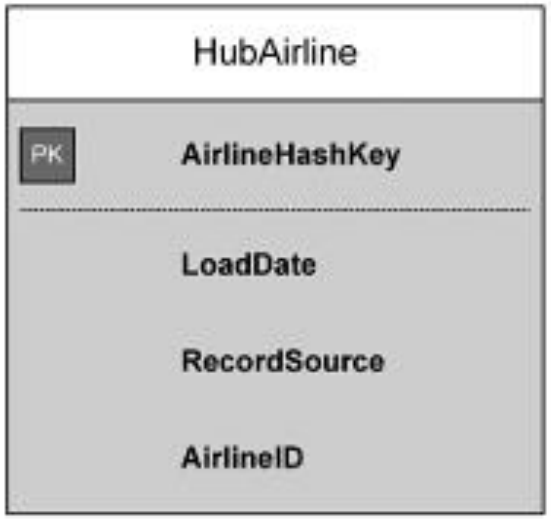


# DV2 Model

*Hubs*

Each hub represents a business key, which is valuable for the overall system and might be different from the single keys found in the operational sources.

P1: Business keys are separated by grain and semantic meaning.




**Hash Key**: generated surrogate key to ease lookup.

**Metadata: Load Date** indicates when the business key first arrived in the EDW, **RecordSource** keeps track of the source.

**Business Key**: one ore more keys, identifying the object.

# DV2 Model

*Links*

Links model transactions, associations, hierarchies and redefinitions of business terms.

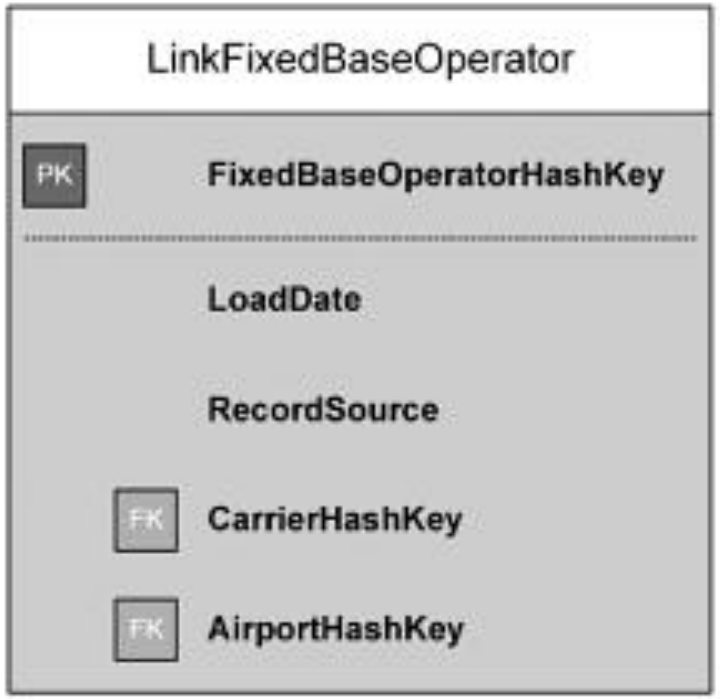
Links capture past, present and future relations among hubs.

P2: intersections across two or more business keys are placed into link structures.

P3: links have no begin or end dates. They are the expression of the relationship at the time the data arrived in the EDW

# DV2 Model

*Links: structure*




**Hash Key**: generated surrogate key to ease lookup.

**Metadata**

**Hash Keys of the hubs connected by the link.**

# DV2 Model

*Links*

Links are many-to-many relationships among two or more hubs.

They absorb data changes.

Flexibility: change in business rules does not require link reengineering.

Example:

Business rule: “one carrier handle more airports, but one airport must be handled but one and only one carrier” → weak entity model

Let’s say, a few years later, any airport can be handled by more than one carrier..
This would require the redesign the existing structures!

The granularity of links is defined by the number of connected hubs.

# DV2 Model

*Satellites*

Satellites store all data that describes a business object, relationship or transaction. They add CONTEXT at a given time over a given hub/link.

P4: Satellites are separated by type of data and classification and rate of change.

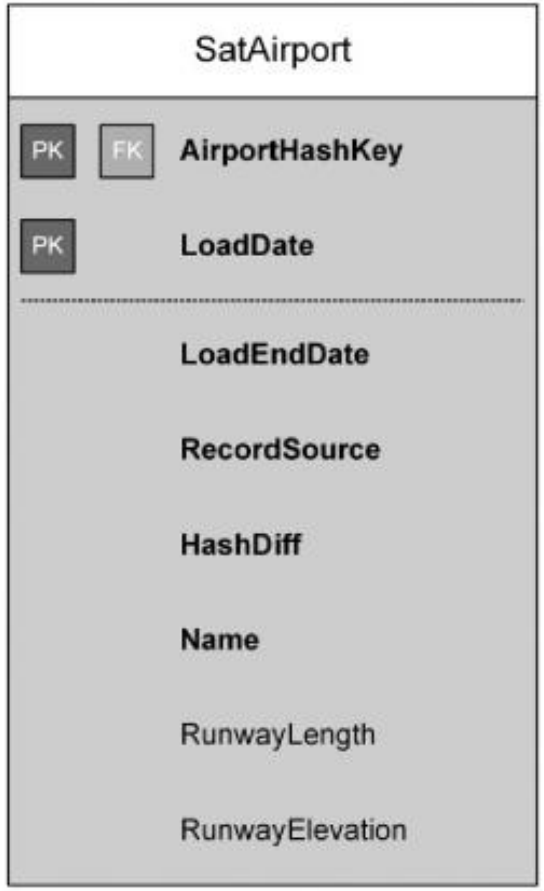
Each satellite is attached to only one hub or link.

A satellite is identified by the parent's hash key and the timestamp of the change. (Remind traditional DW historical data!)

In addition, attributes that describe the context of the business object are contained. Satellites track change!

# DV2 Model

*Satellites: structure*




- **Hash Key** of the parent hub/link
- **Timestamp** of the satellite
- **Timestamp** that determines the end of the SAT's validity.
- **Record Source** keeps track of the source.
- **Hash difference:** hash value of all the descriptive data in a satellite.

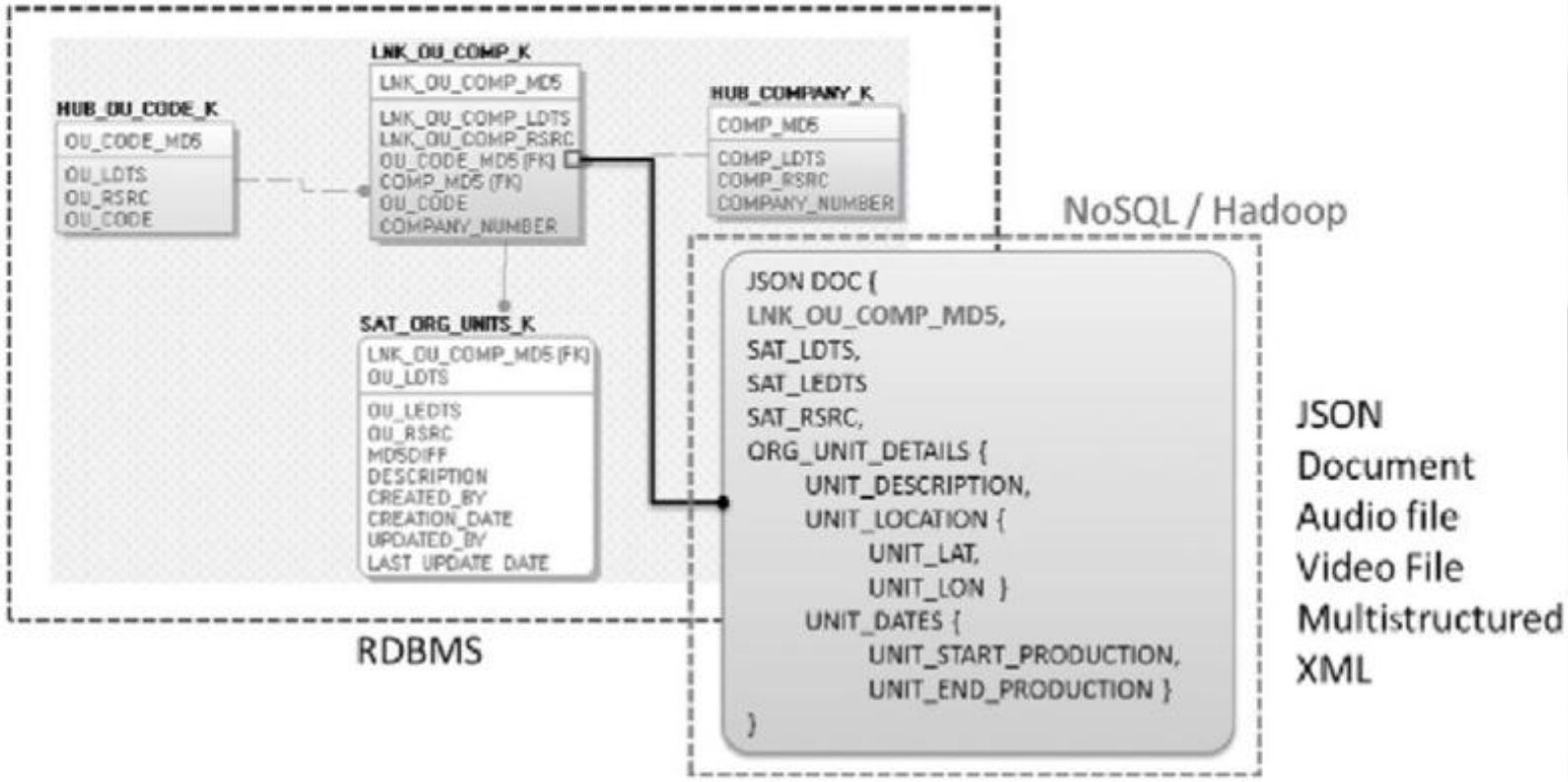
**Name and attributes.**

# DV2 Model

*Heterogeneous satellites*

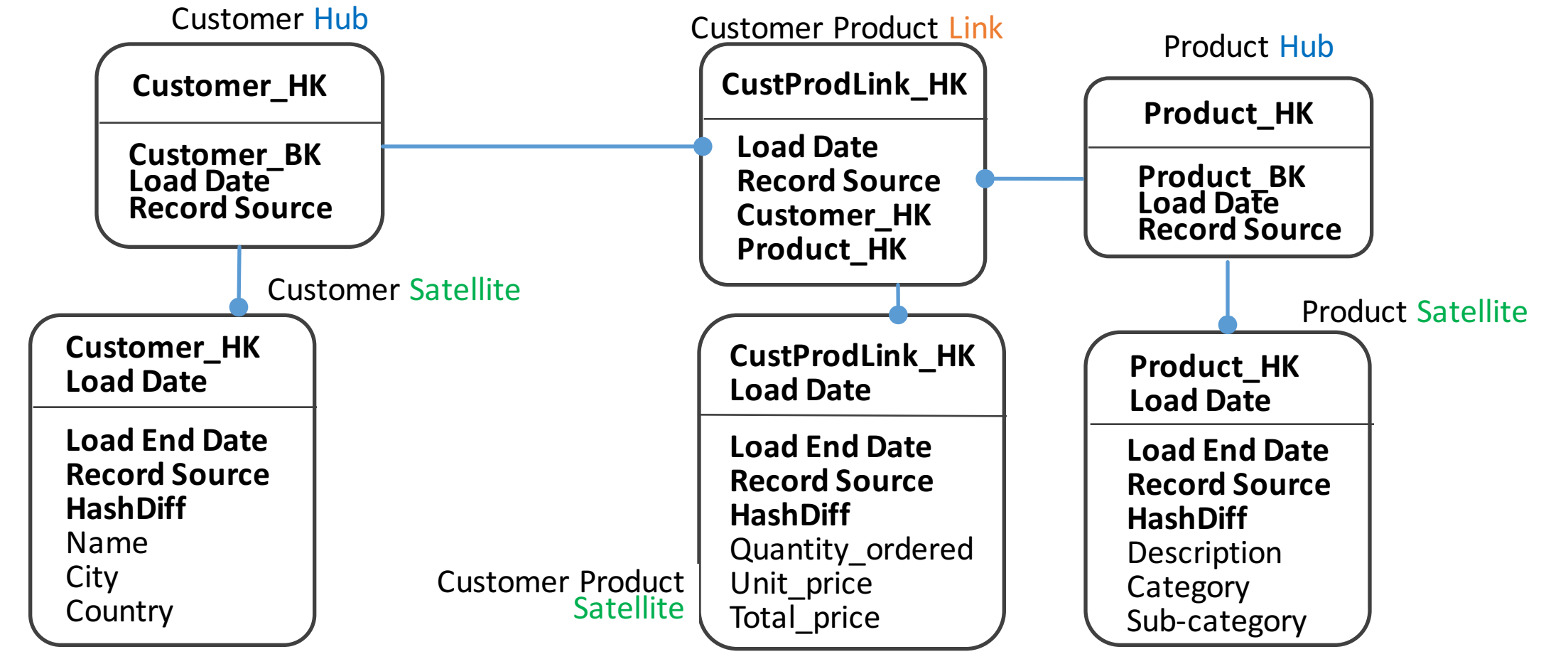
Example of a logical foreign key between RDBMS and Hadoop-stored satellite.

Hash keys allow cross-system joins to occur between RDBMS and NoSQL/Hadoop platforms.



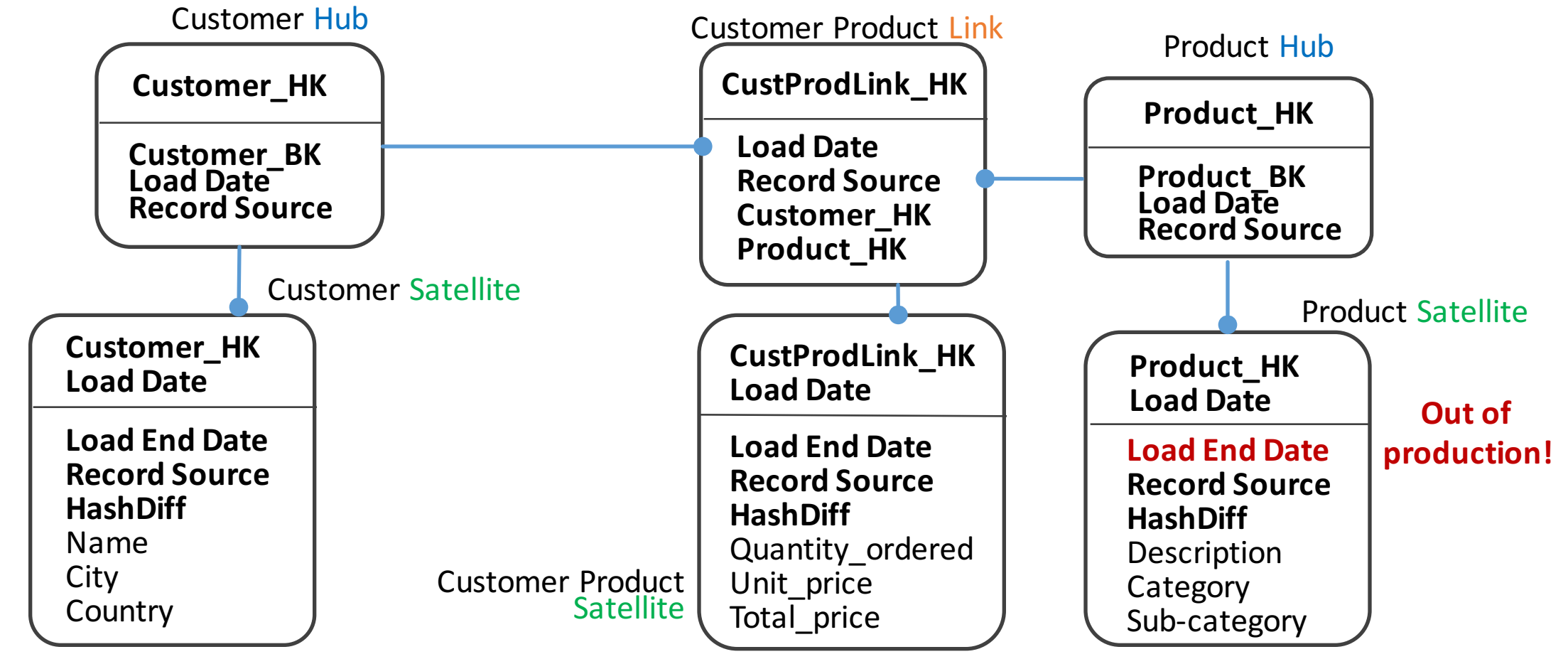
# DV2 Model

## Model example: Customer/Product



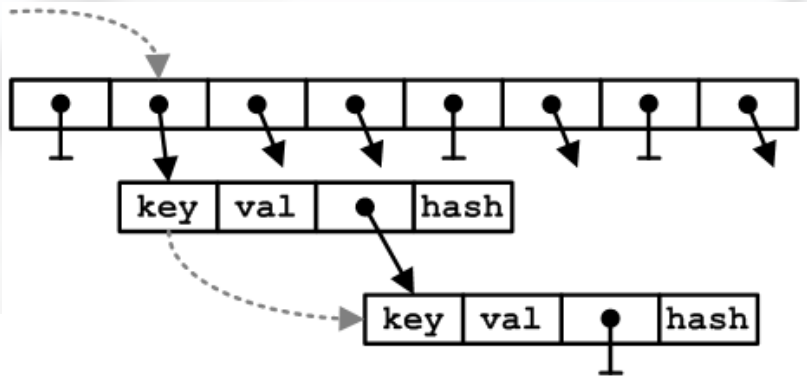
# DV2 Model

## Model example: Customer/Product



# Data Vault 2.0 (DV2)

*Modeling objectives*




- Data integration is based on business keys.
  - Business keys are the keys to the information stored across multiple systems used to locate and uniquely identify records or data.
- Data sets are traceable across multiple lines of business.
- Modeling can support unstructured and structured data:
  - Hash keys allow the connection between heterogeneous data environments, such as Hadoop and RDBMS and remove the dependency on “loading”.

Parallelization of loads: removes dependencies in loading streams.

Example: loading data into Hadoop, perhaps a JSON document, requires looking up the sequence number from a hub in a relational database.

# Conclusions

*When to use a DV2 model*

- The DV2 Model allows split/merge of business keys and data entities:
  - Parallel/distributed system, geographical reasons, security reasons.
  - It is designed to be resilient to environmental changes.
- Seamlessly integrates Big Data technologies with existing RDBMS technologies:
  - Hadoop, MongoDB and many other NoSQL options are easily added.
  - Data cleaning required by a star-schema becomes unnecessary: all data is relevant.

Hadoop and RDBMS are side by side in Big Data Warehouses.

# Big Data Warehouses vs traditional DW

*A summarizing view over the two approaches*

| Design Principle | Traditional Data Warehouse | Big Data Warehouse |
|---|---|---|
| Business Expectations | •Fact based;<br>•Pre-designed for specific reporting requirements;<br>•single source of the business truth; | •Exploratory analysis;<br>•Finding of new insights<br>•Veracity of results might be questionable |
| Design Methodology | •Iterative and waterfall<br>•Integrated and consistent model | •Agile and iterative approach<br>•No data model definition |
| Data Architecture | •Not all data is managed and maintained in the EDW: the data sources are previously known;<br>•Anything new has to go through a rigorous requirements gathering and validation process;<br>•Scales but at a potentially higher cost per byte; | •Integrates all possible data structures;<br>•Scales at relatively low cost;<br>•Analyzes massive volumes of data without resorting to sampling mechanisms. |
| Data Integrity and Standards | •Driven by RDBMS and ETL tools.<br>•Centralized data | •Integration is loosely defined;<br>•Data and data processing programs are highly distributed. |

# Thank you

Thank you
for your attention!

Q & A?

# Quick References

**Books:**

- **DATA ARCHITECTURE: A PRIMER FOR THE DATA SCIENTIST Big Data, Data Warehouse and Data Vault -** W.H. Inmon, Daniel Linstedt
- **Big Data Imperatives Enterprise Big Data Warehouse, BI Implementations and Analytics** – S. Mohanty, M. Jagadeesh and H.Srivatsa
- **Building a Scalable Data Warehouse with Data Vault 2.0 -** Dan Linstedt, Michael Olschimke
- **Advanced Data Warehouse Design - From Conventional to Spatial and Temporal Applications -** E. Malinowski and E. Zimányi

**Other sources:**

- **Hadoop and the Data Warehouse: When to Use Which -** Dr. Amr Awadallah, *Founder and CTO, Cloudera*, Dan Graham, *General Manager, Enterprise Systems, Teradata Corporation*
- **Big Data in Big Companies** - Thomas H. Davenport, Jill Dyché

**Data Vault Support: QUIPU - http://www.datawarehousemanagement.org**