

---

## The Associative Model of Data

---

*The Associative Model of Data is the name given by Simon Williams, Lazysoft's founder, to the database architecture that he has described in his book of the same name, on which LazySoft's database management system Sentences™ is based.*

### The Problem

---

Database applications are costly to develop and maintain. Most companies could benefit from custom-built systems that support their unique value propositions, but few can afford them: most must settle for one-size-fits-all application packages which have to be expensively modified to meet their objectives.

Moreover, managers who must rely on such applications cannot directly influence their behaviour: when changes are needed, they are dependent on programmers whose priorities may differ.

### The Cause

---

The biggest driver of cost and complexity in database applications is the Relational Model of data, the standard database architecture underlying almost all modern applications.

Relational databases store each different type of data in a separate table. Each table is unique, and the programs are hard-coded to fit the tables.

This is labour-intensive and wasteful, for two reasons: new applications always need new programs which have to be written from scratch, and any change to the structure of the data means that programs must also be changed, or new programs created.

### Why a new Model?

---

The quest for greater productivity in application development has focused on using more abstraction in programming languages, which have evolved from binary machine code through second generation symbolic languages and third generation high-level languages.

However, 4GLs have failed to win widespread acceptance, and today's object-oriented programming languages are in truth no more abstract with respect to the real world than their third-generation predecessors. Moreover, the commercial failure of object-oriented databases means that most modern commercial

applications comprise programs designed and developed using object oriented techniques, but implemented over relational databases, an impedance mis-match that imposes further overheads on the development process.

Ted Codd acknowledged the connection between data architecture and programming productivity when in 1970 he introduced the relational model to the world with the words "*Future users of large data banks must be protected from having to know how the data is organized in the machine*" – Dr E F Codd, "A Relational Model of Data for Large Shared Data Banks", CACM, 1970.

**The next major breakthrough in software development productivity will come not from advances in programming languages, but from a new database architecture.**

### The Solution

---

Instead of using a separate, unique table for every different type of data, the Associative Model of Data uses a single, generic structure to contain all types of data. Information about the logical structure of the data and the rules that govern it are stored alongside the data in the database.

This sets the scene for a new way of programming, whereby the data structures and the rules that govern them are no longer hard-coded into the programs, but are obtained by the programs from the database itself. Thus, for the first time, such programs are truly reusable, and no longer need to be amended when the data structures change. This dramatically reduces the cost of application development and maintenance.

**The aim of the Relational Model of Data was to free programmers from having to know the physical structure of data.**

**The aim of the Associative Model of Data is to free them in addition from having to know its logical structure.**

---

*"The Associative Model of Data" by Simon Williams, published by Lazysoft, ISBN 1-903453-01-1.*