

# The Audacity of Hope: Thoughts on Reclaiming the Database Dream\*

Sam Lindley and Philip Wadler

University of Edinburgh

**Abstract.** A venerable line of research aims to provide a general-purpose programming language with a well-defined subset that compiles into efficient queries, perhaps by translation into SQL or some other suitable query language. This talk discusses some older and more recent advances in this direction, including the languages Kleisli, LINQ, Ferry, M, and Links.

Twenty-five years ago, Copeland and Maier decried the “impedance mismatch” between databases and programming languages, and Atkinson and Buneman observed “Databases and programming languages have developed almost independently of one another for the past twenty years”, a situation that has not greatly changed in the intervening time, and spoke of “The need for a *uniform* language” (their emphasis).

The problem is simple: two languages are more than twice as difficult to use as one language. Most programming languages support data abstraction and nested data, while most relational databases support tables over a few fixed scalar types. Any task involving both requires that the programmer keep in mind two representations of the same underlying data, converting between them and synchronizing updates to either. This persistent bookkeeping adds to the mental burden on the programmer and adds to the complexity of the code.

The solution is equally simple: provide a single language with a well-defined subset that compiles into efficient queries, perhaps by translation into SQL or some other suitable query language. It is important that the subset support the abstraction mechanisms of the language, for example allowing one to lambda abstract over a predicate used in a query, where instantiating the abstraction with a specific predicate should result in an efficient query. In the simplest case, only flat data (such as bags of records) can be translated into the database. In a more sophisticated system, nested data (such as records of bags) can be represented in the database via a suitable encoding, the details of which need not concern the programmer. While easy to envision, practical implementation of such languages has proved elusive in practice.

This talk discusses some older and more recent advances in this direction, including the languages Kleisli, LINQ, Ferry, M, and Links.

---

\* With apologies to Barack Obama and David August.