

# Chapter X

## Visual Query Languages, Representation Techniques, and Data Models

**Maria Chiara Caschera**  
*IRPPS-CNR, Italy*

**Arianna D’Ulia**  
*IRPPS-CNR, Italy*

**Leonardo Tininini**  
*IASI-CNR, Italy*

### ABSTRACT

An easy, efficient, and effective way to retrieve stored data is obviously one of the key issues of any information system. In the last few years, considerable effort has been devoted to the definition of more intuitive, visual-based querying paradigms, attempting to offer a good trade-off between expressiveness and intuitiveness. In this chapter, we analyze the main characteristics of visual languages specifically designed for querying information systems, concentrating on conventional relational databases, but also considering information systems with a less rigid structure such as Web resources storing XML documents.

We consider two fundamental aspects of visual query languages: the adopted visual representation technique and the underlying data model, possibly specialized to specific application contexts.

### INTRODUCTION

The retrieval of data from an information system requires specifically designed languages, enabling the user to extract the desired information by specifying properties on the structure and/or on the content of the information itself. These languages, commonly known as *query languages*, have been studied in detail in the literature and are one of the

most important topics of database theory. Most query languages adopted in commercial database management systems (DBMSs) are of the *textual* type (i.e., the request consists of a sequence of words and textual symbols combined according to a well-defined syntax).

Although traditional textual query languages have been shown to enable the user to express complex requests and several techniques have been proposed for the efficient evaluation of the corresponding answers, their ease of use is very limited, particularly for non-expert users. In recent years, there has been a drive to define more intuitive, visual-based querying paradigms, attempting to offer a good trade-off between expressiveness and intuitiveness. In *visual query languages (VQLs)*, words and textual symbols are replaced by graphical objects (e.g., arcs, geometric shapes, tables, icons), the spatial relationships among them (e.g., inclusion, connection, intersection), and some visual properties (e.g., the colour and line type: solid, thick, dotted, etc.).

In this chapter, we analyze the main characteristics of VQLs, mainly concentrating on languages to query conventional relational databases, but also examining information systems with a less rigid structure such as Web resources storing XML documents. We consider two main classifications of VQLs: the adopted visual representation technique (e.g., based on forms and tables, diagrams, icons, sketches, or on some combinations thereof) and the underlying data model (e.g., visual languages to query relational databases, object-oriented databases, collections of XML documents, as well as languages specifically designed for particular data such as geographical and multidimensional data).

## BACKGROUND

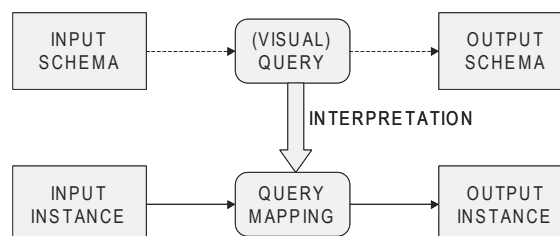
In this section, we review some basic terms and concepts of database theory, especially those related to the area of query languages (see, for

example, Abiteboul, Hull, and Vianu (1995) for a more comprehensive introduction). Given a database  $D$ , we distinguish the (database) *schema*  $S$  and the (database) *instance*  $I$ , representing the structure and the actual contents of the data stored in  $D$ . For example, in a relational database the schema is constituted by the relation (table) names, along with the corresponding attribute names (and possibly types), while the instance is constituted by the sets of tuples (records) having the structures specified by the single schemas.

A *query* is a syntactic object, typically constituted by a text string, a graph, a combination of shapes and icons, etc., constructed using elements of the schema  $S$  (the *input schema*), some specific symbols and according to the rules (the syntax) of a language. The query also describes the *output schema* (i.e., the structure of the data that will be produced by the query's interpretation). In visual queries, the elements of the input and output schema, the operations to combine them, and the constraints that must be satisfied by the required data are represented by visual metaphors, organized according to a visual syntax.

The interpretation of a query according to a predefined semantics determines a *query mapping* (see Figure 1) i.e., a function from the set of possible input instances (on the input schema) of a given database to the set of possible output instances (on the output schema). Given a certain input instance, the construction of the corresponding output instance is commonly known as *query*

Figure 1. Query and query mapping



14 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage: [www.igi-global.com/chapter/visual-query-languages-representation-techniques/28554](http://www.igi-global.com/chapter/visual-query-languages-representation-techniques/28554)

## Related Content

---

### Thirty Years Later: Some Reflections on Ontological Analysis in Conceptual Modeling

Yair Wandand Ron Weber (2017). *Journal of Database Management* (pp. 1-17).

[www.irma-international.org/article/thirty-years-later/181666](http://www.irma-international.org/article/thirty-years-later/181666)

### Sequential Pattern Mining from Sequential Data

Shigeaki Sakurai (2009). *Handbook of Research on Innovations in Database Technologies and Applications: Current and Future Trends* (pp. 622-631).

[www.irma-international.org/chapter/sequential-pattern-mining-sequential-data/20748](http://www.irma-international.org/chapter/sequential-pattern-mining-sequential-data/20748)

### A Requirements Engineering Framework for Software Startup Companies

Sudhaman Parthasarathyand Maya Daneva (2021). *Journal of Database Management* (pp. 69-94).

[www.irma-international.org/article/a-requirements-engineering-framework-for-software-startup-companies/282445](http://www.irma-international.org/article/a-requirements-engineering-framework-for-software-startup-companies/282445)

### Collaboration Matrix Factorization on Rate and Review for Recommendation

Zhicheng Wu, Huafeng Liu, Yanyan Xuand Liping Jing (2019). *Journal of Database Management* (pp. 27-43).

[www.irma-international.org/article/collaboration-matrix-factorization-on-rate-and-review-for-recommendation/232720](http://www.irma-international.org/article/collaboration-matrix-factorization-on-rate-and-review-for-recommendation/232720)

### Developing a Homegrown Course Management System: Community/Course Action/Interaction Management System (CAMS)

Brian G. Mackie, Norbert L. Ziemer, Nancy L. Russoand Wayne E. Mackie (2006). *Cases on Database Technologies and Applications* (pp. 154-170).

[www.irma-international.org/chapter/developing-homegrown-course-management-system/6210](http://www.irma-international.org/chapter/developing-homegrown-course-management-system/6210)