

Set-Accessing I/O for SQL

Relational Capabilities without the SQL Overhead

D L Childs

<iis@umich.edu>

1. INTRODUCTION

SQL^[5] is basically a **Set Query Language**. However, it has never been implemented to fully exploit the advantage of its set-theoretic heritage. Though developers were aware of set-accessing I/O capabilities,^[1] they chose performance inhibiting record-accessing I/O to support set operations.

Existing applications can be *retrofitted* with set-accessing I/O to improve SQL performance by orders of magnitude.

2. Machine-Independence

Since relationships are independent of representations, ARPA, in 1965, initiated research on *machine-independent* data structures. Conclusion: set-accessing I/O provided superior performance and reliability over record-accessing.^[1]

In 1970 the Relational Data Model,^[3] RDM, introduced the advantages of the machine-independence of data using set operations at the application level.

In 1973 SQUARE^[4] proposed the use of the set-theoretic Image operation (mapping function) to formally support the RDM set operations.

Both RDM and SQUARE papers referenced ARPA's research on a set-theoretic data structure, STDS, for I/O.

3. STDS: Set-Theoretic Data Structure

STDS implementations used FORTRAN array structures coupled with ordered column labelings, <A,B,C> or <C,A,B>, allowing physical ordering of columns to be inconsequential to user applications. STDS supported the MICRO RDBMS.^[2]

Using **labeled arrays** allowed application set operations to be defined without concern for storage representations. STDS provided machine-independence, but (at that time) only for data represented as arrays.

RDM set operations were equated with set operations on **labeled arrays**. The RDM operations were just a small subset of set operations already supported by STDS.^[1]

4. SQL & STDS I/O

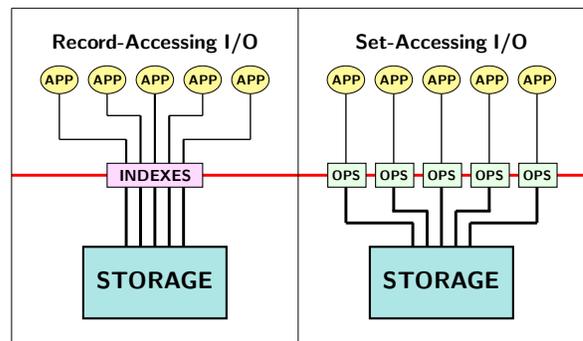
The fundamental SQL retrieval operation is **SELECT**. The fundamental data relationship is a **mapping**. The fundamental data processing operation is a **function**. The fundamental set operation for all three of these is **Image**.

This set-theoretic connection was well appreciated by the early developers of SQL. The primary operation of SQUARE, the precursor of SQL, was **mapping**.^[4]

Set-Processing Research & Development Series SAIOS: 06/28/17
Copyright 2017 Integrated Information Systems, iis@umich.edu.

5. Extended Image

Mappings produce **sets** from **sets** making **Image** the quintessential I/O operation. For Image operations to accommodate all storage representations, extensions to the foundations of set theory were required.^[9]



I/O: Machine-Dependent vs. Machine-Independent

6. XML & STDS

One of the first demonstrable advantages of an extended Image was the ability to treat XML documents as sets.^[6]

7. CONCLUSION

Though the Image operation was known to developers of SQL it was never exploited to support machine-independent I/O. This can be remedied. Adding set-accessing I/O to existing SQL systems can improve performance by orders of magnitude^[7] with access to any and all computer data.^[8,10]

8. REFERENCES

- [1] Childs: *Description of a Set-Theoretic Data Structure* - 1968 http://xsp.xegesis.org/AFIP_1968.pdf
- [2] MICRO & STDS - (1970-1998) http://en.wikipedia.org/wiki/Micro_DBMS
- [3] Codd: *A Relational Model of Data for Large Shared Data Banks*, CACM 13, No. 6 (June) 1970 <http://www.seas.upenn.edu/~zives/03f/cis550/codd.pdf>
- [4] Boyce: *Specifying Queries as Relational Expressions: SQUARE*, IBM 1973, Paper introduced the term **Table**. <http://xsp.xegesis.org/SQUARE.pdf>
- [5] Chamberlin; Boyce: *SEQUEL*, [renamed SQL] IBM 1974 <http://dl.acm.org/citation.cfm?id=811515>
- [6] Champion: *XSP: An Integration Technology for Systems Development and Evolution* - Software AG - 2001 <http://xsp.xegesis.org/Xsp-uxr.pdf>
- [7] Stout: *Information Access Accelerator*, IBI 2005 http://xsp.xegesis.org/Iaa_ibi.ppt
- [8] Childs: *Why Not Sets?* - 2010 <http://xsp.xegesis.org/WNSETS.pdf>
- [9] Childs: *Functions As Set Behavior* (Essential Concepts) - 2016 <http://xsp.xegesis.org/Process.pdf>
- [10] **XSP Technology**: (webpage), <http://xsp.xegesis.org>