Universidad Católica

TAI 2
Bases de Datos Deductivas

49314

Matías Zubiri

Índice

Introducción

En las siguientes páginas se hará referencia a las BDD (Base de Datos Deductivas - también conocidas como Bases de Datos Lógicas). Comenzando con una pequeña reseña histórica para explicar los orígenes de los sistemas de bases de datos, repasando a sus predecesores, los sistemas de ficheros, y los diferentes enfoques que fueron surgiendo a medida que avanzaba del tiempo.

Obviamente el alcance del trabajo va mucho más allá de lo que nos puede decir la historia, es así que me enfocaré en explicar en que consiste una BDD, cuales son sus ventajas y desventajas con respecto a otros enfoques, haré una reseña a algunas implementaciones que hay en el mercado, y también mencionaremos algunas cosas sobre un enfoque hibrido que utiliza conceptos de BDD y bases de datos orientadas a objetos (BDOO) a la que llamaremos BOOD.

Antes de comenzar podemos mencionar las funciones básicas que debe cumplir un sistema de base de datos:

- Permitir a los usuarios crear nuevas bases de datos y especificar su estructura usando un lenguaje o interfaz especializado.
- Permitir a los usuarios consultar los datos y modificarlos, usando un lenguaje de interfaz apropiado.
- Permitir el almacenamiento de grandes cantidades de datos, durante un largo periodo de tiempo. Mintiendo los datos seguros de accidentes o ataques.
- Controlar la concurrencia de acceso a los datos, cubriendo los posibles casos de corrupción de datos por estas causas.

Reseña Histórica

Las bases de datos nacieron alrededor del 1960 con el objetivo de simplificar el proceso que significaba tener que guardar la información necesaria para los procesos en tarjetas perforadas. Esto implicaba un costo enorme, además de gran lentitud y de alta complejidad a la hora de modificar información o tener que trabajar con grandes cantidades de ella.

La idea de un sistema que pudiese manejar grandes cantidades de información a menor costo, tiempo y complejidad estimuló al desarrollo de las bases de datos. En un principio aparecieron los sistemas de ficheros, y es difícil de diferenciar en que momento de la historia dejaron su lugar a los sistemas de bases de datos, incluso hoy día todavía existen sistemas de ficheros en uso. Luego, la evolución de la computación pone al alcance máquinas de mayor potencia y facilidad de manipulación, así como también nuevos dispositivos de almacenamiento como los discos y cintas, en ese momento las bases de datos comienzan a ser de gran utilidad.

Durante los ’70 comenzaron a aparecer distintos modelos de datos para describir la estructura de la información en una base de datos para poder conseguir mayor independencia entre las aplicaciones y la organización física de los datos. Los dos modelos más populares de esa época fueron el modelo jerárquico y el modelo en red, que con el paso de los años se transformaron en el modelo puente para el modelo relacional.

No contento con lo que los modelos existentes ofrecían, en 1970 Edgar F. Codd propuso que los sistemas de bases de datos deberían presentarse a los usuarios como una visión de los datos organizadas en relaciones, las cuales eran definidas como conjuntos de tuplas, con lo que el orden no es importante. Además el usuario no tenia que preocuparse de la estructura del almacenamiento, solo debía preocuparse del que consultar y no de cómo hacerlo. De ese punto en adelante se realizaron muchos avances y características que si bien son en su mayoría independientes del modelo lo complementaban y hacían de el una herramienta mucho mas útil.

Alrededor de la mitad de los ochenta, algunas aplicaciones exigían mayor expresividad en los datos con los que trabajaban, esto coincidiendo con la entrada de lenguajes de programación como el C++ y SmallTalk llevo a algunos investigadores plantearse el transportar estas ideas a las bases de datos y permitir que el tipo de datos marcara cómo se representaba y se manipulaba dependiendo de los métodos que se definían para dicho tipo o clase. La idea de una base de datos orientada a objetos se articuló por primera vez alrededor de 1984, en este modelo la información sobre una entidad se almacena como un objeto persistente. Esto, en principio, lo hacía más eficiente en términos de requerimientos de espacio y aseguraba que los usuarios puedan manipular los datos sólo de las maneras en las que el programador haya especificado. También es más eficiente en el uso de espacio de disco requerido para las consultas, ya que en vez de almacenar la consulta, simplemente se construye una serie de índices a los objetos seleccionados. A esto hay que sumar las ventajas derivadas del modelo orientado a objetos, ya explotadas en sus lenguajes de programación, la mayor expresividad y su adecuación para almacenar muchos tipos de datos diferentes. A pesar de todas estas ventajas y que existen implementaciones hace ya mas de 15 años, los modelos orientados a objetos no han reemplazado a los relacionales, de hecho se cree que solo un 5% del mercado les pertenece.

El modelo objeto-relacional es un desarrollo más reciente y parece haber tenido bastante efecto. No es una tecnología en sí, sino una aglutinación de los modelos relacional y orientado a objetos. De hecho, algunas extensiones objetuales a los sistemas relacionales se pueden datar en los principios de los ochenta. El modelo objeto-relacional se define como una extensión objetual del modelo relacional, permitiendo la definición de nuevos tipos y de relaciones de herencia, entre otras cosas. Permite a las organizaciones continuar usando sus sistemas existentes y sus datos, sin realizar prácticamente cambios y les permiten empezar a utilizar gradualmente características orientadas a objetos, especialmente si se hace en conjunción con aplicaciones desarrolladas en entornos también orientados a objetos. Un buen ejemplo de lo que acabo de mencionar es el JDBC.
Hubieron mas avances y modelos híbridos que fueron apareciendo con el correr del tiempo, pero siendo que los mas importantes ya fueron mencionados ahora pasamos a tocar el tema que nos compete, las bases de datos deductivas.
A pesar que pudiera parecer un tema nuevo, las bases de datos deductivas nacieron con el propio modelo relacional, por lo que el concepto no es tan novedoso. De hecho antes de la introducción del modelo relacional del Codd, un personaje apellidado Kuhns considero el uso de la lógica para realizar consultas, y si observamos detenidamente una relación puede verse como la extensión de un predicado lógico de primer orden.
Con la aparición del lenguaje Prolog (Robinson - Kowalski 1979) y del Cálculo Relacional de Tuplas (Codd 1971), se plantea la posibilidad de definir un subconjunto de la lógica de primer orden que sea suficiente para representar la extensión de la base de datos y las consultas sobre ella. Posteriormente nace el Datalog, que, aunque se restringe (respecto a Prolog) a constantes y variables en los argumentos de las relaciones/predicados, mantiene la recursividad. Proliferan las reuniones y conferencias sobre la conexión entre “Lógica y Bases de Datos”, comenzando por la reunión celebrada en Toulouse en 1977. Se crea la equivalencia entre base de datos relacional y teoría lógica de primer orden (con datos básicos), con lo que se plantea la definición de relaciones en función de otras, es decir, con definiciones intencionales en vez de extensionales. Esta visión extiende el concepto de *vista como predicado derivado deductivamente*. Así, nacen las bases de datos deductivas (Das 1992) que permiten consultar información derivada de la información introducida extensionalmente con anterioridad.
Esta equivalencia ha permitido portar al campo de las bases de datos resultados y desarrollos fundamentales del campo de la lógica, la programación lógica y la inteligencia artificial, como ha sido la comprobación de la integridad de la deducción automática, la asimilación de conocimiento del área de actualización y revisión de programas lógicos, la optimización de consultas a partir de la optimización y transformación de programas lógicos, las bases de datos con restricciones y un largo etcétera. Estos sistemas basados en conocimiento han sido el tema de investigación durante los pasados 25 años aproximadamente, pese aunque su perspectiva ha sufrido algunos cambios en los últimos años.

Las Bases de Datos Deductivas

Introducción

El interés involucrado en el desarrollo de las bases de datos deductivas se centra en buscar la capacidad de definir reglas que permitan inferir información adicional a partir de los datos contenidos en las bases de datos. Los estudios involucrados en el desarrollo de tales sistemas se inspiraron inicialmente en las técnicas desarrolladas en el área de la inteligencia artificial, por lo que podemos decir que nacieron de la intersección de las bases de datos, la lógica y la inteligencia artificial o las bases de conocimientos.

Como dijimos anteriormente, un sistema de base de datos deductivo debe tener la capacidad de derivar nueva información a partir del uso de reglas de inferencia y los datos ya almacenados. Debido a que parte de los fundamentos teóricos se basan en la lógica matemática, también se suele conocer a estos sistemas como bases de datos lógicas. Las reglas deberán ser definidas en un lenguaje declarativo, para definir lo que queremos que el programa realice, y utilizando estas reglas el motor de inferencia se encarga de deducir nuevos hechos en la base de datos.

El modelo empleado en las BDD esta íntimamente relacionado con el modelo de las bases de datos relacionales, y sobre todo con el formalismo del cálculo relacional. También esta relacionado con el campo de la programación lógica y el lenguaje Prolog, los trabajos sobre BDD basados en lógica utilizan Prolog como punto de partida. Una variante del Prolog conocida como Datalog se utiliza para definir reglas declarativamente junto con un conjunto de relaciones existentes que se tratan como literales en el lenguaje. Aunque la estructura de Datalog se parece a la de Prolog, su semántica operativa sigue siendo una cuestión de activa investigación.

En fin, Un SGBD deductivo es un Sistema que permite derivar nueva información a partir de la introducida explícitamente en la Base por el usuario. Este maneja la perspectiva según la teoría de las demostraciones de una base de datos, y en particular es capaz de deducir hechos a partir de la base de datos extensional, es decir, las relaciones base, aplicando a esos hechos axiomas deductivos o reglas de inferencias especificados. Esta función deductiva se realiza mediante la adecuada explotación de ciertos conocimientos generales relativos a las informaciones de la base.

Definición

Por definición un sistema que tenga la capacidad de deducir hechos a partir de reglas incorporadas y los datos existentes es un sistema de bases de datos deductivo. Estas utilizan principalmente dos tipos de especificaciones que son hechos y reglas. Los hechos se especifican de manera similar a como se especifican las relaciones, excepto que no es necesario incluir los nombres de los atributos (una tupla en una relación describe algún hecho del mundo real cuyo significado queda determinado en parte por los nombres de los atributos). En una BDD el significado del valor de un atributo en una tupla queda determinado exclusivamente por su posición dentro de la tupla. Las reglas

se parecen un poco a las vistas relacionales, especifican relaciones virtuales que no están almacenadas realmente, pero que se pueden formar a partir de los hechos aplicando mecanismos de inferencia basados en las especificaciones de las reglas. La principal diferencia entre las reglas y las vistas es que en las primeras puede haber reexcursión y por lo tanto, pueden producir relaciones virtuales que no es posible definir en términos de las vistas relacionales estándar.

Los sistemas Bases de Datos Deductivas intentan modificar el hecho de que los datos requeridos residan en la memoria principal (por lo que la gestión de almacenamiento secundario no viene al caso) de modo que un SGBD se amplíe para manejar datos que residen en almacenamiento secundario.

En las bases de datos deductivas que emplean datalog, el enfoque es hacia el manejo de grandes volúmenes de datos almacenados en una base de datos relacional; por ello, se han inventado técnicas de evaluación que se parecen a las de una evaluación ascendente. Prolog tiene una limitación de que el orden de especificación de los hechos y las reglas es significativo para la evaluación; es mas, el orden las literales dentro de una regla es significativo. Las técnicas de ejecución para los programas de Datalog intentan subsanar estos problemas.

Características

Una Base de Datos Deductiva debe contar al menos con las siguientes características:

- Tener la capacidad de expresar consultas por medio de reglas lógicas.
- Permitir consultas recursivas y algoritmos eficientes para su evaluación.
- Contar con negaciones estratificadas.
- Soportar objetos y conjuntos complejos.
- Contar con métodos de optimización que garanticen la traducción de especificaciones dentro de planes eficientes de acceso.
- Como característica fundamental de una Base de Datos Deductiva es la posibilidad de inferir información a partir de los datos almacenados, es imperativo modelar la base de datos como un conjunto de fórmulas lógicas, las cuales permiten inferir otras fórmulas nuevas.

Reglas de Deducción

Las relaciones de una base de datos relacional se definen por intención y por extensión. En una base de datos en particular el conjunto de leyes define cual es la intención de las relaciones, y la extensión es determinada por cada uno de los estados de la base misma, es decir el conjunto de tuplas para cada una de las relaciones.
En un sistema de bases de datos convencional las restricciones de integridad actúan como leyes generales y son las encargadas de mantener la coherencia de la información, en cambio en los sistemas deductivos casi todas las leyes van a ser utilizadas como reglas de deducción para deducir nueva información a partir de los datos que sí se encuentran dentro de la base de datos.

A continuación se puede ver un ejemplo para demostrar las reglas y relaciones de una base de datos deductiva o lógica:

Tenemos la relación Padre y la regla Abuelo:

Padre(Juan, Luís)
Padre(Luís, Arturo)
Abuelo (x, y):- **Padre**(x, z),**Padre** (z, y)

Una consulta típica consiste en preguntar si un registro ``podría" existir, por ejemplo:

Abuelo(Juan, Luís) la respuesta sería no ya que:

Padre(Juan, Luís) la respuesta sería si
Padre(Luís ,Luís) la respuesta sería no

Esta relación es falsa.

Abuelo(Juan, Arturo) la respuesta sería si, ya que:

Padre(Juan, Luís) la respuesta sería si
Padre(Luís, Arturo) la respuesta sería si

Esta relación es verdadera.

Problemas y utilización de las reglas de Deducción

Si bien desde el punto de vista de un usuario, la idea de una base de datos deductiva puede ser muy interesante, hay que notar que la complejidad de los algoritmos es exponencial respecto a la cantidad de relaciones. De esta manera mantener un sistema de información consistente es prácticamente imposible.

En cuanto a la utilización y/o explotación de las reglas de deducción podemos encontrar dos análisis diferentes que separan las reglas en dos grupos, reglas de derivación o reglas de generación. El primer grupo es se forma con las reglas de deducción cuando su uso se desempeña en la fase de interrogación, buscando información deducible implícita. Por otro lado, para el segundo grupo, hablamos del uso de reglas de deducción en la fase de modificación, cuando se añade información deducible.

Interpretación de las reglas

Podemos usar dos metodologías diferentes para poder interpretar el significado teórico de las reglas, estos son, por la teoría de demostración y por la teoría de modelos. Sin embargo en los sistemas prácticos, es el mecanismo de inferencia que tiene el sistema el que define la interpretación exacta, y esta puede diferir de los modelos teóricos que acabamos de mencionar. Este mecanismo de inferencia es un

procedimiento computacional y por lo tanto provee una interpretación computacional del significado de las reglas.

Interpretación mediante teoría de modelos

La teoría de modelos puede ser tomada como la teoría de la interpretación de los lenguajes formales. Y podemos definir a la interpretación de un lenguaje formal básicamente como una asignación de significados a sus símbolos y/o a sus fórmulas.

Entre los conceptos que se utilizan en la teoría de modelos se encuentran los de verdadero para una interpretación, consecuencia semántica (o consecuencia desde el punto de vista de la teoría de modelos) y validez lógica.

En la teoría de modelos, dado un dominio finito o infinito de valores constantes, se le asigna a un predicado todas las combinaciones posibles de valores como argumentos. Después se debe determinar si el predicado es verdadero o falso. En general, basta con especificar las combinaciones de argumentos que hacen que el predicado sea verdadero, y decir que todas las demás combinaciones hacen que sea falso. Si esto se hace con todos los predicados, se habla de una interpretación del conjunto de predicados. Llamamos modelo a una interpretación para un conjunto de reglas si es que estas siempre se cumplen en esa interpretación, es decir que sean cuales fueren los valores que se le asignan a las variables de las reglas, la cabeza de reglas es verdadera cuando sustituimos los valores de verdad asignados a los predicados en el cuerpo de las reglas según esa interpretación. De este modo, siempre que se aplica una sustitución (enlace) a las variables de las reglas, si todos los predicados del cuerpo de un arreglo son verdaderos en esa interpretación, el predicado de la cabeza de la regla también debe ser verdadero. Cabe señalar que una regla se viola si un determinado enlace de constantes en a las variables hace verdaderos todos los predicados del cuerpo de la regla, pero hace que el predicado de la cabeza de la regla sea falso.

Interpretación mediante teoría de demostraciones

La teoría de la demostración parte de la teoría de los sistemas formales (Lenguajes formales dotados de mecanismos deductivos) y no requiere de ninguna referencia o interpretación de los lenguajes.

Entre los conceptos que pertenecen a la teoría de la demostración se encuentran los siguientes: demostración formal, teorema formal, derivación formal y consecuencia desde el punto de vista de la teoría de la demostración.

En la teoría de la demostración se considerarán los hechos y las reglas como enunciados verdades o axiomas. Los axiomas base no contienen variables. Los hechos son axiomas base que se dan por ciertos. Las reglas se llaman axiomas deductivos, ya que pueden servir para deducir hechos nuevos. Con los axiomas deductivos se pueden construir demostraciones que deriven hechos nuevos a partir de los ya existentes. Los axiomas deductivos, junto con las **restricciones de integridad** constituyen lo que en ocasiones se denomina base de datos intencional, y la base de datos extensional junto con la intencional constituye lo que suele llamarse Base de Datos Deductiva.

Una demostración de teorema es el proceso de determinar si un determinado hecho se cumple. La interpretación por la teoría de demostraciones ofrece un enfoque por procedimientos o computacional para calcular una respuesta a la consulta *Datalog*.

Mecanismos de Inferencia

Existen dos mecanismos de inferencia principales y ambos se basan en la interpretación de reglas mediante la teoría de las demostraciones, que son el mecanismo de inferencia ascendente y el mecanismo de inferencia descendente, mas comúnmente conocidos como encadenamiento hacia delante (forward-chaining) y encadenamiento hacia atrás (backward-chaining) respectivamente.

Mecanismos de inferencia ascendente

En este caso la maquina de inferencia parte de los hechos y genera nuevos hechos mediante la aplicación de reglas. Estos hechos que fueron generados se comparan con el predicado que es el objetivo de la consulta para confirmar si coinciden o no. Lo más conveniente es que la estrategia de búsqueda solo genere hechos que sean relevantes a una consulta, para evitar que se generen todos los hechos posibles y en un orden que nada tendrá que ver con la consulta hecha.

Mecanismos de inferencia descendente

En este mecanismo la inferencia comienza con el objetivo buscado y trata de encontrar hechos que lo satisfagan. O sea que el objetivo de la consulta será el predicado y a partir de él se tratan de encontrar variables que nos lleven a hechos válidos. Este es el utilizado en los intérpretes de Prolog.

Seguridad de los programas

Se dice que un programa o regla es segura si genera un conjunto finito de hechos. El problema se encuentra en que determinar si un conjunto de reglas es o no seguro esta dentro de los que se conocen como *problemas indecidibles*. A pesar de esto, es posible determinar la seguridad de formas restringidas de reglas. Se obtendrán reglas inseguras, que puedan generar un número infinito de hecho, cuando una de las variables de la regla pueda abarcar un dominio infinito de valores y esa variable no esté limitada a abarcar una relación finita.

Existe una manera más formal de generar reglas seguras y para ello se utiliza el concepto de variable limitada. Si X es una variable perteneciente a una regla, decimos que X es limitada si se cumple uno de los siguientes casos:

- Aparece en un predicado normal (no integrado) en el cuerpo de la regla.
- Aparece en un predicado de la forma X=c o c=X o (c1<=X y X<=c2) en el cuerpo de la regla. En este caso c, c1 y c2 representan valores constantes.
- Aparece en un predicado de la forma X=Y o Y=X en el cuerpo de la regla. En este caso Y representa una variable limitada

Podemos decir que una regla es segura en caso de que todas las variables que la componen sean limitadas.

Sistemas de Bases de Datos Deductivas

El evento que puede considerarse fundador del campo de las bases de datos deductivas fue el seminario de Toulouse sobre “Lógica y Bases de Datos”, organizado por Gallaire, Minker y Nicolas en 1977. El siguiente periodo de gran auge comenzó con la creación de MCC (*Microelectronics and Computer Technology Corporation*), que fue una reacción al Proyecto de Quinta Generación Japonés. Se han desarrollado varios sistemas de bases de datos deductivas experimentales y unos pocos se han utilizado comercialmente. A continuación, repasaremos brevemente tres implementaciones diferentes de las ideas presentadas hasta ahora; a conocer son LDL, NAIL! y CORAL.

El sistema LDL

El proyecto LDL (*Logic Data Lenguage*) de MCC (*Microelectronics and Computer Technology Corporation*) se inicio en 1984 con vista a dos objetivos principales:

- Crear un sistema que extendiera el modelo relacional y que a la vez aprovechara algunas de las características deseables de un sistema de gestión de base de datos relacional.
- Mejorar la funcionalidad de un sistema de gestión de base de datos de modo que operase como uno deductivo y además permitiese la creación de aplicaciones de propósito general.

Como resultado de este proyecto nació un Sistema de Gestión de Base de Datos Deductivo (SGBDD) que se encuentra en el mercado. Se hará un breve repaso de los puntos principales del enfoque adoptado por LDL y consideraremos sus características importantes.

El sistema LDL ha intentado combinar la capacidad de expresión y deducción de prolog con la funcionalidad y los recursos de un SGBD de propósito general. La desventaja principal que experimentaron los primero sistemas que acoplaron Prolog a un SGBDR es que Prolog es un lenguaje de navegación (una tupla a la vez) en tanto que en los SGBDR el usuario formula una consulta correcta y deja al sistema la optimización de su ejecución. Para lidiar con este inconveniente los diseñadores de LDL optaron por convertir el Prolog a un lenguaje lógico declarativo de aplicación general, produciendo un lenguaje que es diferente en sus constructores y estilo de programación en los siguientes aspectos:

- Las reglas se compilan en LDL.
- Existe una noción de esquema de la base de hechos en LDL en tiempo de compilación. Esta base se actualiza libremente en tiempo de ejecución.
- LDL no sigue la técnica de resolución y unificación que se emplea en los sistemas Prolog basados en backwards-chaining.
- El modelo de ejecución de LDL es más simple, basado en la operación de comparación y el cálculo del menor punto fijo. Estos operadores, a su vez, utilizan extensiones simples del algebra relacional.

Las primeras implementaciones del LDL, completadas en 1987, se basaban en un lenguaje llamado FAD. Una implementación posterior, concluida en 1988, se llama SALAD y sufrió cambios adicionales al probarse con aplicaciones de la vida real. El prototipo actual es un sistema portátil eficiente para UNIZ que supone una interfaz de una sola tupla de tipo obtener el siguiente entre el programa LDL compilado y un gestor de hechos subyacente.

NAIL!

El proyecto NAIL! (*Not Another Implementation of Logic!*) se inició en la universidad de Stanford en 1985, El objetivo inicial era estudiar la optimización de la lógica mediante el modelo de todas las soluciones, orientado a bases de datos. El propósito del proyecto fue el de obtener una ejecución óptima de los objetivos de Datalog sobre un SGBDR. Considerando que una única estrategia viable era inapropiada para todos los programas lógicos en general, se creó una arquitectura extensible, que podía ampliarse a través de adiciones progresivas.

En colaboración con el grupo MCC, este proyecto fue responsable de la idea de los conjuntos mágicos y del primer trabajo sobre recursiones regulares. Aportó muchas contribuciones importantes para el manejo de la negación y la agregación en las reglas lógicas. También se desarrollaron la negación estratificada, la negación bien fundada y la negación modularmente estratificada en relación con este proyecto.

Se construyó un prototipo de sistema inicial, que posteriormente se abandono porque el paradigma puramente declarativo resulto ser inviable para muchas aplicaciones. El sistema revisado emplea un lenguaje central, llamado GLUE, que se compone esencialmente de reglas lógicas individuales, con la capacidad para sentencias SQL, envueltas en constructores de lenguajes convencionales como ciclos, procedimientos y módulos El lenguaje NAIL! Original se convirtió en un mecanismo de vistas para GLUE; permite especificaciones totalmente declarativas en situaciones en que esto es apropiado.

El sistema CORAL

El sistema CORAL, creado en la Universidad de Wisconsin en Madison, aprovecha las experiencias adquiridas con el proyecto LDL. Al igual que este provee un lenguaje declarativo basado en cláusulas de Horn con una arquitectura abierta. Sin embargo, hay muchas diferencias importantes, tanto en el lenguaje como en su implementación. CORAL puede considerarse como un lenguaje de programación de bases de datos que combina características importantes de SQL y Prolog.

Desde la perspectiva de la implementación, CORAL implementa varias optimizaciones para manejar de manera eficiente las tuplas no base, además de usar técnicas como la de plantillas mágicas para incluir selecciones en consultas recursivas, proyecciones, y optimizaciones especiales de diferentes clases de programas lineales.

Visión de las Bases de Datos Deductivas Orientadas a Objetos

Desde finales de los `80, se han venido creando diversos prototipos de BDDOO en universidades y laboratorios de investigación. VALIDITY, creada en Bull, es el primer producto industrial en este ámbito. Los sistemas LDL y CORAL que se mencionaron en la sección anterior, ofrecen algunas características orientadas a objetos adicionales y pueden considerarse BDDOO. Se han adoptado los siguientes enfoques en el diseño de sistemas BDDOO:

- Ampliación del lenguaje: un modelo de lenguaje deductivo existente se amplía con características orientadas a objetos. Por ejemplo, Datalog se amplía para manejar la identidad, herencia y otras características orientadas a objetos.
- Integración del lenguaje: un lenguaje deductivo se integra con un lenguaje de programación imperativo en el contexto de un modelo de objetos o sistema de tipos. El sistema resultante soporta una variedad de programas estándar, al tiempo que permite que se utilicen paradigmas de programación diferentes y complementarios para diferentes tareas, o partes diferentes de la misma tarea. Este enfoque fue promovido por el sistema Glue-Nail.
- Reconstrucción del lenguaje: un modelo de objetos se reconstruye, creando un nuevo lenguaje lógico que incluye características orientadas a objetos. En esta estrategia, el objetivo es el de crear una lógica de objetos que capture lo esencial del paradigma orientado a objetos y que también pueda emplearse como un lenguaje de programación deductivo en las BDDOO. La base de este enfoque es el argumento de que las ampliaciones del lenguaje no consiguen combinar satisfactoriamente la orientación a objetos la lógica, ya que pierden su naturaleza declarativa o no consiguen captar todos los aspectos del modelo orientado a objetos.

Conclusión

Se presentó en este trabajo a las bases de datos deductivas que es una rama relativamente nueva de la gestión de bases de datos. Es un campo que se ha visto influido por los lenguajes de programación lógica, en especial del Prolog y el Datalog. Se vio de manera extremamente reducida la evolución a través del tiempo de las bases de datos, junto con las nuevas complicaciones que iban surgiendo.

En cuanto a las BDD se mencionó e incluso explico en manera no muy detallada su funcionamiento, sus características principales y ciertos conceptos de gran importancia para el entendimiento básico de lo que es un motor de inferencia y como las BDD logran su objetivo.

Por último mencionamos algunas implementaciones existentes y analizamos cada una de ellas de modo a resaltar los puntos mas importantes que ayuden a entender con que objetivos fueron diseñadas y como lograron cumplir con las características principales que requiere una BDD.

Podemos ver que si bien es un tema investigado durante al menos 20 años, difícilmente este lo suficientemente desarrollado como para competir en el mercado con otros sistemas de bases de datos. Más bien, las bases de datos deductivas podrían ser utilizadas en ambientes donde las existentes no cubran las necesidades básicas del problema, o sea que tienen un fin diferente. Así como las bases de datos orientadas a objetos no son todavía lo suficientemente fuertes como para poseer una buena parte del mercado, tal vez la intención de realizar un híbrido entre estas dos visiones sea una manera de buscar un modelo que realmente pueda ingresar fuerte al mercado.

Bibliografía

- Fundamentos de Sistemas de Bases de Datos – Ramez A. Elmascri,
- La disciplina de los sistemas de Bases de Datos. Historia, Situación Actual y Perspectivas – José Hernández Orallo, Mayo 2002.
- http://es.wikipedia.org/wiki/Base_de_datos
- http://alarcos.inf-cr-uclm.es/doc/bbddavanzadas/Funcionalidad2.pdf
- http://www.dsic.upv.es/~mcelma/conferencia.ppt