

# InterBase

## Comparativo contra Ms SQL Server / Sybase SQL Server

Una breve historia

Mecanismos de Bloqueo

SQL Server: Bloqueos de Páginas

SQL Server: Bloqueos de Índices

SQL Server: Bloqueos de Tablas

SQL Server: Promoción de Bloqueos de Página a Bloqueos de Tablas

SQL Server: Inserciones

InterBase: Arquitectura Multi-Generacional

InterBase: Rendimiento

Confirmación en dos fases

SQL Server

Sybase SQL server

InterBase

CHARS y VARCHARS

SQL Server: (Almacenamiento) CHAR vs. VARCHAR

InterBase (Almacenamiento) CHAR vs. VARCHAR

SQL Server (Rendimiento)

InterBase (Rendimiento)

SQL Server (Edición in situ de VARCHARS)

InterBase (Edición in situ de VARCHARS)

Matrices Multidimensionales

Procesamiento de Transacciones

SQL Server

InterBase

Instalación y Mantenimiento

Instalación y Reservación de Espacio

SQL Server (Instalación)

Sybase (Instalación)

InterBase (Instalación)

Mantenimiento (Copias de Respaldo)

SQL Server (Mantenimiento)

InterBase (Mantenimiento)

Registros de Transacciones

SQL Server

InterBase

Puntos de verificación

SQL Server

InterBase

Configuración

SQL Server

InterBase

Recuperación

SQL Server

InterBase

Interbase: Sombreado de Bases de Datos

Uso de Recursos

[Microsoft SQL Server](#)

[Sybase SQL Server](#)

[InterBase](#)

[Escalabilidad](#)

[Microsoft SQL Server](#)

[Sybase SQL Server](#)

[InterBase](#)

[Portabilidad](#)

[Microsoft SQL Server](#)

[Sybase SQL Server](#)

[InterBase](#)

[Tecnologías Avanzadas \(Alertadores de Eventos\)](#)

[Soporte para Java](#)

[Soporte Internacional](#)

Microsoft SQL Server

Sybase SQL Server

InterBase

## Introducción

Motorola, Nokia, MCI, Northern Telecom, la Bolsa de Philadelphia, Bear Stearns, el First National Bank de Chicago, Money Store, el Ejército de los Estados Unidos, NASA, Boeing... A pesar de lo variado de los nombres en esta lista, todos comparten algo en común: han seleccionado a InterBase como el componente fundamental de sus sistemas de información. InterBase puede encargarse con la misma facilidad de controlar sistemas de misiles de combate, de capturar datos para la investigación aerospacial o gestionar una bolsa de valores. Estas aplicaciones, aparentemente diferentes necesitan ciertas características comunes entre ellas: facilidad de uso y mantenimiento, alto rendimiento, escalabilidad, confiabilidad, disponibilidad de los datos, portabilidad, un adecuado uso de los recursos y recuperación tras desastres. InterBase ha sido diseñado para cumplir con todos estos objetivos.

Aunque no todas las empresas necesitan aplicaciones tan interesantes o dramáticas como las citadas, todavía comparten las mismas necesidades y deben suministrar soluciones reales a problemas reales del mundo empresarial. Las mismas virtudes que hacen de InterBase el mejor Sistema de Gestión de Bases de Datos Relacionales (SGDBR) para las aplicaciones mencionadas, son las mismas virtudes que hacen de InterBase la mejor elección para su uso por grupos de trabajos, departamentos y empresas en cualquier aplicación cliente/servidor. El objetivo de este documento es demostrar por qué InterBase es la mejor herramienta SGDBR para aplicaciones Cliente/Servidor examinando cómo se comportan InterBase, Microsoft SQL Server y Sybase en los aspectos señalados anteriormente y en otros más.

Nota.

Antes de la salida de Microsoft SQL Server 6.0, Sybase SQL Server y Microsoft SQL Server eran el mismo producto. Microsoft SQL Server 4 obtuvo una licencia de Sybase y fue vendido a partir de entonces bajo la etiqueta de Microsoft. En 1995, Microsoft compró el código fuente de Sybase y lo modificó para producir Microsoft SQL Server 6.0. Sybase continuó el desarrollo de su producto, que es comercializado ahora bajo los nombres de Sybase SQL Server System 10 y System 11. En el corazón de Microsoft SQL Server y los productos SQL de Sybase se esconde el mismo código fuente. En la mayoría de los casos, los productos se comportan de la misma manera. Por esta razón, el término "SQL Server" se referirá, a los propósitos de este documento, tanto a Microsoft SQL Server como a Sybase SQL Server. En los casos en que estos dos

productos difieran, se utilizarán sus respectivos nombres.

## Una breve historia

InterBase fue concebido y creado originalmente por un grupo de ex-empleados de la Digital Equipment Corporation [DEC], los cuales querían desarrollar un SGBDR innovador que ofreciera beneficios substancialmente mayores que otras bases de datos existentes. InterBase comenzó su vida en 1985 con el nombre de Groton Database Systems, siendo renombrado poco después como InterBase. Ashton Tate adquirió el producto en 1991, y Borland lo adquirió a su vez en 1992 como parte de la compra de Ashton Tate.

A lo largo de su desarrollo, InterBase ha introducido exitosamente una serie de avances tecnológicos. Entre estos se encuentra la Arquitectura Multi-Generacional, Confirmación en Dos Fases Automática, Sombreado de Bases de Datos, Objetos Binarios Grandes [BLOBs], Índices Dispersos con Representación Binaria, Matrices Multi-dimensionales, Alertadores de Eventos y el primer controlador nativo JDBC.

La mayor parte de los sistemas SGBDR existentes no pueden ofrecer tecnologías equivalentes. Por ejemplo, la arquitectura de SQL Server utiliza una combinación de bloqueos de páginas, tablas e índices para permitir el acceso concurrente a sus datos. SQL Server permite la confirmación en dos fases pero necesita ayuda por parte del programador para gestionar las secuencias de confirmación y de anulación. SQL Server ofrece acceso a BLOBs pero su alcance es muy limitado y es significativamente más lento que el acceso a BLOBs de InterBase.

## Mecanismos de Bloqueo

**Resumen:** *InterBase utiliza su Arquitectura Multi-Generacional para reducir enormemente la necesidad de bloqueos de páginas, índices y tablas, asegurando por lo tanto la mayor concurrencia posible, la disponibilidad de los datos y el mejor rendimiento en situaciones reales.*

**Antecedentes:** La integridad de los datos críticos de una empresa es de importancia incommensurable. Hay muchas formas de asegurar la integridad de los datos en un SGBDR. Dos de los esquemas más comunes son los de bloqueos pesimistas y los bloqueos optimistas. El bloqueo pesimista es la técnica más fácil de implementar y genera bloqueos en páginas, índices y tablas para asegurar la integridad de la transacción. Por otra parte, las técnicas optimistas de bloqueos no detienen a la aplicación cliente. Por el contrario, utilizan mecanismos más sofisticados para asegurar la integridad de la transacción. El bloqueo optimista, a la vez que asegura la integridad de los datos, ofrece el mejor rendimiento y disponibilidad de la información.

- **SQL Server: Bloqueos de Páginas.** Con el propósito de garantizar la integridad de los datos, la arquitectura de SQL Server utiliza un mecanismo de bloqueos que bloquea las páginas de datos. Una página de datos es una colección de bloques físicos de datos en el disco duro de un servidor. Las páginas tienen todas el mismo tamaño, el cual está determinado por la configuración del servidor y la base de datos. En dependencia del ancho de una tabla y del tamaño de la página, una página puede contener cualquier número de registros. Generalmente, los registros se disponen en una tabla con los registros añadidos recientemente al final de la misma. La unidad básica de almacenamiento es la página de 2K, que coincidentemente es la menor unidad de almacenamiento que SQL Server puede también bloquear. El bloqueo a nivel de páginas requiere que los desarrolladores estén más conscientes de la concurrencia y de la necesidad de ajustar su código para obtener la mayor concurrencia posible. Un bloqueo sobre una página inmoviliza todas las filas de datos o de índices existentes en esa página. Por ejemplo, en una tabla que contiene filas de datos de 100 bytes y claves de 10 bytes un bloqueo en estas páginas tendrá el efecto acumulativo de bloquear otras 18 filas de

datos y 180 entradas en el índice.

Para asegurar la integridad de un registro, cuando ocurre una escritura en una tabla, el SGBDR bloquea la página donde está ubicado el registro. Bloquear la página completa es más rápido que bloquear los registros individuales y requiere menos recursos en el servidor para gestionar los bloqueos. Esta técnica, sin embargo, impide el acceso de otros usuarios a la página completa. Para más problemas, sin embargo, SQL Server bloquea, además de la página en la que se encuentra el registro, las páginas adyacentes. Esto quiere decir que no solamente se bloquea la página de interés, sino también la precedente y la siguiente. El acceso a los registros en las páginas bloqueadas se impedirá durante la duración del bloqueo. El bloqueo puede ocurrir en una amplia variedad de situaciones. La más frecuente es mientras se escribe un registro en la base de datos. Sin embargo, el uso de cursores por las aplicaciones clientes puede causar un extenso bloqueo de páginas. En tales casos, los usuarios que están visualizando registros pueden impedir a otros usuarios que modifiquen registros en esas páginas hasta que la aplicación de visualización cierre el cursor o el usuario navegue fuera de la página, liberando el bloqueo.

El hecho de que el lector de un registro, página o página adyacente pueda impedir a otros usuarios la actualización de un registro en cualquiera de las páginas bloqueadas por el lector plantea potencialmente problemas importantes con el rendimiento para los usuarios que necesitan acceder a los registros en las páginas bloqueadas. En vez de poderse concentrar en la implementación de las reglas de negocios de la empresa, los desarrolladores deben malgastar mucho tiempo para la creación de mecanismos sofisticados de bloqueos en las aplicaciones clientes para asegurar que estas respondan adecuadamente a las páginas bloqueadas que encuentren durante el procesamiento. Esto aumenta la complejidad de la aplicación y los costos de desarrollo y mantenimiento.

- **SQL Server: Bloqueos de Índices.** Los índices de SQL Server sufren las mismas limitaciones que sus tablas asociadas, excepto que las implicaciones tienen mayor alcance. Los registros de una tabla se almacenan en una secuencia más o menos aleatoria. La excepción a esto es el uso de índices agrupados. Cuando se actualiza una página de datos, sus índices correspondientes necesitan también ser actualizados. Cuando se actualiza una página de datos, sus índices correspondientes también necesitan ser actualizados. Al igual que las tablas, los índices también se almacenan en páginas. Para actualizar una página de un índice, la página debe bloquearse primero. Tenga en cuenta que, debido a la distribución aleatoria de los datos en una tabla, la página bloqueada del índice puede referirse a cualquier número de páginas de datos. Esto quiere decir que las actualizaciones al índice de un único registro pueden bloquear o diferir las actualizaciones de los muchos registros a los que se refiere una sola página del índice. Tenga también en cuenta que si hay múltiples índices sobre la tabla, y si estos índices representan diferentes criterios de ordenación de registros, quedan afectadas páginas adicionales de índices y datos. En aplicaciones grandes, con muchos usuarios concurrentes, la incidencia sobre el rendimiento puede ser substancial.
- **SQL Server: Bloqueos de Tablas.** Para asegurar lecturas repetibles se necesitan vistas consistentes de una base de datos. Las lecturas repetibles garantizan que dentro del alcance de una transacción determinada, los datos implicados en la transacción permanecen sin alterar por otros usuarios mientras la transacción progresa. Esto es particularmente importante en las aplicaciones financieras o en aplicaciones en las cuales se ejecutan informes o consultas largas mientras otros usuarios actualizan registros. Para asegurar una visión consistente en una base de datos de Sybase o de Microsoft SQL Server, el desarrollador debe utilizar bloqueos de tablas. Un bloqueo de tabla coloca un bloqueo completo sobre una tabla y, en dependencia del tipo de bloqueo (Compartido, Actualización, Exclusivo) impide escrituras o lecturas de la tabla por otros usuarios durante la transacción. Un ejemplo en el cual esto es particularmente importante es en un informe de los registros asientos de contabilidad, donde es muy probable que todos los registros sean utilizados por el informe. Por definición, todos los balances en la tabla de asientos deben sumar cero. La

arquitectura de SQL Server exige que el desarrollador bloquee la tabla completa durante la ejecución del informe. En dependencia de la herramienta de informes utilizada, pueden necesitarse varias pasadas. Es muy probable que se utilicen otras tablas, que deben a su vez ser bloqueadas

- **SQL Server: Promoción de Bloqueos de Página a Bloqueos de Tablas.** Las transacciones largas que afectan a un gran número de registros necesitan un gran número de bloqueos de páginas. En la medida en que aumenta el número de páginas bloqueadas en una tabla, SQL Server promueve automáticamente los bloqueos de páginas a un bloqueo total de la tabla. El propósito de esta promoción es minimizar el trabajo que el SGBDR debe realizar para gestionar los bloqueos y mantener la concurrencia sobre los datos, mientras que mantiene el rendimiento en un nivel aceptable. Desafortunadamente, un bloqueo completo sobre una tabla también bloquea al resto de los usuarios de la tabla. Esta promoción puede ocurrir tanto para actualizaciones como para consultas. Para asegurar un rendimiento óptimo en las situaciones con múltiples usuarios, será necesario un administrador debidamente entrenado para que analice tanto el diseño de la base de datos como las aplicaciones clientes y los patrones de uso. El administrador debe entonces ajustar el nivel de promoción de bloqueos para intentar balancear los efectos de la gestión de un número elevado de bloqueos contra los efectos de impedir el acceso a otros usuarios cuando los bloqueos de páginas se transforman en bloqueos de tablas. Los programadores de aplicaciones deben también estar conscientes de las implicaciones de los bloqueos de páginas y tablas y del proceso de promoción y escribir código que responda adecuadamente a los bloqueos cuando se produzcan. Los requisitos de los desarrolladores para escribir este código de respuesta añade bastante complejidad a las aplicaciones, incrementando por consiguiente los costos de desarrollo y mantenimiento.
- **Soluciones:** Para poder lidiar con las numerosas limitaciones del mecanismo de bloqueo de SQL Server, los desarrolladores han tenido que idear parches para optimizar el balance entre concurrencia y rendimiento. Entre estas técnicas se encuentran:
  - Utilizar un administrador debidamente entrenado, y caro, para realizar un análisis exhaustivo de las aplicaciones clientes, de las estadísticas de la base de datos y de su diseño, en conjunto con una cuidadosa configuración y ajuste del gestor de bloqueos para establecer umbrales óptimos de promoción para cada base de datos.
  - Almacenamiento intermedio de los datos para abreviar la duración de las transacciones lo más posible, reduciendo por lo tanto la contención por bloqueos. Aumenta la complejidad de la aplicación, elevándose el costo asociado de desarrollo y mantenimiento. Los problemas de concurrencia surgen también cuando otros usuarios intentan actualizar registros asumiendo que sus datos en uso son los reales.
  - Movimientos regulares de datos para separar bases de datos en diferentes servidores cuando se están ejecutando informes largos. Este es el uso más frecuente de la replicación de SQL Server. Desafortunadamente, la replicación en SQL Server funciona en un solo sentido; es sencillamente una copia de datos de la base de datos maestra a la esclava. Esta limitación impide el uso práctico de la replicación para otras tareas. En Sybase, la replicación se logra con un producto separado: "Replication Server". La replicación en SQL Server es poco más que un método para sobreponerse a una deficiencia básica del producto.
  - Tablas temporales de trabajo para contener datos en actualizaciones en lote. Esto introduce otros problemas de concurrencia, ya que los datos que otros usuarios están utilizando no son necesariamente los actualizados.
  - Utilizar HOLDLOCK para colocar un bloqueo SHARE sobre elementos seleccionados, para eliminar la posibilidad de que otros usuarios actualicen dicho elemento. Con muchos usuarios en el sistema, es probable que los otros usuarios intenten actualizar los mismos datos o páginas de índices. Se pueden colocar varios bloqueos SHARE sobre los mismos elementos, bloqueando de este modo indefinidamente las actualizaciones. En tal escenario, el segundo usuario puede causar un abrazo mortal (*deadlock*) y uno de los dos

usuarios tendrá que romper este círculo vicioso. El uso de un bloqueo exclusivo evitaría el abrazo mortal pero impediría a otros usuarios la visualización de los elementos bloqueados. Una terminación anormal también dejaría bloqueos sin resolver hasta que el servidor pueda determinar que la conexión ha sido rota.

- **SQL Server: Inserciones.** Microsoft SQL Server 6.5 ha mejorado ligeramente su mecanismo de bloqueo con respecto a la versión 6.0 y a Sybase SQL Server, soportando bloqueos a nivel de fila durante las inserciones. Aunque esto mejora el rendimiento de las operaciones de inserción, no alivia ninguno de los otros problemas relacionados con los bloqueos sobre páginas, índices y tablas. De este modo, en la arquitectura de SQL Server, independientemente de la versión, las actualizaciones siguen necesitando bloqueos de páginas y tablas para asegurar la integridad de los datos.
- **InterBase: Arquitectura Multi-Generacional.** InterBase emplea una estrategia de bloqueo optimista a través de su Arquitectura Multi-Generacional [MGA]. El mecanismo MGA de InterBase crea versiones optimizadas de registros nuevos, borrados o actualizados que son solamente visibles dentro de la aplicación que provoca los cambios en los datos. En realidad, InterBase solamente crea versiones de las columnas cambiadas produciendo deltas. Esto asegura el mayor rendimiento posible y los requisitos mínimos de espacio.

El estado de invisibilidad de los registros versionados permanece solamente durante la ejecución de la transacción. Después de determinar la transacción, los registros cambiados se vuelven visibles para todas las transacciones que han comenzado después que la primera transacción terminase. De igual forma, todas las otras transacciones tienen su propia visión de los registros que alteran durante el curso de su transacción. Una vez que todas las transacciones que leen o alteran registros han terminado, InterBase desecha las versiones antiguas y todas las transacciones a partir de ese momento tienen la misma visión de esos registros. Cuando dos transacciones intentan modificar el mismo registro, la transacción que primero envía sus datos es la propietaria del registro, y se produce una excepción cuando el segundo registro es enviado. Los desarrolladores de SQL Server y los desarrolladores de InterBase tienen la posibilidad de trabajar con tales excepciones. Sin embargo, la aplicación sobre SQL Server debe primero releer el registro para asegurarse de que el registro no ha sido cambiado. En dependencia de la herramienta de desarrollo utilizada en la aplicación cliente, el desarrollador de SQL Server puede necesitar escribir código adicional para controlar este proceso. En vez de escribir código en la aplicación para manejar la muy común ocurrencia de bloqueos de páginas, índices y tablas, o verificar que el registro destino no ha sido cambiado, el desarrollador de InterBase programa las aplicaciones clientes para que manejen las menos frecuentes excepciones de bases de datos que son elevadas cuando un registro destino ha sido modificado por otra transacción. El desarrollador de InterBase obtiene importantes incrementos en productividad, rendimiento, concurrencia y disponibilidad de datos, así como una complejidad reducida. En última instancia esto significa un tiempo de desarrollo y unos costos de mantenimiento más reducidos para las empresas que utilizan InterBase.

- **InterBase: Rendimiento.** Las pruebas para determinar el rendimiento de una base de datos tienen en cuenta pocas veces los efectos de los mecanismos de bloqueos utilizados por el SGBDR. Los mecanismos de bloqueo de páginas y tablas de los servidores de Microsoft y Sybase pueden afectar adversamente el rendimiento cuando muchos usuarios necesitan acceder a la misma página de datos, o incluso a páginas adyacentes. Por ejemplo: en situaciones reales, el bloqueo de páginas empleado por SQL Server puede causar retrasos en la medida en que las aplicaciones clientes esperan por la liberación de los bloqueos. Estos efectos pueden ser bastante visibles en sistemas de grandes volúmenes de datos o en sistemas en los que se ejecutan consultas o informes largos mientras que otros usuarios actualizan registros. La arquitectura MGA de InterBase asegura que todos los registros están disponibles para todos los usuarios del sistema en todo momento. Las aplicaciones clientes nunca tienen que esperar que estén disponibles tablas, registros o índices, independientemente del número de usuarios del sistema o de la longitud y complejidad de las

transacciones que se estén ejecutando. Los desarrolladores de InterBase pueden, por lo tanto, suministrar aplicaciones con el mayor rendimiento posible en cualquier entorno de trabajo con la menor complejidad posible de desarrollo.

## Inicio

### **Confirmación en dos fases**

**Resumen:** InterBase ofrece el mayor nivel de concurrencia entre múltiples bases de datos con el menor esfuerzo de desarrollo, consiguiendo la mayor integridad de datos posible a la vez que minimiza los costos de desarrollo y mantenimiento.

Se dice que la palabra "Transacción" se deriva de la frase "Acción de Transformación". Una transacción es una acción o serie de acciones que transforman un sistema desde un estado consistente a otro. Por ejemplo: María utiliza un cajero automático para transferir 10.000 dólares desde su cuenta corriente y depositarla en su cuenta VISA; ha transformado (cambiado) los balancestanto de su cuenta corriente como de su cuenta VISA. Las transacciones se caracterizan por sus propiedades **ACID**:

**Atomicidad** - Se refiere a la propiedad todo o nada de la transacción. O la transacción se confirma como un todo o no realiza cambio alguno. Si la transacción falla en completarse exitosamente, las operaciones realizadas hasta ese punto son anuladas.

**Consistencia** - La transacción debe transformar el sistema desde un estado consistente a otro estado consistente. La consistencia se define mediante las reglas de negocio del sistema, y son reforzadas por la aplicación.

**Independencia** - Aunque pueden ocurrir múltiples transacciones concurrentes, cada transacción debe pensar que es la única en cada momento.

**Durabilidad** - Los efectos de una transacción confirmada deben ser permanentes.

En el ejemplo de María, todas las propiedades ACID deben cumplirse. Asumiendo que la información acerca de su cuenta corriente y la información sobre su cuenta VISA están en bases de datos separadas, debe tener lugar una transacción coordinada. Esto se controla normalmente por medio de confirmaciones en dos fases. Una confirmación en dos fases es un mecanismo mediante el cual la actualización de las bases de datos afectadas se trata como una transacción ACID. Las dos fases de este tipo de confirmación son la fase de preparación y la confirmación, propiamente dicha. Si, por algún motivo, el proceso falla durante la fase de preparación, esto es, después de que el dinero ha sido extraído de la cuenta corriente y no ha ingresado aún en la cuenta VISA, la transacción debe anularse. Esto asegura que el dinero extraído es devuelto. En otras palabras, hasta que la extracción y el depósito puedan ser confirmados, las operaciones en ambas bases de datos no son permanentes. Una vez que se han realizado las confirmaciones parciales, la transacción se confirma como un todo. Esto es lo que se conoce como confirmación en dos fases.

- **Microsoft SQL Server.** La arquitectura de SQL Server permite a los desarrolladores la programación de confirmaciones en dos fases utilizando The Microsoft Distributed Transaction Coordinator [MSDTC]. MSDTC necesita que los desarrolladores creen objetos de transacción utilizando objetos OLE. Esto fuerza a los desarrolladores a programar en otro lenguaje, como C++, para poder crear y gestionar confirmaciones en dos fases a nivel de empresa. Claro está, en la medida en que ocurren cambios en la aplicación, este código debe verificarse y habrá que darle mantenimiento a lo largo del ciclo de vida de la aplicación. Del mismo modo, si la aplicación se porta desde NT/Intel a NT/PPC, el código mismo debe adaptarse, recompilarse y volver a verificar. No sólo deben mantenerse la base de datos, el código de la aplicación y sus objetos, sino también los

objetos MSDTC. Esto limita la portabilidad de las aplicaciones de Microsoft SQL Server e impone gastos adicionales en desarrollo y mantenimiento. Para los distribuidores de software, esto es de suma importancia, debido a las múltiples versiones de código que deben ser desarrolladas, verificadas, mantenidas y documentadas, aún cuando hay un único sistema operativo. Además, ya que MS SQL Server solo está disponible sobre NT y tiene una implementación única para las confirmaciones en dos fases, no puede ser programado para manejar confirmaciones en dos fases entre sí mismo y las bases de datos de gestión de Sybase que se ejecutan sobre UNIX.

- **Sybase SQL Server.** En Sybase SQL Server, se anima a los desarrolladores a utilizar la confirmación en dos fases para asegurar la integridad de los datos cuando se necesitan transacciones que abarcan múltiples bases de datos. En todo el conjunto de los veinte manuales de Sybase System 10, hay una sola mención de pasada sobre las confirmaciones en dos fases. La página web de Sybase, en cambio, contiene cierta cantidad de documentos que esbozan cómo se pueden escribir confirmaciones en dos fases en C, FORTRAN, Pascal y COBOL. Cada uno de estos ejemplos tiene más de 100 líneas de código. A los desarrolladores se les obliga a escribir rutinas externas complejas, que no están documentadas en los manuales de Sybase, utilizando otros lenguajes, en vez de utilizar las capacidades internas del SGBDR. La migración a otra plataforma puede requerir la recompilación o reprogramación para la nueva plataforma. Como en el caso de Microsoft SQL Server, los ficheros externos deben mantenerse y sincronizarse a nivel de empresa, elevando los gastos de mantenimiento y desarrollo. Una vez más, los distribuidores están obligados a soportar múltiples bases de códigos. Sin embargo, el desarrollador de Sybase se enfrenta a un nivel adicional de complejidad, porque estas bases de código se distribuyen a través de varios sistemas operativos.
- **InterBase.** InterBase ofrece confirmaciones en dos fases automáticas que cumplen con todos los requisitos de una transacción ACID sin programación. Esto permite alcanzar la mayor concurrencia posible con el menor esfuerzo.

### Inicio

## CHARS y VARCHARS

**Resumen:** InterBase utiliza técnicas óptimas para el almacenamiento de datos, y tipos de datos, más flexibles, necesitando significativamente menos almacenamiento, menos complejidad y mayor rendimiento.

- **SQL Server: (Almacenamiento) CHAR vs. VARCHAR.** Hay dos tipos principales de caracteres que se utilizan virtualmente en todas las bases de datos SQL cliente/servidor. El primero es un tipo de datos de longitud fija, conocido como CHAR. En la mayoría de los SGBDRs, un CHAR ocupa un espacio constante en la base de datos, sin tener en cuenta el largo real de los datos. Cualquier espacio que sobre se rellena al final con espacios en blanco. El tipo CHAR es particularmente útil para datos que no varían en longitud, como los códigos de países y estados. Los CHARs pueden también utilizarse para columnas más anchas, como nombres y direcciones, pero de este modo se desperdician grandes cantidades de espacio en la base de datos.

El tipo VARCHAR es un tipo de datos de caracteres de longitud variable. Ofrece un método para almacenar datos de caracteres de longitud variable. La longitud máxima de un VARCHAR se asigna para cada columna VARCHAR cuando se crea la base de datos. En el momento en que se almacenan datos en un columna VARCHAR, el SGBDR determina cuánto espacio debe reservarse para acomodar el dato, y solo este espacio se asigna. Los VARCHARs ofrecen un método eficiente en espacio para almacenar datos basados en caracteres cuya longitud dentro de una misma columna varía de fila a fila.

**Limitaciones de tamaño.** Las columnas de tipo VARCHAR de SQL Server están limitadas a

255 bytes de datos. Si se necesitan cadenas mayores que 255 bytes, debe utilizarse el tipo de datos TEXT. El tipo TEXT se almacena como datos BLOB en segmentos de 2K. En otras palabras, independientemente de si la columna de tipo TEXT utiliza un byte o 1.500 para una fila en particular, SQL Server debe reservar un bloque de 2K para esa fila. SQL Server ofrece acceso a BLOB penalizando el rendimiento. Los desarrolladores deben planificar cuidadosamente sus diseños de bases de datos para lograr el balance óptimo entre almacenamiento y rendimiento, y pueden verse forzados, en situaciones en que una columna contenga regularmente más de 255 bytes pero mucho menos de 2.000 bytes, a combinar dos columnas VARCHAR y a escribir código adicional para recombinar los datos.

- **InterBase (Almacenamiento) CHAR vs. VARCHAR.** Como la familia de bases de datos de SQL Server, InterBase soporta los dos tipos de datos CHAR y VARCHAR. Superficialmente, se comportan del mismo modo que en SQL Server. Esto asegura la compatibilidad con el software cliente que espera que los datos tengan longitud variable o fija. Internamente, InterBase implementa estos tipos de datos de modo diferente a la arquitectura SQL Server. En InterBase, CHAR y VARCHAR se almacenan exactamente de la misma manera. Todos los espacios al final de un valor se eliminan y el resto es lo que se almacena en la base de datos. En el caso de los datos VARCHAR, cuando se solicitan al servidor, el dato de longitud variable es pasado al cliente. En el caso de datos CHAR, InterBase completa los datos con espacios en blanco para asegurar la devolución de los datos con el tamaño correcto. Además, InterBase utiliza un algoritmo de compresión para optimizar la cantidad de espacio requerida para el almacenamiento de los tipos CHAR y VARCHAR.

**Tamaño de los VARCHAR.** El ancho máximo del tipo VARCHAR de InterBase es de 32K. Los desarrolladores de InterBase disfrutan, en consecuencia, de todos los beneficios de los tipos de datos VARCHAR sin estar limitados a 255 caracteres. Esta posibilidad es extremadamente valiosa para los desarrolladores que desean buscar y manipular largas cadenas de texto, tales como campos MEMO, sin tropezar con las limitaciones impuestas para el uso de campos BLOB (como las de Sybase y Microsoft). Si el MEMO puede sobrepasar el límite de 32K, solamente InterBase ofrece un tipo de datos BLOB extremadamente eficiente, con tamaños de segmentos especificables por el desarrollador. En otras palabras, si el desarrollador espera grandes cantidades de textos en incrementos regulares de 512 bytes, el tamaño de segmento de BLOBs de InterBase puede optimizarse para acomodar los datos, en vez de manipular los datos para que se acomoden al método de almacenamiento.

- **SQL Server (Rendimiento).** Teniendo en cuenta la evidente ventaja de la flexibilidad y almacenamiento eficiente ofrecido por el tipo de dato VARCHAR, puede parecer que las columnas VARCHAR son la elección indudable para el almacenamiento de datos basados en caracteres. Sin embargo, el uso de tipos VARCHAR en SQL Server inflige una penalización en el rendimiento que es el resultado de la traducción que debe realizarse. De este modo, los desarrolladores en SQL Server deben elegir entre almacenamiento o recuperación eficiente de los datos.
- **InterBase (Rendimiento).** Por el contrario, debido a que las estructuras internas de ambos tipos es idéntica en InterBase, el desarrollador nunca necesita realizar este tipo de decisiones. En cambio, el desarrollador simplemente puede basar su decisión en qué tipo de datos es el más adecuado para aceptar o devolver datos a la aplicación cliente. El desarrollador tampoco tiene que preocuparse acerca de las técnicas óptimas de almacenamiento, gracias a los algoritmos de compresión de datos de InterBase y su soporte para grandes columnas de tipo VARCHAR.
- **SQL Server (Edición in situ de VARCHARS).** El espacio reservado para el almacenamiento de los valores individuales de tipo VARCHAR se determina cuando el registro se escribe por primera vez en la base de datos. Cualquier edición posterior de una columna VARCHAR probablemente una cantidad diferente de espacio para almacenar los datos. La alteración del tamaño de los datos provoca que SQL Server reasigne espacio basándose en su nuevo tamaño. Por ejemplo, si un usuario final quiere algo tan sencillo como actualizar una lista de distribuidores mayoristas, y modifica una

línea de dirección de "6475 Christie Avenue" a "100 Borland Way", SQL Server debe borrar el registro y añadirlo al final de la tabla para realizar la modificación. Hay una serie de cuestiones asociadas con este proceso:

1. **Registro de Transacciones:** Cada borrado o inserción de un registro afecta al registro de transacciones de la base de datos, al registrar tanto un borrado como una inserción. El mantenimiento de los registros de transacciones requiere acciones por parte del administrador de la base de datos, para limpiar este registro de forma regular. Los registros de transacciones impropios o irregularmente mantenidos pueden desbordarse y provocar un fallo de la base de datos. Para pequeñas instalaciones sin administradores a tiempo completo que realizan el mantenimiento de la base de datos, los resultados pueden ser desastrosos.
  2. **Transacciones Largas:** Las transacciones largas que actualizan una gran cantidad de valores de tipo VARCHAR pueden bloquear las últimas páginas de una tabla por períodos extensos de tiempo. Una transacción que actualiza una columna de tipo VARCHAR en cada registro de una tabla puede conseguir que cada registro sea borrado y añadido al final de la tabla. Esto puede provocar un crecimiento del registro de transacciones a un ritmo superior al de los datos básicos.
  3. **Espacio Libre:** El espacio ocupado por los registros eliminados permanece sin recuperar hasta que se realice el mantenimiento de la base de datos. El resultado es una tabla que puede fácilmente ocupar más del doble de su espacio original, junto a un registro de transacciones que ha crecido para reflejar los borrados e inserciones.
  4. **Transacciones Fracasadas:** La adición de los registros actualizados mueve el registro modificado hacia el final de la tabla. Esta acción bloquea las últimas páginas de la base de datos hasta que la transacción finalice. Otros usuarios que estén creando registros o estén editando columnas VARCHAR en cualquier otro lugar de la base de datos también intentarán escribir a las últimas páginas de la base de datos. En la arquitectura de SQL Server, también se presentan situaciones en las cuales los lectores de datos pueden bloquear a los escritores. De este modo, otros usuarios de la base de datos pueden incluso bloquear la actualización de la columna de tipo VARCHAR debido a los bloqueos de páginas impuestos a las últimas páginas, provocando el fallo de la transacción.
  5. **Rendimiento:** Las restricciones de acceso a las páginas bloqueadas resultan en un menor rendimiento, con usuarios que tienen que esperar por la liberación de las páginas bloqueadas. Con bases de datos de grandes volúmenes de información que contienen grandes cantidades de datos de tipo VARCHAR, la degradación del rendimiento es significativa.
- **InterBase (Edición in situ de VARCHARS).** InterBase reasigna espacio para los valores de tipo VARCHAR al vuelo. Esto quiere decir que el proceso de borrado e inserción utilizado por SQL Server no ocurre en InterBase. Esto implica un trabajo menor para el servidor cuando se realizan actualizaciones a columnas VARCHAR, lográndose un mejor rendimiento. Además, al no existir un registro de transacciones que mantener, y que pueda potencialmente provocar un fallo de la base de datos, hay considerablemente menos trabajo para un administrador especializado. El problema de los bloqueos de páginas planteado por la arquitectura de SQL Server es también evitado por completo, lográndose la mayor disponibilidad posible de los datos.

[Inicio](#)

## Matrices Multidimensionales

InterBase ofrece un tipo de datos único conocido como Matrices Multi-Dimensionales [MDA]. Este tipo de datos no está disponible en otros SGDBR. El tipo de datos MDA permite que el desarrollador almacene matrices de virtualmente cualquier tamaño, hasta 16 dimensiones. Ya que un valor MDA se almacena en un solo campo, únicamente se necesita el acceso a una sola fila y columna para recuperar el dato. Los tipos MDA suministran un método potente de captura y representación de datos en una forma que sería difícil, si

no imposible, utilizando la arquitectura de SQL Server. La razón fundamental es el rendimiento. Considere que un conjunto de datos en particular debe ser representado como una matriz de dimensiones 100x100x100; el número total de elementos de datos sería 1.000.000. Para escribir estos datos al disco del servidor se necesitarían un millón de actualizaciones, más las correspondientes actualizaciones a los índices. Una lectura de estos datos requeriría también un millón de operaciones de lectura. Con un tipo MDA, solamente un registro es necesario para la lectura o la actualización. Además, si un valor matricial es nulo, InterBase no asigna espacio para este dato. En términos relacionales, el acceso a un conjunto de datos con un lado de la relación que no tenga un valor correspondiente requiere el uso de encuentros externos (outer joins) en cualquier consulta que acceda a esta relación. En la mayoría de los SGBDR este tipo de operaciones se realizan con un importante costo de rendimiento. Los tipos MDA no se acceden de esta manera, por lo cual no sufren estos problemas.

Bear Stearns es una firma de corredores de bolsa que utiliza el alto rendimiento de los tipos MDA para obtener ventajas en su negocio. Este tipo de negocios implican la compra de acciones en una bolsa y su venta rápida en otra, aprovechando el pequeñísimo precio diferencial como margen de ganancias. Ya que las variaciones de precios entre las distintas bolsas se coordinan muy estrechamente, la clave para triunfar reside en la habilidad para aprovechar estas pequeñas diferencias antes de que se sincronicen los precios. Los tipos MDA de InterBase ofrecen el rendimiento necesario para aprovechar estas breves ventanas de oportunidad.

Cierto fabricante líder de la industria aeroespacial también utiliza InterBase para capturar datos en tiempo real a partir de disposiciones tridimensionales de micrófonos colocados en pistas de despegue para recoger niveles de ruidos de reactores al aterrizar y despegar. Utilizando tipos MDA, se consiguen intervalos de muestreo extremadamente cortos, ofreciendo una visión muy detallada de los niveles de presión del aire. Igualmente, el análisis de estos datos puede realizarse con mucha velocidad ya que solamente hay que leer para cada instante un registro de la base de datos.

El alto rendimiento y las ricas posibilidades de representación representadas por los tipos MDA permite que los desarrolladores ofrezcan soluciones que serían imposibles con otros SGBDR.

## Inicio

### **Procesamiento de Transacciones**

**Resumen:** InterBase tiene la arquitectura óptima tanto para transacciones complejas cortas como para las largas, ofreciendo poca complejidad en las aplicaciones clientes, la máxima disponibilidad de los datos, concurrencia e integridad.

**Modelos de Transacciones:** La industria de bases de datos ha desarrollado varios modelos para explicar el tipo de entorno en el cual operan los SGBDR.

**OLTP.** Una aplicación de Procesamiento de Transacciones En Línea [OLTP] se caracteriza por un entorno similar al entorno en que trabaja un cajero de un banco. En este tipo de escenario, los cajeros envían una serie de transacciones pequeñas de corta duración al servidor. La aplicación puede necesitar un pequeño informe para la actualización de la libreta del usuario. Los informes largos y las copias de respaldo se realizan normalmente durante los momentos de poco uso de la base de datos.

**DSS.** Los Sistemas de Asistencia a Decisiones [DSS] están diseñados para soportar transacciones largas tales como informes de resumen o análisis estadísticos. Este tipo de sistema depende de una imagen relativamente estática de la base de datos para asegurar la integridad de los datos mientras el proceso largo se ejecuta.

**OLCP.** El Procesamiento Complejo En Línea [OLCP] es una mezcla de los modelos OLTP y DSS. En vez

de concentrarse en los aspectos básicos del modelo OLTP o en los igualmente importantes problemas de los informes y análisis [DSS], este modelo intenta alcanzar un balance entre estos otros dos, y se enfrenta a las necesidades reales de la mayoría de las empresas. Estos requisitos reflejan la necesidad de obtener un alto rendimiento, una completa disponibilidad de los datos, la capacidad de realizar copias de respaldo en línea, y la ejecución de grandes consultas e informes largos mientras los usuarios están actualizando la información diaria de la empresa. La información debe estar disponible en todo momento, sin límite de acceso a los usuarios OLTP y DSS.

- **SQL Server.** La arquitectura de SQL Server ha sido diseñada para soportar los entornos OLTP y DSS; pero no ambos. Esta arquitectura no permite el más frecuente entorno de tipo OLCP, bajo el cual muchas empresas funcionan. Esta limitación se debe fundamentalmente a los mecanismos de bloqueos utilizados por SQL Server.

**Un Escenario Característico.** Considere una tabla cuyos registros son una serie de balances mensuales de contabilidad. Por definición, el total de todos los balances debe ser igual a cero. Esto es: la suma de los créditos debe ser igual a la suma de los débitos. Suponga también que actualmente hay dos usuarios en el sistema: un analista y un grabador de datos. Si el analista desea ejecutar un informe largo, como un balance de prueba para todos los períodos, la consulta que extrae los datos debe leer cada registro de la tabla. Si el analista comienza la consulta y el grabador de datos realiza una actualización (recuerde que el débito total debe igualar al crédito total) y modifica un registro al principio de la tabla después de que la consulta del analista haya explorado esa porción de la tabla, y después modifica un registro al final de la tabla que no haya leído aún la consulta del analista, sucederá que el analista verá solamente la segunda mitad de la transacción del grabador. El total obtenido por el analista estaría fuera de balance por una cantidad igual a la del segundo registro insertado por el grabador de datos.

En un escenario de este tipo, el total al final del informe será evidentemente incorrecto. En muchas situaciones, esto representa sólo un pequeño inconveniente y el analista simplemente volvería a ejecutar el informe. Como puede haber más de una entrada en la transacción correspondiente a la persona que está introduciendo datos, y por cuanto puede haber otros usuarios accediendo simultáneamente a la base de datos, es totalmente posible que ocurran transacciones que alteren el nivel de detalle del informe sin alterar los totales. Algunos informes pueden requerir dos pasadas a la base de datos; una primera para extraer los registros de detalle, y la segunda para obtener los sumarios. Los errores en el informe serán menos obvios porque los cambios serían enmascarados por las transacciones complementarias. Una situación más seria todavía podría presentarse al gestionar información procedente de experimentos científicos o procesos industriales.

En la arquitectura de SQL Server, el único modo de garantizar la integridad y la repetibilidad de una lectura es bloquear la tabla completa. Un bloqueo de este tipo impide el acceso a la tabla a otros usuarios. En el escenario anterior, SQL Server promocionaría automáticamente los bloqueos de página a un bloqueo completo de la tabla. La promoción de bloqueos, sin embargo, no ocurriría hasta que se hubieran leído suficientes páginas. Del mismo modo, el acceso de otros usuarios a la tabla mediante actualizaciones podría bloquear la promoción de bloqueos y provocar que la consulta fallase o tuviese que esperar. El desarrollador deberá entonces estar consciente de las implicaciones del proceso de escalado de bloqueos y comprobar manualmente y aplicar bloqueos en la medida necesaria. Bloquear una tabla completa en un entorno de producción 7 días \* 24 horas puede hacerse muy difícil, si no, imposible, debido a la imposibilidad de obtener los accesos apropiados en transacciones largas o la inaceptabilidad de realizar transacciones largas, al bloquear el acceso de otros usuarios. Muchos desarrolladores no tienen otra posibilidad que programar la impresión de informes largos en tiempos "muertos", donde será mayor la posibilidad de obtener un bloqueo completo o la imposición del bloqueo de una tabla completa no afecte a otros usuarios.

Los desarrolladores deberán asimismo perder un tiempo valioso en crear sistemas de gestión de bloqueos para asegurar que otros procesos no fallarán cuando encuentren una tabla bloqueada. Frecuentemente, el estado de bloqueo no puede ser determinado sin que el usuario intente emitir una transacción. Los desarrolladores deberán entonces repetir el intento de transacción hasta que las tablas bloqueadas queden disponibles, o escribir complejas rutinas de 'caching' locales en los puestos clientes, lo cual tampoco es ideal por varias causas, entre ellas que múltiples usuarios tienen vistas inconsistentes de la base de datos.

En los entornos de negocios altamente competitivos de hoy, la falta de datos exactos equivale a la pérdida de oportunidades, y el mecanismo de bloqueo de la arquitectura de SQL Server causa un impacto negativo sobre dichas características.

- **InterBase.** "El soporte a la toma de decisiones de esta naturaleza requiere una arquitectura modular y flexible, que brinde soporte tanto al procesamiento distribuido como a las bases de datos distribuidas. Por ello hemos elegido InterBase. Este se ha comportado mejor que la competencia y nos ha convencido de que será confiable en situaciones de vida o muerte."

InterBase soporta plenamente el modelo OLTP. Su exclusiva arquitectura Multi-Generacional garantiza que todos los usuarios de tipo OLTP no encontrarán bloqueos cuando actualicen sus datos mientras que a los usuarios de tipo DSS se le garantizan lecturas repetibles. En el escenario anterior: cuando el analista comienza su transacción, InterBase crea un identificador de transacción para la misma. Cuando el grabador de datos comienza su transacción, InterBase asigna a esta un identificador de transacción diferente. Si se encontrase alguna transacción activa en el momento en que su transacción comenzara, InterBase se asegurará de la repetibilidad de las transacciones encontradas, manteniendo versiones separadas de cualesquiera registros alterados o leídos por cualquiera de esas transacciones. Durante la transacción, el analista ve sólo las versiones de registros asociadas a su transacción. Durante la transacción, el grabador de datos ve solamente las versiones de los registros modificados que tienen un identificador de transacción asociado a su transacción. Al final de ambas transacciones, los registros iniciales que eran visibles y fueron usados en el informe del analista serán eliminados de la tabla por InterBase, y los cambios hechos por el grabador de datos se harán visibles al analista.

Cualesquiera otros registros modificados durante el curso de cualquiera de las dos transacciones serán gestionados de la misma manera por InterBase. Este es su comportamiento por defecto, y los desarrolladores no tendrán que hacer nada para beneficiarse de él. Además, ningún usuario encontrará bloqueos de tablas, página o índices en ningún momento. En vez de ello, InterBase suministrará versiones de registros consistentes a lo largo de la duración de la transacción. La Arquitectura Multi-Generacional garantiza la repetición de una lectura de datos, así como de la disponibilidad de los datos en todo momento. Este resulta en la disminución de la complejidad de las aplicaciones y el tiempo de desarrollo, así como el acceso rápido, exacto, y eficiente a los datos corporativos en todo momento.

[Inicio](#)

## Instalación y Mantenimiento

**Resumen:** InterBase posee una arquitectura que requiere significativamente menos mantenimiento que Microsoft SQL o Sybase SQL Server. Ello resulta en menores costos de personal, entrenamiento y mantenimiento, así como la máxima disponibilidad y el mínimo de tiempo "muerto".

El uso óptimo de los recursos es fundamental para la supervivencia y rentabilidad de cualquier organización. Los recursos de grupos de trabajo, departamentos y pequeñas organizaciones están

frecuentemente más limitados y su gestión es, en consecuencia, mucho más importante. Este tipo de entornos pocas veces tienen recursos tales como administradores de bases de datos dedicados a tiempo completo, como los que se pueden encontrar en grandes departamentos de informática. Ni tienen grandes presupuestos para el desarrollo de aplicaciones y compra de hardware. De este modo, las bases de datos utilizadas en tales entornos deben ser fácilmente instalables y mantenibles por personas que no son necesariamente administradores especializados de bases de datos. Los distribuidores, y las organizaciones que se comportan como tales, son más sensibles a los costos de instalación y mantenimiento debido a su falta de capacidad para ofrecer un servicio de administración a tiempo completo en el lugar, para vigilar la base de datos.

Del mismo modo, muchas empresas ya se han normalizado en relación a sus sistemas operativos para redes. La introducción de un nuevo sistema operativo dentro de la mezcla de plataformas de servidores puede no ser posible debido a estándares empresariales; aún cuando sea posible no es siempre deseable por las siguientes razones:

- Cuando se introduce un nuevo sistema operativo para redes, se necesita que las nuevas habilidades sean aprendidas por las personas a cargo de su administración.
- Se introducen nuevos problemas de seguridad y se necesita mantenimiento y soporte adicional.

En pocas palabras, las bases de datos de departamentos y de grupos de trabajo deben operar en plataformas existentes de servidores en vez de forzar la adopción de nuevas plataformas.

## Inicio

### **Instalación y Reservación de Espacio**

**Resumen:** InterBase ofrece el proceso de instalación más sencillo, independientemente de la plataforma, haciéndola la opción ideal para empresas y desarrolladores.

**Antecedentes:** Los requisitos de instalación de un SGBDR varían radicalmente en dependencia del fabricante y del sistema operativo empleado. Algunos sistemas funcionan sin modificar el sistema operativo del servidor, y otros necesitan cambios en el núcleo antes o durante la instalación del software de bases de datos. Claramente, los desarrolladores quieren asegurar el mejor rendimiento y confiabilidad de sus aplicaciones cliente/servidor, por lo que las modificaciones a un nivel tan elemental pueden aumentar el riesgo de fallos e incompatibilidades.

**Reservación de Espacio.** Antes de la instalación, SQL Server y Sybase SQL Server necesitan reservar espacio para el SGBDR y para las bases de datos. La base de datos o el administrador deben reservar el espacio necesario para la base de datos. Si la base crece por encima de lo inicialmente esperado debido a la adición de registros durante el transcurso de una consulta, la operación terminará. Pueden ocurrir varios problemas, como:

- Las consultas largas, que fuerzan la escritura de las tablas temporales al disco.
- Registros de transacciones que se desbordan.
- La adición de más datos de los que permite el espacio reservado.
- Espacio libre excesivo debido a un gran número de ediciones de VARCHAR que provocan los borrados y las adiciones.

- Espacio libre excesivo debido a un gran número de borrados.

**Determinando el espacio necesario.** Determinar el espacio requerido para las bases de datos de SQL Server puede ser un proceso complejo y dependiente de muchos factores. El espacio reservado debe ser suficientemente amplio para tablas de sistema, tablas de usuario, y todos los índices. Algunos de los factores a tener en consideración son:

- De un 20% a un 30% adicional deberá ser calculado y reservado para acomodar el registro de transacciones.
- Añada un 150% extra (en base al tamaño de la tabla) por cada tabla que requiera un índice clusterizado.
- Añada un 5% del tamaño de cada tabla para espacio interno del sistema (por ejemplo, punteros internos).
- Añada 2K adicionales por fila por cada columna de texto o imagen, incluso si la columna está vacía.
- Añada espacio extra en las tablas y los registros de transacciones para las adiciones y eliminaciones resultantes de actualizaciones de VARCHARs.
- Añada espacio extra para tablas cuyas filas son eliminadas frecuentemente.
- **Microsoft SQL Server (Instalación).** Microsoft SQL Server se encuentra disponible solamente sobre Windows NT. Esta falta de disponibilidad sobre sistemas operativos de 64 bits, la falta de capacidad por parte del sistema operativo para manejar más de cuatro CPUs, y la incapacidad del software para ejecutarse sobre los procesadores RISC más avanzados limita necesariamente las posibilidades de los desarrolladores de distribuir soluciones multiplataformas basadas en Microsoft SQL Server.

Todas las bases de datos de Microsoft SQL Server requieren la reservación del espacio suficiente al ser instaladas y un mantenimiento regular para controlar el espacio utilizado, reclamar el espacio libre y asegurar que hay suficiente espacio disponible. La mala gestión del espacio puede traer serias consecuencias. Los problemas de mantenimiento e instalación para monitorear el espacio utilizado hacen a Microsoft SQL Server una opción menos atractiva como servidor de departamento o grupo de trabajo, especialmente cuando se cuenta con recursos limitados. Estos problemas son igualmente una preocupación de los desarrolladores, cuyos clientes pueden ser incapaces de garantizar la disponibilidad de un Administrador azevado para asegurar el mantenimiento adecuado del sistema.

- **Sybase (Instalación).** Sybase se encuentra disponible sobre una variedad de plataformas, incluyendo a Windows NT, Novell NetWare y varias plataformas UNIX. Al igual que Microsoft SQL Server, la instalación de una base de datos Sybase requiere la asignación previa de espacio para el SGBDR y las bases de datos iniciales y sufre, por lo tanto, de los mismos problemas relacionados con el registro de transacciones, ediciones de valores VARCHAR, borrados de registros, carga masiva de datos y consultas largas.

Adicionalmente, con vistas a obtener el máximo rendimiento sobre plataformas UNIX [incluyendo SCO], el administrador tiene que crear particiones. Cuando la base de datos es instalada en una partición, el sistema operativo del servidor es cortocircuitado en las operaciones con el disco y el SGBD controla directamente toda la entrada/salida de la base de datos. Aunque se obtiene mejoras de eficiencia, se introduce una complejidad adicional de instalación y mantenimiento. Como el sistema operativo no tiene acceso a esa partición, las

utilidades UNIX tampoco, haciendo muy difícil la recuperación en caso de desastres. Sobre plataformas UNIX, Sybase hace modificaciones al núcleo del sistema operativo, lo cual provoca preocupación sobre incompatibilidades y fallos al actualizar el SGBD o aplicar parches o actualizaciones al sistema operativo.

Sybase SQL Server está sujeto a los mismos problemas de reservación de espacio que Microsoft SQL Server. Sobre plataformas UNIX igualmente existe la complejidad de las particiones directas y las modificaciones del núcleo del sistema operativo. El resultado es que Sybase es una opción menos atractiva como servidor de departamento o grupo de trabajo sobre todas las plataformas, especialmente UNIX.

- **InterBase (Instalación).** La instalación de InterBase es extremadamente sencilla. InterBase asigna espacio en disco de forma completamente dinámica y automática cuando es necesario. Esto significa que cuando se instala InterBase el espacio en disco no es reservado específicamente para él o para alguna base de datos. Durante el uso de una base de datos, InterBase automáticamente realoja el espacio libre en la medida necesaria, sin intervención de un administrador. Adicionalmente, InterBase no necesita mantener registros de transacciones.

Como InterBase no necesita modificaciones a su núcleo en ninguna plataforma, raramente experimenta problemas de compatibilidad entre cambios de versión de los sistemas operativos. Esto permite al desarrollador mantener el sistema operativo al máximo nivel de estabilidad y robustez, sin temor de reinstalar el SGBDR como resultado de actualizaciones o parches al núcleo del sistema operativo.

## Mantenimiento (Copias de Respaldo)

**Resumen:** InterBase ofrece el menor costo de mantenimiento así como una superior confiabilidad y disponibilidad de los datos al disponer de procesos de respaldo y restauración significativamente más sencillos.

- **SQL Server (Mantenimiento).** Todas las bases de datos deben ser resguardadas regularmente (usualmente por las noches). Se pueden realizar dos tipos de respaldos sobre una base de datos de SQL Server: total o incremental. Un respaldo total (normalmente conocido como vaciado -dump-) crea una imagen completa de la base de datos incluyendo las tablas de sistema y los registros. Un respaldo incremental solamente hace una copia del registro de transacciones. Es importante que el administrador de la base de datos haga una planificación que combine tanto respaldos totales como incrementales, por cuando una copia de respaldo completa NO trunca el registro de transacciones. A pesar de que un administrador con poca experiencia puede pensar que tiene la base de datos bien protegida con respaldos totales periódicos, puede producirse un desastre si el registro de transacciones no es limpiado regularmente por respaldos incrementales. Un fallo en hacer esto puede provocar que el registro de transacciones se desborde y la base de datos se venga abajo. Tales hechos requieren de la intervención manual de un administrador experimentado para rectificar el problema, y deja a los usuarios sin acceso a la base de datos hasta tanto se restablezca la misma.

Las copias de respaldo y la restauración de las mismas consume tiempo. Los respaldos incrementales aseguran la menor ventana de vulnerabilidad de los datos. Sin embargo, frecuentes respaldos incrementales mezclados con respaldos totales poco frecuentes resultan en largos tiempos de recuperación que introducen un riesgo adicional debido a la pérdida, daño o colocación en orden incorrecto de cintas. El administrador de un SQL Server debe encontrar el balance óptimo entre todas estas variantes para asegurar una disponibilidad máxima, funcionamiento óptimo y el menor chance posible de corrupción de los datos. Las copias completas requieren acceso exclusivo a la base de datos, obligando a sacarla de línea para

realizar el respaldo.

- **InterBase (Mantenimiento).** Debido a la arquitectura Multi-Generacional de InterBase, las bases de datos pueden ser copiadas en cualquier momento, mientras cualquier número de usuarios está accediendo a los datos simultáneamente. Un backup de InterBase es una transacción y es tratada exactamente igual que otra transacción cualquiera. La Arquitectura Multi-Generacional suministra una vista instantánea de la base de datos en el momento en que comienza la copia. Durante la misma, InterBase crea nuevas "versiones" de los registros actualizados o eliminados mientras se realiza la copia. El proceso de copia no verá las nuevas versiones de registros durante el transcurso de la copia, por tanto éstas son parte de otras transacciones. Esto garantiza una visión atómica y consistente de toda la base de datos, independientemente del momento en que el resguardo se realiza. InterBase utiliza un solo tipo de backup: el total. Debido a la arquitectura Multi-Generacional de InterBase, la copia es considerada sólo como una transacción más. Una copia de resguardo completa contiene una visión instantánea de toda la base de datos, negando la necesidad de cualquier tipo de respaldo incremental. Esto simplifica grandemente el proceso de copia y restauración y minimiza el impacto de las cintas perdidas o dañadas.

El impacto de una copia de resguardo sobre la productividad del sistema es similar a la de cualquier otra transacción de larga duración. Por lo tanto, en lugar de obligar a que las copias de resguardo se realicen por las noches, esta posibilidad única garantiza la creación de copias de resguardo en cualquier momento.

## Registros de Transacciones

- **SQL Server.** Un registro de transacciones (Transaction log) es una tabla de nivel de sistema que contiene una lista de todas las modificaciones realizadas a todos y cada uno de los objetos de la base de datos. Igualmente contiene la información necesaria para mantener la integridad de los datos relacionados a esos cambios. El espacio necesitado por el registro de transacciones puede ser considerable, así como difícil de predecir. Considere que el registro deberá contener información sobre cada fila actualizada más cada fila de índice afectada. Considérese igualmente que probablemente existirán varios índices por cada tabla. La relación del espacio requerido para el registro en relación con el espacio requerido por los datos varía en función del ancho de las filas. Para filas "estrechas", esta relación puede llegar a ser de hasta 10:1. Las filas más anchas requieren proporcionalmente menos espacio. Otros elementos que afectan la cantidad de espacio necesario para el registro son acciones como la carga o eliminación masiva de registros, así como la edición de VARCHARs, debido a que la edición de columnas VARCHAR afecta no sólo a la columna en cuestión, sino que introduce en el registro una eliminación y una inserción.

La planificación de las copias es otro factor que influye, por cuanto los backups incrementales son el mecanismo utilizado por SQL Server para limpiar el registro de transacciones. Otros factores pueden afectar el espacio necesitado por el registro, como las transacciones que ejecutan durante largo tiempo que actualizan grandes volúmenes de datos.

La localización del registro de transacciones es asimismo crítica. Cualquier base de datos que no tenga su registro de transacciones separado en un dispositivo independiente no podrá volcar el registro de transacciones. Como la base de datos maestra deberá residir totalmente en el dispositivo maestro, el Administrador deberá manualmente "limpiar" el registro de transacciones maestro para evitar que se desborde y destruya la base de datos maestra.

- **InterBase.** InterBase no utiliza registro de transacciones para mantener la información sobre las transacciones confirmadas, pero no escritas. En vez de ello, mantiene la información en Páginas de Información de Transacciones (Transaction Information Pages [TIP]). Esta información determina el estado de cualquier transacción como: activas, confirmadas, retrocedidas, preparadas [para

confirmación en 2 fases]. En el caso de un fallo de sistema, tan pronto el servidor es relanzado, InterBase automáticamente revisa las TIPs buscando las transacciones no confirmadas. Los registros que se encuentren en estado no confirmado serán retrocedidos. Este proceso tarda menos de un segundo para la mayoría de las bases de datos.

Al no existir registro de transacciones, no es necesario determinar los requerimientos de espacio ni la relación entre copias de resguardo incrementales o completas, establecer puntos de verificación o recortar manualmente el registro de transacciones para evitar que se desborde. Adicionalmente, al no haber registro de transacciones, una base de datos de InterBase nunca tiene problemas a causa de un desbordamiento del registro de transacciones.

## Puntos de verificación

- **SQL Server.** La arquitectura de SQL Server exige que el administrador de bases de datos establezca puntos de verificación. Un punto de verificación es una aproximación del tiempo que tomará la recuperación de la base de datos en caso de un fallo de sistema utilizando el registro de transacciones. El SGBD utiliza este intervalo para determinar cuándo los registros de transacciones "sucios" y las páginas de datos serán escritas a disco. El proceso de escritura de las páginas sucias al disco minimiza la cantidad de trabajo a realizar por el servidor para restaurar una base de datos. El establecimiento de puntos de verificación es importante y tiene algunas implicaciones serias. Supongamos que si hay diez bases de datos en un servidor y que el intervalo de verificación es establecido por el administrador en 12 minutos. La restauración del sistema tardará 120 minutos (2 horas), a 12 por cada base de datos. El administrador tiene la opción de establecer un intervalo menor, lo que garantizaría que las bases de datos estuvieran en línea en un menor tiempo. Sin embargo, establecer un valor demasiado bajo puede tener un efecto negativo en el rendimiento, por cuanto el cache estará siendo escrito al disco demasiado frecuentemente. Si se establece demasiado alto, la recuperación puede durar demasiado y el riesgo de desbordamiento del registro de transacciones crecerá, resultando en un fallo inmediato de la base de datos. Cuando esto ocurre, la única manera de limpiar el registro de transacciones es volcarlo y restaurar la base de datos. Esto sacará de línea por un tiempo la base de datos, dejando a los usuarios en incapacidad de desarrollar su trabajo.
- **InterBase.** Al no utilizar InterBase el concepto de registro de transacciones, el administrador no tiene que optimizar los puntos de verificación para prevenir los fallos de la base de datos. Las restauraciones automáticas necesitan por lo general menos de un segundo, incluso para bases de datos grandes, y no necesitan de la intervención del administrador. Esto minimiza los niveles necesarios de habilidades y entrenamiento para el personal que administra InterBase, a la vez que asegura la mayor disponibilidad y el mínimo posible de desconexiones. Estas capacidades únicas hacen de InterBase el sistema ideal cuando la disponibilidad 7x24 es crítica. InterBase es también la elección ideal cuando se necesita un mínimo de labores de mantenimiento como en grupos de trabajos, departamentos y en aplicaciones desarrolladas por distribuidores.

## Configuración

- **SQL Server.** Tanto Microsoft SQL Server como Sybase SQL Server ofrecen millones de opciones de configuración y parámetros de ajuste para optimizar el rendimiento de sus bases de datos. Muchas de estas opciones son complejas y tienen interdependencias tales, que el cambio de un parámetro afecta valores que deberán ser compensados ajustando otros parámetros. Sólo un Administrador altamente entrenado comprenderá totalmente las implicaciones de alterar parámetros. Sybase System 11 ha complicado aún más el asunto al añadir más de 200 nuevos parámetros configurables. Esta complejidad añadida incrementa los costos de mantenimiento, los costos de empleo de administradores, e implica que se requerirán numerosos ajustes de parámetros si se intenta mantener

la base de datos en un estado realmente optimizado.

- **InterBase.** InterBase es casi totalmente autoconfigurable, no requiriendo prácticamente ninguna intervención del administrador ni antes, ni después de la creación de la base de datos. Esto elimina la mayoría de los problemas de mantenimiento, y garantiza un funcionamiento óptimo independientemente de la plataforma o especificidades de la instalación. InterBase ha sido diseñado para garantizar los menores costos de mantenimiento posible. Una vez instalado, excepto por un fallo catastrófico del hardware o los períodos de mantenimiento, una base de datos de InterBase puede ser dejada sin atención. Esto reduce los costos de mantenimiento, asegurando el máximo rendimiento y disponibilidad.

## Recuperación

- **SQL Server.** La recuperación automática de una base de datos de SQL Server provoca la ejecución del registro de transacciones como parte del proceso de recuperación. Este proceso reejecuta las transacciones individuales almacenadas en el registro para re-crear la base de datos a partir del último punto de verificación. El tiempo [en minutos] de verificación es establecido por el Administrador, y el tiempo total requerido para la restauración de todas las bases de datos será igual a la suma de los tiempos determinados por el parámetro de verificación de cada una de las bases de datos en el servidor.

Si el fallo de la base de datos es consecuencia de un desastre, la base de datos deberá ser primero eliminada. La eliminación puede no resultar si existe cualquier corrupción de los datos. En ese caso, el administrador deberá ejecutar la orden `dbcc dbrepair` con la opción `dropdb` para crear una nueva estructura de bases de datos. El administrador deberá entonces cargar la base de datos hacia la nueva estructura usando el backup completo más reciente que tenga. Posteriormente, deberá ejecutar en secuencia todos los backups incrementales realizados desde el último resguardo total. Cada restauración de backup incremental relanza las transacciones incluidas en el registro correspondiente. El proceso se repite para cada una de las bases de datos. Este es un proceso complejo, largo y tedioso que requiere las habilidades de un administrador entrenado.

- **InterBase.** "Hemos descubierto que al ocurrir fallos de sistema, InterBase es capaz de recuperar el sistema en segundos, en vez de las horas que éste ocupaba en el pasado"

La recuperación de una base de datos InterBase es automática y no necesita la intervención de un administrador. Además, no es necesario establecer puntos de verificación ni hacer cálculos para asegurar el balance óptimo de los tipos de copias de respaldo, existiendo solamente la copia total. Las copias de respaldo pueden ejecutarse en línea en cualquier momento.

Como ya se ha mencionado, InterBase mantiene el estado de las transacciones en sus páginas TIP. Tan pronto como la base de datos arranca, se comprueban las TIPs en busca de registros no confirmados. Los registros no confirmados son retrocedidos y el sistema queda disponible inmediatamente. Este proceso generalmente dura menos de un segundo para la mayoría de las bases de datos. Este elemento fue clave para la selección de InterBase para el tanque M1 Abrams del Ejército de los EE.UU. Cuando el cañón del tanque dispara, produce un pulso electromagnético enorme que provoca la caída del sistema informático del tanque. Tan pronto el ordenador es reiniciado, InterBase se restaura inmediatamente. Esta capacidad de restauración rápida no es ofrecida por ningún otro SGBDR. Todos los resguardos de InterBase son copias totales. En caso de fallos catastróficos como un disco duro que falla, una base de datos de InterBase puede ser recuperada por sus usuarios (mínimamente entrenados) mediante el InterBase Server Manager. Aquí un usuario puede seleccionar el archivo que se desea restaurar del dispositivo de resguardo y restaurar la base de datos hacia el mismo u otro dispositivo.

InterBase no necesita utilizar particiones de bajo nivel en sistemas UNIX, lo que simplifica la restauración de InterBase en entornos UNIX.

El Server Manager de InterBase puede utilizarse con InterBase en cualquier plataforma. Al restaurar una base de datos de InterBase, no es necesario configurar ningún dispositivo ni reservar espacio para la base de datos.

**Sombreado de Bases de Datos.** InterBase ofrece la posibilidad de hacer copias de seguridad "en caliente" a través del sombreado de bases de datos. Una sombra de una base de datos es una copia viva duplicada de la base de datos, mantenida en el mismo u otro servidor. En la medida en que se completan las transacciones en el servidor maestro, las sombras se actualizan automáticamente para reflejar la versión más actual de la base de datos. Las aplicaciones cliente pueden, en caso de un fallo del servidor principal, ser programadas para conectarse a la base de datos de sombra, y continuar trabajando así hasta que los problemas en el servidor principal sean resueltos. Una ventaja adicional del sombreado es que las sombras pueden ser utilizadas para dividir la carga. En otras palabras, en aplicaciones de gran volumen, la sombra puede ser puesta en línea para brindar información a usuarios de tipo DSS, maximizando por tanto los recursos del servidor disponibles para los usuarios de tipo OLTP.

### [Inicio](#)

## Uso de Recursos

**Resumen:** InterBase tiene las más bajas necesidades de recursos, haciéndola la opción ideal para grupos de trabajos, departamentos y VARs.

- **Microsoft SQL Server.** Microsoft SQL Server necesita un mínimo de 60M de espacio en disco para su instalación, y 16MB de RAM para cargarse en NT 4. Cada usuario adicional necesita 48K de memoria. Una aplicación de bases de datos con 20 usuarios necesitará 16M para el SGBDR y 960K para los usuarios, para un total de casi 17M de RAM, sin incluir la memoria temporal para la carga de tablas o datos de búffer.

Aunque la instalación de Microsoft SQL Server no exige que el administrador configure la reservación de espacio en el momento de la instalación, Microsoft considera a éste como el parámetro más importante de todos los parámetros de configuración y recomienda encarecidamente que sea configurado manualmente. Aunque el manual "Microsoft SQL Server Administrator's Companion" detalla algunos de los parámetros como sobrecarga del núcleo, caché de datos, caché de procedimientos, etc., que deben ser considerados al configurar la reservación de memoria, Microsoft no ofrece una fórmula para calcular los valores óptimos. En vez de ello, Microsoft recomienda que el administrador utilice el Monitor de Rendimiento SQL para examinar el contador de fallos de página para ver si se están generando fallos de página. En caso afirmativo, SQL Server ha sido configurado con demasiada memoria. Además, como Microsoft SQL Server bloquea la memoria y el tempdb (si se está usando tempdb en RAM), es posible obtener mensajes de "memoria exhausta" al ejecutar otras aplicaciones sobre el servidor. Asimismo, configurar Microsoft SQL Server con más memoria virtual que la física puede resultar en un rendimiento pobre. Todas esas situaciones exigen una prueba en condiciones reales de cada servidor bajo las condiciones de carga esperadas para determinar la configuración de memoria óptima de Microsoft SQL Server. Con vistas a asegurar un rendimiento óptimo, se necesita un administrador de bases de datos entrenado para monitorizar y poner a tono el SGBDR. Estas consideraciones provocan costos adicionales de mantenimiento y configuración para los desarrolladores de aplicaciones de mercados verticales.

- **Sybase SQL Server.** Una instalación de Sybase necesita aproximadamente 50M de espacio en disco

para acomodar el lenguaje, los documentos de ayuda, utilidades, etc. Se requiere espacio adicional para los dispositivos de volcado, espacio de trabajo y otra herramientas. De 2 a 3 MB adicionales se necesitan para cada lenguaje adicional soportado, como Alemán o Francés. Igualmente se necesita espacio adicional para el software compatible con Sybase, incluyendo:

- OPEP Cliente.
- OPEP Server.
- Embedded SQL.

Las necesidades de RAM son diferentes para cada plataforma. Sin embargo, Sybase SQL Server exige que el administrador calcule los requerimientos de RAM a partir de:

- La memoria estática necesitada por el núcleo de SQL Server.
- Cachés para procedimientos y datos que también son configurados por el administrador.
- Búferes de red para cada usuario por cada conexión.
- Búferes para extender los búferes de entrada y salida.

Los requerimientos de memoria deberán balancearse contra:

- La memoria necesaria para el sistema operativo.
- La memoria necesaria para otros procesos, como X Windows.
- Memoria adicional para la red.

Las asignaciones de memoria deben reajustarse cuando:

- Se añade o se quita memoria del sistema.
- El uso del sistema informático cambia.
- Se producen cambios a los búferes de entrada y salida y a la memoria de la red.

Como sucede con Microsoft SQL Server, la configuración de memoria de Sybase SQL Server aumenta la complejidad de la instalación y mantenimiento, añadiendo costos adicionales. Estos problemas son de interés primordial a los desarrolladores de aplicaciones para mercados verticales.

- **InterBase.** "El usuario no tiene que preocuparse acerca de los procesos del servidor que deban ser monitoreados y gestionados, de diferentes ubicaciones de discos para tablas diferentes, de la reestructuración de tablas para recuperar el espacio de las filas eliminadas, o de reindexar para rebalancear los índices."

El núcleo de InterBase ocupa menos de 2 MB, y el cliente completo, el servidor y los ficheros de documentación en línea instalados sobre Windows NT necesitan menos de 10MB de espacio en disco. Del mismo modo, InterBase nunca necesita más memoria que la memoria base requerida por el sistema operativo. InterBase reserva dinámicamente el espacio en el disco duro y los recursos de memoria en la medida en que sean necesarios, sin la intervención de un administrador. Para los servidores distribuidos que ejecutan una variedad de sistemas

operativos, se minimiza la inversión en hardware, entrenamiento y mantenimiento para los Sistemas Informáticos. Las bajas necesidades de recursos de InterBase y la reservación dinámica de los mismos permiten a los desarrolladores ofrecer soluciones significativamente más baratas.

[Inicio](#)

## Escalabilidad

**Resumen:** InterBase ofrece la mejor solución para escalar aplicaciones que aprovechen las nuevas posibilidades de multiprocesamiento simétrico disponibles en muchas plataformas de hardware y Sistemas Operativos, de la misma forma en que ofrece las mejores soluciones para escalar en ambos sentidos los sistemas de toda una empresa.

**Antecedentes:** En la medida en que crecen las necesidades de los grupos de trabajo y las corporaciones, así mismo crecen las demandas que éstos exigen para sus sistemas informáticos. Las aplicaciones de bases de datos deberán ser capaces de satisfacer las necesidades de un número creciente de usuarios y deberán manejar volúmenes de datos cada vez mayores. Para ello, los SGBDR deberán ser capaces de utilizar los Servidores de Multiproceso Simétrico [SMP]. Añadir procesadores adicionales a una máquina existente casi siempre es más deseable que reemplazar la máquina entera.

- **Microsoft SQL Server.** Microsoft SQL Server solamente está disponible sobre Windows NT. Windows NT, independientemente de la CPU sobre la cual esté implementado [Intel, Alpha, MIPS, PPC] es un sistema operativo de 32 bits limitado a 4 procesadores. Por tanto, Microsoft SQL Server y Windows NT no pueden utilizar las arquitecturas de hardware de 64 bits que utilizan los sistemas empresariales, ni pueden soportar arquitecturas con más de 4 CPUs, como las disponibles para Sun, DEC, SGI, e IBM. Igualmente, MS SQL Server no puede ser escalado hacia abajo a ordenadores que utilicen Windows 95 o Windows 3.1, obligando al uso de SGBDR múltiples para la implementación de soluciones corporativas. Claramente, la implementación de soluciones escalables con MS SQL Server significa una inversión mucho mayor en desarrollo y ciclos de mantenimiento, por cuanto un sistema común con un único API no puede ser usado en toda la empresa.
- **Sybase SQL Server.** Sybase ofrece su SQL Server en una amplia gama de plataformas. Sybase SQL Server, sin embargo, no es escalable hacia arriba y hacia abajo para toda la empresa. Sus productos para Windows 3.1 y Windows 95, por ejemplo, son enteramente diferentes del SQL Server NT. Los motores de datos son diferentes y utilizan distintas interfaces, haciendo difícil la migración entre diferentes máquinas y sistemas operativos.
- **InterBase.** InterBase ofrece una escalabilidad excelente y utilizará todos los procesadores que estén disponibles en cualquier plataforma. Esto es de un tremendo valor para los desarrolladores, que se aseguran de que las aplicaciones escritas con InterBase pueden satisfacer las demandas de clientes en cualquier plataforma con 1, 10 o 100 usuarios.

[Inicio](#)

## Portabilidad

**Resumen:** InterBase ofrece el rango más amplio de plataformas posibles para cualquier SGBDR cliente/servidor, asegurando la mayor compatibilidad posible con cualquier plataforma popular para servidores. Este amplio rango de compatibilidad asegura que las bases de datos en InterBase funcionarán de

modo idéntico, independientemente de la plataforma en que residan.

**Antecedentes:** Los entornos distribuidos son por definición heterogéneos, y una disponibilidad de SGBD para NT, Windows y UNIX es crítica para el éxito de las soluciones corporativas cliente/servidor, porque asegura la utilización inmediata de los recursos corporativos preexistentes. La siempre creciente importancia de la informática móvil subraya la necesidad de desarrollar aplicaciones basadas en SGBD a través de las plataformas fundamentales, a saber : Windows 95, Windows NT, Solaris, HP-UX, IBM-AIX, SGI-IRIX, and DEC UNIX.

- **Microsoft SQL Server.** Microsoft SQL Server opera sólo bajo Windows NT y es por tanto no es portable a otras plataformas. Esta ausencia de portabilidad limita severamente el rango de aplicaciones, así como el tamaño de la base de datos y el número de usuarios concurrentes que la base de datos puede soportar. Las corporaciones que eligen Microsoft SQL Server se ven forzadas a elegir Windows NT como plataforma servidora. Las empresas que han venido utilizando Novell NetWare o UNIX como plataformas de servidores se ven forzadas a introducir otro sistema operativo más a la mezcla. Se necesita nuevo hardware, nuevo software y nuevas habilidades para implantar tales soluciones, lo que dispara los costos de soporte y mantenimiento.

La falta de portabilidad a otras plataformas igualmente deja ver que Microsoft carece de una solución viable a las necesidades de los usuarios de portátiles. Para solucionar esto, los desarrolladores necesitan desarrollar otra aplicación, típicamente mediante Paradox o Microsoft Access para sus usuarios de portátiles. Como los SGBD basados en archivos difieren enormemente en enfoque programático de los basados en SQL, diferentes bases de código deberán ser mantenidas. Ello implica un tremendo volumen de trabajo de desarrollo, así como de costos de mantenimiento y soporte.

- **Sybase.** Sybase no suministra una solución única que satisfaga las necesidades de todos los usuarios, desde el propietario de un portátil hasta la empresa completa. En particular, Sybase SQL Server no está disponible para Windows 3.x o Windows 95. En un intento de responder a las necesidades de esos usuarios, Sybase ha reempaquetado y renombrado el Watcom SQL adquirido con la compra de PowerSoft. El producto se ha rebautizado como SQL Anywhere. SQL Anywhere es un producto fundamentalmente diferente de Sybase SQL Server. En un intento de aumentar la compatibilidad, Sybase ha añadido una librería de interfaz que permite a SQL Anywhere soportar la mayoría de las construcciones SQL de Sybase. Desgraciadamente, en eso termina la similaridad. Cada producto utiliza diferentes estructuras internas, tipos de datos y mecanismos de bloqueo. Adicionalmente, el lenguaje utilizado para crear procedimientos almacenados y triggers es totalmente diferente en ambos productos. Ello significa que las aplicaciones que se deberán hacer disponibles para los usuarios de portátiles y las utilizadas por grupos de trabajo en entornos empresariales serán radicalmente diferentes tanto a nivel de bases de datos como a nivel de aplicaciones.

En la medida en que las aplicaciones se modifican para satisfacer las necesidades de los usuarios, lo mismo deberá ocurrirle a las bases de datos. La versión run-time de SQL Anywhere sufre de unas limitaciones extremas en este aspecto, al ser imposible modificar los meta-datos. Una vez instalada la base, el desarrollador no tendrá mecanismos para modificar las estructuras subyacentes.

- **InterBase.** InterBase es un sistema de bases de datos verdaderamente cross-plataforma. Funciona de la misma manera bajo Windows, NT, NetWare y una docena de variantes de UNIX.. InterBase es el único SGBD que suministra portabilidad transparente a través de las plataformas. El Server Manager de InterBase es todo lo que se necesita para mover fácilmente una base de datos de InterBase de una plataforma a otra. Esta portabilidad es un beneficio clave para los desarrolladores, que no necesitan conocer el sistema operativo que sus clientes necesitarán. Adicionalmente, como el servidor de InterBase es muy pequeño [menos de 10MB], y como el servidor no modifica el sistema sobre el que se ha instalado, Borland necesita trabajar muy poco para transportarlo a otras plataformas. Esto

puede contrastarse con el Sybase SQL Server, por ejemplo, que emula el multi-hilado en su servidor cuando el sistema operativo subyacente no lo soporta. No solo se facilita el proceso de porte, sino que se asegura una base de código estable y consistente bajo todas las plataformas. Los desarrolladores pueden estar seguros de obtener un comportamiento idéntico bajo diferentes plataformas.

**Local InterBase para Windows 95/NT.** Local InterBase es una versión monousuario del Servidor de InterBase, y suministra un medio eficiente de construir e implantar aplicaciones cliente/servidor monousuarios. Las versiones para Windows 95 y Windows NT de InterBase son idénticas, y brindan idénticas características, porque tiene igual código binario. Esto es importante para distribuidores que suministran aplicaciones para entornos mono- y multi-usuarios. La conversión de una base de datos mono-usuario a una multi-usuario en InterBase puede realizarse mediante simples operaciones de ratón. Esto hace extremadamente simple la distribución de copias de evaluación de software de mercado vertical.

Las corporaciones encontrarán también la fácil escalabilidad importante para los usuarios de portátiles que necesitan llevar a su trabajo consigo. Todas las estructuras de bases de datos son totalmente portables entre InterBase Local y cualquier otra de las plataformas soportadas. Esto significa que todos los tipos de datos, incluyendo BLOBs y los meta-datos, incluyendo procedimientos almacenados y triggers se implementan exactamente de la misma manera. Ello significa también que las aplicaciones que utilicen bases de datos de InterBase podrán ser transportadas desde la mayor de las máquinas multiprocesadoras de Sun a un portátil sin ningún cambio en la base de datos o el código de la aplicación. Esta sola característica puede brindar una reducción significativa en los costos de desarrollo y mantenimiento.

### Inicio

## **Tecnologías Avanzadas**

**Alertadores de Eventos.** InterBase ofrece una característica única que lo ha hecho muy exitoso en la comunidad financiera. Aunque los Alertadores de Eventos no son exclusivos de InterBase, su implementación sí lo es. Los Alertadores permiten que una base emita eventos a las aplicaciones clientes interesadas a partir de ocurrencias específicas en la base de datos. Donde InterBase es diferente es en que tradicionalmente los mecanismos de otras bases de datos requieren que las aplicaciones interesadas pregunten a la base de datos cada cierto intervalo regular, lo que presenta algunos problemas serios. Primero, como las aplicaciones tienen que consultar a la base de datos, existe una demora entre la ocurrencia del evento y el momento en que la aplicación recibe la notificación. En segundo lugar: como las aplicaciones clientes deben consultar al servidor, se crea una carga y un tráfico en la red adicional. En instalaciones grandes, con cientos o miles de usuarios, esta carga adicional puede causar una seria degradación del rendimiento. En tales casos, el desarrollador debe trabajar en coordinación con el administrador de bases de datos para determinar el ciclo óptimo de interrogación para balancear los efectos de la notificación demorada y sus efectos, tales como la pérdida de oportunidades de negocio, contra las limitaciones prácticas de la red y el servidor. InterBase no sufre de ninguna de estas limitaciones y brinda notificación instantánea y la menor carga posible en el servidor y la red.

## **Soporte para Java**

**Antecedentes:** Java ofrece a los desarrolladores un entorno robusto, simple y orientado a objetos para crear y distribuir soluciones cliente/servidor inter-plataformas de alto rendimiento. Controlado por Sun Microsystems y más de 40 socios industriales, incluyendo a Microsoft, IBM, Borland, Oracle, y Netscape, Java promete ser la más influyente plataforma informática.

- **Microsoft SQL Server.** Microsoft no ha anunciado todavía soporte alguno para Java desde SQL Server, ni suministra conectividad a SQL Server desde Java mediante un JDBC.
- **Sybase SQL Server.** Sybase no ha anunciado todavía soporte alguno para Java desde SQL Server, ni suministra conectividad a SQL Server desde Java mediante un JDBC.
- **Borland InterBase e InterClient.** "La participación de Borland en la creación de la especificación JDBC ha sido de gran utilidad para nuestro equipo JavaSoft. Su implementación de una solución de controlador de JDBC/Net enteramente en Java valida la factibilidad de usar a Java y JDBC como verdaderas arquitecturas cliente/servidor. Esto ayuda a hacer la implantación de soluciones de bases de datos para Internet e Intranet significativamente más simple."

Las posibilidades de auto-configuración de InterClient para InterBase están diseñadas para dar a los gerentes de servicios informáticos de las corporaciones acceso a bases de datos de alta velocidad, lo que se traduce en mejores tiempos de respuesta y caminos más cortos a los datos a través de redes y servidores Web.

Inicio

## Soporte Internacional

El soporte de los conjuntos de caracteres internacionales dentro de una base de datos SQL es imperativo para su utilización fuera de los Estados Unidos. Este soporte puede variar, desde el soporte para un conjunto de caracteres al más alto nivel [p.e. dentro de una tabla de una base de datos], hasta el más bajo [soporte para un juego de caracteres específico dentro de un campo específico de una tabla]. Adicionalmente, para poder ser verdaderamente eficiente, el servidor debe ser capaz de organizar las tablas a partir de diferentes conjuntos de caracteres, permitiendo por tanto la agrupación, ordenación y consulta adecuadas.

- **Sybase SQL Server / Microsoft SQL Server.** En SQL Server, el administrador sólo puede definir un carácter a nivel de tabla, lo que implica que una base de datos cualquiera es incapaz de almacenar conjuntos de caracteres de dos conjuntos diferentes, por ejemplo, en una tabla. Por ello, implementar soluciones tan simples como un diccionario bilingüe requiere dos tablas diferentes. Lo que es más importante, para el desarrollador que desea crear una aplicación controlado por una base de datos [donde los menús y las listas se generan a partir de la base de datos], será necesario utilizar varias tablas para dar soporte a múltiples idiomas, y los entornos con requerimientos para múltiples lenguajes se hacen muy difíciles de implementar.
- **InterBase.** A diferencia de SQL Server, InterBase permite que el administrador de la base de datos aplique múltiples juegos de caracteres dentro de diferentes campos de una tabla, y aplique múltiples secuencias de ordenación para esa tabla. Por tanto, una desarrollador de aplicaciones que esté implantando una aplicación internacional podrá colocartoda la información de mensajes, menús y ayuda en una tabla, y cambiar dinámicamente de lenguaje para los diferentes usuarios que accedan a la aplicación.

## Conclusiones

InterBase es claramente la solución más portable y escalable de todas las soluciones de SGBDR aquí presentadas. InterBase brinda la mayor productividad para los entornos del mundo real, en el que prevalece un modelo de transacciones mixto. InterBase no sufre de complejidades escondidas y costos de instalación, mantenimiento, entrenamiento de personal, requisitos de recursos en el servidor y enormes cantidades de código adicional que debe escribirse para gestionar los mecanismos de bloqueo del SQL Server. InterBase

igualmente garantiza la máxima disponibilidad, el menor tiempo de restauración y el menor tiempo "muerto". Todos estos factores son componentes importantes a la hora de suministrar soluciones para usuarios aislados, grupos de trabajos, departamentos y empresas. InterBase, de este modo, ofrece la mejor solución, de costo más efectivo, a las necesidades de cualquier firma que necesite un SGBDR cliente/servidor.

Realizado por: Michael Lant - Sphere Data Systems Inc.