# Record Linkage in Data Warehousing

**Alfredo Cuzzocrea**
*ICAR-CNR and University of Calabria, Italy*

**Laura Puglisi**
*GESP Geographic Information Systems, Italy*

## INTRODUCTION

"A Data Warehouse is a subject-oriented, integrated, time-variant, and non-volatile collection of data supporting decision-making processes" (Inmon, 2002). At a more practical level, a data warehouse is a repository of information collected from *multiple sources*, stored under a *unified schema*, and that usually resides at a single site (i.e., the Data Warehousing server). Looking into inside, Data Warehouses are characterized by different processes: *data cleaning*, *data integration*, *data transformation*, *data loading*, and *periodic data refreshing*. All these convey in the so-called ETL (*Extraction-Transformation-Loading*) main process (Inmon, 2002), which, essentially, aliments the Data Warehouse.

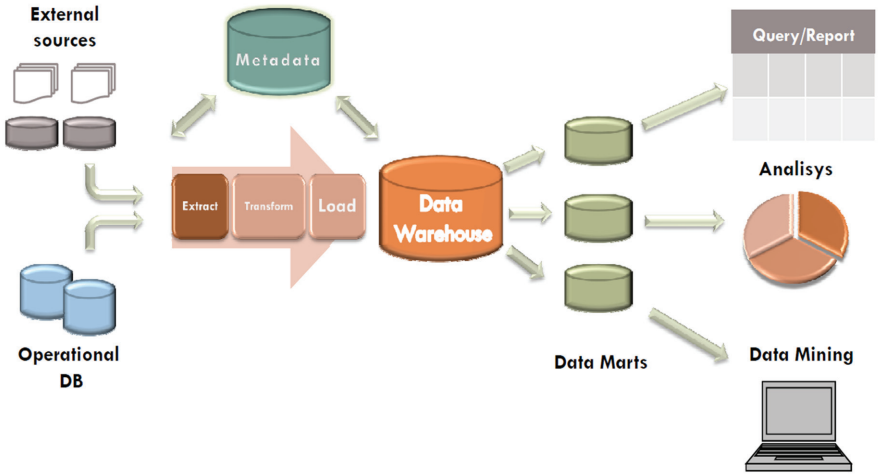
In order to support decision-making processes, data in a Data Warehouse are organized around so-called *subjects*, such as *Customer*, *Item*, and *Activity*, and so forth. Also, data are stored in such a way as to provide information from a *historical perspective* (e.g., since the past 5-10 years) and are typically *summarized* according to a given level of *granularity*. Consider, for instance, the case of a Data Warehouse storing sale data. Here, rather than storing the details of each sale transaction, the Data Warehouse may rather store a summary of transactions per-item-type for each store or, at a higher level, (summarized) for each sale region. In addition to this, Data Warehousing platforms provide *On-Line Analytical Processing* (OLAP) (Gray et al., 1997) tools for supporting interactive data analysis according to a *multidimensional and multi-resolution vision*. Also, many other *Data Mining functionalities* (Fayyad et al., 1996; Frawley et al., 1992), such as *Association Rule Discovery*, *Classification*, *Prediction* and *Clustering*, can be integrated with the OLAP layer in order to enhance interactive (summarized) knowledge discovery and mining at multiple levels of abstraction. Figure 1 shows the reference architecture on a Data Warehousing platform (Inmon, 2002).

According to Inmon (2002), the major distinctive features of a Data Warehouse are the following: (*i*) *subject-orientation*, which refers to the amenity according to which a Data Warehouse focuses on subjects of analysis, and features or data that are not useful to the target decision-making process are excluded from the analysis; (*ii*) *integration*, which refers to the amenity according to which input data for a Data Warehouse come from multiple and heterogeneous sources, such as relational databases, flat files etc. – as a consequence, in order to remove possible *inconsistency and duplicated information*, data cleaning and data transformation processes are exploited to this end; (*iii*) *time-variance*, which refers to the amenity according to which input data in a Data Warehouse have a marked temporal perspective and multi-versioning (e.g., across the past 5-10 years); (*iv*) *non-volatility*, which refers to the amenity according to which, in a Data Warehouse, (summarized) analytical data are maintained separated from (alimenting) transactional data – due to this clear separation, a Data Warehouse server does not require transaction processing and recovery, and concurrency control mechanisms (like conventional DBMS servers) but, rather, it only requires three main operations: (initial) data loading, data refreshing, data accessing.

A major research challenging in Data Warehousing research conveys under the term *data quality*, beyond the classical one focusing on *Data Warehouse query optimization* (e.g., (Cuzzocrea et al., 2004, 2005, 2005; Cuzzocrea & Serafino, 2009). Indeed, quality of data stored in a Data Warehouse can have significant effectiveness and cost implications on a system that *only* relies on information to make decisions and predictions on business organizations and activities. On the other hand, integrating data from multiple and

DOI: 10.4018/978-1-4666-5888-2.ch189

*Figure 1. Reference architecture of a data warehousing platform*




heterogeneous sources into a coherent data store is a complex and time-consuming task, since systematic differences or conflicts can occur easily (Elmagarmid et al., 2007). For instance, misspellings and different conventions for recording the *same* information can result in *different, multiple representations of a unique* (*data*) *object*. This problem, i.e. identifying objects in input data sources that refer to the *same real-world entity*, is known in literature as *duplicate detection* or *record linkage* (Costa et al., 2011). Intuitively enough, it is worthy to figure-out that, in a multiple-data-source application scenario, objects may be duplicated, even though not identical, due to errors and missing data (Lehti & Fankhauser, 2006), like, for instance, in a *Web Warehouse* (e.g., (Bonifati & Cuzzocrea, 2007)).

Within the above-illustrated research context, this article provides an analysis of the state-of-the-art techniques for supporting record linkage in Data Warehousing, with critical discussion, and draws novel directions for possible research perspectives that may stimulate future challenges in the investigated research field.

## BACKGROUND

Record linkage (Fellegi & Sunter, 1969) has given rise to a large body of works in several research communities, where it is referred to with as many umbrella names, such as e.g., *de-duplication* (Sarawagi & Bhamidipaty, 2002) and *object identification* (Neiling & Jurk, 2003). Indeed, the typical scenario in the design of information systems is the availability of multiple data repositories, with different schemas and assumptions on the underlying canonical data representation. Schema differences imply a *segmentation* of data tuples into sequences of strings, which correspond to specific *semantic entities*. However, such a segmentation is not known in advance and this plays the role of a challenging issue for record linkage problems. Moreover, the adoption of various canonical data representations (such as the presence of distinct data separators and/or various forms of abbreviations) coupled with erroneous data entry, misspelled strings, transposition oversights and inconsistent data collection further exacerbates the foresaid difficulties behind the recognition of duplicates in data.

More specifically, as mentioned in Sect. 1, the goal of record linkage is to identify objects that refer to the same real-world entity. The record linkage problem has originated a rich season of results, mainly in the context of *Database Integration research*. State-of-the-art techniques usually aim at solving two types of *data heterogeneity*, namely *structural* and *lexical heterogeneity*. Structural heterogeneity occurs when fields of target tuples in the reference (different) databases have different structures. For instance, in a database, customer addresses may be stored by means of just one field (e.g., *Address*), whereas, in another database, the same information may be recorded by means of multiple fields (e.g., $\langle$*Street*, *City*, *State*, *Zipcode*$\rangle$). Lexical heterogeneity occurs when target tuples have the same structure across (different) databases, but different representations are exploited to refer to the