A brief history of Data
Hugh Darwen, 21 April 2009 (lecture originally prepared for 2007 Kingston BCS conference)

Introduction
This is a personal view of the Past, Present and Future of the relational model. In Hugh’s view, there still isn’t any such thing as a relational database management system!

Past: a personal view and memoir
1960: Data was on punched cards or (wow) magnetic tape. Cards tended to hold the latest transactions, while the tape contained the “database” (there was no such word then). Very highly developed merge-sort algorithms and others were developed, very difficult to understand and of course with obscure loop-related bugs.

1965: Disks arrived, leading to “direct access”. So more bugs arrived with the introduction of pointers!

1970: E.F. Codd (IBM) had the vision of a “Relational Model of Data for Large Shared Data Banks” based on First-Order Predicate Calculus (FOPC). Codd was one of three chief architects of the IBM 360, along with Gene Amdahl and Gerrit Blauw. Relations did away with both loops and pointers, which seemed like a great idea. Codd rightly became very famous.

However, there were a lot of holes in the work produced in 1970, and Codd had little interest in programming language design. Chris Date was teaching PL/1 and Assembler for IBM in the UK and started to spread the word about the relational model. He had some contacts in Hursley and tried (unsuccessfully, alas) to motivate people there to make PL/1 support the relational model.

1975: Relational prototypes developed in IBM: PRTV (ISBL), QBE (query by example) and System R. P in PRTV stands for Peterlee, near Newcastle – Peterlee Relational Test Vehicle. IBM had a lot of investment in the DL1 language on IMS, so wasn’t keen on anyone else coming up with a more attractive name that implied it had a fully-fledged language (ISBL).

People in the UK were devising a language that worked to fill in the holes in Codd’s work; while the US people (System R) were trying to prove that the underlying engine could be made to perform. The language SQL (sequel) arose from this work as an afterthought, unfortunately, not ISBL, which was quietly forgotten. ISBL was modelled a bit on APL – elegant and powerful. The system R team has told Hugh that Ted Codd wasn’t very helpful and they never meant Sequel to become a standard – it was just a stop-gap.

1980: First SQL products on the market, Oracle and SQL/DS (IBM’s “answer” to Oracle).

1986: First international SQL standard (“Intergalactic Dataspeak” – Mike Stonebraker)

1987: OODBMS started to arrive – clearly a sign of progress was to bring back pointers!

1 “… the OODB "jingle" was "persistence is orthogonal to type", meaning that anything you could do with local variables in your programs written in your OO programming language you could do with the database. Your database consisted of a bunch of persistent "objects" (really, variables in disguise) just like the local variables that disappear when your program exits. And variables were usually occupied by pointers (called object identifiers, or oids).” (Hugh in an e-mail)
1989: *The Object Oriented Database Manifesto* – many people thought that OODBMS were the big thing of the future. Stonebraker and friends disagreed and responded with:

1990: *The Third Generation Database Systems Manifesto* – they thought that object oriented was not the way to manage data; but that object oriented notions could be incorporated within relational database management systems – by which they of course meant SQL databases. Darwen and Date didn’t agree with either of those propositions!

1992: SQL:1992 finally became a “relationally complete” language (but only at the full level of the international standard, which nobody could actually support – DB2 and Oracle only fulfilled entry level and a few products only went up to intermediate level).

1995: *The Third Manifesto* published by Darwen and Date.

1999: SQL:1999 standard added user-defined types, making the standard “object/relational”. This made rather a mess of the language, by doing exactly what the *Third Manifesto* said not to do. There were two options – you could do it the right way or the wrong way.

2003: SQL:2003 was “the XML edition”. This brought back the hierarchies that existed before relational databases.


**Hugh Darwen’s CV**

1960: Hugh left school

1967: BSc in Zoology (failed!) followed by work for an IBM Service Bureau in Birmingham. This involved timesharing between customers. Writing programs with loops in them in COBOL to read jobs from cards, modifying applications to particular customer requirements etc. Users would send in paper documents, which were converted to cards by “punch girls” and fed into the machine by operators.

1969: “Terminal Business System” – this put users (at terminals) in direct contact with their databases and applications via telephone lines. This opened the way for small to medium enterprises to make use of computer systems. Application programmers made lots of loop and pointer errors. Hugh’s team developed a “report writer” that could produce really nice reports provided you needed to access only one file with records all in the same format. Hugh had been tasked with devising some kind of language that could produce ad-hoc reports across diverse files with different record formats. So…

1972: Attended C.J. Date’s course on databases – a personal watershed. At the end of the course, Date mentioned the new relational idea, which revealed to Hugh that his approach to the report writing problem was doomed.

1978: Computers had become much smaller. “Business System 12” was a relational dbms service for the bureau service. Hugh was chief architect for the development of this\(^2\). Discovered the ISBL language, which solved all the problems Hugh hadn’t been able to solve

\(^2\) “BS12 was actually developed in The Netherlands, where I lived for 5 years (I didn't mention that), by a multinational group of developers (mostly Dutch and British). [Gerrit] Blauw, by then a professor at a Dutch university, was a consultant to the development team, so I got to know him very well…” *Hugh in an e-mail*
from Codd’s publications. Also got good advice about the relational engine from the System
R people, but found that they already knew most of it! Hugh made a presentation to the whole
lab in the presence of a senior executive responsible for the funding of the project. The senior
exec asked “why ISBL instead of SQL?” Answer: “Rather mickey-mouse, not user friendly,
difficult to use, unfaithful to the relational model, relationally incomplete. It will never catch
on”! Famous last words… [Both ISBL and Business System 12 have entries in Wikipedia,
with language examples.]

1985: Bureau service died – and Business System 12 with it. Customers were unhappy with
the move to SQL – both the restrictions imposed, from which ISBL did not suffer, and the
poor performance. This despite the fact that BS12 was fully interpreted!

1987: Joined IBM Warwick development lab, developing end-user software on top of SQL
databases. Attended a Codd & Date database conference in London in December. These two
had formed their own company by then. Both criticised SQL in their presentations. Codd went
so far as to say that SQL had “fatal flaws”. Hugh was able to add a few more! Upshot was:

1988: “Adventures in Relationland” published under the pseudonym Andrew Warden. This
contained many thinly veiled criticisms of SQL. Also joined the SQL standardisation
committee (made possible by David Gradwell, who was involved in other standardisation
work and thus realised that the SQL committee was looking for contributors), which gave him
a lever to influence the DB2 developers, who were fobbing off the Warwick lab’s
requirements under the pretext that they had to be standards compliant.

1989: Joined O.U as a part-time tutor on database. Also, Hugh’s NATURAL JOIN proposal
was accepted for the SQL standard. Unfortunately DB2 didn’t implement it.

1990: Lecturing debut at a BCS SIG meeting.

1994: SQL standards meeting in Munich precipitated The Third Manifesto. The Open
University had set a SQL expression as an exercise in a paper Hugh had to mark: what did the
query actually mean? This really brought out the “perversity” of SQL. It should not be so hard
to work out what a statement means. In most RDBMS products, though not in the SQL
standard, you have to write “4 < (expression)” instead of “(expression) > 4”. The upshot of
this was a change proposal from the USA to the “technical corrigendum” of the 1992 standard
to make it conform to the broken implementations! Hugh felt sick and ashamed that he had to
support this change proposal, thanks to who was sponsoring him on the standards committee.
In his hotel room in Munich he wrote the first draft of TTM.

Also in 1994, IBM closed the Warwick Lab due to budget constraints, making Hugh
redundant. Thanks to some contacts within IBM, Hugh was kept on as a full-timer on the SQL
standard (along with three others).

2004: Retired from IBM (sigh) and SQL (hooray). Immediately joined Warwick University to
teach Relational theory – without SQL.

Collaboration with CJ Date
From 1985 to 1989, Hugh co-authored Relational Database Writings with C.J. Date. It
appeared with the Andrew Warden pen-name.

The Third Manifesto was published as “by Hugh Darwen and C.J. Date”.

A fourth volume of RDW will be coming out soon. Each volume has included a lot on The Missing Information Problem – where SQL people tend to use NULL (see below). Hugh’s notion is that records with missing fields should be placed into separate tables that don’t contain those fields. People complain about this because of the implied performance hit – but you should never optimise in application code, only in the DBMS itself.

Anagrams
Darwen == Warden == Andrew == Wander == Warned == End War == ReDawn. Andrew Warden was Henry VIII’s fool (?). Hence the “wanderings in Relationland” by Andrew Warden. A Warden is also something of a guardian.

Present: the mess created by “The Askew Wall”
People point out that the “wall” in Hugh’s cartoon illustration of Relationland couldn’t stand up in real life. It was originally intended just to show that you had to vault over a great barrier to get into relationland, but the cracks between the stones soon came to stand for fissures that led down to a dark underworld – peopled by anonymous columns, duplicate rows and similar manifestations of the dark arts.

Relationland was discovered in the 1970s by two strange creatures – a fish (representing Codd) and a palm (representing Date).

Sonnet 112 (Shakespeare): Thy gift, thy tables, are within my brain, full character’d with lasting memory, which shall, above that idle rank, remain beyond all Date, even to eternity…

Fatal Flaws of SQL
- Anonymous Columns
- FROM clause (originally) restricted to named tables – meant that SQL was not relationally complete
- Duplicate column names
- Duplicate rows
- NULL and three-valued logic (PTRC people had managed to avoid this pitfall, which implies that expressions might not be false if the inverse expression is true)
- Failure to recognise degenerate cases (e.g. columnless tables)
- Updating views WITHOUT CHECK OPTION
- Failure to support “=” properly
- CREATE TABLE <table name> OF <type> (only columns should have types)

The Relationlander’s promise
I promise never to use the word “relational” when I mean “SQL”.

Guiding Light
Wittgenstein: all logical differences are big differences.
Darwen’s corollary: All logical mistakes are big mistakes.
Future (not): The dream of “The Third Manifesto”

The dream database language:
- Will faithfully embrace the Relational Model of Data (without extensions)
- Will support user-defined domains (types) and user-defined functions of arbitrary complexity
- Will allow SQL to be implemented in it for temporary use
- Will provide unprecedented chivalry

It might be named D!

Projects based on TTM

Rel: an implementation of Tutorial D (Dave Voorhis)
D4: a commercial implementation by Alphora, Utah – front-ends SQL, alas
Duro: a C API for relational algebra and transactions (Rene Hartmann)

Ingres D: research project at Warwick University – sponsored by Ingres Corporation. Sadly, Ingres has recently sacked the chief proponent of this.

See [www.thethirdmanifesto.com](http://www.thethirdmanifesto.com) for more.

Conceptual Integrity

(principle #7): Conceptual Integrity is the most important property of a software product (Fred Brooks, 1975).

Of course, you must have concepts before you can be true to them! These had better be: (a) few; and (b) agreeable to those invited to share them.

Brooks uses the Cathedral at Reims as an example of conceptual integrity, which the builders followed for the 150 years of construction.

Polonius in Shakespeare’s Hamlet (as altered by Hugh to address D instead of Laertes): “This above all: to thine own self be true, and it must follow, as the night the day, thou canst not then be false to any user”.

[ENDS]