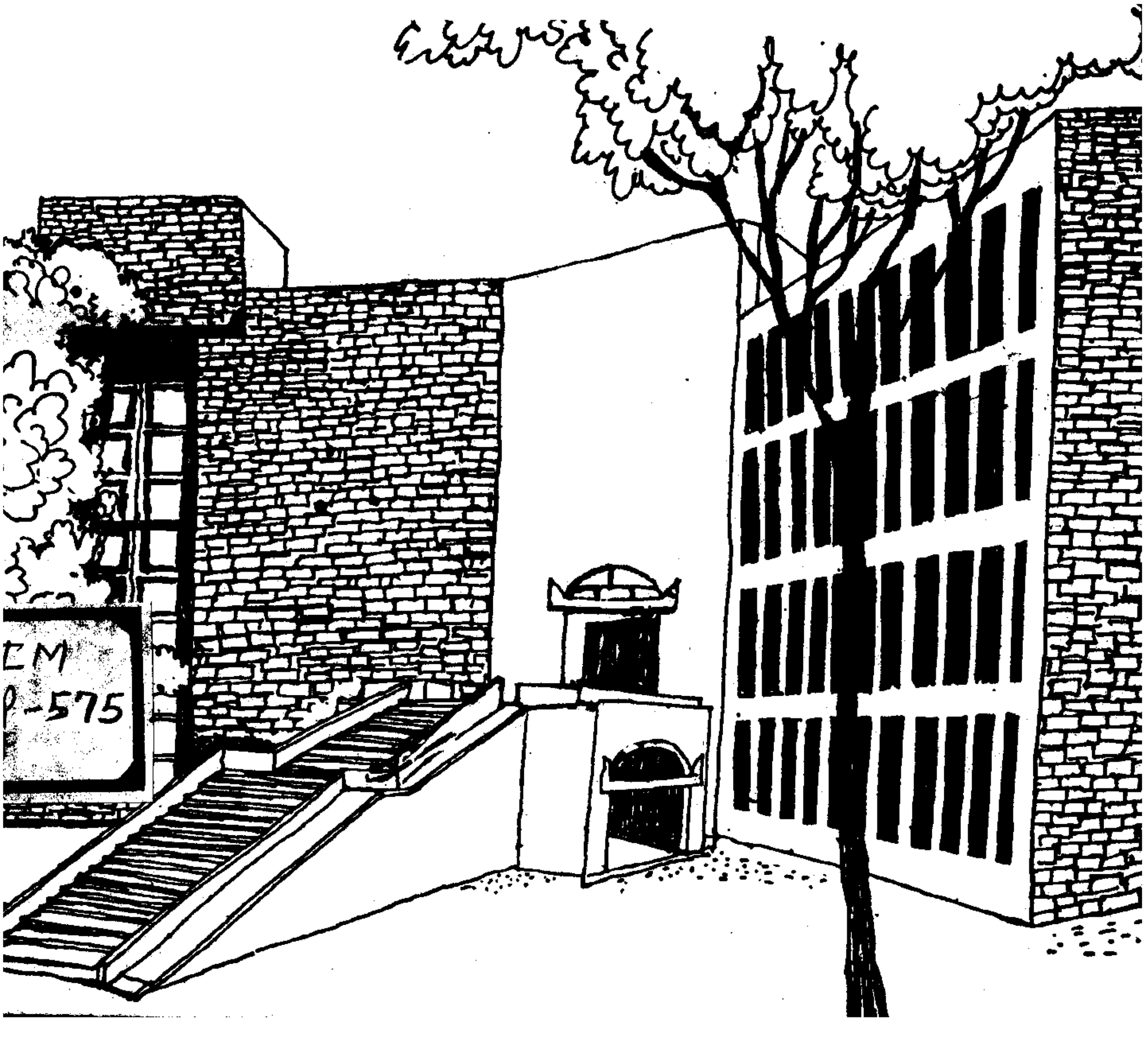
विद्याविनियोगाद्विकासः
IIM
AHMEDABAD

W. P. 57

Working Paper



EM
P-575

A PURE RELATIONAL ALGEBRA QUERY LANGUAGE USING dBASE II

By

Pralay Sen Chaudhuri

&

Suresh Ankolekar

W P No. 575

August 1985

INDIAN INSTITUTE OF MANAGEMENT
AHMEDABAD-380015
INDIA

A Pure Relational Algebra Query Language Using dBASE II

Pralay Sen Chaudhari

Suresh Ankolekar

Indian Institute of Management, Ahmedabad

Abstract

This paper discusses the development of a query language based on relational algebra implemented using dBASE II. It is observed that dBASE II is relationally incomplete in the sense that certain relational algebra operators cannot be directly simulated at the dBASE II command level. Secondly, dBASE II does not remove duplicate tuples leading to redundancy in processing and display. Thirdly, dBASE II is characterised by considerable 'command overheads' consisting of preparatory chores to be performed by the users in terms of activation of relations in appropriate workspaces. The Pure Relational Algebra Query Language (PRAQL) developed using dBASE II overcomes these limitations.

A Pure Relational Algebra Query Language Using dBASE II

Query languages for the relational model are usually based on relational algebra and/or relational calculus concepts. In relational algebra, the queries are expressed as algebraic expressions consisting of relational algebra operators and relations as operands. In relational calculus, the queries are expressed as predicates describing a desired set of tuples. Query languages *providing certain minimum capability in terms of one or both of above* concepts is said to be relationally complete. In case of languages based on relational algebra, this minimum requirement consists of five basic operators, namely,

set union
set difference
cartesian product
projection
and selection.

Additionally, three more operators derived from the five basic operators, are also considered to be primitive enough to form part of minimum requirement in the literature for relational completeness. They are,

set intersection
join
and division.

A brief description of the relational algebra operators along with the sequence of dBASE II commands to simulate the ones that are possible, is given in Table A. It is observed that three of the operators, namely, set difference, set intersection and division cannot be simulated directly using dBASE II commands. Thus we can say that dBASE II is relationally incomplete at the level of dBASE II commands operating directly on relations.

In addition to being relationally incomplete dBASE II has following limitations.

1) Duplicate tuples that may result after commands such as COPY TO - FIELD - and APPEND FROM - cannot be removed at dBASE II command level. This may lead to redundancy in processing and display.

2) It involves considerable 'command overheads' in the sense that activation of relations in appropriate workspaces is required to be done by the user. For example, before JOINing two relations, the user must SELECT appropriate workspace (PRIMARY/SECONDARY) and then USE relations corresponding to each of the workspaces. This verbosity of the language can considerably effect user efficiency.

However, dBASE II has some powerful features such as structured programming control structures, string manipulation, macro substitution and program level access to structure of a relation. Using these features it is possible to develop a query language processor to overcome the limitations.

Pure Relational Algebra Query Language (PRAQL) Processor

The PRAQL processor consists of a command interpreter and a set of procedures for command execution.

The command interpreter, after accepting a PRAQL command, interpretes it, extracts corresponding parameters from it, and invokes appropriate procedure. A list of PRAQL commands is given in Table B. The appropriate set of parameters consisting of relation names, conditional expression and list of fields are passed on to the procedure invoked.

The set of procedures consists of parameterised dBASE II command files for various relational algebra and other operations. Of these, procedures for union, selection, projection, cartesian-product and join are fairly straightforward except that duplicates, if any, are automatically removed after union and projection. The procedures for removing the duplicates, set difference, set intersection and division are developed using features of dBASE II such as programming control structures, macro substitution and program level access to structure of relations. Among these procedures, set difference is the most crucial since other procedures are either closely related to or can be derived using set difference in conjunction with others, as remarked in Table B.

PRAQL has been successfully used in the teaching environment where its emphasis on purity (i.e. duplicate tuples not allowed) and expressive power of relational algebra with minimum verbosity is very well appreciated. It can be very easily adapted to commercial environment by enhancing the command set and making certain structural validity checks of operand relations which are now mandatory into optional.

A sample PRAQL session to answer "Who are the suppliers supplying all parts?" using the relations NSP (S:N), P:NO) and P(P:NO, P:NAME) is given in Table C. The query is answered both using division operator and its equivalent sequence of other relational operators so that most of the operators could be illustrated.

Conclusions

dBASE II has following major limitations.

1) It is relationally incomplete at the dBASE II command level.

2) It does not remove resulting duplicate tuples after operations, leading to redundancy in processing and display.

3) It involves considerable command overheads consisting of preparatory chores to be performed by the users in terms of activation of relations in appropriate workspaces.

It is possible to overcome above limitations by developing a query language processor using dBASE II features such as programming control structures, string manipulation, macro substitution and program level access to structure of relation.

References

Ullman, J.D.(1984), Principles of Data Base Systems, Computer Science Press, Maryland.

Wayne Ratliff (1983) dBASE II: User Manual, Ashton-Tate, California.

Table A

Relational Algebra Operations and Equivalent dBASE II Commands

Name	Description	Notation	Equivalent dBASE II Commands
Set Union	Set of tuples that are in P or S or both	$R = P \cup S$	USE P COPY TO R USE R APPEND FROM S
Set Difference	Set of tuples that are in P but not in S	$R = P - S$	Not Available
Cartesian-Product	Set of tuples formed by concatenating every tuple of S to every tuple of P	$R = P * S$	SELECT PRIMARY USE P SELECT SECONDARY USE S JOIN TO R FOR T
Projection	Set of tuples consisting of only specified subset of fields of relation P	R = P (list of fields)	USE P COPY TO R FIELD (list of fields)
Selection	Set of tuples satisfying a specified condition in relation P	R = P (condition)	USE P COPY TO R FOR (condition)
Intersection	Set of tuples common to both P and S	$R = P \cap S$	Not Available
Join	Set of tuples formed by concatenating every tuple of P with every tuple of S provided the specified condition is satisfied	R = P * S (join condition)	SELECT PRIMARY USE P SELECT SECONDARY USE S JOIN TO R FOR (join condition)
Division	Useful in answering queries such as "Who are the suppliers supplying all parts?" The operator gives set of tuples in P (all fields except field A) such that corresponding tuples in its cartesian-product with S (field B) are all contained in P	R = P/S (field A of P; field B of S)	Not Available

Table B: PRAQL Commands*

Command	Action	Remark
PRINT <rA>	Lists relation A	Implementable directly at dBASE II command level
<rC> = <rA>	Simply copies relation A into C	
<rC> = <rA> (condition)	SELECTION Operation	
<rC> = <rA> (fields)	PROJECTION Operation	
<rC> = <rA> * <rB>	CARTESIAN-PRODUCT Operation	
<rC> = <rA> * <rB> (condition)	JOIN Operation	
<rC> = <rA> + <rB>	SET UNION Operation	
<rC> = <rA> - <rB>	SET DIFFERENCE Operation	All matching tuples are removed

<rC> = <rA> & <rB>	SET INTERSECTION Operation	Implemented as
		X = A - B
		C = A - X

<rC> = <rA> / <rB>(fN of rA; fD of rB)	DIVISION Operation	Implemented as
		XA = A (all except field N)
		XB = B (field D)
		XAB = XA + XB
		YAB = XAB - A
		YA = YAB (all except field N)
		C = XA - YA

PURIFY <rA>	Removes duplicates in relation A	Similar to SET DIFFERENCE operation on relation A with itself except that all but one matching tuples are removed.
EXIT	out of PRAQL	

* We make use of the following abbreviations.
rA : relation A
fN : field N

Table C: Sample PRAQL Session

```
C>DBASE
       *LIST OF THE RELATION NSP
S:NO    P:NO

   S1      P1
   S1      P2
   S1      P3
   S1      P4
   S1      P5
   S1      P6
   S2      P1
   S2      P2
   S3      P2
   S4      P2
   S4      P4
   S4      P5
   *LIST OF THE RELATION P

P:NO   P:NAME

   P1      NUT
   P2      BOLT
   P3      SCREW
   P4      SCREW
   P5      CAM
   P6      GOG
   DO QUERY
PRAQL>:XS=NSP(S:NO)
PRAQL>:PRINT XS
S1
S2
S3
S4
PRAQL>:XP=P(P:NO)
PRAQL>:PRINT XP
P1
P2
P3
P4
P5
P6
PRAQL>:XSP=XS*XP
PRAQL>:YSP=XSP-NSP
PRAQL>:YS=YSP(S:NO)
PRAQL>:PRINT YS
S2
S3
S4
PRAQL>:R1=XS-YS
PRAQL>:PRINT R1
S1
PRAQL>:R=NSP/P(P:NO;P:NO)
PRAQL>:PRINT R
S1
PRAQL>:EXIT
. QUIT
*** END RUN      dBASE II        ***
```