A LESS COSTLY CONSTRAINTS CHECKING FOR JOIN DEPENDENCY

K.P. Tan

Department of Computer Science, National University of Singapore,
Lower Kent Ridge Road, Singapore 0511, Singapore.

Abstract

A set of n tuples in a relation of a relational database design is tested upon the constraints of the join dependency. Some constraint equalities are found to be redundant. To remove this superfluity, the universe of attibutes is partitioned into n disjoint sets and a new notation of join dependency is introduced. The checking time in each run of n tuples is significantly reduced by a factor of $(n-1)/2$ when $n > 3$. The result of less costly constraints checking is of great importance for a large number of tuples in a relation.

Key Words and Phrases: database design, join dependency, constraint equalities

1. Introduction

In relational database theory, a relation is simply a set of tuples or a table with one column for each attribute and one row for each tuple. Normalization rules provide guidelines for relational schema design in order to prevent update anomalies and data inconsistencies. The study of integrity constraints was initiated by Codd in the functional dependency (FD) on the second and the third normal forms in 1971 [Co1] and [Co2]. Six years later, the fourth normal form was defined in terms of multivalued dependency (MVD) in a many-to-many relationship [Fa1]. These two classes of dependencies have captured a great amount of semantic information of the real world. As an extension of MVD, mutual dependency (MD) [Ni] and hierarchical dependency (HD) [De] were proposed. In 1978, Rissanen introduced a very powerful data constraint, known as the join dependency (JD), [Ri]. The characteristics of its lossless join decomposition leads to the formation of the fifth normal form [Fa2]. Except FD, the other above mentioned dependencies are just the special cases of JD. Because of the particular importance of JD in the database design, a great many of the papers have explored the properties and the complete axiomatization of the JD, [ABU], [BV] and [Sc]. Checking whether a relation instance is in the fifth normal form, we need to check it against the JD. In this paper, we will provide a method to reduce the cost of constraints checking for a particular class of JD which is to be characterized.

2. Join Dependency

The join dependency [Ma] and [Ul] is defined below:

Let $R = X_1, X_2, \dots X_n$ be a set of relation schemes over the universe U. A relation r(U) satisfies the n-JD, $*[X_1, X_2, \dots X_n]$ if r decomposes lossless onto n projections, $X_1, X_2, \dots X_n$:

$$r = \pi_{X_1}(r) \bowtie \pi_{X_2}(r) \bowtie \dots \pi_{X_n}(r) \quad (1)$$

that is, r is the natural join of its projections onto the X_i's.

In other words, if r contains tuples $t_1, t_2, \dots t_n$ such that

$$t_i(X_i \cap X_j) = t_j(X_i \cap X_j) \quad \text{for all } i \text{ and } j \quad (2)$$

then

r must contain a tuple t in R

such that

$$t(X_i) = t_i(X_i) \quad 1 \leq i \leq n \tag{3}$$

3. Redundant Constraints Checking in JD

Assume R(U) is a relation schem over 4 subsets, X_1, X_2, X_3, and X_4 in the universe U and each X_i is made up of some of the attributes in the universe, such as A, B, C, D, E and F. For instance,

$$\begin{aligned} X_1 &= ABCDE \\ X_2 &= CDEF \\ X_3 &= ABDEF \\ X_4 &= ABCF \end{aligned} \tag{4}$$

According to the definition of 4-JD, all instances of R will obey JD if any set of 4 tuples, say t_1, t_2, t_3 and t_4, is in an instance such that

$$t_1\,[CDE] = t_2[CDE] \tag{5}$$
$$t_1\,[ABDE] = t_3[ABDE] \tag{6}$$
$$t_1\,[ABC] = t_4[ABC] \tag{7}$$
$$t_2\,[DEF] = t_3[DEF] \tag{8}$$
$$t_2\,[CF] = t_4[CF] \tag{9}$$
$$t_3\,[ABF] = t_4[ABF] \tag{10}$$

there exists a tuple t in the instance such that

$$t[DE] = t_1[DE] \tag{11}$$
$$t[F] = t_2[F] \tag{12}$$
$$t[AB] = t_3[AB] \tag{13}$$
$$t[C] = t_4[C] \tag{14}$$

When the first two constraint equalities are satisfied, then it follows that $t_2[DE] = t_3[DE]$ and therefore testing equality (8). Obviously, given four tuples and checking whether they obey constraint equalities (5) to (10) in the above example will lead to some redundant checkings (i.e., equalities (6) and (9)). However, testing the following set of four equations:

$$t_1[DE] = t_2[DE] = t_3[DE] \tag{15}$$
$$t_2[F] = t_3[F] = t_4[F] \tag{16}$$
$$t_3[AB] = t_4[AB] = t_1[AB] \tag{17}$$
$$t_4[C] = t_1[C] = t_2[C] \tag{18}$$

is sufficient to ensure that equalities (5) to (10) are satisfied. Suppose we refer to the comparison of an attribute value for two tuples (e.g. $t_i[B] = t_j[B]$) as an elementary checking. Testing the first set of equalities requires 18 elementary checkings whereas only 12 elementary checkings are needed for the second set. Although the saving of 6 elementary checkings for a set of 4 tuples is not much, similar equality checkings have to be done for every set of 4 tuples taken from the relation and the number of tuples in a relation may be very large. Therefore, the cost of checking whether a relation obeying the JD can be diminished significantly by using the second set of equalities. In the following section, we will introduced a new notation which is a particular class of JD and will yield directly the reduced set of constraint equalities.

4. Notation of New n-JD

The new n-JD is defined below:

Let R(U) be a relation over subsets X_1, X_2, ... X_n in the universe U,

where $U = \bigcup_{i=1}^{n} X_i$

and it is possible to partition U into n disjoint sets, Y_1, Y_2 ... Y_k

where $U = \bigcup_{k=1}^{n} Y_k$

such that

1) each X_i is a union of (n-1) Y_k's in the cyclic form,

2) each Y_k is a union of some attributes in the universe

The n-JD, $*[X_1, X_2, \ldots X_n]$ holds in R(U) iff whenever there are n tuples t_1, t_2, ... $t_n \in R$

such that

$$\forall_{k=1,\ldots n}\; t_k\Big[\bigcap^{k \oplus_n p} X_i\Big] = \ldots = t_{k \oplus_n p}\Big[\bigcap^{k \oplus_n p} X_i\Big] \tag{19}$$

where p=n-2, the number of intersections among X_i's.

$\exists$ a tuple $t \in R$

such that

$$\forall_{i=1,\ldots n}\; t[X_i] = t_i[X_i] \tag{20}$$

Remarks:

1) the notation $\oplus_n$, (plus modulo n), is defined as follows:
given integers q and $q' \leq n$

$$q \oplus_n q' = \begin{cases} q+q' & \text{if } q+q' \le n \\ q+q'-n & \text{if } q+q' > n \end{cases} \quad (21)$$

2) $$\bigcap_{i=k}^{k \oplus_n p} X_i = X_k \cap X_{k \oplus_n 1} \cdots \cap X_{k \oplus_n p} \quad (22)$$

3) Y_k's in the cylic form in X_i means that

$$\forall_{i=1,\dots n} \quad X_i = \bigcup_{k=i}^{i \oplus_n p} Y_k \quad (23)$$

Besides the modification on the constraint equations, the result (i.e. the new tuple generated) of the new n-JD is the same as that of the old n-JD.

Consider the example in Section 3. The new 4-JD is satisfied whenever there are tuples t_1, t_2, t_3 and t_4 in R such that

$$t_1[X_1 \cap X_2 \cap X_3] = t_2[X_1 \cap X_2 \cap X_3] = t_3[X_1 \cap X_2 \cap X_3] \quad (24)$$

$$t_2[X_2 \cap X_3 \cap X_4] = t_3[X_2 \cap X_3 \cap X_4] = t_4[X_2 \cap X_3 \cap X_4] \quad (25)$$

$$t_3[X_3 \cap X_4 \cap X_1] = t_4[X_3 \cap X_4 \cap X_1] = t_1[X_3 \cap X_4 \cap X_1] \quad (26)$$

$$t_4[X_4 \cap X_1 \cap X_2] = t_1[X_4 \cap X_1 \cap X_2] = t_2[X_4 \cap X_1 \cap X_2] \quad (27)$$

there exists a tuple t in R

such that

$$t[X_1] = t_1[X_1] \quad (28)$$
$$t[X_2] = t_2[X_2] \quad (29)$$
$$t[X_3] = t_3[X_3] \quad (30)$$
$$t[X_4] = t_4[X_4] \quad (31)$$

As a result, equations (15) to (18) are derived from the new constraint equations from (24) to (27) and are equivalent to those useful constraints, such as (5), (7), (8) and (10) in the old JD. Also, equations (28) to (31) lead to the results in equations (11) to (14).

5. Comparison of Old JD with New JD

In order to compare the old JD with the new JD precisely, let n=4. The effect of the cyclic characteristics of the disjoint sets, Y_k's, in the subsets X_i's, is shown in the Appendix. The comparison of the constraint equalities and elementary checkings of the old n-JD with the new n-JD is shown in Table 1.

In the case of old n-JD, there are n(n-1)/2 constraint equations with one constraint equality in each equation and one intersection between 2 X_i's in each equality. Each disjoint set occurs (n-1)(n-2)/2 times in all the equations. It leads to n(n-1)(n-2)/2 disjoint set checkings or (n-1(n-2)xk/2 elementary checkings. On the other hand, the new n-JD has n(n-2) constraint equalities in n constraint equations. There are (n-2) intersections among (n-1) X_i's in each equality. However, each disjoint set occurs only (n-2) times. Consequently, there are n(n-2) disjoint set checkings or (n-2)xk elementary checkings. Therefore, in terms of elementary checkings,

$$\text{old n-JD} : \text{new n-JD} = (n-1)/2 : 1 \quad (32)$$

That is, for any set of n tuples in a relation, the constraint checking upon the new n-JD is (n-1)/2 times faster than that upon the old n-JD. In other words, we need only 2/(n-1) of the total time required originally. Therefore, the new n-JD is less time consuming and is more efficient.

6. Conclusion

In the new n-JD, the universe U is partitioned into n disjoint sets, Y_k's. Each subset X_i contains (n-1) disjoint sets in the cyclic combination and (n-2) intersections of X_i's are implied in each constraint equality. As a result, the checking time is reduced by a factor of (n-1)/2 in each run of n tuples in the relation. For a large value of n and a large number of tuples in a relation in the database design, the checking using the new notation of JD is always less costly than the old notation.

Acknowledgement

The author is grateful to J.M. Nicolas for many helpful discussions while the author visited him at CERT at Tonlouse, France in June 1983. The manuscript was prepared while the author was on his sabbatical leave at the Department of Computer Science, University of Maryland at College Park in early 1984.

Table 1. Comparison of Constraint Equalities and Elementary Checkings of Old n-JD with New n-JD

Where n is the number of subsets and k, the total number of attributes in the universe U. X_i and Y_k denote a subset and a disjoint set of attributes respectively.

	Old n-JD				New n-JD			
No. of X_i's	3	4	5	n	3	4	5	n
No. of Y_k's in each X_i	2	3	4	n-1	2	3	4	n-1
No. of constraint equations	3	6	10	$\frac{n(n-1)}{2}$	3	4	5	n
No. of equalities in each equation	1	1	1	1	1	2	3	n-2
No. of intersections in each equality	1	1	1	1	1	2	3	n-2
No. of resultant Y_k's in each equality	1	2	3	n-2	1	1	1	1
No. of occurrences of each Y_k	1	3	6	$\frac{(n-1)(n-2)}{2}$	1	2	3	n-2
No. of Y_k checks	3x1	6x2	10x3	$\frac{n(n-1)(n-2)}{2}$	3x1	8x1	15x1	n(n-2)
No. of elementary checks	1xk	3xk	6xk	$\frac{(n-1)(n-2)xk}{2}$	1xk	2xk	3xk	(n-2)xk

Appendix: Comparision of the Constraints Between Old 4-JD and New 4-JD

In the new 4-JD, we have

1) 4 subsets of attributes, X_1, X_2, X_3, and X_4 in the universe,

2) each subset contains 3 out of 4 disjoint sets, Y_1, Y_2, Y_3 and Y_4 in the cyclic union.

3) each Y_k may contain any number of attributes, A_j, say, j=1 to 10, e.g., $Y_1 = A_1A_2A_3$, $Y_2 = A_4A_5$, $Y_3 = A_6A_7A_8$, and $Y_4 = A_9A_{10}$

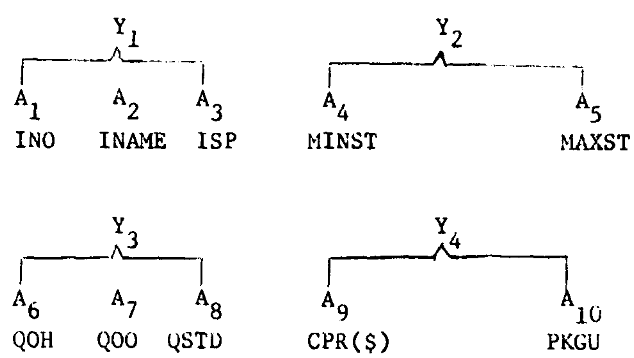
Thus, we have

$$X_1 = Y_1Y_2Y_3$$
$$X_2 = Y_2Y_3Y_4$$
$$X_3 = Y_3Y_4Y_1$$
$$X_4 = Y_4Y_1Y_2$$

Take the supermarket inventory as an example, we may have 10 attributes in the universe U, such as

INO = Item number
INAME = Item name
ISP = Item specification
MINST = Minimum stock
MAXST = Maximum stock
QOH = Quantity on hand
QOO = Quantity on order
QSTD = Quantity of total sale-to-date
CPR = Cost price
PKGU = Package unit

Thus,



So, each Y_k is a domain, containing some attributes which are related to each other. Based on these given X_i's and Y_k's the comparison of the constraints between the old 4-JD and the new 4-JD is as below:

In view of each elementary attribute checking, the ratio of the checking time required between thenew 4-JD and the old 4-JD for each run of 4 tuples in the relation is 2/3. That is, by

Old 4-JD	New 4-JD
The constraint equations are: $t_1[X_1 \cap X_2] = t_2[X_1 \cap X_2]$ $t_1[X_1 \cap X_3] = t_3[X_1 \cap X_3]$ $t_1[X_1 \cap X_4] = t_4[X_1 \cap X_4]$ $t_2[X_2 \cap X_3] = t_3[X_2 \cap X_3]$ $t_2[X_2 \cap X_4] = t_4[X_2 \cap X_4]$ $t_3[X_3 \cap X_4] = t_4[X_3 \cap X_4]$	The constraint equations are: $t_1[X_1 \cap X_2 \cap X_3] = t_2[X_1 \cap X_2 \cap X_3] = t_3[X_1 \cap X_2 \cap X_3]$ $t_2[X_2 \cap X_3 \cap X_4] = t_3[X_2 \cap X_3 \cap X_4] = t_4[X_2 \cap X_3 \cap X_4]$ $t_3[X_3 \cap X_4 \cap X_1] = t_4[X_3 \cap X_4 \cap X_1] = t_1[X_3 \cap X_4 \cap X_1]$ $t_4[X_4 \cap X_1 \cap X_2] = t_1[X_4 \cap X_1 \cap X_2] = t_2[X_4 \cap X_1 \cap X_2]$
That is, $t_1[Y_2Y_3] = t_2[Y_2Y_3]$ $t_1[Y_3Y_1] = t_3[Y_3Y_1]$ $t_1[Y_1Y_2] = t_4[Y_1Y_2]$ $t_2[Y_3Y_4] = t_3[Y_3Y_4]$ $t_2[Y_2Y_4] = t_4[Y_2Y_4]$ $t_3[Y_4Y_1] = t_4[Y_4Y_1]$	That is, $t_1[Y_3] = t_2[Y_3] = t_3[Y_3]$ $t_2[Y_4] = t_3[Y_4] = t_4[Y_4]$ $t_3[Y_1] = t_4[Y_1] = t_1[Y_1]$ $t_4[Y_2] = t_1[Y_2] = t_2[Y_2]$
1) 6 equalities in 6 equations 2) 1 intersection in each equality 3) 2 resulting disjoint sets in each equality 4) each disjoint set occurs 3 times in the 6 equations 5) need (3x4) or (6x2) disjoint set checkings	1) 8 equalities in 4 equations 2) 2 intersections in each equality 3) 1 resulting disjoint set in each equality 4) each disjoint set occurs 2 times in the 4 equations 5) need (2x4) or (1x8) disjoint set checkings
Suppose Y_1, Y_2, Y_3 and Y_4 contains n_1, n_2, n_3 and n_4 attributes respectively. There are k attributes altogether, i.e.. $k = n_1+n_2+n_3+n_4$	e.g. k = 10 in the above example
6) need 3xk elementary checkings	6) need 2xk elementary checkings

adopting the new JD, we need only two-thirds of the time required for the old JD. Therefore, the checking using the new JD is always less time consuming.

References

[ABU] Aho, A., Beeri, C. & Ullman, J., "The Theory of Joins in Relational Database," ACM TODS 4 3, Sept. 1979, pp. 297-314.

[BV] Beeri, C. & Vardi, M., "On the Properties of Total Join Dependencies," H. Gallaire, J. Minker, & J.M. Nicolas (Eds.), Advances in Database Theory, Vol. 1, Plenum Press, New York, 1981, pp. 25-71.

[Co1] Codd, E.F., "Normalized Data Base Structure: A Brief Tutorial," ACM SIG-FIDET Workshop on Data Description, Access, and Control, San Diego, Ca., Nov. 11-12, 1971, E.F. Codd and A.L. Dean (Eds.).

[Co2] Codd, E.F., "Further Normalization of of the Data Base Relational Model," Database Systems, R. Rustin (Ed.), Courant Computer Science, Symposium, Vol. 6, Prentice-Hall, Englewood Cliffs, N.J., 1972, pp. 33-64.

[De] Delobel, C., "Normalization and Hierarchichical Dependencies in the Relational Data Model," ACM TODS 3 3, Sept. 1978, pp. 201-222.

[Fa1] Fagin, R., "Multivalued Dependencies and A New Normal Form for Relational Databases," ACM TODS 2 3, Sept. 1977, pp. 262-278. Also IBM Research Report RJ1812

[Fa2] Fagin, R., "Normal Forms Relational Data base Operators," ACM SIG-MOD Int. Conf. on Management of Data, Boston, Mass., May 1979. IBM Research Report RJ2471, February, 19 1979.

[Ma] Maire, D., The Theory of Relational Databases, Computer Science Press, 1983.

[Ni] Nicolas, J.M., "Mutual Dependencies and Some Results on Indecomposable Relations," Proc. 4th Int. Conf. on Very Large Data Bases, West Berlin, Germany, Sept. 1978, pp. 360-367.

[Ri] Rissanen, J., "Theory of Joins for Relational Databases - A Tutorial Survey," Proc. 7th Symposium on Mathematical Foundations of Computer Science, Lecture Notes in Computer Science 64, Winkowski (ed.), Spring-Verlag, New York, 1978, pp. 537-551.

[Sc] Sciore, E., "A complete Axiomization of Full Join Dependencies," J.ACM 29 2, April, 1982, pp. 373-393.

[Ul] Ullman, J.D., Principles of Database Systems, 2nd ed., Computer Science Press, Potomac, Maryland, 1982.