

Visual SQL

An ER-Based Database Querying

Bernhard Thalheim

Computer Science Institute, Christian-Albrechts-University Kiel,
Olshausenstrasse 40, 24098 Kiel, Germany
thalheim@is.informatik.uni-kiel.de

download:
<http://www.informatik.uni-kiel.de/en/is/miscellaneous/visualsql/>

Abstract

Query formulation is still a difficult task whenever a database schema is large or complex. The user has to entirely understand the schema before a correct and complete formulation of the query will be found. Furthermore, users may overlook types in the SQL schema that must be used in the query. We show in this demo that visualization led in this case to higher conceptual correctness and conceptual completeness. Visualization is based on Visual SQL. Visual SQL follows the paradigm of entity-relationship representation. At the same time, it has the same expression power as SQL-92. The quality of query formulation is, however, higher. Visual SQL is an ER interface to database systems. Queries can be formulated in Visual SQL and then translated to SQL-92. Queries in SQL-92 can be represented by Visual SQL queries after translating them to Visual SQL based on the a given database schema.

Database querying and programming based on Visual SQL

Database programming usually tough in university courses is mainly based on simple examples. Database programming in practice is far more complex. Applications lead often to queries that cannot be placed on a full sheet of paper, that range over more than a dozen of relations, and that have a complex join structure. Visualization is one of the features that may substantially improve correctness and completeness of query formulation.

Visual SQL is at the same time

- as powerful as SQL-92 and SQL-99 and

- simpler to use and to comprehend, and
- less error-prone in complex settings.

The demo shows what we would gain after realization of Chen's dream on ER database systems and to demonstrate the power of visual programming already for existing database technology. The demo demonstrates how to master real difficult SQL queries based on ER techniques.

Characterization of database system quality is based on a number of parameters such as functionality (suitability, accuracy, interoperability, security), reliability (maturity, fault tolerance, recoverability), efficiency (time behavior, resource utilization), maintainability (analyzability, changeability, stability, testability) portability (adaptivity, installability, co-existence, replaceability), and last but not least usability (understandability, learnability, operability, attractiveness).

Visual SQL supports the last quality parameter, *usability*. Users must be able to request a large database system without complete understanding of a large database schema, without understanding of the impact of specific values such as null values and without deep consideration of integrity constraints. The quality of SQL query interface can be evaluated by referring to **correctness** of query formulation, **completeness** of query expression depending on the underlying schema, its suitability for use, its adherence to a predetermined set of expectations, or its freedom from mistakes or flaws.

A demonstration example

Let us compare the facilities of Visual SQL with SQL-92 based on the database schema given in Figure Appendix B.

The SQL-92 query

Let us consider the following query:

Provide data on students who have successfully completed those and only those courses which have successfully been given or which are currently given by the student's supervisor?

This query is given by the following SQL query:

```
SELECT  P1.Name, P1.BirthDate, P1. Address,
        P2.Name AS "Name of supervisor"
FROM    Person P1, Professor P2, Student S1,  Supervisor, Lecture L,
        Enroll E
WHERE   P1.Name = Student.Name AND P1.BirthDate = Student.BirthDate
        AND S1.StudNo = E.StudNo
        AND E.Result NOT NULL
        AND S1.StudNo = Supervisor.StudNo
        AND Supervisor.Name = Professor.Name
```

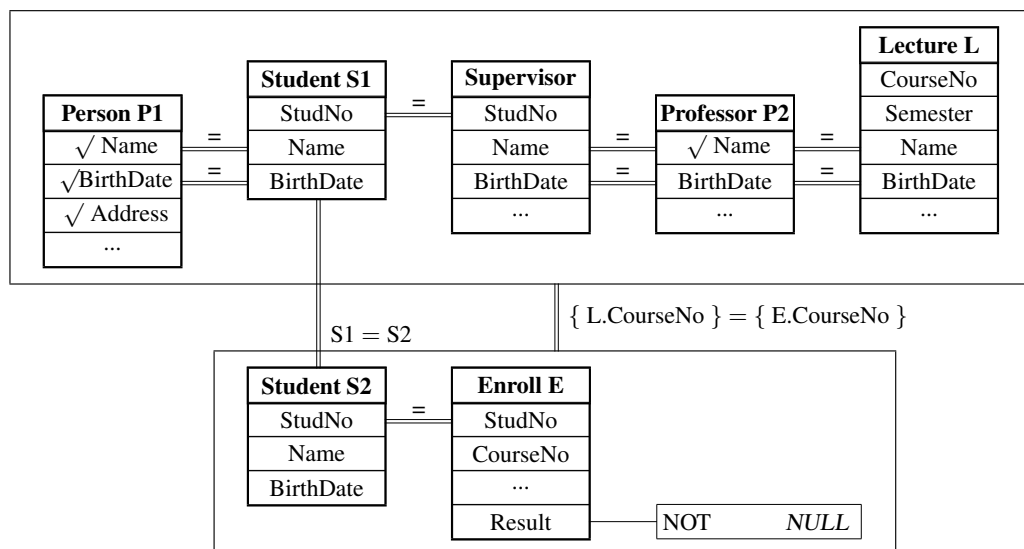
```

AND Supervisor.BirthDate = Professor.BirthDate
AND P2.Name = Professor.Name AND P2.BirthDate = P2.BirthDate
AND L.Name = Professor.Name AND L.BirthDate = Professor.BirthDate
AND
  L.CourseNo
  IN      (SELECT E2.CourseNo
           FROM  Enroll E2
           WHERE  S1.StudNo = E2.StudNo AND
                 E2.Result NOT NULL )
AND
  E.CourseNo
  IN      (SELECT  L2.CourseNo
           FROM    Lecture L2
           WHERE
             L2.Name = P2.Name AND
             L2.BirthDate = P2.BirthDate );

```

The Visual SQL query

The Visual SQL query is far simpler and depicted in the following picture.



Quality of querying depends on visualization

The visualization of complex conceptual structures is a key component of support for many applications in science and engineering. An ER schema is an abstract structure that is used to model information. ER schemata are used to represent information that can be modeled as objects and connections between those objects.

Graphical structures have been investigated by mathematicians for centuries, for visualization and intuition, as well as for the pure beauty of the interplay between graph theory and geometry.

Visualization of structures is only useful to the degree that the associated diagrams effectively convey information to the people that use them ¹. A good diagram helps the reader to understand the system, but a poor diagram can be confusing and misleading. Diagrams should be of practical value. If diagrams are only used for graphical visualization and later not used they become only illustrations. If we can, however, translate the diagram to such languages as SQL we have a graphical languages of high utility.

Visualization of database structuring offers a number of advantages for querying purposes:

- Visualization enables in surveying and understanding the query and the associations among relations that are used in one query.
- Visualization enables in direct manipulation applying changing restrictions or viewpoints.
- Visualization is allows to coordinate multiple viewpoints. It supports to understand complex and multivariate data and the exploration of complex databases.

Features of Visual SQL

1. Visual SQL as database query description language
2. Retranslation of SQL-92 queries to Visual SQL based on database schemata given in DBMain
3. Visual SQL and intelligent diagramming of queries
4. Visual SQL and view specification
5. Visual SQL and diagramming of integrity maintenance
6. Translation profile of Visual SQL specifications to SQL'92, SQL'99 and SQL'2003
7. Visual SQL and SQL tuning
8. Formal foundations of Visual SQL

¹The advertisement for Doan's Backache Kidney Pearls used the slogan: *Every picture tells a story.*

Appendix A: Additional information

Applications

Applications for teaching.

The system is currently used in database courses at different universities in Germany and New Zealand. It is also used for introductory courses to database technology in high schools in Lower Saxonia and Schleswig-Holstein in Germany.

Tests made with students in the years 2003, 2005, 2006 and 2007 have shown that completeness and correctness of SQL queries is significantly higher if Visual SQL queries are formulated.

Applications in projects.

The Visual SQL system has used in about one dozen of database refactoring and tuning projects for visualisation of queries. The largest SQL-92 query that has been redesigned consisted of about 250 lines. After visualisation the cleverness and the pitfalls of this query could be analysed.

Previous demonstrations

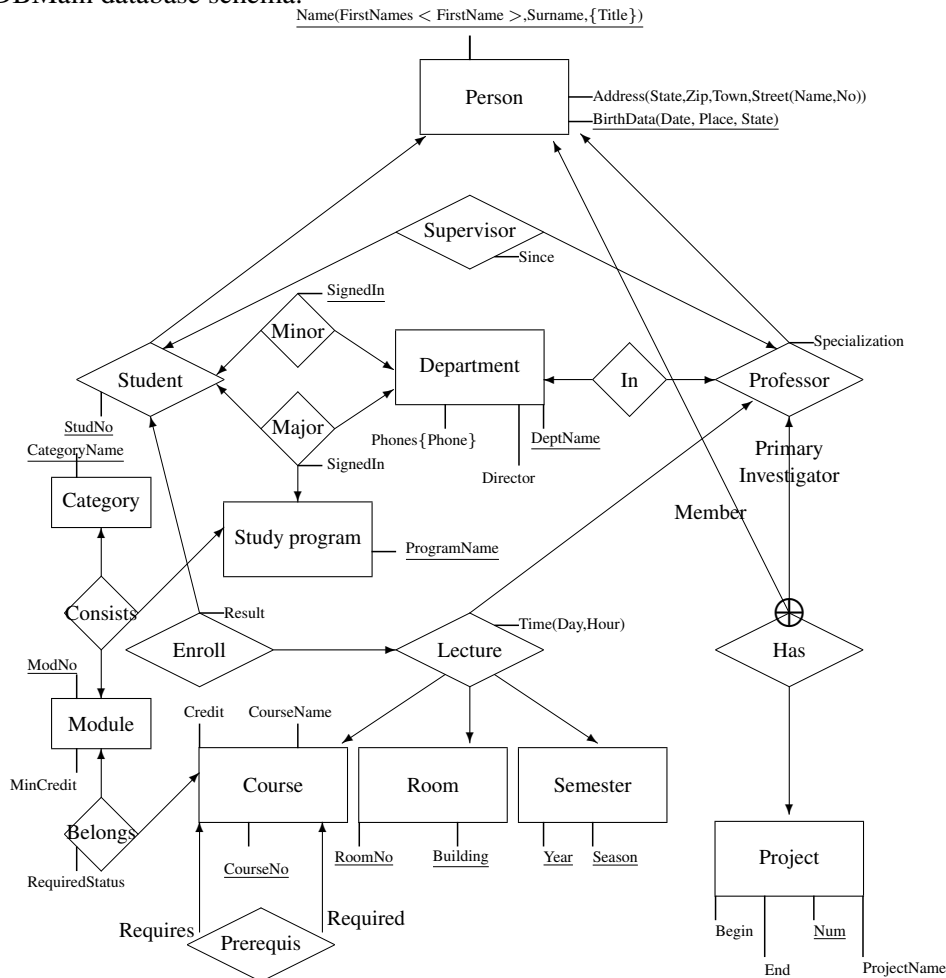
The version 1.5 of the system has been exhibited at CeBIT 2006.

The version 1.6 is available from teaching path of the website

<http://www.informatik.uni-kiel.de/~fiedler/>.

Appendix B: The EER schema of the example

The following EER² is used for demonstration. This schema is transformed to a DBMain database schema.



²We use the extended entity relationship model. It generalizes the classical entity-relationship model by adding constructs for richer structures such as complex nested attributes, relationship types of higher-order i which may have relationship types of order $i - 1, i - 2, \dots, 1$ as their components, and cluster types that allow disjoint union of types (displayed by \oplus). Further, HERM extends the ER model by an algebra of operations, by rich sets of integrity constraints, by transactions, by workflows, by views and by techniques for translating or compiling HERM schemes to relational and object-relational specification. For introduction: <http://www.informatik.uni-kiel.de/~thalheim/slides.htm>