PRACTICAL ISSUES IN DATABASE MANAGEMENT

FOR THE THINKING PRACTITIONER

Fabian Pascal

DATABASE DEBUNKINGS

www.dbdebunk.com

PRELIMINARIES

DATABASE

- A database is a set of axioms.
- The response to a query is a theorem.
- The process of deriving the theorem from the axioms is a proof, which
  - is made by manipulating symbols according to agreed mathematical rules.
  - can only be as sound and consistent as the rules are.

-- H. Darwen

ASSERTIONS OF FACT

Employee has employee number 100, is named Spenser, works in department E21, was hired on 6/19/1980, earns a salary of $26,150

Employee has employee number 110, is named Lucchessi, works in department A00, was hired on 5/16/1958, earns a salary of $38,170

Employee with employee number (EMP#), named (ENAME), works in department (DEPT#), was hired on (HIREDATE), earns a salary of (SALARY)

EMP#	ENAME	DEPT#	HIREDATE	SALARY
100	Spenser	E21	06-19-1980	26150
110	Lucchessi	A00	05-16-1958	38170

DBMS

- A DBMS is a deductive logic system
  - Derives new facts from database facts
  - The derived facts are facts if and only if:
    - database assertions are true (facts)
    - derivation rules are sound and consistent

DATA MODEL = SEMANTICS

- Reality --> Logical Models
  - Types
  - Organization (structure)
  - Integrity
  - Manipulation
- General
- Formal (scientific)
- As simple as possible (but not simpler!)

THE RELATIONAL DATA MODEL

- Complete
  - Domains
  - R-Tables
  - Integrity Constraints
  - Manipulation
- Formal
  - Predicate Logic
  - Set Theory
  - Dependency Theory

DATA TYPES AND COMPLEXITY

CAREFUL WHAT YOU WISH FOR

CHAPTER 1

pp. 1-24

THE REAL WORLD

"The company was using a [SQL] RDBMS ... to handle data transactions for its trading applications. However, the applications required arbitrary data types, which is nearly impossible for relational systems, according to experts."

– *TRADE MAGAZINE ARTICLE*

THE REAL WORLD (cont'd)

"Object-oriented DBMSs ... support the storage and processing of any type of data,such as text, graphics, diagrams, video, audio, and user-defined data."

– *VENDOR PRESS RELEASE*

THE REAL WORLD (cont'd)

"There is an increasing need of enterprises for complex and function-related data ... this is [t]he most serious challenge so far ... to the relational database management system (RDBMS) model, [which] understands only simple types of data ... [the] solution ... is arriving in the form of Universal Server[s] ... RDBMS[s extended with] support for complex data types."

– *TRADE MAGAZINE ARTICLE*

DATA TYPES

- THE ISSUE
- FUNDAMENTALS
- PRACTICAL IMPLICATIONS
- CONCLUSIONS AND RECOMMENDATIONS

THE ISSUE

The data type concept is one of the least understood by database practitioners. This is both a cause and a consequence of the failure by SQL and its commercial dialects to implement relational domains, which are nothing but data types of arbitrary complexity.

Consequently, blame is being misplaced on the relational approach for the SQL products' lack of support for so-called "complex" types, which permits proponents of the object approach to claim with impunity that object DBMSs are superior in this respect to relational DBMSs.

FUNDAMENTALS

1. "Simple" Types
2. System-defined Types
3. User-defined Types
4. DBMS Type support
5. Type "Atomicity"
6. "Complex" Types

ASSERTIONS OF FACT

Employee has employee number 100, is named Spenser, works in department E21, was hired on 6/19/1980, earns a salary of $26,150

Employee has employee number 110, is named Lucchessi, works in department A00, was hired on 5/16/1958, earns a salary of $38,170

Employee with employee number (EMP#), named (ENAME), works in department (DEPT#), was hired on (HIREDATE), earns a salary of (SALARY)

EMP#	ENAME	DEPT#	HIREDATE	SALARY
100	Spenser	E21	06-19-1980	26150
110	Lucchessi	A00	05-16-1958	38170

TYPE CONSTITUENTS

- Name
- Possible Representation(s)
  - Actual
  - Declared
- Type constraints (optional)
- Operators (applicable to type' s values)

REPRESENTATIONS

Type: TEMPERATURE

Possible
- * *degrees Fahrenheit*
- * *degrees Celsius*
- * *hot/warm/cool*

Declared: degrees Fahrenheit

(Actual: 4 bytes)

Operators: set applicable to degrees Fahrenheit

"SIMPLE" TYPES

- Numbers
- Strings

SYSTEM-DEFINED TYPES

INTEGER	CHARACTER(n)
Declared: INTEGER (Actual: 2 bytes) Type constraint: >=-231 AND <=231 Operators: set applicable to integers	Declared: CHAR(n) (Actual: n bytes) Type constraint: n>=0 AND n<=255 Operators: set applicable to character strings

USER-DEFINED TYPES

EMP#	SALARY
Declared: CHAR(3) Type constraint: [A-Z][0-9][0-9] Operators: set applicable to employee numbers (selector, =, ..., but not <, >)	Declared: INTEGER Type constraint: =>15000 AND <=100000 Operators: set applicable to salary values (selector, =, +, <, >, ..., but not SQROOT)

RESULT TYPES

Project (PROJ#) has staff size (STAFF)

STAFF

Declared: INTEGER

Type constraint: >= 0 AND <= 450

Operators: set applicable to STAFF values (selector, =, +, /, ...)

RESULTS

Monadic

Dyadic (STAFF_RATIO: DECIMAL(4.2)

Triadic

:

N-adic

TRUTH-VALUED TYPE

TRUTH-VALUED
Declared: CHAR(1)
(Actual: SMALLINT)
Type constraint: T/F
Operators: set applicable to T/F values

"ATOMICITY"

DATE

Declared: CHAR(10)

Type constraint: [Jan,Feb,...,Dec]-[1-31]-[0-...]

Operators: set applicable to date values (selector, =, <, >, ..., but not *, /)

```
SELECT *
FROM employees
WHERE hiredate = DATE('Nov-21-1973');
```


"ATOMICITY" (cont'd)

DAY	MONTH	YEAR	DATE
Declared: INTEGER Constraint: >=1,<=31 Operators for days	Declared: CHAR(3) Constraint: [Jan,Feb,..,Dec] Operators for months	Declared: INTEGER Constraint: [0-...] Operators for years	Declared: (M MONTH, D DAY, Y YEAR) Operators for dates

```
SELECT *
FROM employees
WHERE hiredate = DATE('Nov",21,1973)
```

COMPLEX REPRESENTATIONS

"Values of any type are designatable, but such designations might be long-winded when it comes to pictures, videos, sound recordings, engineering drawings, and the like ... e.g. two-dimensional arrays of pixels, where each pixel is represented by three numbers designating amounts of red, green, and blue."

– *H. Darwen*

COMPLEX OPERATORS

"Different instances of ["complex"] types require widely different types of processing [operators], and about the only thing the[se types] have in common is that they are hard to deal with in today' s DBMS products."

– *C.J. Date*

"COMPLEX" TYPES

- User-defined
  - Representations
  - Operators
- Encapsulation
  - Programmed
  - DBMS user-extendible

PRACTICALIMPLICATIONS

1. R. Domains vs. Object Classes
2. Database Design
   1. Relational structure vs. Object Manipulation
3. DBMS Design
   1. SQL "Domains"
   2. "Universal" DBMS

RELATIONAL REPRESENTATION

Employee with employee number (EMP#) has name (ENAME), works in department (DEPT#), was hired on (HIREDATE), earns a salary of (SALARY), has fingerprint (EFP)

EMP#	ENAME	DEPT#	HIREDATE	SALARY	EFP
100	Spencer	C21	06 19 1980	26150	
150	Pienka	D11	10 11 1977	27250	

SELECT emp#,ename
FROM employees
WHERE efp = [EFPx]

OBJECT REPRESENTATION?

"It is odd that so many [object proponents] tend to use employees, departments, and so forth as examples of object classes. An object class is a type, of course, and so those [proponents] are forced to define a "collection" for those employees. What is more, those "collections" typically omit the all-important attribute names, so they are not relational tables. As a consequence, they do not lend themselves very well to the formulation of ad hoc queries, declarative integrity constraints, and so forth -- a fact that advocates of the approach themselves often admit, apparently without being aware that it is precisely the lack of attribute names that causes the problems."

– *C.J. Date*

DESIGN CHOICES

- Employee Type
  - Component representation
  - One-column table
- EMPLOYEES Table
  - no components
  - six-column table (one "complex")

THE FILM RIGHTS COMPANY

- Movie rights sales
  - Contracts(transactions)
  - "What rights, for what films, in what regions, have been sold to what customers, for what price?"

NO FREE LUNCH

- Tackling Complexity
  - Structure and Integrity (RM)
  - Manipulation (OO)

THE PLIGHT OF NASA

"... struggling with ... how to capture and analyze the [terabytes] of data beamed down to earth daily from orbiting satellites ... [a] problem is the way in which the raw data must be assigned to tables in order to be processed. This process inherently requires a degree of rationalization and some predisposition toward the ultimate use of the data. This is difficult because the scientist may not know ahead of time what analysis to run on the data. This lack of knowledge severely limits the usefulness of the system."

– *TRADE MAGAZINE ARTICLE*

FORMAL DATA MODEL

- Atomicity
- Selectivity
- Addressability

THINGS TO REFER TO

"The most appropriate design choices will emerge if careful consideration is given to the distinction between (a) declarative sentences in human language and (b) the vocabulary used in the construction of such sentences ... it is [rows in relational tables] that stand for those sentences, and it is domain values [in columns] that stand for particular elements typically nouns in that vocabulary. To say it slightly differently: Domains (types) give us values that represent things we might wish to refer to, [relational tables] give us ways of referring to those things in utterances about them."

– *C.J. Date*

DBMS DESIGN

1. SQL "Domains"
2. "Universal" DBMS

SQL SYSTEM-DEFINED "DOMAINS"

- Numbers
- Character Strings
- Bit Strings
- Dates
- Times
- Timestamps
- Year-Month Intervals
- Day-Time Intervals

SQL "DOMAINS"


```
CREATE DOMAIN emp# AS CHAR(3);


CREATE TABLE employees
(emp# emp#,
ename. ename,   ...);
```


SQL DBMS "COMPLEX DOMAINS"

- IMAGE
- TEXT
- BINARY
- VARBINARY
- (XML !)

DOMAIN IMPERSONATION

- Representation-based comparability
- No user-defined domains
- No operators (system- or user-defined) for system-defined "complex" types
- No type constraints
- No truth-valued domain

"UNIVERSAL" DBMS

- User-defined DBMS Type Extensions
  - Problem resolution?
  - Integrity, data independence violations?
  - Catalog integration?
  - Performance optimization?
  - Upgrades, maintenance, support?

CONCLUSIONS

Representations	Type
* At least one stored * Of arbitrary complexity * Can have components	* Multiple possible representations * Not necessarily stored * Operator set (encapsulated) * "Atomic" by definition

RECOMMENDATIONS

- No panacea
  - "Universal" misleading
  - ODBMS = DBMS "building-kit"
- No casual deployment to avoid structuring
  - Structure/integrity vs. manipulation
  - Dependence on programming
  - DBMS extendability

THE REAL WORLD REVISITED

"The company was using a [SQL] RDBMS ... to handle data transactions for its trading applications. However, the applications required arbitrary data types, which is nearly impossible for relational systems, according to experts."

– *TRADE MAGAZINE ARTICLE*

REAL WORLD REVISITED (cont'd)

"Object-oriented DBMSs ... support the storage and processing of any type of data,such as text, graphics, diagrams, video, audio, and user-defined data."

– *VENDOR PRESS RELEASE*

REAL WORLD REVISITED (cont'd)

"There is an increasing need of enterprises for complex and function-related data ... this is [t]he most serious challenge so far ... to the relational database management system (RDBMS) model, [which] understands only simple types of data ... [the] solution ... is arriving in the form of Universal Server[s] ... RDBMS[s extended with] support for complex data types."

– *TRADE MAGAZINE ARTICLE*

MISSING INFORMATION

WHAT YOU DON'T KNOW CAN HURT YOU

CHAPTER 10

pp. 225-47

THE REAL WORLD

MISSING INFORMATION 10

"The "SQL Language Reference" manual contradicts itself right away with the definition of NULL. Null is not equivalent to zero or to blank... Empty strings have a null value ... In subsequent section "Nulls and search conditions" the manual doesn't distinguish NULL and zero-length string. It goes on to explain that

SELECT ... WHERE x IS NULL

returns null rows as well as empty strings. The problem is, the converse is not true. Such a statement is guaranteed to return nothing

SELECT ... WHERE x = ''

This is a very nasty exception! Some further confusion in the manual can be found in the following: If any item in an expression contains a null value, then the result of evaluating the expression is null. However, this doesn't seem to apply to NULL string. These expressions simply treat NULL as empty strings.

@length(x)'[' ||x||']'

Furthermore, x IS NULL never results in null! It seems the vendor has major confusion on NULL."

MISSING INFORMATION

- THE ISSUE
- FUNDAMENTALS
- PRACTICAL IMPLICATIONS
- CONCLUSION AND RECOMMENDATIONS

THE ISSUE

As attested to by the volume of writings and the heat of the debate on the subject without an end in sight, how to treat missing information has possibly been the thorniest aspect of database management.

Users are left between a rock and a hard place: They can either rely on SQL' s badly flawed version of three-valued logic based on NULLs and risk hard to interpret database answers and/or hard to detect errors in query results, or take upon themselves the burden of what is a complex database function, which belongs in the DBMS.

FUNDAMENTALS

1. Meaningless Assertions
2. Empty Assertions
3. Many-Valued Logic
4. Missing Information as Metadata
5. DBMS Support

DATABASES AND 2VL

Employee identified by employee number (EMP#), named (NAME) works in department (DEPT#), was hired on (HIREDATE), earns a salary of (SALARY).

EMP#	NAME	DEPT#	HIREDATE	SALARY
100	Spenser	E21	06-19-1980	26150
110	Lucchessi	A00	05-16-1958	38170
120	O'Connell	A00	12-05-1963	37950
130	Quintana	C01	07-28-1971	33800
240	Marino	D21	12-05-1979	33760
250	Smith	D21	10-30-1969	39180
260	Johnson	D21	09 11 1975	17250
290	Parker	D31	05-30-1980	15340
310	Setright	D31	09-12-1964	15900

MEANINGLESS ASSERTIONS

EMP#	NAME	DEPT#	HIREDATE	SALARY	COMMISSION
100	Spenser	E21	06-19-1980	26150	
110	Lucchessi	A00	05-16-1958	38170	
120	O'Connell	A00	12-05-1963	37950	
130	Quintana	C01	07-28-1971	33800	
240	Marino	D21	12-05-1979	33760	
250	Smith	D21	10-30-1969	39180	
260	Johnson	D21	09-11-1975	17250	
290	Parker	D31	05-30-1980	15340	1780
310	Setright	D31	09-12-1964	15900	3200

ENTITY SUBTYPE/SUPERTYPE

MISSING INFORMATION 10.2.1

EMP#	NAME	DEPT#	HIREDATE	SALARY
100	Spenser	E21	06-19-1980	26150
110	Lucchessi	A00	05-16-1958	38170
120	O'Connell	A00	12-05-1963	37950
130	Quintana	C01	07-28-1971	33800
240	Marino	D21	12-05-1979	33760
250	Smith	D21	10 30 1969	39180
260	Johnson	D21	09-11-1975	17250
290	Parker	D31	05 30 1980	15340
310	Setright	D31	09-12-1964	15900

EMP

EMP#	COMMISSION
290	4780
310	3200

COM_EMP

EMPTY ASSERTIONS

MISSING INFORMATION 10.2.2

EMP#	NAME	DEPT#	HIR-DAT-	SALARY
100				
110				
120				
130				
240				
250				
260				
290				
310				

EMP#	NAME	D-PT#	HIREDATE	SALARY
100	Spenser	E21	06 19 1980	26150
110	Lucchessi	A00	05 16 1958	
120	O'Connell	A00	12-05-1963	37950
130	Quintana	C01	07-28-1971	33800
240	Marino	D21	12 05 1979	33760
250	Smith	D2	0 30 1969	
260	Johnson	D21	09-11-1975	
290	Parker	D31	05-30-1980	15340
310	Setright	D3_	09 12 1964	15900

MANY-VALUED LOGIC

- Value not valid (e.g. unprevented integrity violation): unknown
- Value not supplied (refusal to answer a question): unknown
- Value does not exist (e.g. some employees do not earn a commission): inapplicable
- Value undefined: e.g: (e.g. calculated column with 0 divisor): inapplicable
- Value is empty set (e.g. outer-joins): inapplicable

GUARANTEED CORRECTNESS

- Relational model
  - First Order Predicate Logic (2VL)
    - Accurate representation
    - Correct derivation

3VL

"[Even if] it would be possible to define a 3VL that is logically self-consistent ... in a [database] system based on [it], certain conclusions will follow that are ' logically incorrect' in the real world"

– *C.J. Date*

ALL BETS OFF

- Propositions true in 3VL not necessarily true in the real world:
- Tables with missing values are not relational tables
- Normalization rules break down;
- Msissing value marks not typed;
- No criteria for comparing missing values;

PROHIBITIVE COMPLEXITY

	2VL (n=2)	3VL (n=3)
Monadic operators	4	27
Dyadic operators	16	19,683

- Counterintuitiveness
- Implementation errors
- Interpretation errors

USER DISCIPLINE

"Suppose employee Joe is not a salesperson and so does not qualify for a commission [value inapplicable]. Then Joe's commission [would] be misrepresented as 'value unknown' ... One simple consequence of such misrepresentation is that Joe's total compensation (salary plus commission) will incorrectly evaluate to 'unknown' instead of to just the salary value".

– *C.J. Date*

"BUNDLING"

EMP#	NAME	DEPT#	HIREDATE	SALARY
100	Spencer	E21	06-19-1980	26150
110	Lucchessi	A00	05-16-1958	
120	O'Connell	A00	12-05-1963	37950
130	Quintana	C01	07-28-1971	33800
240	Marino	D21	12-05-1979	33760
250	Smith	D21	10-30-1969	
260	Johnson	D21	09-11-1975	
290	Parker	D31	05-30-1980	15340
310	Setright	D31	09-12-1964	15900

- Existence of salary
- Knowledge of salary amount

METADATA

"The kind of data we are concerned with here is most commonly encountered in audit trails in which the source of the data, data entry operator ID, time of entry, and so on are captured. Extending an audit trail concept to include metadata about missing information would seem natural. To accomplish this goal, we would need to capture [for each table] ... the primary key [value] of the row involved, a code classifying the data entry operator's belief, and an identifier for the column involved (but not a column value). This metadata can be recorded in separate lookup tables. While this solution can be implemented manually, why should not RDBMS products support audit trails in which the information to be captured is declaratively specified by the user?"

— *David McGoveran*

DBMS SUPPORT: MANIPULATION

MISSING INFORMATION 10.2.4

EMP#	NAME	DEPT#	HIREDATE
100	Spenser	F21	06-19-1980
110	Lucchessi	A00	05-16-1958
120	O'Connell	A00	12 05 1963
130	Quintana	C01	07-28-1971
240	Marino	D21	12-05-1979
250	Smith	D21	10-30-1969
260	Johnson	D21	09-11-1975
290	Parker	D31	05-30-1980
310	Setright	D31	09-12-1964

EMP

EMP#	SALARY
100	26150
120	37950
130	33800
240	33760
290	15340
310	15900

EMP SAL

SELECT emp#, name, salary
FROM emp, emp_sal
WHERE emp.emp# = sal_emp.emp#

DBMS SUPPORT: META-TABLES

MISSING INFORMATION 10.2.4

EMP#	NAME	DEPT#	HIREDATE	SALARY
100	Spenser	E21	06 19 1980	26150
110	Lucchessi	A00	05-16-1958	
120	O'Connell	A00	12-05-1963	37950
130	Quintana	C01	07-28-1971	33800
240	Marino	D21	12-05-1979	33760
250	Smith	D21	10-30-1969	
260	Johnson	D21	09-11-1975	
290	Parker	D31	05-30-1980	15340
310	Setright	D31	09-12-1964	15900

PKVALUE	COLUMN	STATUS
110	SALARY	UKN
250	SALARY	UKN
260	SALARY	UKN

MANIPULATION (cont'd)

EMP#	NAME	SALARY
110	Lucchessi	
100	Spenser	26150
120	O'Connell	37950
130	Quintana	33800
240	Marino	33760
250	Smith	
260	Johnson	
290	Parker	15340
310	Setright	15900

A

EMP#	NAME	SALARY
100	Spenser	26150
120	O'Connell	37950
130	Quintana	33800
240	Marino	33760
290	Parker	15340
310	Setright	15900

B1

EMP#	NAME
110	Lucchessi
250	Smith
260	Johnson

B2

PRACTICAL IMPLICATIONS

1. SQL NULLs
2. User Options
    1. NULLs and 4VL
    2. NULLs and 3VL
    3. 2VL and Metadata

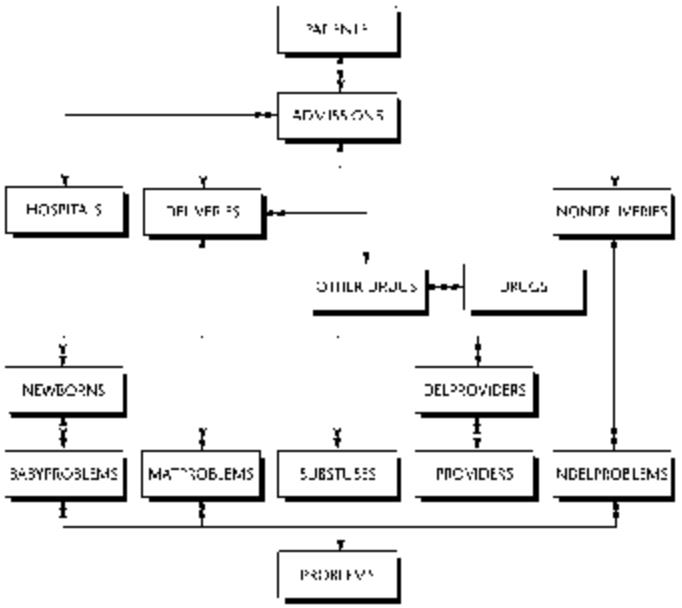
SQL NULLS

- Aggregate functions (SUM(), AVG()) ignore NULLs (except for COUNT());
- Scalar expressions on empty tables evaluate to NULL, instead of 0;
- "NULL=NULL" evaluates to NULL, but is actually invalid in SQL;
  - ORDER BY treats NULLs as equal;
  - Precede or follow "regular" values left to the DBMS vendor;
- "x IS NOT NULL" <> "NOT(x IS NULL)";
- SQL' s NOT <> natural language 'not';
  - "NOT is not Not"!
- SQL' s EXISTS does not behave like 3VL EXISTS;
- For integrity purposes NULL is treated as true;

NULLS (cont'd)

- 3VL
  - Implementation errors
- No truth-valued domain
  - NULL not typed
  - No full support of all 3VL operators
    - 27 monadic
    - 19,683 dyadic

EXAMPLE



NULLS AND 4VL

- ADMISSIONS {ADM#,PAT_ID,CLASS,SPONSOR,HOSP_ID,THOSP_ID}
  - THOSP_ID: 3/95
- DELIVERIES {DEL#, ...,}
  - 1/76, 1/78, 5/ 78, 1/80, 9/80, 10/80, 7/86, 8/89, 4/94, 3/95

NULLS AND 3VL

ADM76:
{ADM#,PAT_ID,CLASS,SPONSOR}

ADM95:
{ADM#,THOSP_ID}

- Inapplicable values
  - Outer-Joins
- 3VL problems

2VL AND METADATA

- Table proliferation
  - 5 attributes --> 15
  - 6 attributes --> 36
- Manual meta-table system?
- Manual manipulation modification?
- SQL --> NULLs

CONCLUSIONS

- Imperfect knowledge --> Limited assertability
- Real solution: 0 tolerance for missing values*
  - No DBMS metadata support --> disincentive
- SQL (NULLs)
  - Allows missing values
  - 3VL without operations
  - Flawed implementation

RECOMMENDATIONS

- Avoid/Minimize NULLs in base tables
- Exercise extreme care
  - Formulating queries
  - Interpreting results

THE REAL WORLD REVISITED

MISSING INFORMATION 10

"The "SQL Language Reference" manual contradicts itself right away with the definition of NULL. Null is not equivalent to zero or to blank... Empty strings have a null value ... In subsequent section "Nulls and search conditions" the manual doesn't distinguish NULL and zero-length string. It goes on to explain that

SELECT ... WHERE x IS NULL

returns null rows as well as empty strings. The problem is, the converse is not true. Such a statement is guaranteed to return nothing

SELECT ... WHERE x = ''

This is a very nasty exception! Some further confusion in the manual can be found in the following: If any item in an expression contains a null value, then the result of evaluating the expression is null. However, this doesn't seem to apply to NULL string. These expressions simply treat NULL as empty strings.

@length(x)'[' ||x||']'

Furthermore, x IS NULL never results in null! It seems the vendor has major confusion on NULL."