Title: W11-019: ANY_VALUE
Date: 2021-05-08
Author: Peter Eisentraut
Status: change proposal

References:
[Foundation IWD] 9075_8IWD14-02-Foundation_2021-05-02.pdf

Abstract

Add aggregate function ANY_VALUE.

1 Introduction

The following ballot comment was submitted:

112.	P02-USA-400		1-Major Technical	*P02-No specific location*	The SQL standard should provide an ANY_VALUE aggregate function that randomly returns a single value from the group. Though non-deterministic, the function is useful when the exact value returned is not that important (e.g., when the only difference are the presence or absence of trailing blanks in a character string) or when there is a functional dependency in the data and all values in the group are the same to start with. **Solution** None provided with comment.

This aggregate function exists in a number of implementations, so it makes sense to attempt to standardize it.

Some notes on the proposed specification:

- Similar to most unary aggregate functions,
  - Null values in the input are ignored.
  - If there are no input rows, the result is null.
- ALL and DISTINCT can be specified, but in this case they do not affect the result.

2 Proposal for [Foundation IWD]

2.1 Changes to subclause 4.18.4, "Aggregate functions"

1. Add the following list entry (after the shown list entry):

- If LISTAGG is specified, then a character string value that is the concatenation of the <character value expression> evaluated for each row that qualifies.
- If ANY_VALUE is specified, then the <value expression> evaluated for an implementation-dependent row that qualifies.

2.2 Changes to subclause 5.2, "<token> and <separator>"

1. Add the following word to <reserved word> in the appropriate position:

- ANY_VALUE

2.3 Changes to subclause 9.16, "Potential sources of non-determinism"

1. Add the following Syntax Rule:

an.1) An <aggregate function> that specifies ANY_VALUE.

2.4 Changes to subclause 10.9, "<aggregate function>"

1. Changes to the format:

```
<computational operation> ::=
  ...
  | COUNT
  | ANY_VALUE
  | STDDEV_POP
  ...
```

2. Modify Syntax Rule 7) f) thus:

f) If MAX,~~ or~~ MIN, or ANY_VALUE is specified, then the declared type of the result is *DT*.

3. Add the following General Rule:

7) d) iv.1) If ANY_VALUE is specified, then the result is an implementation-dependent value in *TXA*.

4. Add the following Conformance Rule:

1.1) Without feature T*nnn*, "ANY_VALUE", conforming SQL language shall not contain a <computational operation> that immediately contains ANY_VALUE.

2.5 Changes to Annex C, "Implementation-dependent elements"

1. Add the following entries:

4.1) Subclause 4.18.4, "Aggregate functions"

a) If ANY_VALUE is specified, then the qualified row for which the <value expression> is evaluated is implementation-dependent.

48) Subclause 10.9, "<aggregate function>"

a.1) The result of ANY_VALUE is implementation-dependent.

2.6 Changes to Annex E, "Incompatibilities with ISO/IEC 9075:2016"

1. Add the new reserved word ANY_VALUE to the existing list of added reserved words.

2.7 Changes to Annex F, "SQL feature taxonomy"

1. Add a new row to Table 44, "Feature taxonomy for optional features", as shown here:

	Feature ID	Feature Name
x	T*nnn*	ANY_VALUE

3 Disposition of ballot comments

Ballot comment #112 can be resolved, referencing the changes made by this paper.

4 Checklist

1.	Interactions with other concurrent proposals identified and editorial assistance given	none
2.	Concepts	**yes**
3.	Access Rules	n/a
4.	Conformance Rules, including the relevant Annexes	**yes**
5.	Lists of statements by category	n/a
6.	Collation coercibility determination for changes related to character strings	n/a
7.	Closing Possible Problems when a proposal resolves them	none
8.	Any new Possible Problems clearly identified	none
9.	Reserved and non-reserved keywords	**yes**
10.	Information and Definition Schemas	n/a
11.	Implementation-defined and –dependent Annexes	**yes**
12.	Incompatibilities Annex	**yes**
1.	Table of identifiers used by diagnostics statements	n/a
2.	Embedded SQL bindings and host language implications	n/a
3.	Dynamic SQL issues: including Dynamic descriptor areas	n/a

4.	PSM impact	none
5.	Schemata impact	none
6.	XML impact	none
7.	MDA impact	none
8.	PGQ impact	none

--- end of the paper ---