TEMA 42

SISTEMAS DE BASES DE DATOS DISTRIBUIDOS

ÍNDICE

1. INTRODUCCIÓN
2. CONSTRUCCIÓN DE UNA BASE DE DATOS DISTRIBUIDA
3. DISEÑO DE UNA BASE DE DATOS DISTRIBUIDA
4. TRANSPARENCIA DE LA RED Y AUTONOMÍA LOCAL
5. VENTAJAS DE LAS BASES DE DATOS DISTRIBUIDAS
6. DESVENTAJAS
7. COMPONENTES DEL SGBD DE LAS BASES DE DATOS DISTRIBUIDAS
8. BASES DE DATOS DISTRIBUIDAS ORIENTADAS A OBJETOS
9. APLICACIONES CLIENTE/SERVIDOR
10. BIBLIOGRAFÍA

1. INTRODUCCIÓN

Una de las principales tendencias de la computación a lo largo de los últimos años ha sido el pasar de grandes ordenadores centralizados a redes distribuidas de Sistemas Informáticos.

El incremento explosivo de la popularidad del ordenador personal en los 80 trajo la potencia del ordenador directamente a las mesas de despacho de millones de personas.

Conforme los ordenadores y las redes de ordenadores se extendían a través de organizaciones, los datos ya no residían en un único sistema bajo el control de un único SGBD. En lugar de ello, los datos fueron extendiéndose a través de muchos sistemas diferentes, cada uno con su propio Gestor de Bases de Datos. Con frecuencia los diferentes Sistemas Informáticos y los diferentes SGBD procedían de diferentes fabricantes.

Estas tendencias han conducido a una fuerte centralización en la industria informática y en la comunidad de gestión de datos sobre los problemas de la gestión de las Bases de Datos Distribuidas.

Un **Sistema de Bases de Datos Distribuidas** esta formado por un conjunto de *localidades*, cada una de las cuales puede participar en la ejecución de transacciones que procesan datos de una o varias de ellas, e incluso los datos pueden estar fragmentados en diversas localidades.

En las localidades, llamadas también **nodos**, los procesadores del sistema distribuido pueden variar en cuanto a su tamaño y función: pueden ir desde microordenadores a grandes ordenadores. Cada nodo, además del ordenador, debe tener un SGBD para poder administrar su propia base.

Las BDD pueden ser:

Homogéneas: El SGBD utiliza un sólo lenguaje para describir los esquemas.

Heterogéneas: El SGBD utiliza más de 1 lenguaje para describir los esquemas.

En un sistema de BDD la información se encuentra dispersa por varios lugares. Desde cada una de ellos se pueden ejecutar los datos de la propia localidad o los datos situados en varias localidades diferentes, dando lugar a dos **tipos de transacciones**:

- **Transacciones locales**: Procesan información situada en el mismo nodo.
- **Transacciones globales**: Procesan información situada en nodos diferentes. En este tipo los ordenadores deben estar interconectados entre sí.

Según la situación geográfica, la **conexión entre los nodos** se realiza mediante:

- Redes de larga distancia.
- Redes de área local.

La **unión entre los nodos** se puede realizar de las siguientes formas:

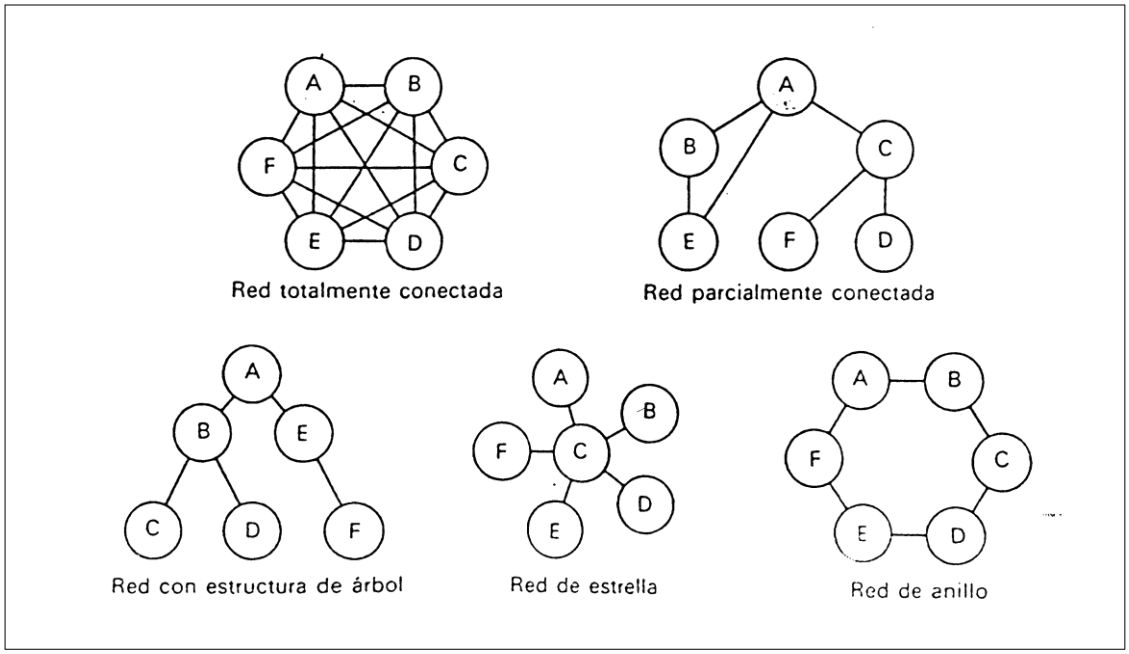
Red *totalmente conectada*.

Red *parcialmente conectada.*

Red con *estructura de árbol*.

Red con *estructura de estrella*.

Red con *estructura de anillo*.



2. CONSTRUCCIÓN DE UNA BASE DE DATOS DISTRIBUIDA

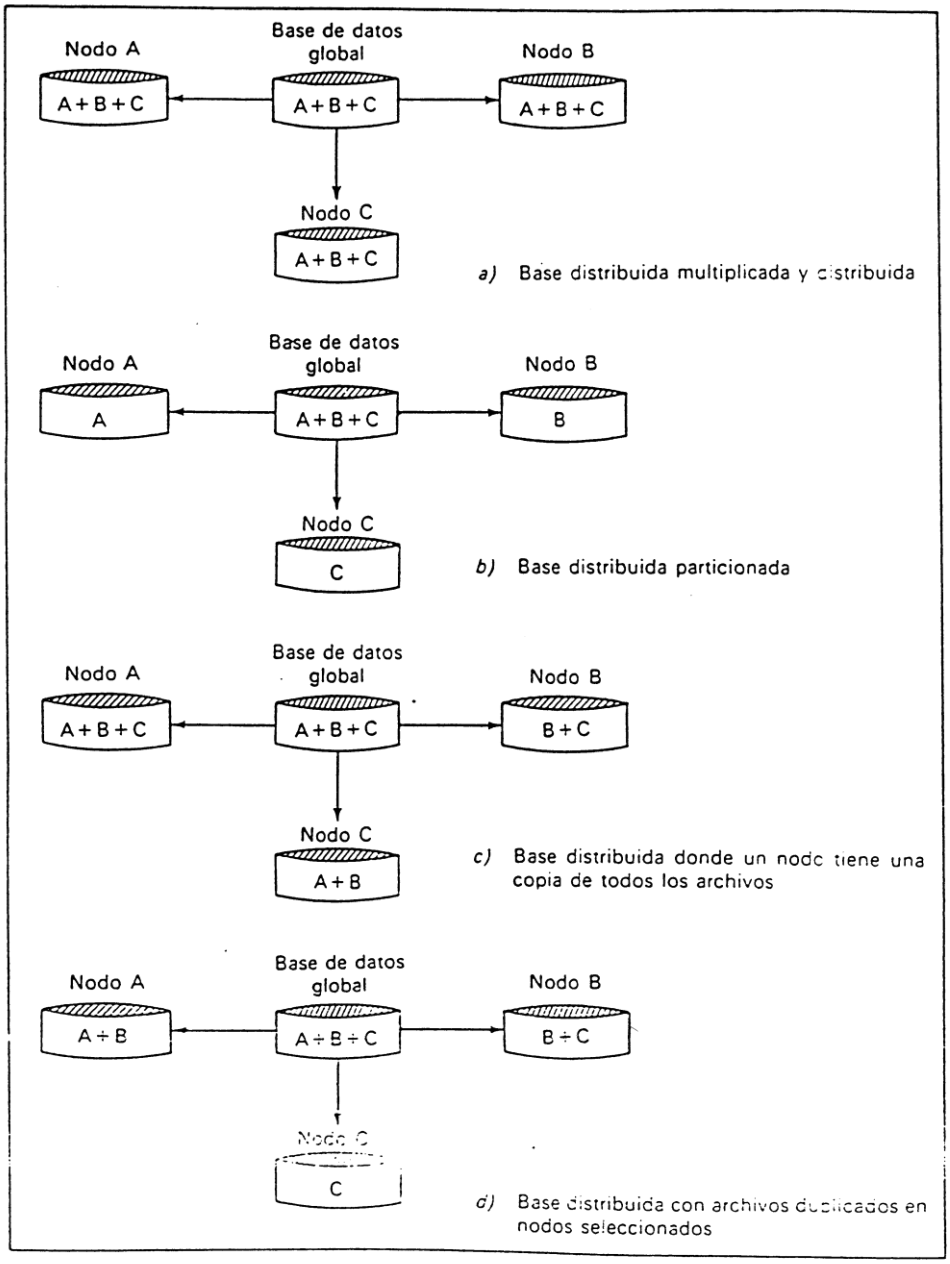
El punto de partida de las BDD es el enfoque relacional. La construcción se refiere a la distribución de ficheros relacionales por los distintos nodos de la red.

1. **Base de Datos Multiplicada y Distribuida**. Todos los ficheros están duplicados en cada nodo con el fin de reducir los costes de comunicación y aumentar el desempeño, ya que cada operación de recuperación se puede hacer en cada nodo. Presenta el inconveniente de la cantidad de espacio necesario para su almacenamiento, redundancia externa y los problemas en la actualización de los datos redundantes en cada nodo, por lo que este tipo de distribución se utiliza raramente.

2. **Base de Datos Distribuida Particionada**. Es la forma más simple de distribuir los datos en una red. Ficheros específicos se distribuyen en cada nodo sin duplicidad de datos. Presenta la ventaja de tener que actualizar sólo un fichero cada vez que se produce una transacción. Pero cuando se desea hacer una consulta que implica la necesidad de acceder a datos situados en otras localidades diferentes lleva consigo un aumento del coste de tiempo para su realización. Un fallo en cualquier nodo puede dejar a toda la red sin servicio.

3. **Bases de Datos Distribuidas en donde los ficheros se ubican en un nodo específico**. Y además, algunos ficheros se almacenan en otros nodos. La elección del nodo para acoger a todos los ficheros, así como de los ficheros que se guardan en cada nodo, debe realizarse con el fin de reducir al mínimo los costes de comunicación en las transacciones globales, es decir, escoger la localidad de referencia. Se han reducido los costes de comunicación, pero aún quedan por resolver los problemas de actualización. Si la localidad que acoge a todos los ficheros sufre un fallo, toda la red puede quedar inutilizada.

4. **Bases de Datos Distribuidas con ficheros duplicados seleccionados**. Es la más flexible de todas las construcciones. Considera que un fichero que se utiliza prácticamente en una localidad se almacena sólo en ella y que un fichero que se consulta en varias localidades está ubicado en todas ellas. Este sistema, a pesar de tener ventajas, también presenta inconvenientes: la actualización y hallar datos que se encuentran en varias localidades diferentes.



Distintas formas de almacenar los archivos en una red

3. DISEÑO DE UNA BASE DE DATOS DISTRIBUIDA

Aparte de las consideraciones propias de cualquier Base de Datos, cuando se trata de diseñar una BDD es necesario tener en cuenta otros aspectos como son:

- La **repetición**: Consiste en mantener varias copias de ficheros en nodos diferentes, como se ha visto en el punto anterior.

- La **fragmentación**: De la información contenida en las relaciones. Cada fragmento se almacena en nodos diferentes. Un fragmento contiene información suficiente para poder restituir el fichero.

Las diferentes *formas en que se puede fragmentar una relación* son:

a) **Horizontal**: Cada fragmento está formado por una serie de tuplas (filas).

b) **Vertical**: Cada fragmento esta formado por una serie de columnas de la tabla global. La fragmentación vertical se lleva a cabo por medio de un campo especial *denominado identificación de tupla* (un puntero (ID) con la dirección física o lógica del resto de las columnas en los diferentes fragmentos).

c) **Mixta**: es un compendio de las dos anteriores.

La repetición y la fragmentación comprenden a las anteriores.

4. TRANSPARENCIA DE LA RED Y AUTONOMÍA LOCAL

Se denomina así, a la forma en que el sistema oculta los detalles de la distribución de los fragmentos por los distintos nodos de la red; en definitiva, el grado en que los usuarios pueden ignorar cómo está diseñada la BDD.

La transparencia está íntimamente ligada a la *autonomía local*, es el grado en que un diseñador de un nodo puede ser independiente del resto del sistema.

Estos dos conceptos se deben tener en cuenta en:

Los **nombres de datos**, que deben ser únicos en la red global. Para que ningún diseñador de distintas localidades utilice el mismo nombre para designar a datos diferentes, a los nombres de datos se les incorpora habitualmente información acerca de la localidad que los genera y sobre los fragmentos que la componen.

La **actualización**, que debe realizarse en todas las copias y en todos los fragmentos afectados.

El acceso a datos distribuidos es útil y pasará a ser crítico cuando se confirme la tendencia hacia la computación distribuida. Los principales vendedores de SGBD están comprometidos en la entrega de gestión de BDD, y algunos de ellos ya ofrecen capacidades limitadas de BDD. Los expertos en la industria y otros han escrito extensamente acerca de las características que debería proporcionar un SGBD distribuido, y existe un acuerdo general sobre estas *características* "ideales":

Transparencia de ubicación. El usuario no debería tener que preocuparse acerca de dónde están localizados físicamente los datos. El SGBD debería presentar todos los datos como si fueran locales y debería ser responsable de mantener esa ilusión.

Sistemas heterogéneos. El SGBD debería soportar datos almacenados en sistemas diferentes, con diferentes arquitecturas y niveles de rendimiento, incluyendo PCs, estaciones de trabajo, servidores de LAN, miniordenadores y macroordenadores.

Transparencia de red. Excepto por las diferencias en rendimiento, el SGBD debería trabajar de la misma manera sobre diferentes redes, desde las LANs de alta velocidad a los enlaces telefónicos de baja velocidad.

Consultas distribuidas. El usuario debería ser capaz de componer datos procedentes de cualquiera de las tablas de la BDD, incluso si las tablas están localizadas en sistemas físicos diferentes.

Actualizaciones distribuidas. El usuario debería ser capaz de actualizar datos en cualquier tabla para que la que tenga privilegios necesarios tanto si la tabla está en un sistema local como si está en un sistema remoto.

Transacciones distribuidas. El SGBD debe soportar transacciones (Utilizando COMMIT y ROLLBACK) a través de fronteras de sistema, manteniendo la integridad de la BDD incluso en presencia de fallos de red y fallos de sistemas individuales.

Seguridad. El SGBD debe proporcionar un esquema de seguridad adecuado para proteger la BDD completa frente a formas no autorizadas de acceso.

Acceso universal. El SGBD debería proporcionar acceso uniforme y universal a todos los datos de la organización.

Ningún producto de SGBD distribuido actual ni siquiera se acerca a satisfacer este ideal. De hecho, existen formidables obstáculos que hacen difícil proporcionar incluso las formas más sencillas de gestión de una BDD. Estos obstáculos incluyen:

- Rendimiento.
- SQL estático.
- Optimización.
- Compatibilidad de Datos.
- Catálogos de sistema.
- Entorno de vendedores mixtos.
- Interbloqueos distribuidos.
- Recuperación.

A causa de estos obstáculos, los principales vendedores de SGBD han adoptado un planteamiento de paso a paso para la gestión de datos distribuida. Al principio el SGBD puede permitir que un usuario en un sistema consulte una Base de Datos localizada en algún otro sistema. En versiones posteriores las capacidades de SGBD distribuido pueden crecer, acercándose cada vez más al objetivo de proporcionar acceso de datos transparente y universal.

5. VENTAJAS DE LAS BASES DE DATOS DISTRIBUIDAS

Entre las **ventajas** de los sistemas que soportan BDD están:

1. Reducir el coste de comunicación de datos, ya que la mayoría de las consultas se pueden realizar sin necesidad de acceder a los archivos situados en localidades diferentes.

2. Autonomía local.

3. Confiabilidad. Cuando el sistema de una determinada localidad sufre un fallo y se “cae”, toda la red no tiene por qué dejar de funcionar, debido a que cada localidad cuenta con su propio ordenador y fragmentos de datos.

   Una vez solucionado el fallo, la localidad afectada deberá reintegrarse a la red con el mínimo de complicaciones.

6. DESVENTAJAS

Las BDD por su mayor complejidad presentan una serie de desventajas entre las que se encuentran:

1. Coste del software por la propia estructuración de los datos.

2. Actualización de datos duplicados.

3. Mayor posibilidad de errores como consecuencia de una mayor concurrencia desde distintas localidades a un mismo dato, lo que requiere complicados protocolos de control.

4. Transparencia de ubicación. Gran parte de los sistemas distribuidos no logran la transparencia de ubicación y deben ser los propios usuarios los que transfieran los archivos para el procesamiento local.

5. Mayor tiempo extra en el procesamiento, sobre todo cuando se trata de transacciones globales.

7. COMPONENTES DEL SGBD DE LAS BASES DE DATOS DISTRIBUIDAS

Al estar los nodos de la red distribuidos por diversos lugares es imprescindible un **Gestor de Comunicaciones** para garantizar que los mensajes enviados a través de la red, llegan a su destino sin sufrir ningún tipo de interferencias o duplicidad de la información. Entre los componentes propiamente dichos del SGBD se encuentran:

En los **puntos de acceso**

- Los **catálogos** con la información que se refiere a los esquemas.

- El **procesador de transacciones fuente**, comprueba que sólo accedan a la red los usuarios autorizados y obtengan la información requerida, así como en qué puntos se encuentran los datos pedidos. A cada transacción la descompone en subtransacciones para reducir al mínimo el uso de las líneas de comunicación.

En los **puntos de almacenamiento**:

- El **SGBD** respectivo.

- El **gestor de transacciones** recibe las transacciones, y termina de compilar si es preciso, y las ejecuta. Cuando finaliza su ejecución, lo comunica al gestor de transacciones de la localidad que lo requirió y manda los resultados a los nodos reseñados en el plan de ejecución.

8. BASES DE DATOS DISTRIBUIDAS ORIENTADAS A OBJETOS

La independencia de la distribución es el último de una serie de pasos hacia la independencia que comenzó a finales de los años 60 con la independencia de dispositivo y continuó desde finales de los 70 y principios de los 80 con la independencia de datos. Los temas más técnicos relativos a la realización de Bases de Datos Relacionales Distribuidas se aplican también a las Bases de Datos Orientadas a Objetos. Los sistemas de Bases de Datos Distribuidas se explican normalmente en el contexto de su modelo Cliente/ Servidor.

Una Base de Datos se dice que satisface el **modelo Cliente/Servidor** si la mayoría de los datos y lógica de procesamiento relativa residen en uno o más servidores, aunque cada cliente procese localmente solo los datos suficientes y la lógica de consulta para permitirle enviar consultas y recibir subconjuntos de datos a partir del servidor(es).

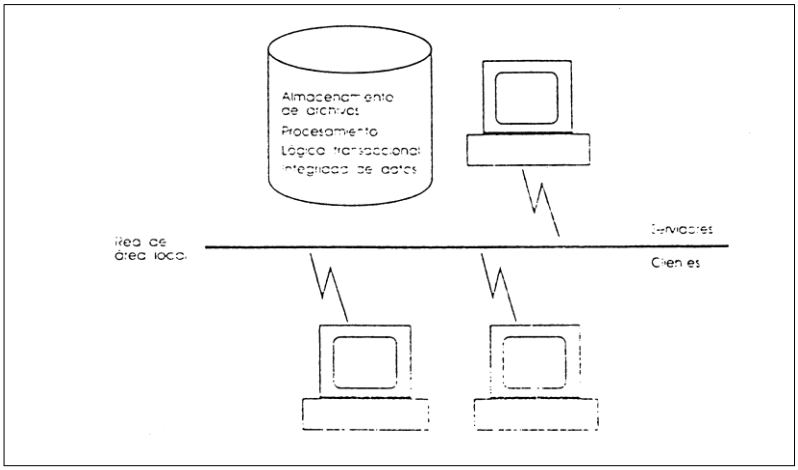
Cliente: Un ordenador o estación de trabajo conectado a una red y que se utiliza para acceder a los recursos de esa red.

Servidor: Un ordenador que suministra a los clientes servicios tales como bases de datos, conexiones a red o grandes controladores de disco. Los servidores pueden ser grandes ordenadores, miniordenadores, estaciones de trabajo o dispositivos LAN. Más de un servidor puede estar involucrado en suministrar servicios a los clientes.

Hay que tener en cuenta, que el modelo Cliente /Servidor no implica una Base de Datos Distribuida; un modelo Cliente /Servidor se distribuye solamente si tiene más de un servidor.

Existen varias ventajas en la arquitectura Cliente /Servidor. Una de ellas es la mayor facilidad para imponer la integridad de los datos, el bloqueo y la seguridad, puesto que los datos se comprueban en la entrada y salida de un único lugar. Otra ventaja es la mayor facilidad de ampliar a sistemas más grandes

añadiendo servidores y clientes. Además, puede ser menos cara, porque se necesita menos potencia de procesamiento y capacidad de almacenamiento para cada cliente. Finalmente, existe un reducido tráfico de red ya que el servidor devuelve solamente al cliente el subconjunto de datos solicitado.



Permanecen sin resolver un cierto número problemas en el método distribuido. Por ejemplo, no está todavía muy claro cómo identificar mejor los objetos y difundir los identificadores entre las máquinas para hacer que la identidad sea válida en la totalidad de la red. Se necesita una estrategia para hacer transparentes los objetos en un esquema distribuido. La estrategia debe permitir a un único objeto comportarse correctamente con interfaces múltiples o incluso con sistemas de archivos múltiples. Efectuar el seguimiento de las posiciones y movimientos de los objetos presenta otro problema. Las sub-bases de datos, como por ejemplo una Base de Datos que gestione las posiciones (base de datos del corredor, bolsista, agente comercial de posiciones), pueden facilitar estas exigencias funcionales.

Una ampliación propuesta para el modelo Cliente/Servidor se denomina *modelo cliente y corredor* (broker). En este modelo, los clientes son consumidores de varios servicios como acceso a ficheros e impresoras. Los objetos residentes en la red contienen simple información sin interpretar o procedimientos reales, por ejemplo, procedimientos para colocar un documento en una sola cola de impresión. El corredor crea una función que denomina y sitúa los objetos y usuarios del sistema, identifica los clientes de los objetos, localiza los recursos de cálculo inactivos en la red, etc.

Otros métodos de distribución incluyen *llamadas remotas a procedimientos* (RPC's remote procedure calls) entre gestores de objeto locales y Bases de datos objeto en servidores. Las RPC's como forma de interacción entre objetos proporcionan la ventaja de ser al mismo tiempo independientes de la red e independientes de la aplicación. Aunque la Base de Datos objeto de cada servidor funciona como un corredor o gestor de posiciones, las Bases de Datos Orientadas a Objetos en diferentes servidores se comunicarán utilizando el mismo mecanismo RPC utilizado para la interacción cliente/servidor.

9. APLICACIONES CLIENTE/SERVIDOR

La explosiva popularidad y la agresiva política de precios de los ordenadores personales han producido un cambio importante en el modo en que las organizaciones están implementando nuevas aplicaciones de procesamiento de datos. Cada vez más, los diseñadores de aplicaciones dejan de asumir un mini o un macroordenador central que se comunica con docenas de usuarios sentados ante terminales “tontos”. En vez de ello, las aplicaciones están emigrando a ordenadores personales, en donde pueden tomar ventaja de la interfaz de usuario gráfica del PC. La Base de Datos que soporta la aplicación está almacenada en un Servidor de Base de Datos, ligada a los ordenadores personales por una red de área local. Las redes de área local de PCs con Servidores de Bases de Datos presentan, por tanto, una seria amenaza competitiva frente a los miniordenadores para nuevas aplicaciones de procesamiento de datos.

El planteamiento Cliente/Servidor para diseño de aplicaciones esta acelerando la aceptación de Bases de Datos Relacionales, con el lenguaje SQL actuando de interfaz entre el ordenador personal y el Servidor de la Base de Datos. Algunos analistas de la industria informática, al advertir este fenómeno, han concluido que los servidores de bases de datos SQL se han convertido en una mercancía, y que su

estandarización permitirá a una aplicación basada en SQL acceder a datos almacenados en cualquier Base de Datos basada en SQL.

10. BIBLIOGRAFÍA

Gary W. Hansen
Diseño y Administración de Bases de Datos
Prentice Hall, 1998

Korth
Fundamentos de bases de datos
Mc Graw-Hill, 1993