Equivalence of the Template Dependencies and the Join Dependencies*

LI Xing-ye, WANG Shu-ning, YUE Zhan-feng

(*Department of Automation, Tsinghua University, Beijing* 100084, *China*)

E-mail: lixingye@tsinghua.org.cn

Received December 22, 2001; accepted May 13, 2002

Abstract: The relationship between the total template dependencies and the total join dependencies is probed into by means of abstract algebra. First, two equivalence relations are defined in the set of the total template dependencies and the set of the total join dependencies respectively. The equivalence relations regard the dependencies that function is the same as equivalent dependencies. Then, it is proved that two quotient sets under two equivalence relations constitute monoids respectively and there is an isomorphism mapping between the monoids, which shows that the class of the total join dependencies is essentially identical with the class of the total template dependencies. Finally, an interesting result about the total acyclic join dependencies is given. The relevant results will play active role in designing relational database schemes.

Key words: template dependency; join dependency; implication problem; isomorphism; monoid

One of the important issues in designing relational database schemes is the specification of the constraints that the data must satisfy to model correctly the part of the world under consideration.

Of particular interest is the constraints called data dependencies. The first dependencies to be studied were the functional dependencies[1], which were followed by the multivalued dependencies and join dependencies[2~8]. Later, several types of dependencies were investigated in the literature. One of them is the template dependencies[9].

In this paper, we probe into the relationship between the total template dependencies and the total join dependencies by means of abstract algebra. At first, we give the equivalence relations ρ and σ in the set S_T of the total template dependencies and the set S_J of the total join dependencies respectively. The equivalence relations regard the dependencies that function is the same as equivalent dependencies. Then, we prove that both the quotients S_T/ρ and S_J/σ constitute monoids respectively and there is an isomorphism mapping between the monoids, which shows that the class of the total join dependencies is essentially identical with the class of the total template dependencies. Finally, an interesting result about the total acyclic join dependencies is given.

1 Basic Concepts

The set composed of all attributes is denoted *U*, which is a finite non-empty set containing different symbols. It

---

* Supported by the National Natural Science Foundation of China under Grant No.69974023 (国家自然科学基金); the Natural Science Foundation of Tsinghua University (清华大学科学基金); the Doctoral Foundation of Tsinghua University (清华大学博士基金)

LI Xing-ye was born in 1958. He is a Ph.D. candidate at the Department of Automation, Tsinghua University. His research interests are in data dependency theory in relational database and nonlinear system modeling theory. **WANG Shu-ning** was born in 1956. He is a professor and doctoral supervisor of the Department of Automation, Tsinghua University. His current research areas are complexity system modeling, robust identification, nonlinear system identification, decision analysis, system optimization. **YUE Zhan-feng** was born in 1974. He received his M.S. degree at the Department of Automation, Tsinghua University in 2000. His research interests are in multisensor fusion.

is often called *universe* and its non-empty subsets are attributes sets.

We denote the domain of the attribute A by $DOM(A)$.

It is a convention in this paper that the domains of distinct attributes are assumed to be disjoint. This convention limits the dependency mentioned in this paper to the typed dependency.

A Z-value is a mapping $w:Z\rightarrow DOM(Z)$, such that $w(A)\in DOM(A)$ for all $A\in Z$. Particularly, a U-value is a tuple and the set of tuples is called relation. For a tuple w and a set $Z\subseteq U$, the restriction of w on Z is a Z-value, denoted $w[Z]$. Let I be a relation, then the projection of I on Z, denoted $\pi_Z(I)$, is a Z-value set $\{w[Z]|w\in I\}$. We denote the set $\{w[A]|A\in U\}$ by $VAL(w)$ and the set $\bigcup_{w\in I} VAL(w)$ by $VAL(I)$.

Assuming that $R_i\subseteq U$, $1\le i\le l$. We define the projection join of the relation I on $R_1,R_2,\ldots,R_n$ as $\{t|t[R_i]\in\pi_{R_i}(I),1\le i\le l\}$, denoted $^*[R_1,\ldots,R_n](I)$. If $^*[R_1,\ldots,R_n](I)=I$, it is said that I satisfies the join dependency $^*[R_1,\ldots,R_n]$. Particularly, if $\bigcup_{1\le i\le l}R_i=U$, I satisfies the total join dependency $^*[R_1,\ldots,R_n]$, otherwise, I satisfies the embedded join dependency $^*[R_1,\ldots,R_n]$. It is manifest that when $l=2$, $^*[R_1,R_2]$ is the multivalued dependency $R_1\cap R_2\rightarrow R_1-R_2|R_2-R_1$ or $R_1\cap R_2\rightarrow R_2-R_1|R_1-R_2$.

Let h be the mapping defined on $\times_{w\in I}DOM(A)$. h is called *valuation* if $h(DOM(A))\subseteq DOM(A)$ for all $A\in U$. The valuation h can be extended to tuples and relations as follows. Let w be a tuple; then the valuation h of w is still a tuple and it is the result of the composite mapping of w and h on U, denoted $h(w)$. The valuation h of relation I is $h(I)=\{h(w)|w\in I\}$.

A tableau is a pair $\langle I,u\rangle$, where I is a relation and u is a tuple. The operation result of tableau $\langle I,u\rangle$ on the relation J is defined as $\{h(u)|h(I)\subseteq J\}$, denoted $\langle I,u\rangle(J)$. Apparently, the tuple in the relation $\langle I,u\rangle(J)$ is decided by the valuation h that satisfies $h(I)\subseteq J$. Generally, the number of the valuation h that satisfies $h(I)\subseteq J$ equals the number of tuples $h(u)$ in $\langle I,u\rangle(J)$. If $\langle I,u\rangle(J)=J$, we can say J satisfies the template dependency $\langle I,u\rangle$. Particularly, if $VAL(u)\subseteq VAL(I)$, J satisfies the total template dependency.

2 Implication Problem

The implication problem for dependencies is to decide whether a given dependency is logically implied by a given set of dependencies. This problem is recursively unsolvable in general[10,11], and is solvable but computationally intractable if all template dependencies are total[10,11].

For a set of dependencies D we denote by $SAT(D)$ the set of relations that satisfy all dependencies in D. D implies a dependency d, denoted $D|=d$, if $SAT(D)\subseteq SAT(d)$. That is, if d is satisfied for every relation which satisfies all dependencies in D. If $D|=d$ and $d|=D$, we say that D is equivalent to d, denoted $D|=|d$.

Assuming that $\langle I,u\rangle$ is the total template dependency and v is the tuple. We define $h(w[A])=\begin{cases} v[A] & w[A]=u[A] \\ w[A] & w[A]\ne u[A] \end{cases}$, $w\in I$, $A\in U$. Then h is the valuation on I and $h(I)$ is denoted by $I(u/v)$.

If $VAL(v)\cap VAL(I)=\varnothing$, h is apparently a one-to-one valuation on I.

Let $I=\{u_1,\ldots,u_l\}$ be a relation and $\langle J,v\rangle$ be a total template dependency. Defining $I(J,v)=\bigcup_{1\le i\le l}J_i(v_i/u_i)$, where J_i, v_i, $1\le i\le l$ satisfies: I) for any $1\le i\le l$ there exists a one-to-one valuation h_i on J such that $J_i=h_i(J)$, $v_i=h_i(v)$. II) $VAL(J_i)\cap VAL(I)=\varnothing$, $1\le i\le l$; $VAL(J_i)\cap VAL(J_j)\ne\varnothing$, $i\ne j$.

Lemma 1. Let $\langle I,u\rangle$ and $\langle J,v\rangle$ be total template dependencies. Then, for any relation K, we have $\langle I,u\rangle(\langle J,v\rangle(K))=\langle I(J,v),u\rangle(K)$ [12].

Lemma 2. Let the total join dependency $^*[R_1,...,R_l]$ and the total template dependency $\langle I,u\rangle=\langle\{u_1,...,u_l\},u\rangle$ have the relation $R_i=\{A|u[A]=u_i[A]\}$, $1\le i\le l$. Then, for any relation K, we have $\langle I,u\rangle(K)$=$^*[R_1,...,R_l](K)$ [5].

The following lemma is the key conclusion of the paper.

Lemma 3. Let both $^*[R_1,...,R_l]$ and $^*[S_1,...,S_m]$ be join dependencies. Then we have $\{^*[R_1,...,R_l],^*[S_1,...,S_m]\}$ $|=^*[R_1\cap S_1,...,R_1\cap S_m,...,R_l\cap S_m]$.

Proof. Constructing total template dependencies $\langle I,u\rangle$, $\langle J,v\rangle$, which satisfy $I=\{u_1,...,u_l\}$, $R_i=\{A|u_i[A]=u[A]\}$, $1\le i\le l$ and $J=\{v_1,...,v_m\}$, $S_j=\{B|v_j[B]=v[B]\}$, $1\le j\le m$.

We know from lemma 2 that for any relation K, $\langle I,u\rangle(K)$=$^*[R_1,...,R_l](K)$ and $\langle J,v\rangle(K)$=$^*[S_1,\dots,S_m](K)$.

If K satisfies the set $\{^*[R_1,...,R_l],^*[S_1,...,S_m]\}$ of dependencies, apparently $\langle I,u\rangle(\langle J,v\rangle(K))=\langle I,u\rangle(K)=K$. Then from lemma 1 we have $\langle I(J,v),u\rangle(K)=K$.

Constructing the join dependency $^*[T_1,...,T_n]$ that satisfies the condition that T_k is the attribute set in which u has the same value with the *k*-th tuple in $I(J,v)$.

From lemma 2 we know that $\langle I(J,v),u\rangle(K)$=$^*[T_1,...,T_n](K)$ for any relation K.

Let $J_i(v_i/u_i)=\{v_{1i},...,v_{mi}\}$. Assuming that the *k*-th tuple in $I(J,v)$ is the *j*-th tuple in $J_i(v_i/u_i)$, i.e. $T_k=\{C|v_{ji}[C]=u[C]\}$. From the structure of $J_i(v_i/u_i)$ we know that $T_k=\{C|v_{ji}[C]=u[C]\}=\{A|u_i[A]=u[A]\}$ $\cap\{B|v_{ji}(B)=u_i(B)\}=\{A|u_i[A]=u[A]\}\cap\{B|v_j(B)=v(B)\}=R_i\cap S_j$.

So K satisfies $^*[R_1\cap S_1,...,R_1\cap S_m,...,R_l\cap S_m]$.

Because K is arbitrarily taken, we get the conclusion of this lemma.

Lemma 4. Let $^*[R_1,...,R_l]$ and $^*[S_1,...,S_m]$ be total join dependencies. Then $^*[R_1,...,R_l]|=^*[S_1,...,S_m]$ iff for any $1\le i\le l$ there exists $1\le j\le m$ such that $R_i\subseteq S_j$ [6].

Corollary. Let $^*[R_1,...,R_l]$ be the total join dependency. If $R_i\subseteq R_j$ we have $^*[R_1,...,R_{i-1},R_i,R_{i+1},...,R_l]$ $|=|^*[R_1,...,R_{i-1},R_{i+1},...,R_l]$.

Assuming that JD_1=$^*[R_1,...,R_l]$ and JD_2=$^*[S_1,...,S_m]$, we denote $^*[R_1\cap S_1,...,R_1\cap S_m,...,R_l\cap S_m]$ by $JD_1\cap JD_2$.

The following theorem is the main result of the paper.

Theorem 1. Let $JD_1,JD_2,...,JD_r$ be the join dependencies. Then $\{JD_1,JD_2,...,JD_r\}|=|JD_1\cap JD_2\cap...\cap JD_r$.

Proof. We prove the conclusion by the method of induction.

From lemma 3 and lemma 4, we directly get $\{JD_1,JD_2\}|=|JD_1\cap JD_2$.

Let $\{JD_1,...,JD_{r-1}\}|=|JD_1\cap...\cap JD_{r-1}$, then $\{JD_1,...,JD_{r-1},JD_r\}|=|\{JD_1\cap...\cap JD_{r-1},JD_r\}|=|JD_1\cap...\cap JD_{r-1}$ $\cap JD_r$.

The above lemmas 1 and theorem 1 had respectively solved the implication problems of the total template dependencies and the total join dependencies. They are the foundation of our further discussion on the relationship between the template dependencies and the join dependencies.

3 Template Dependencies and Join Dependencies

We denote the set of the total template dependencies by S_T.

Assuming that $\langle I,u\rangle,\langle J,v\rangle\in S_T$. We say $\langle I,u\rangle$ and $\langle J,v\rangle$ satisfy the relation ρ, denoted $\langle I,u\rangle\rho\langle J,v\rangle$, If

$\langle I,u\rangle(K)=\langle J,v\rangle(K)$ for any relation K.

Apparently ρ is an equivalence relation on S_T. Consequently we obtain the quotient set S_T/ρ by the equivalence relation ρ. In S_T/ρ the equivalence class containing $\langle I,u\rangle$ will be denoted by $[\langle I,u\rangle]$.

For any $[\langle I,u\rangle],[\langle J,v\rangle]\in S_T/\rho$, the product of $[\langle I,u\rangle]$ and $[\langle J,v\rangle]$ is defined as $[\langle I,u\rangle]\circ[\langle J,v\rangle]=[\langle I(J,v),u\rangle]$.

Let $\langle I_0,u_0\rangle\in[\langle I,u\rangle],\langle J_0,v_0\rangle\in[\langle J,v\rangle]$. Then $\langle I_0,u_0\rangle(\langle J_0,v_0\rangle(K))=\langle I,u\rangle(\langle J,v\rangle(K))$ for any relation K. Thus we have $[\langle I_0,u_0\rangle]\circ[\langle J_0,v_0\rangle]=[\langle I_0(J_0,v_0),u_0\rangle]=[\langle I(J,v),u\rangle]=[\langle I,u\rangle]\circ[\langle J,v\rangle]$, which means that the multiplication $\circ$ is well defined.

Let M be a nonempty set and $\circ$ be a binary operation defined on M. We say M constructs the monoid by the multiplication $\circ$, If I) $a\circ b\in M$ for any $a,b\in M$; II) $(a\circ b)\circ c=a\circ(b\circ c)$ for any $a,b,c\in M$; III) there exists $e\in M$, such that $a\circ e=e\circ a=a$ for any $a\in M$.

Theorem 2. The quotient set S_T/ρ constructs monoid by the multiplication $\circ$.

Proof. From the definition of the total template dependency we know that $\langle w,w\rangle(K)=K$ for any relation K, where w is a tuple. So $[\langle w,w\rangle]\circ[\langle I,u\rangle]=[\langle I,u\rangle]\circ[\langle w,w\rangle]=[\langle I,u\rangle]$ for any $[\langle I,u\rangle]\in S_T/\rho$, which means that $[\langle w,w\rangle]$ is the unit element of S_T/ρ.

Assuming that $[\langle H,t\rangle],[\langle I,u\rangle],[\langle J,v\rangle]\in S_T/\rho$. Then for any relation K, $\langle H(I(J,v),u),t\rangle(K)=\langle H,t\rangle(\langle I(J,v),u\rangle(K))=\langle H,t\rangle(\langle I,u\rangle(\langle J,v\rangle(K)))$ and $\langle H(I,u)(J,v),t\rangle(K)=\langle H(I,u),t\rangle(\langle J,v\rangle(K))=\langle H,t\rangle(\langle I,u\rangle(\langle J,v\rangle(K)))$.

Thus we have $[\langle H,t\rangle]\circ([\langle I,u\rangle]\circ[\langle J,v\rangle])=([\langle H,t\rangle]\circ[\langle I,u\rangle])\circ[\langle J,v\rangle]$ which means that $\circ$ satisfies the associativity. Consequently S_T/ρ constructs the monoid by $\circ$.

We denote the set of total join dependencies by S_J, and then we similarly define an equivalence relation σ on S_J and a binary operation $\bullet$ on S_J/σ such that S_J/σ construct the monoid by $\bullet$.

Let $^*[R_1,\dots,R_l],{}^*[S_1,\dots,S_m]\in S_J$. If $^*[R_1,\dots,R_l](K)={}^*[S_1,\dots,S_m](K)$ for any relation K, we say that $^*[R_1,\dots,R_l]$ and $^*[S_1,\dots,S_m]$ satisfy the relation σ, denoted $^*[R_1,\dots,R_l]\sigma{}^*[S_1,\dots,S_m]$. It is evident that σ is an equivalence relation.

In S_J/σ the equivalence class containing $^*[R_1,\dots,R_l]$ will be denoted by $[^*[R_1,\dots,R_l]]$.

For any $[^*[R_1,\dots,R_l]],[^*[S_1,\dots,S_m]]\in S_J/\sigma$, the product $[^*[R_1,\dots,R_l]]\bullet[^*[S_1,\dots,S_m]]$ of $[^*[R_1,\dots,R_l]]$ and $[^*[S_1,\dots,S_m]]$ is defined as $[^*[R_1\cap S_1,\dots,R_l\cap S_1,\dots,R_l\cap S_m]]$, where $R_i\cap S_j$ means the intersection of attributes set R_i and S_j.

Theorem 3. S_J/σ is isomorphic to S_T/ρ.

Proof. If $\langle I,u\rangle(K)={}^*[R_1,\dots,R_l](K)$ for any relation K, we will define $f([\langle I,u\rangle])=[^*[R_1,\dots,R_l]]$.

Assuming that $f([\langle I,u\rangle])=[^*[R_1,\dots,R_l]]$ and $f([\langle J,v\rangle])=[^*[S_1,\dots,S_m]]$.

If $[^*[R_1,\dots,R_l]]\neq[^*[S_1,\dots,S_m]]$, then there exists a relation K such that $^*[R_1,\dots,R_l](K)\neq{}^*[S_1,\dots,S_m](K)$. From the definition of f and lemma 2 we have $\langle I,u\rangle(K)\neq\langle J,v\rangle(K)$, and consequently $[\langle I,u\rangle]\neq[\langle J,v\rangle]$.

Thus f is well defined.

For any $[^*[R_1,\dots,R_l]]\in S_J/\sigma$, apparently there exists $[\langle I,u\rangle]\in S_T/\rho$ such that $I=\{u_1,\dots,u_l\}$ and $R_i=\{A|u[A]=u_i[A]\}$, $1\le i\le l$. Thus we have $f([\langle I,u\rangle])=[^*[R_1,\dots,R_l]]$. That means that f is a surjection.

Let $f([\langle I,u\rangle])=[^*[R_1,\dots,R_l]]$ and $f([\langle J,v\rangle])=[^*[S_1,\dots,S_m]]$. If $[\langle I,u\rangle]\neq[\langle J,v\rangle]$, there exists a relation K such that $\langle I,u\rangle(K)\neq\langle J,v\rangle(K)$. So $^*[R_1,\dots,R_l](K)\neq{}^*[S_1,\dots,S_m](K)$. Consequently, $[^*[R_1,\dots,R_l]]\neq[^*[S_1,\dots,S_m]]$. That means that f is an injection.

Assuming that $f([\langle I,u\rangle])=[^*[R_1,\dots,R_l]]$ and $f([\langle J,v\rangle])=[^*[S_1,\dots,S_m]]$. From the proof procedure of lemma 3 we know that $f\left(\left[\langle I,u\rangle\right]\circ\left[\langle J,v\rangle\right]\right)=f\left(\left[\langle I(J,v),u\rangle\right]\right)=\left[{}^{*}\left[R_1\cap S_1,...,R_l\cap S_1,...,R_l\cap S_m\right]\right]=\left[{}^{*}\left[R_1,...,R_l\right]\right]\bullet\left[{}^{*}\left[S_1,...,S_m\right]\right]$ $=f\left(\left[\langle I,u\rangle\right]\right)\bullet f\left(\left[\langle J,v\rangle\right]\right)$.

In conclusion, f is the isomorphic mapping between S_T/ρ and S_J/σ.

This conclusion shows that the class of total template dependencies is essentially identical to the class of total join dependencies.

4 Discussion

We can further obtain the interesting result as follows by use of the algebraic properties of the total join dependencies.

Let $R_i\subseteq U$, $1\le i\le l$ and $\bigcup_{1\le i\le l}R_i=U$. If the hypergraph taken U as a vertex set and $R_1,R_2,...,R_l$ as edges is an acyclic hypergraph, we say that ${}^{*}\left[R_1,...,R_l\right]$ is a total acyclic join dependency.

Lemma 5. A total join dependency is a total acyclic iff the total join dependency is logically equivalent to a set of multivalued dependencies[13].

From abstract algebra we know that all element taken shape ${}^{*}\left[R_1,R_2\right]$ in S_J, namely the multivalued dependencies subset of S_J, generates a submonoid of S_J. Apparently a multivalued dependency is a total acyclic join dependency. Thus we have the conclusion as follows.

Theorem 4. All total acyclic join dependencies in S_J constitute a proper submonoid of S_J.

The total acyclic join dependencies have many good properties[14]. These properties get attracting more and more researcher. Some properties had been applied to the design of database schemes. After all, it is a new exploration to investigate the properties of data dependencies by means of the abstract algebra. We hope that the method will be extended.

In fact, the above results can immediately play role in designing relational database schemes.

References:

[1] Codd, E.F. Further normalization of the database relational model. In: Rustin, R., ed. Database Systems. Englewood Cliffs, NJ: Prentice-Hall, 1972. 33~64.

[2] Delobel, C. Semantics of relations and the decomposition process in the relational data model. ACM Transactions on Database Systems, 1978,3:201~222.

[3] Fagin, R. Multivalued dependencies and a new normal form for relational databases. ACM Transactions on Database Systems, 1977, 2:262~278.

[4] Rissanen, J. Theory of relations for databases —— a tutorial survey. Lecture Notes in Computer Science, Vol. 64, 1978. 537~551.

[5] Beeri, C., Vardi, M.Y. Formal systems for join dependencies. Theoretic Computer Science, 1980,10:62~83.

[6] Beeri, C., Vardi, M.Y. On the properties of join dependencies. In: Gallaire, H., *et al*., eds. Advances in Database Theory. New York: Plenum Publishing Company, 1981. 25~72.

[7] Fagin R., Vardi M.Y. The theory of data dependencies —— a survey. Proceedings of Symposia in Applied Mathematics, Vol.34, 1986. 19~71.

[8] Gorchinskaya, O.Y. Join dependencies in relational database design. Information Science, 1989,47:372~387.

[9] Sardi, F., Ullman, J.D. Template dependencies: a large class of dependencies in relational databases and their complete axiomatization. Journal of ACM, 1981,29:363~372.

[10] Beeri, C., Vardi, M.Y. The implication problem for data dependencies. Lecture Notes in Computer Science, Vol. 115, 1981. 73~85.

[11] Chandra, A.K., Lewis, H.R., Makowsky, J.A. Embedded implication dependencies and their inference problem. In: Proceedings of the 13th ACM Annual Symposium on Theory of Computing. 1981. 342~354.

[12] Beeri, C., Vardi, M.Y. Formal systems for tuple and equality generating dependencies. SIAM Journal on Computing, 1984,13: 76~98.

[13] Fagin, R., Mendelzon, A.O., Ullman, J.D. A simplified universal relation assumption and its properties. ACM Transactions on Database Systems, 1982,7:343~360.

[14] Beeri, C., Fagin, R., Maier, D. On the desirability of acyclic database schemes. Journal of ACM, 1983,29:479~514.

样本依赖与连接依赖的等价性

李星野， 王书宁， 岳占峰

(清华大学 自动化系,北京 100084)

摘要: 以抽象代数为工具,探索了全样本依赖与全连接依赖之间的关系.首先,分别在全样本依赖集和全连接依赖集上建立等价关系,这两种等价关系都将作用相同的依赖视为等价依赖.然后证明了在这两个等价关系下的商集分别构成么半群,并且这两个么半群是同构的.这就等于证明了全样本依赖类本质上等同于全连接依赖类.最后给出了一个关于全无环连接依赖的有趣结果.有关结果可以在关系数据库的设计中发挥积极作用.

关键词: 样本依赖;连接依赖;蕴含问题;同构;么半群

中图法分类号: TP311 文献标识码: A

SIGMOD / PODS 2003

San Diego, California June 9~11, 2003

PODS 2003 Call for Papers

http://www.db.ucsd.edu/SIGMODPODS03/PODScfp.html

The 22nd ACM SIGACT-SIGMOD-SIGART Symp. on Principles of Database Systems (PODS)

The symposium invites papers on fundamental aspects of databases. Original research papers on the theory, design, specification, or implementation of databases are solicited. Papers emphasizing new topics or foundations of emerging areas are especially welcome. The symposium will be held in Town and Country Hotel in San Diego, in conjunction with the ACM SIGMOD International Conference on Management of Data (SIGMOD), and as part of the Federated Computing Research Conference (FCRC'03).

Topics: Suggested topics include the following (this list is not exhaustive and the order does not reflect priorities): Access Methods and Physical Design; Complexity and Performance; Concurrency Control Evaluation; Transaction Management; Integrity and Security; Data Models; Logic in Databases; Query Languages; Query Optimization; Database Programming Languages; Database Updates; Active Databases; Deductive Databases and Knowledge Bases; Object-oriented Databases; Multimedia Databases; Spatial and Temporal Databases; Constraint Databases; Real-Time Databases; Distributed Databases; Data Integration and Interoperability; Views and Warehousing; Data Mining; Databases and Information Retrieval; Semistructured Data and XML; Information Processing; Databases and Workflows.

Important Dates:

* November 15, 2002 - Paper titles and short abstracts due.

* November 22, 2002 - Papers due.

* February 11, 2003 - Notification of acceptance/rejection.

* March 8, 2003 - Camera-ready due.

More information can be achieved at the homepage.