Matthew Wragg

Are Nulls Evil: A Discussion

The nature of missing information and how it is handled by a database management system (DBMS) is a topic of discussion among database professionals. One such method to handle missing information is the "null marker", suggested by E. F. Codd to be part of a proper implementation of the relational model (Codd, 1985) (Codd, 1985). This method has proven to be a particularly controversial concept in the discussion of handling missing information and the views and arguments about the aptitude of the null marker will be explored below.

Codd suggested the use of a special marker "null" in order to allow an undefined value to take advantage of multi-valued logic (MVL) in relational algebra (Codd, 1986, p. 55). Specifically, the undefined value is denoted as "unknown", creating an extra column in the two-valued logic (2VL) truth table in addition to "true" and "false" to create a three-valued logic (3VL) truth table (Fesperman, 1998, p. 3). This means the DBMS has a system-based way of handling missing information.

C. J. Date argues that the systematic and uniform method of handling unknown values via 3VL hinders the database system's ability to produce meaningful information about the real world it is modelling (Rubinson, 2007, p. 13). He puts forward the example of two tables: one for "parts" and one for "suppliers", each with a column of "cities". An SQL query can be built to return the primary keys of the part and supplier where "either the supplier and part cities are different", or where the supplier city is not equal to "Paris". If one of the cities is flagged as null for a transient reason, the SQL logical operator "NOT" will return null. In the real world, the city must either be Paris or not, which means that the 3VL handling of nulls is incomplete and gives rise to ambiguous results (Rubinson, 2007, p. 14). In other words, a 2VL SQL query is processed using 3VL when it happens to invoke a null (Date, 2004, p. 576), and thus cannot guarantee a propositional truth to the end-user (Date, 2004, p. 584).

In response to this flaw, Date recommended that database designers should instead require intelligent default values as part of the model, and use real values to define unknown information in order to handle missing information in a more semantic way (Codd, 1986, p. 70). When one considers this route, however, certain advantages of the null marker become evident. Before the null marker, database designers and end-users already used arbitrary methods to denote unknown values, which created problems (Fesperman, 1998, p. 2). For instance, there is no uniform 3VL processing provided by the system, meaning that more of a burden is placed on the application programmers and query-writers to implement checks and methods to ensure the unknown values are properly handled (Fesperman, 1998, p. 5).

Further complicating this burden is the misconception that a value of "zero" is a good default value in lieu of a null marker. Aggregate functions such as "AVG" would be unable to distinguish between an arithmetic zero and a default value "zero" and would produce inconsistent results when averaging a selection of row values (Fesperman, 1998, p. 2).

In spite of Date's notion that 3VL does not apply to real-world situations, L. Fesperman argues that missing information will always be a problem in the real world and so it is in fact the real world that necessitates the 3VL in the first place. Therefore building a consistent way to handle the 3VL in the DBMS is an advantage of the null marker (Fesperman, 1998, p. 2). C. Rubinson states that to expect a meaningful result from a 2VL statement is fallacious. Instead the query should be phrased in such a way to take into account the possibility of missing values, albeit acknowledging that it will give rise to more complicated queries (Rubinson, 2007, p. 14). J. Grant agrees with Rubinson that 3VL is not a problem in itself, but argues that Date's example is justified in that the implementation of

SQL in mainstream DBMS uses flawed 3VL truth tables (Grant, 2008, p. 24). Nevertheless, the way a DBMS handles 3VL can be updated in future releases, whereas it is unlikely that missing information in real world situations will ever go away (Fesperman, 1998, p. 2).

In conclusion, it is apparent that the need to handle missing information will always remain a problem, and that the two proposed methods of handling unknown values (null markers versus default values) both tend to produce ambiguous results and complications for different reasons. The special null marker is inflexible and assumes that all unknown information is alike. The default value creates confusion by resembling known information and lacks system-level support. Both methods require more complicated statements to be built by end-users and application programmers.

Codd has suggested that there should ideally be more than one type of null marker to denote different types of missing information, such as whether the information is contextually inapplicable, or whether the information is simply temporarily not known (Codd, 1986, p. 56). This implies that to truly reduce ambiguity using null values, the logic needs to be extended past 3VL and into MVL. The null marker necessitates the use of more complicated SQL queries while not truly solving the problem of missing information. The solution is therefore for the database designer and end-user to reduce the amount of missing information entering the database in the first place so that the null marker is invoked as little as possible.

Bibliography

Codd, E. F., 1985. Does Your DBMS Run By The Rules?. *Computerworld*, 21 10.

Codd, E. F., 1985. Is Your DBMS Really Relational?. *Computerworld*, 14 10.

Codd, E. F., 1986. Missing information (applicable and inapplicable) in relational databases. *SIGMOD Rec. 15, 4*, pp. 53-78.

Date, C. J., 2004. *An Introduction To Database Systems*. London: Addison-Wesley.

Fesperman, L., 1998. *In Defense of Nulls*. [Online]
Available at: http://www.firstsql.com/idefend.htm
[Accessed 22 11 2012].

Fesperman, L., 1998. *Should Nulls be considered harmful?*. [Online]
Available at: http://www.firstsql.com/inulls.htm
[Accessed 22 11 2012].

Grant, J., 2008. Null values in SQL. *SIGMOD Rec. 37, 3*, pp. 23-25.

Rubinson, C., 2007. Nulls, three-valued logic, and ambiguity in SQL: critiquing date's critique. *SIGMOD Rec. 36, 4*, pp. 13-17.