No More Nulls!

Yisu Remy Wang
University of Washington
Seattle, WA, USA
remywang@cs.washington.edu

ABSTRACT

Since the inception of SQL, nulls have frustrated database users and builders alike. Those writing SQL must painstakingly guard their queries agaist surprising results caused by nulls, while those building database engines constantly struggle to implement the complex semantics of 3-valued logic. Given that the relational model already provides a way to represent missing information, namely, the absence of a tuple in a relation, one may step back and ask: "Are nulls really necessary?" We answer "No!" by proposing a new semantics for SQL that completely eliminates nulls. Our semantics, called Columnar Semantics, is as expressive as the standard 3-valued logic semantics, and behaves the same when the data and query are null-free. Where the two semantics differ, Columnar Semantics results in simpler queries.

ACM Reference Format:
Yisu Remy Wang. 2023. No More Nulls!. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Nulls in SQL are a pain. Among many others, both the founder of the relational model, Codd [3], and a co-inventor of SQL, Chamberlin [2], have lamented the countless bugs caused by nulls, in both the database engine and the application code. The current SQL standard supports nulls via 3-valued logic, which is a common source of confusion for developers [11]. Surprisingly, many attempts to address the problem of nulls have focused on increasingly complex many-valued logics [4, 5, 7, 8, 12] (all the way to 6-valued logic!). These proposals have not seen wide adoption, because they are even harder to understand than 3-valued logic. A recent work by Peterfreund and Libkin [9] goes the other direction, towards simplicity: they show that the textbook 2-valued logic suffices to capture the semantics of SQL in the presence of nulls. In this paper, we go a step further and argue that **nulls can be removed altogether** from the SQL language. Our key insight is simple: nulls were invented to represent missing information, yet the relational model without nulls already provides a way to indicate information is missing, namely, with the absence of a tuple in a relation. But what if only part of a tuple is missing? Our solution is to decompose each relation into a collection of (correlated) columns, and an absent entry in a column thus represents missing information in the corresponding attribute of the tuple. Specifically, **we propose Column Normal Form, a new data normal form** inspired by Sixth Normal Form [6] and Graph Normal Form [10]. Based on Column Normal Form, **we propose Columnar Semantics, a new semantics for SQL** where the query operates on a collection of columns instead of a collection of rows.

Column Normal Form improves upon the previous normal forms by allowing missing information in any part of the tuple, even when the relation already satisfies no non-trivial dependencies. Columnar Semantics satisfies the desiderata put forward by [9]:

(1) It is as expressive as the standard 3-valued logic semantics.
(2) For null-free data and query, the behavior is identical to the standard semantics.
(3) When the two semantics differ, Columnar Semantics results in simpler queries.

While the first two criteria can be defined formally, the third one is rather subjective. Peterfreund and Libkin [9] provides one interpretation by measuring the size of the query. We achieve simplicity by completely eliminating the complexity of nulls from all queries.

The idea of handling nulls via normalization is not new. LogicBlox and RelationalAI have built successful commercial databases based on Sixth Normal Form and Graph Normal Form [1, 6, 10]. In his keynote speech [2], Chamberlin also pointed to normalization as one of the two candidate solutions to the problem of missing information. He also brought up the common criticism of normalization: decomposing the relations introduces additional joins, degrading query performance. We follow a simple solution to this problem proposed by Peterfreund and Libkin [9]: since our semantics is as expressive as the standard one, every query under our semantics can be *compiled* into another query under the standard semantics, which is then executed by existing database engines. In other words, we provide our null-free semantics as a "front-end", or "user interface", to the programmer, while the "back-end" database engine remains unchanged. On the other hand, the simpler semantics may also enable more sophisticated query optimization and execution techniques, as evident in [1, 9, 10]. In the future, more innovative systems can directly implement Columnar Semantics to take advantage of such opportunities.

The rest of this paper is organized as follows. Section 2 reviews background on missing information in SQL and discusses related work. Section 3 introduces Column Normal Form and Columnar Semantics. Section 4 compares Columnar Semantics with the standard 3-valued logic semantics, and show them to be equally expressive. Finally, Section 5 lays out future research directions and concludes.

2 BACKGROUND AND RELATED WORK

The history of nulls in SQL is almost as old as SQL itself, stretching back to the inception of the relational model some 50 years ago.

Conference'17, July 2017, Washington, DC, USA
© 2023 Association for Computing Machinery.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

We therefore do not attempt to provide a comprehensive survey of the literature, but rather focus on the fundamentals and the most relevant research. In this section, we first review the standard semantics of SQL based on 3-valued logic. Then, we discuss prior work on missing information that directly inspired our approach, including data normalization and semantics based on 2-valued logic.

3 COLUMN NORMAL FORM AND COLUMNAR SEMANTICS

4 COLUMNAR SEMANTICS AND 3-VALUED LOGIC

5 CONCLUSION

REFERENCES

[1] Molham Aref, Balder ten Cate, Todd J. Green, Benny Kimelfeld, Dan Olteanu, Emir Pasalic, Todd L. Veldhuizen, and Geoffrey Washburn. 2015. Design and Implementation of the LogicBlox System. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, Melbourne, Victoria, Australia, May 31 - June 4, 2015*, Timos K. Sellis, Susan B. Davidson, and Zachary G. Ives (Eds.). ACM, 1371–1382. https://doi.org/10.1145/2723372.2742796

[2] Don Chamberlin. 2023. 49 Years of Queries. In *Companion of the 2023 International Conference on Management of Data, SIGMOD/PODS 2023, Seattle, WA, USA, June 18-23, 2023*, Sudipto Das, Ippokratis Pandis, K. Selçuk Candan, and Sihem Amer-Yahia (Eds.). ACM, 1. https://doi.org/10.1145/3555041.3589336

[3] E. F. Codd. 1990. *The Relational Model for Database Management, Version 2.* Addison-Wesley.

[4] Marco Console, Paolo Guagliardo, and Leonid Libkin. 2016. Approximations and Refinements of Certain Answers via Many-Valued Logics. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Fifteenth International Conference, KR 2016, Cape Town, South Africa, April 25-29, 2016*, Chitta Baral, James P. Delgrande, and Frank Wolter (Eds.). AAAI Press, 349–358. http://www.aaai.org/ocs/index.php/KR/KR16/paper/view/12813

[5] C. J. Date. 2008. A critique of Claude Rubinson's paper nulls, three - valued logic, and ambiguity in SQL: critiquing Date's critique. *SIGMOD Rec.* 37, 3 (2008), 20–22. https://doi.org/10.1145/1462571.1462574

[6] C. J. Date, Hugh Darwen, and Nikos A. Lorentzos. 2002. *Temporal data and the relational model.* Elsevier.

[7] G. H. Gessert. 1990. Four Valued Logic for Relational Database Systems. *SIGMOD Rec.* 19, 1 (1990), 29–35. https://doi.org/10.1145/382274.382401

[8] Yingxian Jia, Zhuopeng Feng, and Mirka Miller. 1992. A Multivalued Approach to Handle Nulls in RDB. In *Proceedings of the Second Far-East Workshop on Future Database Systems 1992, Kyoto, Japan, April 26-28, 1992 (Advanced Database Research and Development Series, Vol. 3)*, Qiming Chen, Yahiko Kambayashi, and Ron Sacks-Davis (Eds.). World Scientific, Singapore, 71–76.

[9] Leonid Libkin and Liat Peterfreund. 2023. SQL Nulls and Two-Valued Logic. In *Proceedings of the 42nd ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS 2023, Seattle, WA, USA, June 18-23, 2023*, Floris Geerts, Hung Q. Ngo, and Stavros Sintos (Eds.). ACM, 11–20. https://doi.org/10.1145/3584372.3588661

[10] RelationalAI. 2023. *RelationalAI Documentation.* https://docs.relational.ai/rel/concepts/graph-normal-form

[11] Toni Taipalus. 2023. SQL: A Trojan Horse Hiding a Decathlon of Complexities. In *Proceedings of the 2nd International Workshop on Data Systems Education: Bridging Education Practice with Education Research* (Seattle, WA, USA) *(DataEd '23)*. Association for Computing Machinery, New York, NY, USA, 9–13. https://doi.org/10.1145/3596673.3603142

[12] Kwok-bun Yue. 1991. A More General Model for Handling Missing Information in Relational Databases Using a 3-Valued Logic. *SIGMOD Rec.* 20, 3 (sep 1991), 43–49. https://doi.org/10.1145/126482.126487