

Up to a Point, Lord Copper

A response to Tom Johnston's article, "More to the Point"

(Database Programming & Design, October 24th, 1995)

by

C. J. Date, Hugh Darwen, and David McGoveran

Tom Johnston's recent article "More to the Point" [2] was a response to a critique by the present authors [3] of an earlier two-part article by Johnston [4] in support of many-valued logic. What follows is a response to that response. We begin with a slightly edited excerpt from that letter. (Note: Following normal convention, we use MVL, 2VL, 3VL, ... throughout this paper to stand for many-valued logic, two-valued logic, three-valued logic (and so on). Our comments tend to focus on 3VL specifically, though they often apply, sometimes with even more force, to 4VL, 5VL, and the rest.)

"Probably many readers are bored to tears with this whole subject. Certainly the editor of *Database Programming & Design* seems to think so; in his introduction to our previous critique, he wrote: '[Johnston's response to this critique] will appear [soon] ... With that, we'll all shake hands and end this chapter of The Great MVL Debate.'

Would that we could! But Johnston's response simply cries out for further rebuttal. The sad truth is that the topic of our debate is fundamentally important. What's more, it isn't going to go away (indeed, nor should it), so long as MVL advocates such as Johnston fail even to address-let alone answer-our many serious and well-founded objections to their position. We stand by our contention that MVL is not only not a good approach to the problem at hand, but has the potential to be a disastrously bad one.

This isn't the time or place for a detailed response to Johnston's latest comments. Suffice it to say that we find them almost entirely without merit, and that we plan to respond in detail as soon as we can. "

The present article is that promised-perhaps threatened would be a more appropriate word-detailed response.

OVERVIEW

We've already admitted that readers are probably fed up with this whole topic, so we obviously need to justify our decision to contribute yet another installment to the saga. The plain fact is that we simply can't leave Johnston with the last word. To do so would be to suggest that we acquiesce in the opinions he expresses in his latest article, and indeed that we accept his position as a valid one. *Au contraire*-we find nothing in that article to undermine our own position (in fact, we find nothing new in it at all). But to leave it unrefuted would give the dangerous impression that the whole debate is just a matter of opinion, not one of science and logic.

Let's get down to specifics. Our overriding criticisms of the article in question are twofold:

1. Portions of it are simply irrelevant-they have nothing to do with the problem at hand.
2. Those portions that do pertain to the problem at hand don't say anything new, and they don't address our fundamental objections.

To elaborate:

1. The "problem at hand" is-to spell it out one more time-the question of whether MVL is suitable as a basis for dealing with missing information. Much of Johnston's article, by contrast, concerns itself not with this question but with the question of using certain "extended" logics for purposes that have little or nothing to do with missing information per se. In the blow-by-blow response that follows, we choose not to bother much with points of Johnston's that seem to us irrelevant to the problem at hand.
2. By his own admission, Johnston does not really address-let alone answer satisfactorily-our major objections to MVL. To quote: "I will attempt to answer [those objections] in a future article. I simply ask that [Date, Darwen, and McGoveran] provide me with a list of [those objections]" (presumably to facilitate preparation of a corresponding list of responses).

Well, such a list did in fact appear in our earlier critique [3]. However, we repeat it here for convenience:

- Mathematically speaking, a "relation" that "contains a null" isn't a relation (and so relation theory doesn't apply).
- A "relation" that "contains a null" violates Codd's own Information Principle.
- If TABLE_DEE represents true and TABLE_DUM represents false, what represents unknown?
- What's the relation predicate for a "relation" that permits nulls in one or more of its columns?
- The fundamental theorem of normalization breaks down in 3VL.
- What's the justification for treating nulls as duplicates of one another for the purposes of union and projection but not for the purposes of restriction and join? In other words, are two nulls equal or aren't they?
- Even if such a justification can be found, why is it necessary for the relational algebra to suffer such complication?
- Expressions that evaluate to true in 3VL are not necessarily true in the real world (in other words, "3VL doesn't match reality").

To this list David McGoveran now adds the following challenges:

- Many benefits have been claimed for MVL. Please give a solid definition of an MVL that is supposed to provide these benefits, including its vocabulary, axioms, connectives (logical operators, including quantifiers), rules of inference, truth valuation semantics (including a definition of which truth values are designated, which anti-designated, and which neither), and an explanation of how the user is to understand the meaning of each of those truth values.
- Prove that this MVL either is or is not
 - (a) truth-functional,

- (b) truth-functionally complete,
 - (c) deductively complete,
 - (d) expressively complete,
 - (e) negation-consistent,
 - (f) relatively or absolutely consistent, and
 - (g) decidable.
- Provide a consistent (that is, uniform) interpretation of this MVL. In other words, explain the intended meaning in database terms of each and every axiom, operator, and rule of inference in this MVL.
 - If the MVL is not truth-functionally complete, then how many connectives must the user learn? Specify precisely which ones those are, and prove that what is lost by ignoring the others is unimportant.
 - If it is claimed that SQL implements some such MVL, then repeat all of the above for that particular MVL, and show that SQL/92-or some vendor's dialect thereof-implements that MVL consistently. (It would be particularly nice to pin down the meanings of IMPLIES and BI-IMPLIES in such an "SQL MVL.")

If Johnston cannot respond adequately to the foregoing requests, then he cannot substantiate his claims that MVL is useful in the field of database management. Nobody can evaluate the costs and benefits of a system of logic that isn't adequately defined.

The remainder of this article consists of a blow-by-blow rebuttal of various points in Johnston's article [2]. This article's structure, such as it is, thus mirrors that of Johnston's article. All otherwise unattributed quotations are taken from that source; we've numbered them for ease of reference. The section headings are also taken from that same source. Note: We're aware that such a blow-by-blow rebuttal, by getting heavily into details of various kinds, might serve to obscure "the big issue." But Johnston's article contains so many errors and misleading remarks that we thought that a blow-by-blow response had to be attempted, despite that risk [5].

AN AGREEMENT WITH DATE: SCOPE AND LIMITS

1. *"For example, it is precisely the counterintuitivity of IMPLIES, in basic logic, that motivated ... C. I. Lewis's formalization of modal logic ... In fact, [Lewis] pointed out that if you take any two statements from the newspaper, at random, either the first IMPLIES the second, or vice versa. This is very different from our commonsense notion of implication!"*

So what? We believe we adequately addressed the question of "the counterintuitivity of IMPLIES" (in conventional 2VL) in our previous critique; we also pointed out that the problem isn't remedied by MVL. In any case, Lewis's systems of logic suffer from counterintuitivities of their own. For example, in all of those systems, if you take any two statements from the newspaper at random, the first either implies the second or implies the negation of the second [6]. *This is very different from our commonsense notion of implication!*

David McGoveran adds: If Johnston wants to see modal logic used in a DBMS, I invite him to convince a vendor to implement such a system. Let the market decide this one; it has nothing to do with the problem at hand. Much more to the point: Johnston criticizes the 2VL version of IMPLIES, but neither he, nor Codd, nor the SQL standards committee, nor any RDBMS vendor has defined a 3VL or 4VL version! We might assume that material implication is intended (meaning that $p \text{ IMPLIES } q$ is defined as $\text{NOT } p \text{ OR } q$), but several other definitions are possible; just which one does Johnston assume? He is willing to make claims for MVL in general, but he has never, so far as I know, specified a particular MVL (by defining its vocabulary, rules of inference, logical operators, and so forth) that we might thereby examine and criticize scientifically.

2. *"... to minimize errors ... due to using relational DBMSs, we must (a) minimize ambiguities in our assertions that are recorded in those databases [sic], (b) minimize errors in manipulating the formal language with which we query the DBMSs [sic], and (c) minimize misinterpretations of the databases' [sic] responses to our queries ... I claim that it is more to the point to adopt this wider focus than to adopt Date's narrower focus on (b) alone."*

We have several comments on this extract. First, some minor editorial points: The phrase "databases' responses" would

be better shortened to just "responses," and the two occurrences of "DBMSs" should be replaced by "databases." (Perhaps these points aren't so minor. The difference between a database and a DBMS is most certainly a logical difference, and hence a big one [7].)

Anyway, to continue:

- a. "Assertions ... recorded in databases" are unambiguous by definition (see David McGoveran's further remarks on this point under Quotation 29). We contend rather that what we need to "minimize" here is ambiguities-and ideally errors too, of course-in real-world assertions, before trying to enter counterparts to those real-world assertions into our formal databases. We believe 3VL hinders rather than helps us in our attempts to meet this goal.
- b. We also contend that errors are demonstrably more likely to occur in "manipulating the formal query language" if that language is based on 3VL instead of 2VL. Moreover, such errors can be-and historically have been, and still are.to this day!-committed by both the user and the DBMS.
- c. We further contend that "misinterpretations of query responses" are, again, demonstrably more likely to occur in a 3VL context than in a 2VL one.
- d. Finally, it's not at all clear to us that Johnston's article has much constructive to say regarding "this wider focus" anyway.

DATE'S POSITIONS AND DATE'S OBJECTIONS

3. *"... although [Date's] position may have 'wide support' in some sense, it is certainly not a consensus position ... many approaches to handling missing information are still being put forward in the academic journals ... many authors have extensively studied the relevance of various kinds of null values to functional and other dependencies ... indexing nulls, in order to distinguish null values which are known to be distinct, has been studied by several different authors ... If Date's default values alternative were accepted, these investigations would be moot."*

By moot here, Johnston presumably means not debatable but irrelevant. But if so, why does he act so surprised?-it wouldn't be the first time some research direction or other has later turned out to be a dead end. Readers can surely provide plenty of their own examples [8].

4. *"A second issue Date raises is why I suggested that he apparently 'wants to remain within the confines not just of 2VL,' but also of propositional and first-order predicate logic. Was this a 'disingenuous' or 'perfunctory' attribution on my part? I don't think so ..."*

Johnston has misconstrued our original critique. Our original text read as follows:

"[Johnston] claims that "Date apparently wants to remain within the confines not just of 2VL, but of [basic logic]"- though he does also go on to say that he doesn't 'know whether [Date] has ever said this' but has 'never seen an example [in which Date] went beyond these confines.' This disclaimer seems a little perfunctory, however, or even disingenuous, since [he] then devotes virtually the whole of Part II of his article to criticizing [our] imputed position that DBMSs should 'remain within the confines ... of basic logic.' "

In other words, it was the disclaimer, not the imputation (or "attribution"), that we suggested might be disingenuous or perfunctory. Furthermore, we pointed out that it would be more accurate to say that "remaining within the confines ... of basic logic" was Codd's original position, rather than ours. Of course, we certainly support Codd in this regard.

5. *"Date objects to MVL because, among other things, it would be difficult to teach to 'naive end users.' Well, let him try to teach those users basic modal logic, or temporal modal logic! Tarskian semantics-which is the semantics for first-order predicate logic-is difficult enough for beginning students of logic to grasp. How well, then, would Date's 'naive end user' understand Kripkean semantics-the semantics for modal logic-which subsumes Tarskian semantics as merely intrapossible world semantics?"*

We find this excerpt puzzling on at least two counts. First, it seems, if anything, to be arguing in favor of our position

(that is, that we should stay with simple 2VL). Second, it was Johnston, not us, who raised the question of "exotic logics" (modal logic and the rest); we point out once again that they have nothing to do with the case-they do not represent an attack on the missing information problem.

6. *"The third and final point on which Date feels I have misrepresented him is my statement: 'That RDBMSs could eliminate these (join) costs, or at least keep them to a manageable level, is a misconception put forward by vendors and, I regret to say, by Date himself.' To this statement, Date cries 'FOUL!' ... My comment was based on two related observations ... [First,] far from criticizing 'the vendors ... for delivering inadequate levels of independence between the base and stored table levels,' Date has claimed that 'relational technology ... decouples the logical and physical levels of the system to a greater extent than did previous technologies' ... [Second,] in none of [Date's] publications have I been able to find these criticisms ... [Although Date has described] two of the ... techniques that are part of my proposed solution to this problem, [he has] said nothing about the fact that no vendor has provided an implementation of either technique."*

There are several points to be made by way of response here:

- a. We (especially Date) certainly have criticized the vendors in print for "delivering inadequate levels of independence between the base and stored table levels." Examples of such criticism-not the only ones-can be found in references [9-11].
- b. Regarding "[Date's claim] that relational technology ... decouples the logical and physical levels of the system to a greater extent than did previous technologies": Not only is this claim 100 percent true, but in fact such decoupling is 100 percent required for Johnston's "proposed solution to this problem."
- c. It's incorrect to say that "no vendor has provided an implementation of either technique." Oracle provides what is probably the best known counterexample (though we hasten to add that Oracle's implementation is very far from perfect, and we can agree with Johnston that vendors could do better in this area).

7. *"Date, however, publicly accepted the need for denormalization when he said ... that: 'Sometimes, in order to improve retrieval performance, it may be desirable to put two facts in one place, or one fact in two places.' It seems to me, therefore, that instead of criticizing vendors on this point, Date has actually acquiesced in the architectural flaw that imposes these join costs!"*

The quotation attributed to Date here is correct as stated. However, it was made-admittedly tacitly!-in the context of the implementations that existed at the time (1983, when SQL products were very new).

A LOGICAL HOWLER?

8. *"... the phrase 'infinite regress of truth values' is fairly common in the philosophical literature on MVL. And although Date did not use that phrase, and instead used the phrase 'infinite regress of nulls,' I believe that Date is quite wrong to imply that these are very different concepts."*

Nulls aren't truth values. Truth values are, by definition, values; as such, they belong to a certain domain (sometimes called the Boolean domain). Nulls are not values [12]; they do not belong to any domain. This is a logical difference with a vengeance! It follows that an infinite regress of truth values is not the same thing as an infinite regress of nulls (and any contributor to "the philosophical literature on MVL" who thinks otherwise is thoroughly confused). We stand by our original claim.

9. *"Null markers play the same role that the traditional 'T' and 'F' markers play; that is, nulls belong, with 'T' and 'F', in the characteristic matrices for sentential connectives ... If a metalinguistic marker for null is actually stored in the database, one might wonder ... why the two other metalinguistic markers-'T' and 'F'-are not stored in the database. The answer is that 'F' isn't needed so long as the Closed-World Assumption is used, since that assumption designates all and only those assertions not stored in the database, [and not] deducible from assertions stored in the database, as false assertions. And 'T' isn't needed either ... since any atomic assertion in the database in which the null marker does not appear can be assumed to have the 'T' marker."*

Johnston seems to be seriously confused here. "Null markers" apply to row-and-column intersections within relations. "T" markers, if there were such a thing, would apply to whole rows. Of course, one might imagine a "U" ("unknown") marker that would also apply to whole rows-presumably it would be used to mark all and only those rows that included one or more nulls-but that "U" marker would not itself be a null. Once again: Nulls aren't truth values.

DOES DR. CODD UNDERSTAND WHAT A RELATION IS?

10. *"Hugh Darwen's basic response to my arguments appears to be that 'relations just do enshrine basic logic by definition ... Well, if this is true, then we must apparently conclude that Dr. Codd himself does not understand what a relation is! ... Darwen's position seems to be that of a neighbor of his good friends, Tweedledum and Tweedledee: 'When I use a word,' Humpty Dumpty said, in a rather scornful tone, 'it means just what I choose it to mean-neither more nor less.' In fact, it is only on the basis of Date's 'own Version 1 of the relational model' (or that version extended by numerous essays by Date and Darwen) that relations enshrine basic logic by definition."*

It's hard to know how to respond to these truly extraordinary claims! Surely Johnston agrees that the real numbers together with the operators "+", "-", "*", and so on, "just do enshrine" basic arithmetic by definition. Well, in a precisely analogous manner, relations together with the operators union, join, difference, and so on, just do enshrine basic logic by definition-and that's the end of the matter. It's not a question of our asserting, a la Humpty Dumpty, that this is so; it is so. By the way, the "numerous essays by Date and Darwen" that Johnston refers to here have been aimed not so much at "extending" the relational model as explaining it.

Hugh Darwen adds: My co-initialee, Humpty Dumpty, is one of my favorite literary characters, and I've often used Carroll's joke at my own expense when I've felt a need to confess to some possible abuse of a word. But I can think of nothing more galling than to have the same jibe thrown at me in connection with the one word whose precise meaning I value perhaps above all others: relation. Johnston's seems to imply, "Who does Darwen think he is, claiming to know better than Dr. Codd what a relation is?"-a kind of *ad-hominery* that I find is often resorted to by people who are running short of good technical argument [13].

Well, in Codd's book [20] I find (a) a perfectly acceptable definition of relation in Chapter 1, but also (b) much text in Chapter 8, "Missing Information," that is inconsistent with that definition. The definition runs like this: "Given sets S_1, S_2, \dots, S_n (not necessarily distinct), R is a relation on these n sets if it is a set of n -tuples each of which has its first element from S_1 , its second from S_2 , and so on." To repeat, I have no quarrel with this slightly informal definition (except that I would have written what I'm sure Codd intended, namely "if and only if," in place of just "if"). What I mean by relation is also completely consistent with Chapter 4, Section 5 ("Properties of Relations"), of *Beginning Logic*, by E. J. Lemmon, first published by Thomas Nelson and Sons in 1965-four years before Codd's first paper on the subject.

11. *"Dr. Codd, of course, does not think that he has misunderstood what a relation is. According to him, 'Chris (Date) has said that a relation is not a relation if it contains missing values. This, of course, is not true at all.'"*

By definition, "a relation that contains missing values"-we take this expression to mean, more specifically, a relation that contains nulls, because of course missing values as such (again by definition) don't exist, and so can hardly be contained in anything!-is an object that contains something that does not come from any of the applicable domains. A true relation, by contrast, contains only values that do come from the applicable domains. This is a logical difference.

12. *"... Darwen is certainly right that I haven't 'given us a replacement for ... the Relational Model of Data.' And I take Date to be making the same point when he says that 'Johnston's articles ... totally ignore most of our documented objections to the idea of using three-valued logic.'"*

Yes, it's the same point. To repeat, this is the big issue! (See the second of our "overriding criticisms" in the "overview" section, earlier.) It is, accordingly, the issue that we desperately want readers to keep sight of as they wade through the point-counterpoint arguments of this article.

Hugh Darwen adds: Johnston actually goes on to say that "... it would have been foolhardy to attempt to provide an alternative version of the relational model in the space of two articles." *Alternative version?* I said replacement! The Relational Model might have needed some clarification over the years, but it's not up for versioning (pace Codd). Perhaps I should have been a little less demanding and asked not for a replacement per se, but merely for some indication of what Johnston's replacement might look like (in other words, some idea as to what sort of abstract machine he envisions). And Johnston could make a good start on a response by giving acceptable answers to our specific questions, instead of beating about the bush.

Johnston might make an even better start by justifying the decision to use "null markers" in place of values in the first place. Instead he tries to justify a certain consequence of that decision-namely, many-valued logic. And precisely because he concentrates on what is really a secondary effect, I find so much of his contribution to be beside the point. To justify the idea of "null markers," you would have to show them to be an improvement over values as a way of dealing with the problem at hand. The great advantage of using values is that (a) we already have a highly acceptable abstract machine based on nothing but values, and (b) we've had technology capable of delivering much of the promise of that abstract machine for at least two decades.

McGOVERAN'S POINTS

13. *"I challenge McGoveran to name a single example of a system of modal or multivalued logic that is inconsistent. I assure our readers that logicians don't waste time creating inconsistent systems of logic!"*

David McGoveran responds:

- a. Actually, logicians do create inconsistent logics on occasion. Initial attempts at axiomatizing a system often include inconsistencies. Johnston himself provides us with an excellent example-namely, Frege's attempt to axiomatize arithmetic.
- b. But Johnston missed my point. Logicians have proved that a large class of MVLs are not truth-functionally complete (a point Johnston concedes). They have also proved that if such an MVL is made complete (for instance, by adding the Slupecki T-function), it becomes inconsistent! Therefore, for this class of MVLs-which includes ones that Johnston apparently advocates-we can have either truth-functional completeness or consistency, but not both. I invite Johnston to read one of the best known texts on this issue [14].
- c. As an aside, I note that an inconsistency in one of the MVL systems Johnston cites was pointed out at the very conference where it was first presented [15]. That system has the bizarre (and inconsistent) property that $(R \text{ WHERE } p) \text{ UNION } (R \text{ WHERE NOT } p) \neq (R \text{ WHERE } p \text{ OR NOT } p)$!

14. *"... McGoveran is correct that several systems of 3VL are truth-functionally incomplete ... However, this assumes that truth-functional completeness is a sine qua non of a system of logic. But professional logicians don't make this assumption ... For example, modal logic is not truthfunctionally complete, lacking finite characteristic matrices entirely. Intuitionist logic, and supervaluational logic, are also not truth-functionally complete."*

David McGoveran responds: I can't fathom why Johnston has so much trouble understanding the importance of truth-functional completeness. I never claimed such completeness was "a sine qua non of a system of logic," but I did claim it was essential to a logical foundation for a DBMS! As I noted in my original articles on this topic [16], such completeness guarantees that there will be a truth-valued expression for every fact that can be deduced, under the intended interpretation, from the primitive assertions (that is, rows) in the database. Without this property, there is no guarantee that the system will be able to derive, mechanically, the answer to any query! Imagine telling your end users that - occasionally and unpredictably - the DBMS might respond to a syntactically correct query with "Huh?" and a shrug of the shoulders! I wouldn't run my business on such a system.

15. *"[McGoveran asserts] that 'Johnston confuses the need for a computational apparatus with the issue of the ambiguities of the English language.' ... On this issue, I disagree with McGoveran. It is, perhaps, the very crux of our disagreement. I stated repeatedly in my original articles what the relevance of natural-language ambiguities to 'formulas of logic' is. I will defer my last attempt to [a later section of the present article]."*

David McGoveran responds: Johnston doesn't seem to understand that the value of automated systems is that they compute. In particular, RDBMSs compute results from stated assertions. (Those stated assertions are rows, in database terms.) Of course an assertion *per se* is non-computational. Of course the natural language in which an assertion is expressed "in the outside world" is context-sensitive, and therefore ambiguous when taken out of context. But the person making the assertion knows what the assertion means! Once a row is inserted into the database - that is, once an assertion has been made - the human element is removed from the picture. The system can now compute (in a predictable and optimizable fashion) results from those assertions without human interference. This is precisely the value of non-procedural processing: Human beings don't have to tell the computer what to do each step of the way.

16. *"McGoveran has 'never seen any evidence that the development of 3VL was due to an attempt to make the meanings of the connectives more [natural]' ... But the evidence is there. The desire to develop a logical system more in accordance with ordinary discourse modes of reasoning was clearly Aristotle's and L/ukasiewicz's motivation for three-valued logic. Both believed that in ordinary discourse, statements about future contingent events do not have any truth value until those events actually occur. And Bochvar's three-valued logic 'was intended to supply a solution to the semantic paradoxes' that appear in natural language when a 2VL is assumed."*

David McGoveran responds: Here Johnston seems to be equating two very different things - connectives (logical operators) and future contingent statements. Try a simple grammatical diagramming of the future contingent statement "It will rain in 2001." I'm sorry, but I don't see any connectives here! Of course logicians have been concerned with "ordinary discourse," but my claim was that redefining the connectives had nothing to do with the evolution of logical systems, 3VL included.

17. *"[McGoveran claims that my] 'reference to Russell's Paradox-Is the set of all sets a member of itself?-and Frege is gratuitous and irrelevant.' First of all, that's not Russell's Paradox, which concerns, instead, the set of all sets that are not members of themselves. More to the point, my argument was that usage errors are likely to occur with familiar logics precisely because they are familiar ... I therefore reject McGoveran's familiarity objective, and the emphasis Date puts on the unfamiliarity of NOT's multi-valued interpretation. Instead of being 'gratuitous and irrelevant,' my reference to Russell and Frege illustrates why I do not agree that familiarity ... is a sine qua non of a logical apparatus for relational DBMSs."*

David McGoveran responds: Johnston has correctly stated Russell's Paradox; I simply stated the crux of the issue it raised without going into the original construction. Anyway, I fail to see what the discovery by Russell of a paradox (inconsistency) in Frege's very complex logical system has to do with familiarity, and I fail to see how it supports Johnston's argument that familiarity is unnecessary. The 2VL system we support is known to be free of such inconsistencies - by the way, SQL isn't! - and is very simple; Frege's system is and was very complex (just as MVLs are very complex). Familiarity is a *sine qua non* for simplicity. Simplicity in turn is highly desirable in the foundations of a DBMS, because a DBMS must be used and understood by many people.

18. *"[McGoveran claims that my] 'definition [of the Closed World Assumption] leads to logical systems that are infinitely extensible with new axioms.' Huh? And, so what?"*

David McGoveran responds: Each time we add a new axiom, we'll have a new logical system, with new logical properties and probably a new intended interpretation as well. How does Johnston expect users to deal with such a wildly dynamic notion of logic? How can we expect to run our businesses using such a system?

19. *"... basic logic is not truth functional because the quantifiers are not. [Date says this] claim is 'correct as stated' but is 'irrelevant' because 'in databases ... we deal only with finite sets.' Well, at the same time that I made that comment, I said that my point was 'to provide some motivation for keeping an open mind about truth functionality.' In order to call my point 'irrelevant,' Date would have had to miss this passage, which occurred just one line before the challenged point."*

We didn't "miss this passage." We continue to find the point irrelevant.

20. *"... here's the algorithm McGoveran requested [for identifying tautologies and contradictions in the*

supervaluational approach]. Start with a standard two-valued interpretation ... If the statement in question is a tautology or a contradiction under the two-valued interpretation, then proceed no further ... Otherwise, evaluate it using the multivalued characteristic matrix ... "

David McGoveran responds: I fear I didn't express myself clearly, and I apologize to both Johnston and the readership. What I wrote was: "I would love to see an algorithm for identifying tautologies and contradictions." Johnston understood me to be asking for a mechanical test for whether a particular expression was a tautology or a contradiction, but what I wanted was an algorithm that would generate all tautologies or all contradictions for a given MVL. Such an algorithm (a) is required to meet the needs of a logical foundation for a DBMS, and (b) does indeed exist for the form of 2VL we support, but (c) is known not to exist for the brands of MVL Johnston apparently advocates.

In any case, Johnston's "algorithm" doesn't do what he says it does. In fact, it isn't an algorithm, because it begins by assuming part of what it's supposed to be (algorithmically) specifying - namely, that we can identify the class of expressions that will be tautologies or contradictions under a 2VL interpretation. Further, it also assumes that the MVL under consideration has a characteristic matrix-but most MVLs don't.

21. *"[McGoveran says my] 'claims that RDBMSs cannot optimize for the join I/O problem are ludicrous.' On the contrary, they aren't even wrong. Optimization techniques optimize the access path to data, whereas the join I/O problem exists because frequently co-referenced rows are not physically colocated on DASD ... Furthermore, no RDBMSs today can maintain such co-location over a series of updates, even if they permit procedurally specified loading that initially establishes such co-location. In addition, no extensions to SQL DDL (other than those that I have proposed) exist to permit the declarative specification of such co-location. "*

David McGoveran responds: There are two aspects to relational optimization: logical and physical. Johnston's remark that "optimization techniques optimize the access path" has to do purely with physical optimization. If the optimizer did logical optimization too, it would recognize join as a logical operation and separately recognize that no I/O was necessary for tables that were to be joined but were in fact co-located. This separation of logical and physical aspects is a key element of the optimizer: Data can be physically stored wherever it makes the best performance sense (witness Oracle's clustering keys and Rdb's support for hashing multiple tables to the same storage area), without the user having to be aware of such physical issues at all.

DATE'S POINTS

22. *"[Date claims that] 'Johnston misrepresents, or misconstrues, the default values approach and then attacks his own misconstruction.' ... my assertion that Date's default values approach may sometimes require a value to take on two meanings was in error, and I retract the faulty criticism. But what is the difference between 'misrepresents' and 'misconstrues' that leads Date to use both terms ... ? In ordinary language, the difference is one of intention; one intentionally misrepresents, but only inadvertently misconstrues. "*

We thank Johnston for the retraction. As for "misrepresents" vs. "misconstrues," our intent was merely to give Johnston the benefit of the doubt. A full description of the default values scheme does now exist in draft form. We hope to publish it soon.

23. *"... when Date claims that a system using basic logic and a default value 'certainly is capable of recognizing this default value as meaning not applicable and acting accordingly,' I disagree. "*

Consider the example from our previous critique, in which the property "hourly rate" doesn't apply to certain employees. In the default values scheme, the domain for the HOURLY_RATE column would be, say, X, where X consists of all possible valid hourly rates plus the default value, say, "N/A". The system is informed of these facts (that the domain and the "not applicable" value for HOURLY_RATE are X and "N/A", respectively) by means of the database definition. Now the system can obviously deal appropriately with queries such as:

- Find all employees for whom the hourly rate property applies
- Find the average hourly rate for those employees

- Find all employees with an hourly rate in excess of some specified value

and so on. Doesn't this type of behavior on the part of the system effectively mean that the system is recognizing that the default value means "not applicable" and acting accordingly? If not, in what sense does it not?

24. *"The whole purpose of a logical system is to compute a metavalue assignment for a wff based on the meta-value assignments for other wffs, using the syntactic patterns and the given assignments."*

We don't see that this remark in any way contradicts or undermines our claims regarding the system's ability to "deal appropriately" with default values.

25. *"It is precisely this poverty of formal expressiveness that is the cost of adopting the logical conservatism Date recommends-a cost which he appears not to recognize."*

We believe the cost of embracing MVL is much greater, and our various publications on the subject have tried to show why. (Perhaps we should stress yet again that our concern here is solely with the suitability of MVL as a basis for dealing with missing information-not with the possibility of using "extended logics" as a basis for dealing with temporal data and the like, which we see as a totally separate issue.)

26. *"... McGoveran has proposed an elaborate framework for dealing with missing information ... However, he [makes] no reference to the dozens of publications in the academic journals on the topic of missing information, indicating that his proposals are either unresearched or else based on unacknowledged previous work by others. It does seem to me a very selective use of the 'Principle of Cautious Design' to (a) reject even a modest three-valued logic, whose mechanics were developed well over half a century ago, while (b) recommending this elaborate and possibly unresearched framework ..."*

But the whole point of that "framework"-which, by the way, we do not see as particularly "elaborate"-is to stay firmly within the boundaries of the relational model and 2VL! It seems to us, therefore, to rest on a very solidly researched foundation indeed and need no further apology on our part.

Of course, it's true that that "framework" might lead to, or possibly even require, the development of certain new database design principles. If so, however, such new principles are in all probability to be welcomed. And in any case, the question of database design has little to do with research into the relational foundation per se. (By the way, it's also true that nulls tend to encourage bad database designs, in our opinion.)

David McGoveran adds:

- It is incorrect to state, as Johnston does, that in my original articles [17] I "make no reference to ... publications in the academic journals on the topic of missing information."
- My proposal did reference the work of others (for example, Date's default values approach), but I was unable to find a description anywhere in the literature of an approach that was exactly similar to mine. If it turns out after all that my proposal isn't "original" (but see point c. below), I apologize to those on whose shoulders I unwittingly stand.
- Nothing in my approach is unresearched. Each and every database design recommendation is a part of the standard database design framework; all I did was to point out how that framework can deal with "missing information," without needing an MVL. I have used these techniques for years, and the only criticisms anyone has been able to bring have to do either with physical issues (performance or storage space, both of which I addressed in my original articles [17]) or with administration issues (current products could be more user-friendly with regard to physical design, a point I've addressed elsewhere [17]).
- Using such familiar techniques to address the problem of "missing information" will very likely lead to new understanding in the field of database design and to further clarification of the relational model, and thereby to better (more elegant and more consistent) presentation of the ideas in my articles.

27. *"I said in my second article that 'It may surprise some to realize that MVL is consistent with bivalence.' Date says 'It*

certainly surprises me!' Bivalence is the principle that every statement is either true or false. To make a long story about a peripheral issue short, Susan Haack states that 'the use of a many-valued system needn't even require denial of bivalence.' She goes on to add that a four-valued interpretation she described 'entails that every wff is either true or else false.'"

This seems to be "an explanation that doesn't explain." Here is a slightly edited version of what we said in our earlier critique:

" 'Bivalent' means two-valued; the 'MV' in MVL means, well, many-valued, where 'many' means greater than two ... [The standard version of the principle of bivalence refers to two-valuedness.] Perhaps Johnston is referring to a system in which every truth value except true itself is (at some point) effectively converted to false. SQL's 3VL might be thought of as behaving in this way, since (for example) SQL queries retrieve only those rows for which the defining condition evaluates to true, not those for which it evaluates to false or unknown. In other words, SQL effectively converts unknown to false ... If this is all that Johnston's remark means, then his point is irrelevant, and our criticisms of 3VL are still germane."

Simple dogmatic assertions such as those attributed to Haack in Johnston's response here do not in our opinion constitute an adequate refutation of these arguments, nor are they an adequate explanation or justification of what still seems to us a rather surprising remark on Johnston's part.

David McGoveran adds: It is truly frustrating to be presented with a quotation (from Susan Haack's book) that shows such little understanding of context. Haack reviews many nonstandard logics in her book and discusses their characteristics in general. In particular, she discusses the complexities of an MVL extension to the principle of bivalence. She doesn't say that all MVL systems can avoid the denial of bivalence, only that some can. Those that do, like Bochvar's, usually have an internal MVL and an external 2VL. Johnston takes Haack's statement and a grab-bag of properties of numerous MVLs and uses them to argue that some particular MVL can exist that has all the nice properties he would like to see. By contrast, I've tried to show that no single MVL can exist that does in fact have all those properties.

If Johnston will reread my articles [17], he'll find that even if we agree to forgo the familiarity requirement, those systems that satisfy the other requirements defy description in business terms. For example, I'm certain he can't even explain to a business person what Post's third truth value means in terms that business person could use! I took a comprehensive approach in my articles by classifying every MVL into one of three categories and then demonstrating the properties of each.

NOTHING TO DO WITH THE CASE?

28. "In my first article, I argued that there is no sense in which MVL is 'logically flawed' but basic logic is not. In his response, Date retracted 'logically flawed,' and substituted the notion of being conducive to generating 'conclusions that are logically incorrect ... in the real world' ... Like its predecessor-'logically flawed'-Date's new phrase seems to me to have no more cash value than 'somewhat puzzling, on first sight'; and its forcefulness, and air of authority ('logically incorrect!'), therefore seem to me very misleading. Nonetheless, given this retrenchment [sic], my argument can be rephrased as follows: there are no grounds for saying that MVL is conducive to generating such erroneous conclusions while basic logic is not."

We have published examples of queries in which the answer that is correct according to 3VL is incorrect in the real world [18]. (Yes, logically incorrect! We're talking about another logical difference.) We continue to disagree with the position Johnston articulates in the final sentence of this excerpt.

29. "... at the beginning of my first article ... I distinguished between mathematical logic and philosophical logic. I granted that McGoveran's articles were full of impressive stretches of mathematical logic, but stated that the real issue was one of philosophical logic."

David McGoveran responds: The real issue is most emphatically not one of philosophical logic (though it's clear that it would be to Johnston's advantage if it were). The whole point of automation is to eliminate the need for the human element (or at least to relegate it to a realm outside that of the system), and the situation is no different when it comes to automating database management. Every system of logic is subject to problems of human error in translating between human discourse and logical expressions. Most of those problems arise because human discourse is context-sensitive and ambiguous; indeed, much of its flexibility (in terms of linguistic evolution and expressiveness) derives from these very qualities. Human reasoning is often nonlinear; it can even capitalize on its errors without acknowledging them (much as a genetic mutation can create DNA coding errors with positive benefits).

Johnston fails to comprehend that all of these problems are confined to the mind of the user-as indeed they must be! No system of logic can choose its own axioms or provide an intended interpretation of its own symbols. The insertion of a row into a database-by definition, an operation performed by the user-establishes an axiom (possibly redundant). The user must understand the intended subject of the database (certainly no computer can) and should be honest in efforts to make assertions that others can rely upon. But once a row is in the database, the DBMS simply relies upon the truth of the corresponding assertion (axiom) and computes from it. Johnston's ambiguities are not visible to the system of logic: They are completely encapsulated in the meanings assigned to atomic domain values by users.

Claiming that the issues surrounding MVL are issues of philosophical, not mathematical, logic is rather like claiming that a program used for accounting is bad because users can't understand how to represent "three-ness" within it. We, on the other hand, would say that the program is bad because it's not an accounting program, but (perhaps) a drawing program. And we could support our position, using mathematical logic, by showing how the drawing program fails to meet the accounting need. (Analogously, I tried to show in my original article [17] that MVL fails to meet the database management need.) In other words, using "philosophical logic," you can make philosophical claims regarding such issues, but you would be hard-pressed to supply philosophical proofs.

30. *"Date also claimed that my discussion of extended logics, such as modal and temporal logic, was 'a complete red herring!' ... But the point of that discussion was to illustrate how much expressive power lay within the range of formal logic, but beyond the range of the basic logic that Date and company advocate."*

David McGoveran responds: If Johnston wants nothing more than expressive power, he should stick to English or some other natural language. Alternatively, he might try to prove (not just claim) that some modal or temporal logic is more expressive than predicate logic. (Such logics are two-valued, by the way; "red herring" was right.)

Be that as it may: I want correct (accurate), reliable, computational, predictable, understandable, complete, consistent systems that produce results I can capitalize on. If Johnston thinks some particular system of logic does all this better than a finite propositional 2VL (with equality and quantifiers that iterate AND and OR over finite sets of values), then he should write the papers that prove it. His philosophy be damned if he can't prove that "two plus two" will still be equal to "four" in his chosen system of logic.

WHAT SHOULD WE DO?

31. *"I believe that we should give up the idea of query formulation as a 'one-shot' process. Instead ... the interface should use artificial intelligence techniques ... which would present other formal queries back to the user, queries that he or she might have meant."*

On the face of it, these remarks are quite astounding. Is Johnston really arguing that we should give up the benefits (so dearly achieved) of non-procedurality? Surely not! Perhaps he is merely asking for an interactive "query helper" layer on top of an underlying interface that is indeed nonprocedural. If so, we agree that this could be a fruitful path to explore. However, database technology and the database industry clearly aren't there yet, and we would have to question the wisdom of designing and building systems now-that is, systems based on MVL-if the success of those systems were conditional on research, as yet unperformed, into new kinds of interfaces and "AI techniques."

32. *"... a good illustration of the danger of unassisted query formulation is the many 'SQL puzzles' that have appeared in*

Date's columns in this magazine. I seldom get them right! So I thank Date for these examples, because I certainly could not come up with better ones to show that basic logic itself is quite conducive to generating conclusions that are 'logically incorrect in the real world,' and therefore just as in need of AI expert assistance as any extended logic."

Do we detect a slight hint of "gotcha" in this extract? It seems likely to us that the difficulties Johnston has encountered in solving these puzzles arise as much from the quirks and idiosyncrasies of SQL as from the difficulties of "basic logic." Certainly such has been our own experience.

33. *"I think that an initial foray into an implementation of three-valued logic is warranted."*

Do we understand this suggestion correctly as implying that-in Johnston's opinion - none of today's SQL products constitutes such an implementation?

34. *"And I would call that third [truth] value simply 'null.'"*

If you call some truth value "null," what do you call a null? (see our comments on Quotations 8 and 9). One last time: Nulls aren't truth values!

35. *"Indeed, because of the many business rules that involve temporal notions ... I think that a real need exists to introduce propositional modal logic into database administration."*

Why? Why not just treat time as another domain? (Also, why the emphasis on administration?)

36. *"What Should We Do About Using Extended Logics With Relational Databases?"*

David McGoveran responds: Johnston can do whatever he likes with extended logics. But until he can conclusively demonstrate - using formal arguments - that their costs do not outweigh their benefits, I sincerely hope no commercial DBMS will use them as a foundation. When vendors demonstrate that they've used 2VL correctly, I'll support research into extended logics. At this time, however, I want to caution users: If Johnston can't show that the proposed system of logic is truth-functionally complete, don't buy it, unless you want to do a lot of work manually; the query language will, of necessity, be more procedural, and the optimizer won't be able to do very much for you.

CONCLUSION

By way of conclusion, we cannot do better than paraphrase some remarks from our previous critique:

"The problem of how best to deal with missing information in databases has long been a thorny one. Johnston's articles don't move us any closer to solving that problem ... We find absolutely nothing in those articles to cause us to modify our position ... At least with 2VL, we know what needs to be taught, we know what needs to be included in DBMS query languages and optimizers, and we know how to avoid mechanical errors. Thus, we do at least have some hope of identifying other errors as human and striving for better use of database products."

Of course, we're not so naive as to think this article will signal the end of the debate; doubtless we'll hear from other MVL defenders in due course. We ask only that any such future defenses be couched-if possible-in the form of cogent, rational, and technical arguments, so that we can respond to them in a like manner.

NOTES AND REFERENCES

1. Portions of this paper directly contributed by Hugh Darwen or David McGoveran are so flagged.
2. Tom Johnston: "More to the Point," Database Programming & Design 8, No. 11 (November 1995).
3. C. J. Date, Hugh Darwen, and David McGoveran: "Nothing to Do with the Case," Database Programming & Design

8, No. 9 (September 1995).

4. Tom Johnston: "MVL: Case Open," Database Programming & Design 8, No. 2 (February 1995); "The Case for MVL," Database Programming & Design 8, No. 3 (March 1995).

5. We note in passing that certain arguments made by us in our previous critique—for example, our argument that Johnston's "proof" that basic logic is non-monotonic was specious - are simply ignored in his latest article. Are we to infer that Johnston now agrees with us on all such points?

6. James D. McCawley: Everything that Linguists have Always Wanted to Know about Logic. Chicago, Ill.: University of Chicago Press (1981). Page 297.

7. Ludwig Wittgenstein (attrib.; quoted in Hugh Darwen and C. J. Date: "The Third Manifesto," ACM SIGMOD Record 24, No. 1, March 1995).

8. The results reported in some of the references cited by Johnston on research into nulls and MVL specifically are most certainly moot (that is, debatable). Reference [17] is a case in point.

9. C. J. Date: An Introduction to Database Systems (6th edition). Reading, Mass.: Addison-Wesley (1995). Page 72.

10. C. J. Date: Relational Database Writings 1991-1994. Reading, Mass.: Addison-Wesley (1995). Page 11.

11. Ibid. Pages 85-86.

12. Despite the fact that Johnston uses the term "null values" (emphasis added) several times in his article; see Quotation 3, for example. Incidentally, it's worth noting that Codd himself has frequently stated that a null is not a value; see, for instance, reference [22], page 179 and elsewhere.

13. Humpty Dumpty defines *ad-hominery* as the practice of indulging in *ad hominem* remarks.

14. Nicholas Rescher: Many-Valued Logic. New York, N.Y.: McGrawHill (1969). Page 163.

15. Yannis Vassiliou: "Functional Dependencies and Incomplete Information," in Proc. 6th International Conference on Very Large Data Bases, Montreal, Canada (October 1st-3rd, 1980).

16. David McGoveran: "Nothing from Nothing" (in four parts). Database Programming & Design 6, No. 12 (December 1993); 7, No. 1 (January 1994); 7, No. 2 (February 1994); 7, No. 3 (March 1994).

17. David McGoveran: "The Relational Model Turns 25," DBMS 7, No. 11 (October 1994).

18. C. J. Date: "Three-Valued Logic and the Real World," in C. J. Date and Hugh Darwen, Relational Database Writings 1989-1991. Reading, Mass.: Addison-Wesley (1992).

19. Nicholas Rescher: Op. cit.

20. E. F. Codd: The Relational Model for Database Management Version 2. Reading, Mass.: Addison-Wesley (1990).

21. For the benefit of readers who have had the endurance to stay with us this far and might be wondering about our choice of title: Lord Copper is an all-powerful and overbearing newspaper tycoon in Evelyn Waugh's novel Scoop; Mr. Salter is his foreign editor. Mr. Salter's contribution to any conversation with his employer is "limited to expressions of assent. When Lord Copper was right he said, 'Definitely, Lord Copper'; when he was wrong, 'Up to a point.'" The phrase "Up to a point, Lord Copper" has since passed into the language as an elegant variation on "I disagree," or, more simply, NO.