

SISTEMA DE BASES DE DATOS INFERENCIALES ORIENTADAS A PROBLEMAS

*MSc. Laureano Rodríguez Corvea
Lic. Ida J. Rodríguez - Amaya Fernández¹
Dra. Luisa M. González González²
Dr. Ramiro Pérez Vázquez
Dr. Rogelio Silverio Castro*

RESUMEN

En el trabajo hacemos un estudio de las limitaciones de los sistemas de gestión de base de datos (**SGBD**) y de los conceptos fundamentales de la teoría de las demostraciones, la cual constituye la base teórica de los SGBD deductivos, que son los que superan en cierta forma las limitantes de los SGBD tradicionales. Además se describe la herramienta que se diseñó e implementó para poder hacer consultas deductivas utilizando una estrategia de acoplamiento débil, en los SGBD de la familia xBase.

INTRODUCCION

En el campo de las bases de datos ha surgido tendencia dirigida a los "Sistemas de Bases de Datos basados en la lógica". En la literatura actual se encuentran con FRECUENCIA expresiones como "base de datos lógica", "SGBD inferencial", "SGBD experto", "SGBD deductivo", "sistema de administración de bases de conocimientos", "lógica como modelo de datos", "procesamiento recursivo de consultas", etc. Pero no siempre es fácil relacionar esos términos con los términos y conceptos habituales de las bases de datos.

Las investigaciones sobre la relación entre la teoría de las bases de datos y la lógica se remontan por lo menos a finales de la década de 1970, si no antes, sin embargo, el estímulo principal para el reciente aumento de interés por el tema parece haber sido la publicación en 1984 de un artículo de **Reiter** que marcó rumbos. En este artículo Reiter caracterizaba la percepción tradicional de los sistemas de bases de datos según la teoría de modelos. A continuación Reiter argumentó que existía también la posibilidad de una perspectiva según la teoría de demostraciones, la cual era de hecho preferible en algunos aspectos. En esta perspectiva la base de datos se contempla como un conjunto de axiomas y la ejecución de una consulta se considera como la demostración de que una fórmula especificada es un teorema.

Un SGBD se considera deductivo si mejora la base de datos desde la perspectiva de la teoría de las demostraciones. Los axiomas deductivos o reglas de inferencia, junto con las restricciones de integridad, constituyen lo que en ocasiones se denomina base de datos intensional y ésta junto con la base de datos extensional forman lo que suele llamarse base de datos deductiva. Este término no es muy adecuado pues es SGBD,

¹ Facultad de Ciencias Médicas de Sancti Spiritus, Cuba. Olivos III, Sancti Spiritus. Cuba.

² Universidad Central de Las Villas, Cuba. Carretera de Camajuaní, Km 8. Santa Clara. Villa Clara. Cuba.

no la base de datos, es el que se ocupa de hacer deducciones.

En los SGBD convencionales no existen posibilidades de recuperación deductiva, aunque se conocen intentos para dotar a estos sistemas de tales posibilidades. Un ejemplo de ello es el sistema **DEDUCE** (Chang (1976) y (1978)) en el que es posible declarar reglas de inferencia, restricciones de integridad y consultas a las que se responde deductivamente.

En nuestro medio estos sistemas no son definitivos y cuando se desea hacer algún sistema que necesita una consulta deductiva no se integra a los SGBD. Es decir, se manipulan los datos con un SGBD convencional y las posibles consultas se hacen desde fuera. En otras palabras no cuentan los especialistas con SGBD deductivos.

Por otra parte los SGBD convencionales dan pocas facilidades para especificar reglas de integridad semántica y la verificación de éstas corre a cuenta de los especialistas que se encargan de diseñar los sistemas de bases de datos.

Es importante, entonces, adelantarse en las especificaciones de los SGBD deductivos y comenzar a desarrollar herramientas para, o bien incorporarlas a SGBD convencionales o utilizarlas en la creación de un SGBD, de forma que permitan realizar consultas deductivas y manejar de manera natural reglas de integridad semánticas.

En esta integración se pueden utilizar dos estrategias.

La primera es la que brinda un acoplamiento débil entre el SGBD y el lenguaje lógico; éste es el caso en que se toma un SGBD y un

sistema de lenguaje lógico de programación ya existente y se crea una interfaz de comunicación entre los dos. El usuario está consciente de la existencia de dos sistemas distintos. Esta estrategia se puede realizar en forma más directa.

En la segunda estrategia se tiene un acoplamiento fuerte, es decir, existe una verdadera integración del SGBD y el sistema lógico de programación, o sea, el lenguaje del SGBD incluye el manejo directo de las operaciones de influencia lógica; así, el usuario se enfrentará a un solo lenguaje, no a dos. En este caso el diseño e implementación tienen más dificultades pero se puede lograr un mejor desempeño (optimización de consultas, por ejemplo).

DESARROLLO

Limitaciones de los sistemas de gestión de bases de datos existentes.

El diseño de cualquier aplicación de base de datos implica generalmente disponer de facilidades para realizar las siguientes tareas:

- Especificación del esquema conceptual.
- Especificación de los requerimientos de entrada/salida.
- Especificación del esquema lógico de la base de datos.
- Especificación de la estructura física de almacenamiento.
- Diseño del subsistema de privacidad.
- Diseño del subsistema de integridad.
- Diseño del subsistema de tratamiento de fallos por hardware u otros.
- Diseño de los programas de entrada/salida y de los lenguajes de consulta.

Como se puede observar, este proceso es complejo y engorroso; afortunadamente las aplicaciones en bases de datos no necesitan

todas estas facilidades puesto que podemos utilizar un sistema de gestión de bases de datos que automatiza muchas de ellas.

Los componentes principales de un sistema de gestión de base de datos son:

- Lenguaje de definición de datos.
- Sistema de diccionario de datos para el análisis de la información.
- Lenguaje de especificación de paquetes de entrada/salida.
- Módulo de transformación lógica - física.
- Subsistema de privacidad de propósito general.
- Subsistema de integridad de propósito general.
- Generador de programas de aplicación.
- Generador de reportes.
- Lenguaje de consulta de propósito general.

Desafortunadamente los sistemas de gestión de base de datos disponibles no desarrollan completamente todas las componentes descritas anteriormente. Aunque se investiga intensamente en la obtención de nuevos SGBD cada vez más potentes, aún se reconoce que usualmente tienen las siguientes limitaciones:

La independencia de datos no está garantizada

Por independencia de datos se entiende hasta qué punto la estructura física del almacenamiento de un sistema de base de datos es independiente de los programas de aplicación (Ullman, 1982).

La ventaja principal de la independencia de datos es que las bases de datos pueden ampliarse o trasladarse a una nueva estructura física de almacenamiento que produce un efecto mínimo sobre la salida del sistema.

Existen dos niveles de independencia de los datos:

- * **Independencia física de los datos.**
- * **Independencia lógica de los datos.**

A continuación explicaremos estos aspectos:

En una aplicación de base de datos bien diseñada, el esquema físico puede ser modificado sin alterar el esquema conceptual ni requerir una redefinición de subesquemas. Esta independencia se refiere precisamente a la independencia física de datos. Se debe comprender que las modificaciones a la organización física de las bases de datos puede afectar la eficiencia de los programas de aplicación pero nunca será necesario reescribir estos programas.

Otro tipo de independencia deseable es la independencia lógica, la que se logra cuando cambios de esquema conceptual y sus visiones tampoco afectan los programas de aplicación.

La recuperación deductiva no está garantizada

Solamente pocos sistemas de manipulación de Bases de Datos disponibles en la actualidad proporcionan medios de recuperación deductiva que requieran la clausura transitiva.

Las restricciones de integridad no están garantizadas

Todo SGBD se diseña con base en un modelo de datos, esto es, con base en una colección de conceptos bien definidos que ayudan a considerar y expresar las propiedades estáticas y dinámicas de una aplicación. La mayoría de los modelos de datos han evolucionado intuitivamente; no obstante, se supone que deben caracterizarse por proporcionar medio para expresar:

- * propiedades estáticas: objetos, atributos y relaciones entre objetos.
- * propiedades dinámicas: operaciones sobre los objetos, propiedades de estas operaciones, así como interrelaciones entre las mismas (transiciones de estado).
- * reglas de integridad (tanto para los objetos como sus operaciones): éstas describen estados y transiciones de estados de la base de datos válidas.

Las restricciones son reglas usadas para definir las propiedades estáticas y dinámicas de una aplicación. Una restricción frecuente relaciona ambas propiedades (Brodie, 1986).

Se reconocen tres tipos de restricciones:

- *Inherentes al modelo*: Son las propiedades que son representadas directamente y que no pueden ser violadas. Por ejemplo, las restricciones entre objetos en el modelo de datos jerárquico deber constituir una jerarquía.
- *Explícitas*: Son las propiedades que deben ser explícitamente definidas usando los mecanismos proporcionados por el modelo de datos. Ejemplo: la estructura de los objetos, propiedades específicas de los mismos, etc.
- *Implícitas*: Son las propiedades que se derivan como consecuencia de las otras. Por ejemplo, inferencias lógicas pueden ser deducidas de un conjunto de hechos y de un conjunto de dependencias funcionales.

Sin una definición formal de un modelo de datos es difícil identificar o analizar las restricciones de integridad. Numerosas investigaciones sobre modelos de datos han intentado desarrollar medio para expresar las restricciones con una uniformidad tal que permitan su definición, análisis y modificación. También se han hecho

esfuerzos considerables dirigidos a realizar diversos tipos de chequeos con el menor costo de ejecución posible. Uno de los medios utilizados para alcanzar este objetivo ha sido intentar que la mayor parte de las restricciones sean inherentes al modelo (se chequean automáticamente); una segunda tendencia ha sido integrar datos y conceptos procedimentales en mecanismos uniformes usando lenguajes de programación existentes y resultados de la inteligencia artificial.

A pesar de los esfuerzos mencionados, los SGBD comercialmente disponibles frecuentemente no satisfacen en forma adecuada estos requisitos.

Otras limitaciones de los sistemas disponibles en la actualidad consisten en que no utilizan el esquema conceptual como una componente integrada a los programas de aplicación o en que los propios sistemas de gestión de bases de datos no permiten la verificación de la integridad en forma automática ni la realización de determinadas inferencias.

Las bases de datos desde la perspectiva de la teoría de las demostraciones

Como se dijo antes, las investigaciones sobre la relación entre la teoría de las bases de datos y la lógica se remontan por lo menos a finales de la década de 1970, si no antes (Gallaire (1978), Minker (1988) y Dahl (1982)). El artículo publicado por Reiter en 1984 caracterizaba la percepción tradicional de las sistemas de bases de datos como según la teoría de modelos, con lo cual quería decir, en términos muy informales, que la base de datos se contempla como un conjunto de relaciones explícitas, cada una de las cuales contiene un conjunto de tuplas explícitas, y la ejecución de una consulta puede considerarse como la evaluación de una

fórmula especificada (o sea, una expresión con valor lógico) sobre esas relaciones y tuplas explícitas.

A continuación, Reiter argumentó que existía también la posibilidad de una perspectiva según la teoría de demostraciones, la cual de hecho era preferible en algunos aspectos. Desde esa otra perspectiva - una vez más, en términos muy informales - la base de datos se contempla como un conjunto de axiomas (axiomas "fundamentales", correspondientes a las tuplas de las relaciones base, junto con ciertos axiomas "deductivos"), y la ejecución de una consulta se considera como la demostración de que una fórmula especificada es una consecuencia de esos axiomas; en otras palabras, la demostración de que es un teorema.

Como se sabe, una cláusula es una expresión de la forma:

$A_1 \text{ AND } A_2 \text{ AND } \dots \text{ AND } A_m \Rightarrow B_1 \text{ OR } B_2 \text{ OR } \dots \text{ OR } B_n$ donde A_i y B_i son todas términos de la forma

$$R(x_1, x_2, \dots, x_i)$$

Aquí R es un predicado (relación) y x_1, x_2, \dots, x_i son los argumentos de ese predicado (tuplas de la relación). De acuerdo con la referencia Gallaire (1984), consideramos seguidamente dos casos especiales importantes de esta construcción general.

Caso 1

$m = 0, n = 1$. En este caso la cláusula puede simplificarse a

- B_1 en otras palabras (desechando el símbolo de implicación) a $R(x_1, x_2, \dots, x_i)$

para algún predicado r y algún conjunto de argumentos x_1, x_2, \dots, x_i . Si las x_i son constantes, la cláusula representa un axioma fundamental, es decir, declara algo que es verdad en forma inequívoca. En términos de bases de datos, una proposición corresponde a una tupla de alguna relación.

Caso 2

$m > 0, n = 1$. En este caso la cláusula adopta la forma

$$A_1 \text{ AND } A_2 \dots \text{ AND } A_m \Rightarrow B$$

La que puede considerarse como un axioma deductivo; presenta una definición (parcial) del predicado ubicado a la derecha del símbolo de implicación en términos de los predicados de la izquierda.

Las tuplas de las relaciones ("axiomas fundamentales"), las relaciones derivadas ("axiomas deductivos") y las restricciones de integridad se pueden considerar como casos especiales de la construcción general de cláusulas. Ahora trataremos de ver cómo estas ideas pueden conducir a la perspectiva "según la teoría de las demostraciones" de las bases de datos.

En primer lugar, la perspectiva tradicional de una base de datos puede considerarse como una perspectiva según la teoría de los modelos. Con "perspectiva tradicional", nos referimos tan sólo a un punto de vista desde el cual la base de datos se percibe como un conjunto de relaciones nombradas en forma explícita (relación base) cada una de las cuales contiene un conjunto de tuplas explícitas, junto con un conjunto explícito de restricciones de integridad que dichas tuplas no deben violar.

El procedimiento de consultas en la perspectiva según la teoría de modelos es en esencia un proceso de evaluar una cierta

FBF abierta para descubrir cuáles valores de las variables libres de esa FBF hacen que la FBF produzca un valor verdadero en el modelo (Gardarin, 1989).

Para aplicar las reglas de inferencia se hace necesario adoptar un perspectiva diferente, una en la cual la base de datos se considere en forma explícita como una teoría lógica, es decir, como un conjunto de axiomas. Así, el "significado" de las base se convierte, precisamente, en el conjunto de todas las proposiciones verdaderas deducibles de los axiomas en todas sus combinaciones posibles, es decir, es el conjunto de teoremas que pueden demostrarse a partir de esos axiomas. Esta es la perspectiva según la teoría de las demostraciones. Desde este punto de vista, la evaluación de consulta se convierte en un proceso de demostración de teoremas (al menos en lo conceptual; no obstante, en aras de la eficiencia, es probable que el sistema utilice técnicas más convencionales para el procesamiento de consultas).

Sistemas de bases de datos deductivas

Un SGBD deductivo es un SGBD que maneja la perspectiva según la teoría de las demostraciones de una base de datos, y en particular que es capaz de deducir hechos adicionales a partir de "bases de datos extensionales" (es decir la relación base) aplicando a esos hechos axiomas deductivos o reglas de inferencia especificadas. Los axiomas deductivos junto con las restricciones de integridad constituyen lo que en ocasiones se denomina base de datos intensional.

Como se acaba de indicar, los axiomas deductivos forman parte de la base de datos intensional. La otra parte consiste en axiomas adicionales para representar las restricciones

de integridad (o sea, reglas cuyo propósito principal es restringir las actualizaciones, aunque de hecho tales reglas se pueden emplear también en el proceso de deducción para generar nuevos datos).

Los SGBD deductivos aplicarán por lo regular sus deducciones a las bases de datos convencionales ya existentes y construidas de la manera convencional. Adviértase, que ahora es más importante que nunca que la base de datos no viole ninguna de las restricciones de integridad declaradas, porque una base de datos que viole cualquiera de estas restricciones representará (en términos lógicos) un conjunto inconsistente de axiomas, y es de sobra conocido que ninguna proposición en absoluto se puede demostrar partiendo de semejante base (dicho de otro modo, pueden derivarse contradicciones). Por esa misma razón también es importante que el conjunto declarado de restricciones de integridad sea consistente.

Entre las restricciones más frecuentes se encuentran las siguientes:

- Restricciones de dominio. El valor de cada atributo debe corresponderse con el dominio que se ha definido para el mismo, procedimiento conocido como validación.
- Restricciones de clave primaria. Si uno o varios atributos constituyen una clave primaria entonces debe cumplirse que si dos tuplas tienen iguales valores para dichos atributos, los valores para los restantes atributos también deben coincidir.
- Restricciones de claves ajenas. Verificar un estado consistente para determinadas relaciones en que alguno de sus atributos sea a su vez la clave primaria de otra. No tendría sentido un uso no controlado de este tipo de restricción.

- Restricciones propias del sistema. Son restricciones inherentes al problema al que se da solución.

Descripción del sistema diseñado

El sistema que se discute en este trabajo constituye una primera aproximación al problema por resolver, por lo que se utilizó la estrategia del acoplamiento débil. Para realizar las inferencias usamos el método de Resolución.

Lenguaje de consultas

Se construyó en primer lugar un lenguaje mediante el cual se pudieran hacer consultas deductivas a una base de datos relacional. La gramática de dicho lenguaje se muestra a continuación. Ella describe solicitudes para la recuperación de información de un artículo en forma similar al modelo relacional. Por ejemplo, los operadores lógicos que permite son: .AND., .OR., .NOT.

```

<regla> ::= <expresión> ⇒ <functor>
<expresión> ::= <factor> | <factor> .AND.
<expresión> | <factor> .OR. <expresión>
<factor> ::= <functor> | <functor>
{ <op.lógico> <relación> }
<relación> ::= <identificador>
<op.relación> <identificador> |
<identificador> <op.relación> <contante>

<functor> ::= <identificador>
[ <lista de campos> ]
<lista de campos> ::= <identificador> |
<identificador>, <lista de campos>
<identificador> ::= <letra> í<letra> |
<dígito> }
<constante> ::= <número entero> |
<número real> | <cadena>

```

Es importante señalar que lo que aquí se denomina “functor” o “cabecera” no es más

que el nombre de las relaciones de la base de datos, seguido por los atributos seleccionados de las mismas, encerradas entre corchetes.

Para escribir correctamente las reglas deductivas hay que tomar en consideración, además, las siguientes reglas semánticas:

1. Los nombres de los predicados de la izquierda deben corresponder con los nombres que existan.
2. El nombre del predicado de la derecha es el nombre de la relación donde se almacenará el resultado de la consulta.
3. Los atributos del predicado de la derecha deben haber aparecido como atributos de algún predicado de la izquierda.
4. Si en una regla aparece el mismo nombre predicado en varias ocasiones, los atributos en todos los casos tiene que tener el mismo significado semántico.
5. Los elementos de las expresiones lógicas que acompañan a los predicados deben ser atributos de éstos.

Generador de consultas

A pesar de haberse diseñado una gramática para escribir las consultas deductivas que se desean hacer a una base de datos, se utilizaron las herramientas que tiene incorporado el Lenguaje Arity/Prolog para hacer que el usuario entre estas reglas de forma más cómoda y no tenga que escribirlas completamente [Arity, 1988].

Generador de deducciones

Para poder realizar una deducción, en primer lugar es necesario convertir los artículos de los ficheros de datos en hechos Prolog, posteriormente es necesario convertir la regla entrada por el usuario en una cláusula Prolog. Tanto los hechos como la cláusula se

guardan en el fichero de extensión IDB que utiliza el Prolog.

Por último se toma la cabeza de la cláusula y se invoca. Esto desencadena el mecanismo del Prolog que se encarga de hacer las deducciones. Cuando cada una de las deducciones es completada se añade un nuevo artículo al fichero resultado.

Para la realización de este proceso es necesario utilizar predicados externos creados en el Lenguaje MicroSoft C que permitan:

- Tener acceso a los ficheros de la base de datos y convertir cada uno de los artículos con los campos seleccionados, en hechos en Prolog.
- Crear el fichero de datos de los resultados también de la familia xBase.

Explotación del sistema

El objetivo general del sistema consiste en hacer consultas a la base de datos extensional de la familia xBase, demostrando que la fórmula generada de dicha consulta es una consecuencia lógica de los axiomas fundamentales.

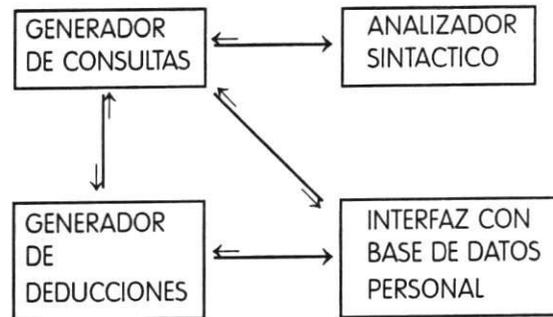
Para explotar el sistema se necesita una micro-computadora IBM o compatible con 640 Kb de memoria interna. Para la ejecución del sistema se requiere que en un mismo directorio se encuentren los ficheros ejecutables del sistema y el fichero IDB, generados por el compilador del Arity/Prolog que contiene el ejecutable y el fichero de datos interno.

Breve descripción de la implementación

Como se mencionó anteriormente, el sistema está programado en el lenguaje Arity/Prolog,

ya que el mismo se basa en el método de resolución y éste es, precisamente, el método utilizado para deducir si una fórmula generada de una consulta es consecuencia lógica de los axiomas fundamentales.

En la figura siguiente se muestra la arquitectura del sistema.



El funcionamiento general es como sigue: el generador de consultas capta una petición del usuario; para ello utiliza la interfaz con la base de datos extensional (se hace acceso a la información sobre la estructura de la base). Cuando completa la regla la envía al analizador sintáctico para probar su validez. Luego de comprobar que la regla está bien escrita desde el punto de vista sintáctico y semántico, la envía al generador de deducciones que se encarga, como su nombre la indica, de hacer las inferencias correspondientes. Para ello es necesario hacer acceso a la base de datos y esto se hace a través de la interfaz con la misma.

CONCLUSIONES

- En este trabajo se implementó una herramienta para poder hacer consultas a partir de SGBD del tipo xBase. Dicho módulo se obtuvo como ejecutable por lo que puede ser llamado desde cualquier sistema que se implemente en estos gestores.

- Se utilizó la herramienta desarrollada en una aplicación concreta de las Ciencias Médicas: complicación del recién nacido en partos gemelares, Hospital Materno Infantil de Sancti Spiritus, Cuba.

BIBLIOGRAFIA

- ABRIAL, J. R: "Data semantica". In Klimbie, J, W. K. and Koffeman, K.L. Data Management Systems. Amsterdam; North Holland. 1974.
- AGRAWAL Rakezh: "Alpha An Extension of Regional Algebra to Express a Class of Recursive Queries". IEEE Trans. on Software Eng. Vol. 14. No. 7, 1988.
- Arity Corporation. Arity/Prolog. Interpret and Compiler. Donirno Drive. Concord M.A.. Development Corporation. 1988.
- Arity Corporation. The Arity/Prolog. Language Reference Manual. Donirno Drive. Concord M.A. Development Corporation. 1987.
- Base de Datos y Sistemas Expertos. La Habana. 1991.
- BRODIE Michael L. Mylopoulos J.: "On Conceptual Modelling". Springer Verlag USA. 1984.
- CHANG, C.L. "DEDUCE: a deductive query laguage for relational datas bases". In Chen, C.H. Pattern Recognition and Artificial Intelligence. New York: Academic Precess. 1976.
- CHANG, C. L. "DEDUCE II: further inetigations of deduction n relational data bases". IBM Research Report RJ2 147. San Jose California. 1978.
- DATE C. J.: "Introducción a los sistemas de Bases de Datos" Addison - Wesley Iberoamericana, S.A. E.U.A. 1990.
- DAHL, Véronica: "On Database Systems Development Throug Logic". ACM TODS 7, número 1, Marzo de 1982.
- FLORENTIN, J. J. "Consistency auditing of databases". The Computer Journal 17, 52 - 58. 1974.
- GALLAIRE H. y J. MINKES. Logic and Data Bases. Nueva York N.Y.: Plenum Publishing Corp. 1978.
- GARCÍA Luciano, O. PADRÓN, R. MARIANO. "Programación Lógica". Universidad de la Habana. 1991.
- GARDARIN, Goerges y Petrik VALDURIEZ. Relational Databases an d Knowledge Bases Reading Mass.: Addison - Wesley. 1989.
- KOWALSKI, Robert. "Logic for Problem Solving". New York: Elsevier, North Holland. 1992.
- LOZINSHII, Eliezer, L. "A Problem - Oriented Inferential Databases System". ACM TODS 11. Núm. 3. 1986.
- MINKER Jack (de). "Foundations of Deductive Databases and Logic Programming". San Mateo, Calif. : Morgan Kaufmann. 1988.
- REITER, Raymond. "Towards a Logical Reconstruction of Relational Database Theory". On Conceptual Modelling: Perepectives from Artificial Intelligence, Databases, and Prgramming Languages (Editores, Michael L. Brodie, John Mylopoulos y Joachim W. Schimidit) Nueva York, N.Y.: Springer - Verlag. 1984.
- STONEBRAKER, Michael.: "Intruction to Integration of Knowledge an d Data Management". En M. Stonebraker (de), Reading in Database System, San Mateo Calif.: Morgan Kaufmann.
- ULLMAN, Jeffrey D. "Database and Knowledge - Base System". Rockville, Md.: Computer Science Press. 1989.
- ULLMAN, Jeffrey D.: "Principles of Database System". Copyright Computer Science Press. 1982. Secon Edcition.
- WINSTON, Prattrick, H. "Prolog Programming For Artificial Intelligence". Cambridge, Massachusetts. 1986.