ISBL's Relational Operators

Hugh Darwen

This table (in three pages) shows the heavy influence of ISBL (1975[1]) on the semantics of relational operators for Business System 12 (BS12, 1982) and **Tutorial D** (1998, revised 2010). The **Tutorial D** example in each row defines the semantics for the other two.

Tutorial D	ISBL	Business System 12[2]
r { A, B, C } *projection*	*r*%A, B, C	PREsent(*r*, A, B, C) or PREsent(*r*, INClude(A, B, C))
r { ALL BUT A, B }	r%^A, B[3]	PREsent(*r*, EXClude(A, B, C))
r { }	*Not supported*[4]	PREsent(*r*, NONe)
r RENAME { A AS X, B AS Y } *attribute renaming*	*r* %%X := A, Y := B	PREsent(*r*, ALL, REName(A,X), RENname(B,Y))
r{A,B,C} RENAME {C AS X} *combined projection and attribute renaming*	*r%*A, B, X:= C	PREsent(*r*, A,B, REName(C,X))
r WHERE *c* *selection*	*r*;*c*[5]	SELect(*r*,*c*)
r1 JOIN *r2* *natural join*	*r1***r2*	JOIn(*r1*, *r2*)
JOIN{*r1,r2,r3*}	*r1***r2***r3*	JOIn(*r1*, *r2, r3*)
Not supported	*r1**/C1,C2/*r2* *join if {C1,C2} are the common attributes, otherwise error*	*Not supported*
r1 TIMES *r2*	*r1**//*r2*	QUAd(*r1, r2*)

[1] The publication date for the Hall/Hitchcock/Todd paper. Codd's 1991 book gives 1972 as the year of the first version of PRTV but Stephen Todd affirms that a working version existed when he joined the Scientific Centre in October, 1971.

[2] All BS12 key words could be abbreviated by truncation from the right down to a minimum of 3 letters Optional letters are shown in lower case.

[3] This syntax was added later—here it was BS12 that influenced ISBL.

[4] But they were aware of the possibility and it was they who drew the attention of BS12 developers to the existing of zero-degree relations.

[5] The semicolon was apparently the result of a typographical error. It should have been a colon!

Tutorial D	ISBL	Business System 12
JOIN{ *r1* RENAME {SUFFIX " AS '1'} RENAME {C11 AS C1, C21 AS C2}, *r2* RENAME {SUFFIX " AS '2'} RENAME {C12 AS C1, C22 AS C2}}	*r1**/1/C1,C2/2/*r2* *appends '1' to all attribute names of r1 except C1 and C2, appends '2' to all attribute names of r2 except C1 and C2, and takes the join of the two resulting relations.*	no support for prefix/suffix renaming
EXTEND *r* : {X := *expr1*, Y := *expr2,…* }	*r*#(X := *expr1*, Y := *expr2,…*[6]	CALculate(*r*, X = *expr1* Y = *expr2, …*)
SUMMARIZE *r* BY{A,B}: {X := SUM(*expr*), Y := COUNT()}	*r*$(A,B)(X := SUM(*expr*),[7] Y := SUM(1))	SUMmary(*r*, GROup(A,B), X = SUM(*expr*), Y = COUnt)
SUMMARIZE *r1* PER(*r2*): {…}	*No direct counterpart*	*No direct counterpart*
r1 UNION *r2* *operands of same type*	*r1 + r2*	UNIon(*r1*,*r2*) *operands projected on common attributes*
r1 INTERSECT *r2* *operands of same type*	*r1.r2*	INTersection(*r1, r2*)
UNION{*r1, r2, r3*} INTERSECT{*r1, r2, r3*}	*r1 + r2 + r3* *r1.r2.r3*	UNIon(*r1*,*r2*,*r3*) *no n-adic intersection*
r1 MATCHING *r2* *semijoin*	*No direct counterpart*	INTersection(*r1, r2*)
r1 NOT MATCHING *r2* *antijoin or semidifference*	*r1–r2* or *r1*-/C1, C2/*r2 to confirm common attributes*	DIFference(*r1*,*r2*)
COUNT(*r*)	*No counterpart, but see next entry*	CTTv(SUMmary(*r*, GROup(NONe), X = COUnt))
RELation{TUPle{CARD COUNT(*r*)}}	CARD(*r*)	SUMmary(*r*, GROup(NONe), CARD = COUnt)
Not supported:	DEGREE(*r*) *Degree of r, given as a one-tuple relation with a single attribute named DEG.*	SUMmary(COLumns(*r*), GROup(NONe), DEGREE = COUnt)

[6] Incidentally, ISBL supported user-defined functions that could be invoked in such expressions.

[7] Builtin aggregate functions included SUM, AVERAGE, BIGGEST, SMALLEST, ALL, ANY, PRODUCT, STRING (concatenation, obviously suspect!), SD (standard deviation), RMS (root mean square), CORREL(X,Y) (correlation of X and Y), SLOPE(X,Y), YINTER(X,Y) (intersection with Y axis). User-defined aggregate functions were also supported.

Tutorial D	ISBL	Business System 12
Not supported.	DOMAINS(*r*) *Gives heading of r as a ternary relation with attributes CNO ("column number"![8]), DNAME ("column name", and SNAME (type, where C means "character" and N means "numeric")*	PREsent(COLumns(*r*), RENname(COLNUM, CNO), RENname(COLNAME, DNAME), RENname(COLTYPE, SNAME))[9]
TUPle{A *expr1*, B *expr1*, … }	<: A = *expr1*, B = *expr1, ...* :>	Not supported
WITH (t1 := *rel-expr*) : …	t1 := N!*rel-expr* *not exact counterpart as the name t1 persists until dropped*	[DEFine][10] t1 = *rel-expr* *exact counterpart of the ISBL example*
VAR t1 INIT(*rel-expr*) KEY { ALL BUT } ;	t1 := *rel-expr* *assuming variable t1 not already defined.*	CREate t1 LIKE(*rel-expr*) STOre (t1, *rel-expr*)
t1 := *rel-expr* ;	t1 := *rel-expr* *assuming variable t1 already defined.*	STOre (t1, *rel-expr*, REPlace[11])

[8] This looks nonrelational but both ISBL and BS12 supported attribute ordering to give users control over the left-to-right order in which columns would appear when results were returned to the application, perhaps for display in tabular form—*and for no other purpose!* (unlike SQL, where column order is used, for example, in the definition of UNION, EXCEPT, and INTERSECT).

[9] Attribute names COLNO, COLNAME and COLTYPE are from vague memory—this operator is sadly missing from the BS12 reference manual. Also, COL(*r*) had several more attributes, for things like default value.

[10] Here the keyword could be omitted (and usually was), so DEFINE was the "default command".

[11] The key word APPEND gave the equivalent of **Tutorial D**'s INSERT.