

Presentación e instalación de SQL Server

En este capítulo, conoceremos la historia del desarrollo de motores de bases de datos para redes, las diferentes versiones de SQL Server y las funcionalidades disponibles en cada una de ellas.

Además, repasaremos los elementos necesarios para instalar un servidor SQL Server 2005 con todas las herramientas precisas para realizar desarrollos.

Sistemas de Gestión de Bases de Datos Relacionales	16
Los primeras versiones de SQL Server	19
SQL Server 2005 y sus ediciones	23
Funcionalidades disponibles por Edición	25
Instalación de SQL Server 2005	26
Revisión del hardware mínimo requerido	26
Sistema Operativo de la plataforma destino	27
Cuentas de Dominio para el servidor y la instalación	28
Definición de instancias de servidor a instalar	29
Definición del conjunto de caracteres	29
Tipos de instalación disponibles	30
Cómo instalar SQL Server 2005 - Beta 2 1	31
Prestaciones de SQL Server para sistemas OLTP	41
El motor de SQL Server y T-SQL	41
El optimizador de consultas	42
Procedimientos almacenados extendidos	43
Resumen	47
Actividades	48

SISTEMAS DE GESTIÓN DE BASES DE DATOS RELACIONALES

Un **Sistema de Gestión de Bases de Datos Relacionales** (*SGBDR* o *RDBMS*, por sus siglas en inglés) tiene como objetivo servir de interfaz de comunicación entre una base de datos y las aplicaciones que la utilizan, de modo de asegurar el cumplimiento de los siguientes objetivos.

- Abstracter la información acerca de cómo se almacenan físicamente los datos.
- Mantener la independencia entre la vista física y lógica de los datos.
- Mantener la mínima redundancia, dado que los diseños de bases de datos que soportan (el modelo relacional) no son propensos a representar información repetida.
- Asegurar la consistencia de la información.
- Proveer seguridad ante usos malintencionados, ataques, robos de información, etc.
- Asegurar la integridad de los datos almacenados.
- Proveer mecanismos de respaldo y recuperación.
- Regular y controlar la concurrencia de usuarios.
- Proveer buen tiempo de respuesta.

Para evaluar los **SGBDR**, el **doctor E. F. Codd** enunció, en 1970, las siguientes reglas (publicadas en 1969 bajo el título **Derivability, Redundancy, and Consistency of Relations Stored in Large Data Banks**, y un año más tarde bajo el título **A Relational Model of Data for Large Shared Data Banks**).

Regla 0. Para que un sistema se denomine sistema de gestión de bases de datos relacionales debe utilizar (exclusivamente) sus capacidades relacionales con el propósito de gestionar la base de datos.

Regla 1. Toda la información contenida en una base de datos relacional (BDR) se representa explícitamente en el nivel lógico de una manera exacta: con valores en tablas. Por este motivo, los metadatos (diccionario, catálogo) se representan en forma idéntica que los datos de usuario y puede usarse el mismo lenguaje (ej., SQL) para acceder a los datos y a los meta datos (**Regla 4**).

Por otra parte, un valor posible es el valor nulo, con sus dos interpretaciones: valor desconocido o valor no aplicable.

Regla 2. Regla del acceso garantizado

Para todos y cada uno de los datos (valores atómicos) de una base de datos relacionales se garantiza que son accesibles lógicamente, utilizando una combinación de nombre de tabla, valor de clave primaria y nombre de columna.

Cualquier dato almacenado en una bases de datos relacionales tiene que poder ser direccionado en forma unívoca. Para hacerlo, se debe indicar en qué tabla se encuentra, cuál es la columna y cuál es la fila (por medio de la clave primaria). Por lo tanto, se necesita el concepto de clave primaria.

Regla 3. Tratamiento sistemático de valores nulos

Los valores nulos (aquellos valores que son distintos de la cadena vacía, blancos, 0,...) se soportan en los SGBD totalmente relacionales con el objeto de representar información desconocida o no aplicable de manera sistemática, independientemente del tipo de datos en cuestión.

Regla 4. Catálogo dinámico en línea basado en el modelo relacional

La descripción de la base de datos relacional se representa lógicamente de la misma manera que los datos normales, de tal manera que los usuarios autorizados se encuentran en condiciones de aplicar el mismo lenguaje relacional a su consulta, igual que lo aplican a los datos normales.

Regla 5. Regla del sublenguaje de datos completo

Un sistema relacional debe soportar varios lenguajes y varios modos de uso de terminal (ej., rellenar formularios, etc.). Sin embargo, debe existir al menos un lenguaje cuyas sentencias sean expresables mediante una sintaxis bien definida como cadenas de caracteres y que sea completo, soportando:

- definición de datos;
- definición de vistas;
- manipulación de datos (interactiva y por programa);
- limitantes de integridad;
- limitantes de transacción (iniciar, realizar, deshacer) (**Begin, commit, rollback**);
- además de poder tener interfaces más amigables para hacer consultas, etc., siempre debe de haber un modo de hacerlo todo de manera textual, es decir que pueda ser incorporada en un programa tradicional. Un lenguaje que cumple con esta característica medida es SQL.

III PERSISTENCIA DE DATOS

En la jerga informática, se entiende por **Persistencia de Datos** o de Información el proceso de almacenamiento de datos o de información de un proceso de manera que puedan ser recuperados en cualquier otro momento tal y como fueron almacenados.

Regla 6. Regla de actualización de vistas

Todas las vistas que son teóricamente actualizables se pueden actualizar por el sistema.

Regla 7. Regla de inserción, actualización y borrado de alto nivel

La capacidad de manejar una relación base o derivada como un solo operando se aplica no sólo a la recuperación de los datos (consultas), sino también a la inserción, actualización y borrado de datos. Esto significa que el lenguaje de manejo de datos también debe ser de alto nivel (de conjuntos).

Regla 8. Regla de independencia física de datos

Los programas de aplicación y actividades del terminal permanecen inalterados en el ámbito lógico independientemente de los cambios que se realicen en las representaciones de almacenamiento o en los métodos de acceso. El modelo relacional es un modelo lógico de datos y oculta las características de su representación física.

Regla 9. Regla de independencia lógica de datos

Los programas de aplicación y actividades del terminal permanecen inalterados en el ámbito lógico independientemente de los cambios que se realicen en las tablas base que preservan la información. Cuando se modifica el esquema lógico preservando información (no valdría, por ejemplo, eliminar un atributo), no es necesario modificar nada en niveles superiores.

Regla 10. Regla de independencia de integridad

Los limitantes de integridad específicos para una determinada base de datos relacional deben ser capaces de definirse en el sublenguaje de datos relacionales, y almacenables en el catálogo, no en los programas de aplicación. El objetivo de las bases de datos no consiste únicamente en almacenar los datos, sino también en guardar sus relaciones y evitar que estas limitantes se codifiquen en los programas. Por tales motivos, se deduce que en una base de datos relacional debe ser posible definir limitantes de integridad.

Como parte de los limitantes inherentes al modelo relacional (formando parte de su definición) podemos hallar los siguientes:

III REGISTRO

Cuando hablamos de **registro**, nos referimos a la mínima unidad cohesiva y coherente de información acerca de una entidad. Así, por ejemplo, hablaremos de **Registros de Clientes**, donde un registro nos aporta toda la información que necesitamos sobre un cliente en una aplicación.

- una BDR tiene **integridad de entidad** o, expresado en otros términos, toda tabla debe tener una clave primaria, y
- una BDR tiene **integridad referencial**, es decir, toda clave externa no nula debe existir en la relación donde es primaria.

Regla 11. Regla de independencia de distribución

Una **BDR** tiene independencia de distribución. Las mismas órdenes y programas se ejecutan igual en una base de datos (BD) centralizada que en una distribuida.

Las bases de datos relacionales son fácilmente distribuibles: se parten las tablas en fragmentos que se distribuyen y, cuando se necesitan las tablas completas, se recombinan por medio de operaciones relacionales con los fragmentos.

Esta regla es responsable de los siguientes tres tipos de transparencia de distribución.

- **Transparencia de localización:** el usuario tiene la impresión de que trabaja con una base de datos local (aspecto de la regla de independencia física).
- **Transparencia de fragmentación:** el usuario no se percata de que la relación con la cual está trabajando se encuentra fragmentada (aspecto de la regla de independencia lógica de datos).
- **Transparencia de replicación:** el usuario no advierte que pueden existir copias (réplicas) de una misma relación en diferentes lugares.

Regla 12. Regla de la no-subversión

Si un sistema relacional posee un lenguaje de bajo nivel (un registro de cada vez), ese bajo nivel no puede ser usado para saltarse (subvertir) las reglas de integridad y los limitantes expresados en los lenguajes relacionales de más alto nivel: una relación (conjunto de registros) de cada vez.

Los primeras versiones de SQL Server

Tanto las organizaciones como los individuos necesitan guardar información de manera fiable y segura durante el curso de sus actividades. El volumen de esta información estará relacionado siempre, de manera directamente proporcional, con el volumen de transacciones que realice.

III ¿QUÉ ES UN CAMPO?

Un **campo** es una unidad de información incluida en un registro, que por sí sola no aporta información. Por ejemplo, **Aguas Claras** puede ser un nombre propio, una localidad o el nombre de un río. Nada nos aporta por sí solo, si no está relacionado directamente con el campo de un registro. En un Registro de clientes, **Aguas Claras** puede ser un dato que contiene el campo **Nombre**.

En este sentido, las principales empresas de tecnología han contribuido, desde la década de los setenta, con el aporte de mejoras sorprendentes sobre las herramientas de bases de datos y de las aplicaciones de gestión.

La historia de los Sistemas de Gestión de Bases de datos Relacionales (SGBDR) se remonta al año 1970, cuando el doctor E. F. Codd publicó el artículo con su **Teoría sobre el Modelo Relacional**.

En 1974, IBM desarrolla, sobre la base de las reglas de Codd, el primer lenguaje orientado hacia el modelo relacional: el lenguaje **SEQUEL** (*Structured English Query Language*) para su prototipo **System/R**. El proyecto se cerró en 1979 (se retomaría luego como base de su proyecto SQL/DS), pero dejó dos hitos en el mundo de los sistemas de bases de datos relacionales: la visibilidad del modelo relacional en un prototipo y un lenguaje estructurado de consultas que más tarde, por razones legales, cambiaría su nombre a SQL.

Rational Software Inc. (*RSI*) sale al mercado con **Oracle II**, basado en el desarrollo del modelo relacional en el que estuvo trabajando IBM. La firma cambió su nombre por el de **Oracle** y avanzó con las mejoras hasta 1986, cuando salió al mercado con la primera versión de bases de datos distribuida.

En 1986, el comité **ANSI** (*Instituto Nacional Americano para Estándares*) publicó el primer estándar **SQL** (*Lenguaje Estructurado de Consultas*) como el lenguaje de acceso a bases de datos relacionales, basándose en el estándar del proyecto **System/R** de IBM. Este estándar fue revisado nuevamente en 1992 y en 1999.

SQL Server tuvo sus orígenes en acuerdos comerciales entre las empresas **Microsoft, IBM, Sybase** y **Ashton-Tate**.

Con la finalidad de ubicarse rápidamente en una posición competitiva, en 1986 Microsoft se asoció con Sybase Inc., que ya tenía un producto corriendo en UNIX que se encontraba muy bien conceptualizado en el mercado. Tal reputación se debió al haber introducido elementos innovadores como **desencadenadores** (*triggers*)



IDIOMA

Para beneficio del lector, traduciremos los conceptos que se encuentren en idioma inglés, siempre y cuando dicha traducción no altere el sentido del concepto. La jerga informática está poblada de palabras en inglés de difícil traducción al español, dado que representan conceptos del mundo real ajenos a esta área.

y procedimientos almacenados; estaba diseñado sobre el paradigma **cliente/servidor**. Como fruto de esta alianza, surgió **Sybase** para OS/2.

En 1988, Microsoft se asoció con Ashton-Tate y anunció un nuevo producto: **Ashton-Tate/Microsoft SQL Server**. La idea era escalar **Dbase** con el fin de crear una versión cliente/servidor que permitiera utilizar su lenguaje de consultas para crear aplicaciones que se comunicaran con SQL Server. En 1989, surgió la primera versión de SQL Server para OS/2 bajo el nombre Ashton-Tate/Microsoft SQL Server 1.0.

El intento de escalar **Dbase** e integrarlo con SQL Server fracasó.

Dbase intentaba acceder a los registros de SQL Server mediante la técnica **ISAM** (*Indexed Sequential Access Method*), que desaprovechaba las potencialidades de SQL Server. Éste, a su vez, no podía ofrecer la capacidad de navegar registros hacia atrás y adelante como **Dbase** (los cursores aún no se habían desarrollado). Microsoft y Ashton-Tate disolvieron la sociedad de comercialización.

En 1991, Microsoft ya estaba trabajando en **Windows NT** (que originalmente sería lanzado como **OS/2 3.0** por su asociación con IBM), basado en la interfaz de programación de aplicaciones **API Win32** (*Application Programming Interface*).

Dos años más tarde, Microsoft y Sybase anunciaron el fin del desarrollo conjunto. A partir de ese momento, ambas empresas continuarían con sus desarrollos en forma independiente (Sybase ya había lanzado su **System10** para Windows NT).

La siguiente versión de SQL Server, la 6.0, se liberó al mercado en 1995. Este producto ya trabajaba elementos clave, como la duplicación y los cursores, que daban la posibilidad de trabajar registros uno a uno.

La versión 6.5, liberada en abril de 1996, incluía más facilidades de uso, la certificación ANSI-SQL, más posibilidades a las transacciones distribuidas, y tuvo en cuenta la creciente demanda de almacenes de datos.

Durante el transcurso del año 1995, Microsoft comenzó a desarrollar un nuevo procesador de consultas al que se le sumó el de **OLE DB**. Se trabajó originalmen-

||| SERVICIOS DE NOTIFICACIÓN

Notification Services proporciona un entorno para el desarrollo de aplicaciones basadas en suscripciones, a través de las cuales se notifica a los usuarios acerca de eventos.

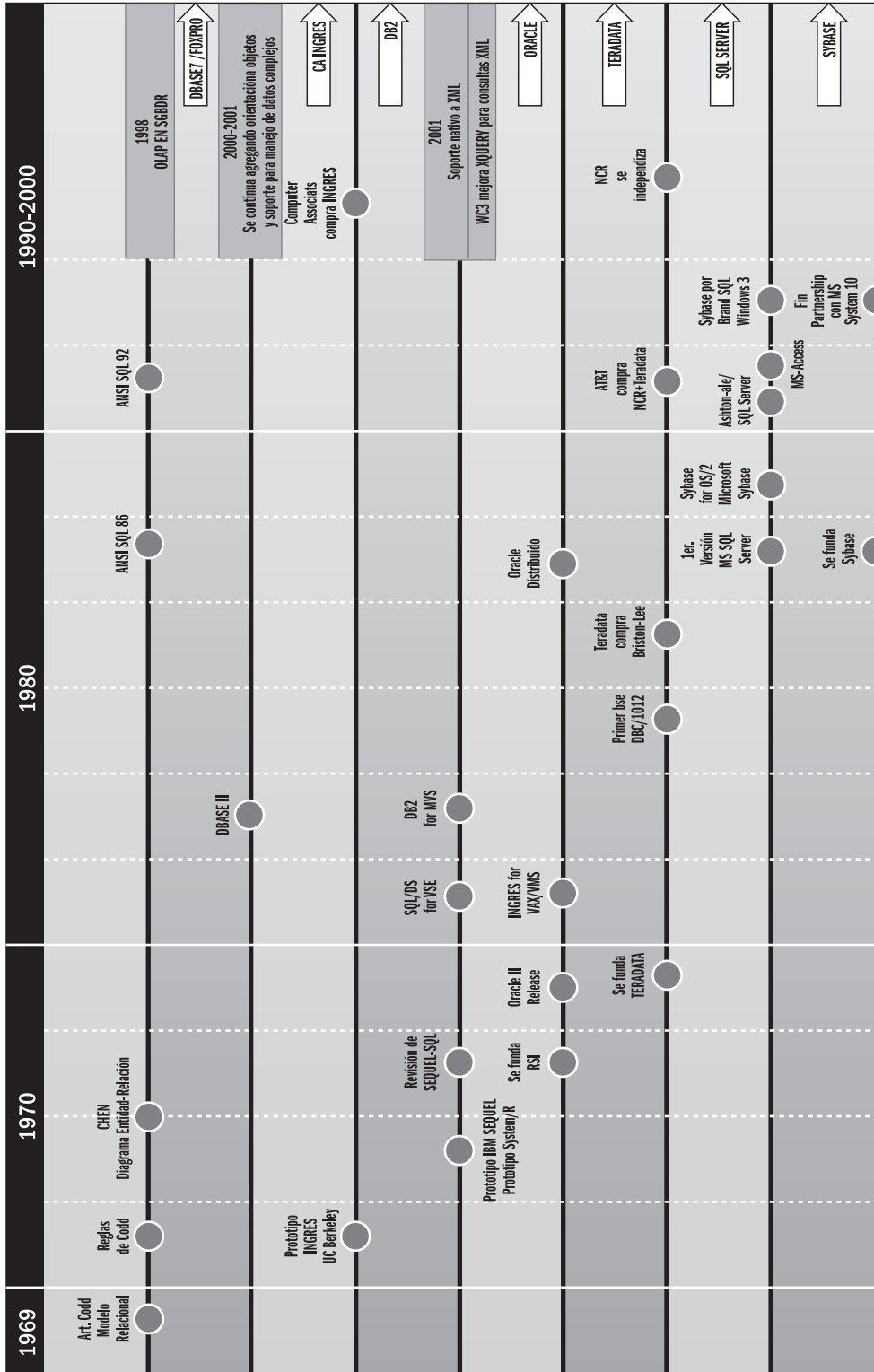


Figura 1. Primeros hitos en el desarrollo de Sistemas SGBDR.

te sobre una nueva versión de SQL Server, que fuese escalable al aprovechar los procesadores y la memoria disponible, que incluyera la posibilidad de efectuar bloqueos a nivel fila con un administrador de bloqueos inteligente, con un procesador de consultas que facilitase las consultas distribuidas y aplicaciones basadas en Internet y de almacenes de datos. El producto fue liberado en enero de 1999 como la versión **7.0**.

La versión 2000 de este producto, presentada en seis ediciones, introduce las características que se enumeran a continuación:

- soporta hasta 32 procesadores y 64 Gbytes de memoria RAM sobre Windows 2000 Datacenter Server;
- se integra con el Directorio Activo de Windows 2000 y aprovecha sus características de seguridad;
- provee ayuda al usuario en la creación de aplicaciones de **Almacén de Datos** (*Datawarehouse*) y **Minería de Datos** (*Data Mining*) mediante asistentes;
- ofrece soporte nativo al almacenamiento, uso y actualización de documentos XML, orientándose hacia el servicio de aplicaciones .NET.

SQL Server 2005 y sus ediciones

SQL Server 2005 es la última entrega de este popular SGBDR. Es mucho más que un simple gestor de bases de datos relacionales: Microsoft considera esta edición como la “próxima generación”.

El producto fue mejorado para:

- proveer alta escalabilidad y disponibilidad;
- suministrar herramientas de análisis de información avanzadas;
- proveer un ambiente seguro, escalable y productivo mediante una plataforma de administración mejorada;
- reducir los costos de propiedad y aprovechar todas las características de integración disponibles al formar parte del **Windows Server System**.



VERSIÓN DE SQL SERVER

En el transcurso de este libro, trabajaremos con la versión **Beta 2** de este producto, cuya liberación al mercado está prevista para finales de 2005. Si bien es esperable que no se introduzcan cambios mayores de última hora, deberá consultarse la documentación técnica del producto liberado.

Dentro de las principales características y prestaciones con que cuenta esta versión, destacamos las siguientes:

- soporte para varios procesadores, RAID y balanceo de carga de trabajo;
- particionado de bases de datos para soportar bases de datos muy grandes;
- operaciones de indexado paralelas;
- **Mirroring** (*espejado*) de bases de datos para proveer protección ante fallos y redirección automática de clientes;
- optimización automática de bases de datos, herramientas de administración sencillas y herramientas de desarrollo avanzadas;
- servicios de autenticación, auditoría y autorización avanzados;
- encriptación avanzada de datos;
- servicio de revisión de buenas prácticas de programación;
- integración con **Microsoft Baseline Security Analyzer**;
- integración con **Microsoft Update**;
- soporte de consultas recursivas, manejo de errores y nuevos tipos de datos;
- integración con **.NET**;
- tipos definidos por el usuario;
- indexado y búsqueda de texto completo sobre **XML** (*Extensible Markup Language*);
- servicios de notificación;
- transformaciones sencillas y avanzadas mediante una nueva herramienta de extracción y transformación de datos;
- replicación transaccional utilizando un Server **Oracle** como Publicador;
- soporte para **Web Services** nativos, **WSDL** y autenticación web;
- herramientas avanzadas para el desarrollo de aplicaciones de inteligencia de negocios.
- desencadenadores DDL que pueden ser utilizados para ejecutar procedimientos almacenados cuando se emiten sentencias de modificación de estructuras (ejemplo: CREATE).
- conjuntos de resultados múltiples (MARS) que permiten que los clientes tengan más de una petición por conexión.
- sentencias TRY y CATCH en T-SQL para manejo de excepciones.
- arquitectura de seguridad mejorada a través de principales, asegurables y permisos.
- la funcionalidad de replicación ha sido mejorada para hacer más sencilla la configuración, y proveer soporte de replicación transaccional para peer-to-peer, replicación sobre HTTP y mejoras para replicación heterogénea.
- mejoras en tareas de mantenimiento, con online restore y online index operations.

El producto se presenta en las siguientes ediciones de 32 bits.

- **SQL Server 2005 Express Edition:** provee un cliente de bases de datos robusto, simple de usar e integrado con Microsoft Visual Studio 2005. Permite rápi-

das implementaciones a un amplio rango de usuarios. A la vez, por ser de licencia libre, es la mejor elección para desarrollos independientes.

- **SQL Server 2005 Mobile Edition:** extiende las capacidades de gestión de datos empresarios a dispositivos móviles. Tiene la capacidad de replicar datos con SQL Server 2005 y con SQL Server 2000, de manera que en estos dispositivos los datos permanezcan sincronizados con la base de datos maestra. Es la única edición que provee manejo relacional de los datos para dispositivos móviles.
- **SQL Server 2005 Personal Edition:** viene incluido en las ediciones **Enterprise** y **Standard**. Requiere **Licencia de Cliente (CAL)** y provee las mismas prestaciones que la edición Standard, con la diferencia de que soporta menor escalabilidad (alrededor de cinco usuarios conectados en forma concurrente). Asimismo, la capacidad de replicación se encuentra reducida a un solo suscriptor.
- **SQL Server 2005 Developer Edition:** ofrece la misma capacidad que la edición Corporativa, pero en modalidad de Licencia Restringida. Se lo utiliza para instalar ambientes de desarrollo, pero no puede ser utilizado en ambientes de producción.
- **SQL Server 2005 Desktop Engine:** esta edición ha sido diseñada para aplicaciones que requieren fuerte soporte de almacenamiento local. Ofrece seguridad, escalabilidad y compatibilidad, tanto con SQL Server 2000 como con Microsoft Access y SQL Server 7.0
Es ideal para mantener situaciones de replicación donde intervienen sólo suscriptores. No requiere licencia CAL, excepto si se conecta con un servidor.
- **SQL Server 2005 for Windows CE Edition:** ofrece las funcionalidades elementales de SQL Server 2005 para dispositivos móviles. Sus componentes sólo pueden ser instalados en estos dispositivos, para accesos de lectura y escritura.

Funcionalidades disponibles por edición

En la siguiente tabla se resumen las características y funcionalidades disponibles en cada una de las ediciones de distribución de SQL Server.

CARACTERÍSTICA	EDICIÓN CORPORATIVA	STANDAR EDITION	PERSONAL EDITION	DEVELOPER EDITION	DESKTOP ENGINE	CD EDITION
Múltiples instancias de servidor	x	x	x	x		
Conmutación por fallo	x			x		
Log Shipping	x			x		
DBCC paralelo	x			x		
CREATE INDEX paralelo	x			x		
Read ahead & scan	x			x		
Vistas indexadas	x			x		
Servidores federados	x			x		
System Area Network	x			x		

CARACTERÍSTICA	EDICIÓN CORPORATIVA	STANDAR EDITION	PERSONAL EDITION	DEVELOPER EDITION	DESKTOP ENGINE	CD EDITION
Herramientas gráficas de administración	x	x	x	x		
Utilidades gráficas de lenguaje	x					
Búsqueda de texto completo	x	x	x (No Win 98)	x		
Replicación SnapShot	x	x	x	x	x	
Replicación transaccional	x	x	Suscriptor	x	Suscriptor	
Replicación de fusión	x	x	x	x	x	Suscriptor
Actualización inmediata de suscriptores	x	x	x	x	x	
Actualización encolada de suscriptores	x	x	x	x	x	
Analysis Services	x	x	x	x		
Particiones OLAP definidas por usuario	x			x		
Ayudante de particionado	x			x		
Cubos OLAP vinculados	x			x		
Soporte para dimensiones ROLAP	x			x		
Soporte HTTP Internet	x			x		
Custom rollups	x	x	x	x		
Celdas calculadas	x			x		
Write-back dimensions	x			x		
Soporte para dimensiones muy grandes	x			x		
Acciones	x	x	x	x		
OLAP en tiempo real	x			x		
Cubos particionados distribuidos	x			x		
Minería de Datos	x	x	x	x		
English Query	x	x	x	x		

Tabla 1. Características de instalación a tener en cuenta cuando se decide utilizar **Microsoft SQL Server 2005**.

INSTALACIÓN DE SQL SERVER 2005

En las líneas subsiguientes, se incluyen algunas consideraciones que se deberán tener en cuenta antes de instalar SQL Server 2005.

Revisión del hardware mínimo requerido

Exceptuando la edición para Windows CE, todas las ediciones de SQL Server soportan la arquitectura común que se expone en la siguiente tabla, difiriendo únicamente en las funcionalidades.

CARACTERÍSTICA	TODAS LAS EDICIONES
Procesador	Intel Pentium 166 mhz o compatible
Memoria RAM	64 MB
Espacio en disco (SQL Server y componentes adicionales seleccionados)	95 Mb - 270 Mb
Desktop Engine	44 Mb
Analysis Services	50 Mb - 130 Mb
English Query	Mínimo 80 Mb
Protocolo de Red	TCP/IP
Microsoft Internet Explorer	Versión 5.0 o superior
Consola de Administración	Depende de su instalación con Internet Explorer
Administrador Corporativo y Ayuda HTML	Depende de su instalación con Internet Explorer
Acceso Web a Analysis Services y SQL Server	Requiere Internet Information Server 4.0 o superior
Instalación de Conectividad Cliente	ActiveX Data Objects 2.6 Admite Internet Explorer 4.01 SP2

Tabla 2. Componentes de hardware y software necesarios para la instalación de SQL Server 2005.

Sistema Operativo de la plataforma destino

Las diferentes opciones de instalación variarán en torno del sistema operativo utilizado en el equipo donde se pretenda instalar SQL Server. En la tabla que se incluye a continuación, es posible determinar cuáles son los sistemas operativos soportados por cada edición de SQL Server.

CARACTERÍSTICA	EDICIÓN CORPORATIVA	STANDAR EDITION	PERSONAL EDITION	DEVELOPER EDITION	DESKTOP ENGINE	CD EDITION
MS Windows NT Server 4.0 - SP 5 o superior	X	X	X	X	X	
Windows NT Server 4.0 Enterprise Edition - SP 5 o superior	X	X	X	X	X	
Windows 2005 Server	X	X	X	X	X	
Windows 2005 Advanced Server	X	X	X	X	X	
Windows 2005 Datacenter Server	X	X	X	X	X	
Microsoft Windows 98			X		X	
Windows Me			X		X	
Windows NT Workstation 4.0 - SP 5 o superior			X	X	X	
Windows 2005 Professional			X	X	X	
Windows 2005 Terminal Server				X		X
Windows CE devices						X

Tabla 3. Sistemas Operativos requeridos por cada edición de SQL Server.

Cuentas de Dominio para el servidor y la instalación

Cuando instale SQL Server 2005 sobre Windows NT o Windows 2005, tanto SQL Server como SQL Server Agent se ejecutarán como servicios y requerirán de una cuenta en el dominio (usuario y contraseña) para ser iniciados.

La cuenta de dominio permitirá que puedan autenticarse en las comunicaciones entre servidores que utilizan los servicios de replicación, los servicios de backup remoto, las consultas distribuidas, SQL Mail y los jobs del SQL Server Agent.

Por lo general, se crea una cuenta de dominio especial para ejecutar estos servicios (evitemos utilizar cuentas de usuario, debido a que, si en algún momento se modifica la contraseña de la cuenta, los servicios fallarán al iniciarse).

Por otra parte, la instalación de SQL Server deberá llevarla a cabo un usuario que posea una cuenta en el dominio de Windows y que disponga de los servicios que se enumeran a continuación.

- Permisos para cambiar el nivel de acceso al directorio de SQL Server 2005 (\Program Files\Microsoft SQL Server\Mssql).
- Permisos para cambiar el nivel de acceso a los archivos con extensión .MDF, .NDF y .LDF.
- Permisos para iniciar servicios.
- La cuenta de usuario con la que se intenta instalar el servidor tiene privilegios de administrador del sistema.

Actualización de componentes

Durante la fase de actualización de componentes en la instalación, el programa de setup de SQL Server 2005 analiza e instala, si es necesario, los siguientes componentes:

- Windows Installer 2.0
- The .NET Framework 2.0
- SQL Server Setup Support Files

Si la instalación es cancelada luego de completarse la actualización de componentes, éstos no serán eliminados.



TAMAÑO DE LOS CARACTERES

Un carácter no UNICODE ocupa, para SQL Server, 1 byte (8 bits). De esta manera, se podrán representar hasta 2^8 (256) caracteres distintos. Este conjunto es limitado para el total de caracteres que forman un idioma. A su vez, un carácter UNICODE ocupa 2 bytes, y si bien permite almacenar más caracteres, ocupa el doble de espacio en disco.

Definición de instancias de servidor a instalar

SQL Server soporta múltiples instancias propias en un mismo servidor. Cada una de ellas tendrá un nombre y una configuración específicos, aunque sólo una será considerada la instancia por omisión y podrá ser referida como **localhost**; el resto de las instancias será referenciadas por su nombre.

Esta flexibilidad que soporta SQL Server se utiliza para mantener ambientes de desarrollo y test en un mismo servidor, para habilitar distintas configuraciones de lenguajes, testear aplicaciones internacionales y reducir los requerimientos de hardware necesarios para armar ambientes de desarrollo, preproducción (**staging**) y producción.

Definición del conjunto de caracteres

Durante la instalación, será necesario seleccionar el conjunto de caracteres y el criterio de ordenamiento. Debido a que los caracteres **NO UNICODE** no son capaces de almacenar todos los caracteres disponibles para un lenguaje del mundo y a que la utilización de caracteres **UNICODE** requiere el doble de espacio de almacenamiento, se deberá elegir, durante la instalación, el **collation** (conjunto de caracteres) que contenga los que deseamos utilizar.

El conjunto predeterminado para el hemisferio occidental y Europa occidental es el **ISO**, compatible con las instalaciones de Windows (también conocido como **Conjunto de Caracteres de Windows, ISO 8859-1** o **Latin-1**). Es necesario, entonces, como paso previo a la instalación de SQL Server, verificar que las estaciones clientes que se conectarán al servidor utilicen el mismo conjunto de caracteres.

Otro factor de suma importancia que debe tenerse en cuenta es el **Modo de Ordenamiento** o **Secuencia de Intercalado** (*Sort Order*), es decir, de qué manera se van a comparar los caracteres y si dicho orden será por **Diccionario** o **Binario** (orden cultural o por valor binario interno).

Esta decisión es muy importante ya que afectará la forma en que SQL Server devolverá los **Conjuntos de Resultados** (*resultsets*), cuando no se especifique una cláusula **ORDER BY**, y cómo se asignen los valores de datos.



CONJUNTO DE CARACTERES

Las **Páginas de Código de MS-DOS** [437] y los **Conjuntos de caracteres de Windows** pueden diferir. SQL Server guarda los caracteres en formato binario y en espacios de 1 byte. Cuando una aplicación de MS-DOS accede a ese carácter, el carácter mostrado en pantalla puede diferir del almacenado en Windows, si no está entre los 128 primeros caracteres **ASCII** (caracteres sin signos diacríticos).

En algunos diseños de bases de datos se prefiere usar la distinción entre mayúsculas y minúsculas. En varios casos, puede tratarse simplemente de un efecto de visualización estética (que debería eliminarse), aunque existen diseños muy particulares que exigen la coexistencia de tablas con el mismo nombre, pero que refieren a entidades distintas. Según este criterio, en una base de datos con un conjunto de caracteres con distinción de mayúsculas podríamos tener una tabla **listaclientes**, otra **ListaClientes** y una tercera **LISTACLIENTES**, siendo todas distintas para SQL Server. También sucederá que al buscar registros con la cláusula:

```
Where apellido = 'Perez'
```

sólo recuperaremos las filas (todas) donde el campo **apellido** contiene **Perez** capitalizado en mayúsculas, no así aquellas donde el campo contiene **perez**, **Pérez** o **PEREZ**.

Por otra parte, los modos de ordenación son propios de cada Conjunto de Caracteres. La instalación predeterminada de SQL Server no distingue mayúsculas de minúsculas a los efectos de comparación y ordenamiento.

Si se selecciona la ordenación binaria, los caracteres se representarán basándose en su peso binario interno. Este tipo de ordenación distingue entre mayúsculas, minúsculas y caracteres diacríticos. Ello se convierte en un aspecto crítico a la hora de instalar SQL Server, puesto que los caracteres con signos diacríticos (como la Á o la Â) aparecerán después de la Z.

El conjunto de caracteres y el modo de ordenamiento seleccionados para la instalación de SQL Server (que pueden seleccionarse en forma independiente para las bases de datos) también afectará las cláusulas de manejo de cadenas (**LIKE**), las de valores máximos y mínimos (**MAX**, **MIN**), las de contar valores distintos (**COUNT DISTINCT**), las de agrupación y agregación (**GROUP BY**, **CUBE**, **UNION**, etc.).

Como consideración técnica de desempeño del motor, la ordenación binaria requiere menos ciclos de procesamiento que la ordenación por diccionario, por lo cual es la selección ideal si no se necesita ordenación semántica y se utilizan los caracteres estándar **ASCII**.

En una ordenación de tipo **Diccionario** con distinción de mayúsculas, las letras **abcABC** se ordenarán **AaBbCc**, mientras que en una ordenación **Binaria** lo harán bajo la forma **ABCabc**.

Tipos de instalación disponibles

Durante la instalación, SQL Server le ofrecerá distintas opciones de componentes a instalar, según el sistema operativo detectado.

Los grupos de componentes disponibles para instalar son:

- **SQL Server:** elementos de la instalación que nos permitirán instalar y configurar un Servidor de Bases de Datos SQL Server.
- **Analysis Services:** elementos de la instalación que nos ofrecerán la posibilidad de configurar un Servidor **OLAP** (*On Line Analytical Process*), es decir, un servidor de **Datawarehouse**.
- **Reporting Services:** elementos que nos permitirán instalar el servicio de generación de reportes.
- **Notification Services:** permite configurar todos los elementos que participarán en los servicios de notificación como aplicaciones.
- **Data Transformation Services:** permite instalar la aplicación para diseñar paquetes de extracción, transformación y manipulación de datos de otras fuentes desde y hacia SQL Server.
- **WorkStation Components:** elementos de configuración de puestos de desarrollo (librerías de red, ayuda, etc.)

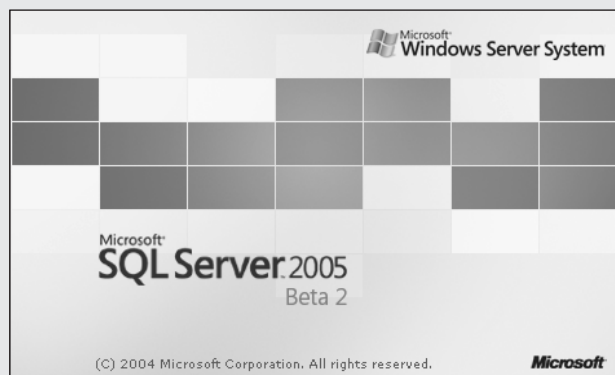
Cómo instalar SQL Server 2005 - Beta 2 1

Una vez que efectuada la verificación de la disponibilidad de la arquitectura de hardware y los aspectos relacionados con el conjunto de caracteres, comenzaremos a instalar **SQL Server Edición Corporativa**.

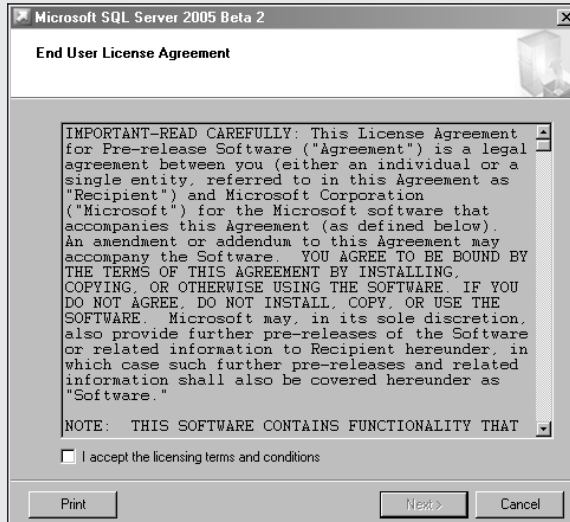
■ La instalación

PASO A PASO

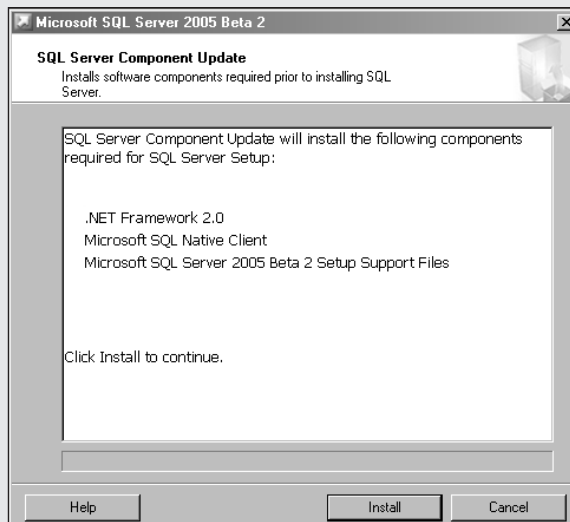
- 1 Introduzca el CD-ROM en la unidad de CD. El programa de instalación se iniciará en forma automática mostrando la ventana de bienvenida; si no sucede así, ejecute el programa **setup.exe**, situado en el directorio raíz del CD de SQL Server.



- 2 Aparecerá la pantalla **End User License Agreement** (Acuerdo de Licencia de Usuario Final): lea los términos del acuerdo y tilde la opción **I Accept the licensing terms and conditions** (Acepto los términos y condiciones de la licencia) y presione el botón **Next** (Siguiente).



- 3 La pantalla siguiente, **SQL Server Component Update** (Actualización de componentes de Servidor) le indica la lista de componentes necesarios para iniciar la instalación del servidor en sí misma. Presione el botón **Install** (Instalar) para continuar.



- 4 La próxima pantalla, **SQL Server Components Update**, le mostrará el progreso de la instalación de los componentes requeridos.

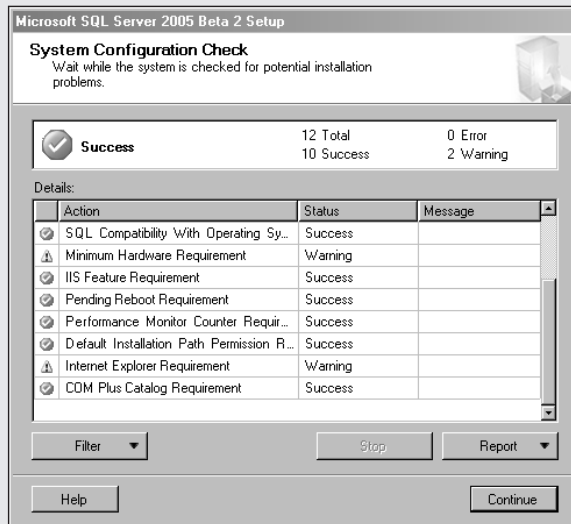


- 5 Una vez concluida la instalación de los componentes requeridos, se iniciará la instalación del servidor propiamente dicha. La ventana **Welcome to the Microsoft SQL Server Installation Wizard** (Bienvenido al asistente de instalación de Microsoft SQL Server) le ofrece el botón **Next** para continuar con la instalación.



6

La pantalla **System Configuration check** (Chequeo de configuración del sistema) ejecuta la verificación de que se han cumplido los requerimientos de sistema mínimos para ejecutar SQL Server y, por ende, la instalación. A cada elemento de sistema requerido para poder instalar SQL Server, el sistema le muestra un tilde en color verde, con la leyenda **Success** (requisito cumplido) si su equipo cumple el requisito; un símbolo amarillo de **Warning** (advertencia) si se cumple el requerimiento pero puede resultar insuficiente el recurso requerido y un símbolo en color rojo **Not Success** (requisito incumplido) si su sistema no cumple un requisito requerido, por ejemplo un parche de actualización sobre el Internet Explorer o recursos de hardware. El botón **Continue** (continuar), pasará a la pantalla de avance en el registro de la instalación en el Registro de Eventos de Windows.

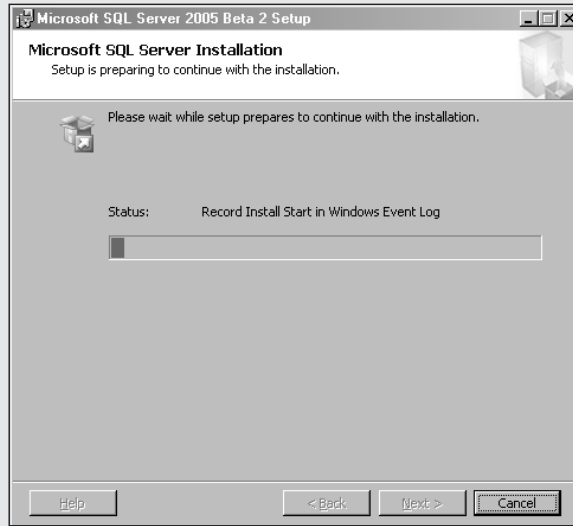


7

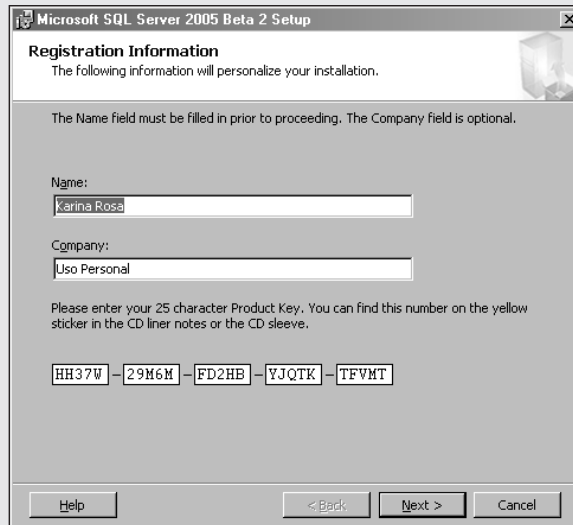
La pantalla titulada **Microsoft SQL Server Installation** le mostrará el avance del registro de eventos.

REPORTING SERVICES

Reporting Services permite crear informes de datos de SQL Server, que pueden ser diseñados usando Visual Studio .NET-based Report Designer y accedidos usando un IIS Web site. Incluye la herramienta Report Manager, que puede utilizarse para administrar la seguridad, generación y distribución de informes según agenda; y administrar la historia del informe.



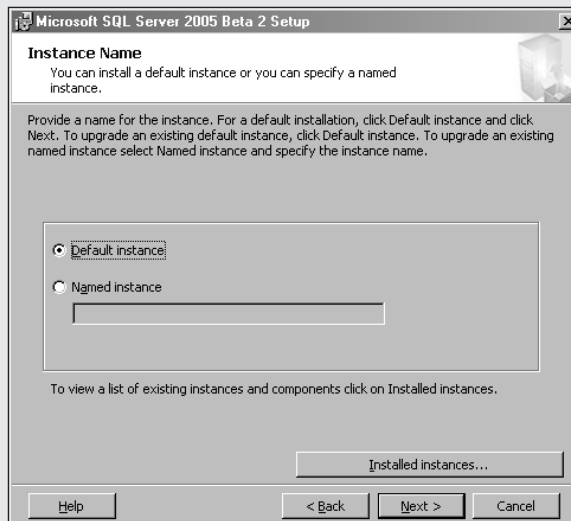
8 Una vez que ha finalizado el registro de la instalación, la pantalla denominada **Registration Information** (Información de Registro) le solicita el ingreso de los datos de registro de la instalación.



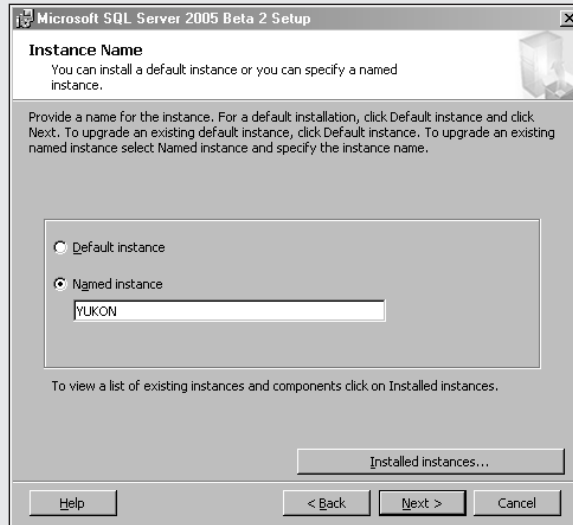
9 La pantalla **Components to Install** le permite seleccionar los productos a instalar, descritos en el apartado anterior. Si está ejecutando el instalador en un puesto de desarrollo, es decir, sobre un sistema operativo no servidor, el instalador le ofrecerá únicamente instalar la opción **WorkStation Components**. Asumimos que está instalando sólo un servidor SQL. Seleccione, por tanto, **SQL Server** y presione **Next**.



- 10** En la pantalla siguiente, **Instance Name** (Nombre de Instancia), defina el nombre de la instancia que instalará. Tilde la opción **Predeterminada** si instala SQL Server con una sola instancia o asigne un nombre a la instancia que instalará. Ahora, presione el botón **Next**. Si desea verificar qué otras instancias hay instaladas, presione el botón **Installed Instances...**



- 11** Si tiene instaladas otras versiones de SQL Server, por ejemplo, la versión 2000, SQL le ofrece la posibilidad de migrarla a la versión 2005. Si ya instaló una instancia 2005 y desea instalar otra, deberá asignarle un nombre. Presione el botón **Next**.



12

La pantalla **Service Account** le solicita indicar las cuentas de usuario con las que se ejecutará la instancia SQL Server. Si tilda la opción **Customize for each Service Account** (Personalizar para cada Cuenta), deberá especificar para cada servicio (instancia de SQL Server, OLAP, etc.) la cuenta de usuario con la cual desea que se ejecute ese servicio en el cuadro inferior.

En el caso de que no tilde esa opción, la cuenta que indique en el cuadro inferior, ejecutará la totalidad de los servicios.

La opción **Use the Local System Account** (Usar la cuenta de sistema local) ejecutará los servicios con la cuenta local con la que se inicia el servidor.

La opción **Use a Domain User Account** (Usar cuenta de usuario del Dominio) requerirá una cuenta válida del dominio al que está asociado el servidor.

La opción **Auto start service** (autoiniciar servicio) arrancará los servicios del servidor, es decir, el servidor mismo, cada vez que se inicie el sistema. Presione el botón **Next** cuando haya configurado las cuentas.

III DEFINIR OBJETOS CON CÓDIGO ADMINISTRADO

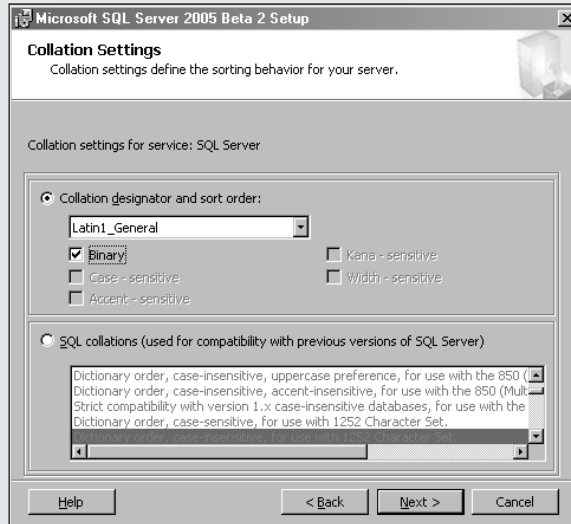
La unidad de despliegue de aplicaciones para .NET es la assembly, archivo DLL o EXE compilado, que contiene clases de definiciones y código que se puede importar a una base de datos usando el comando CREATE ASSEMBLY. Los desarrolladores pueden crear objetos de base de datos en el entorno de Visual Studio .NET y desplegarlos directamente en la base.



- 13** En la pantalla **Authentication Mode** elija el modo de autenticación (seguridad) que utilizará la instancia que está instalando. Escoja **Windows Authentication Mode** (Modo de autenticación de Windows) si desea que sólo puedan conectarse al servidor usuarios del dominio de Windows; seleccione **Mixed Mode / Windows Authentication and SQL Server Authentication** (Modo Mixto) si desea permitir las conexiones de usuarios del dominio de Windows y las de usuarios propios de SQL Server. Si elige **Mixed Mode**, se habilitará el cuadro de texto para asignar la contraseña al usuario **sa**. Es aconsejable no dejar esta contraseña en blanco ya que dejará el control total de la instancia en manos de cualquier conexión **sa**. Pulse el botón **Next**.



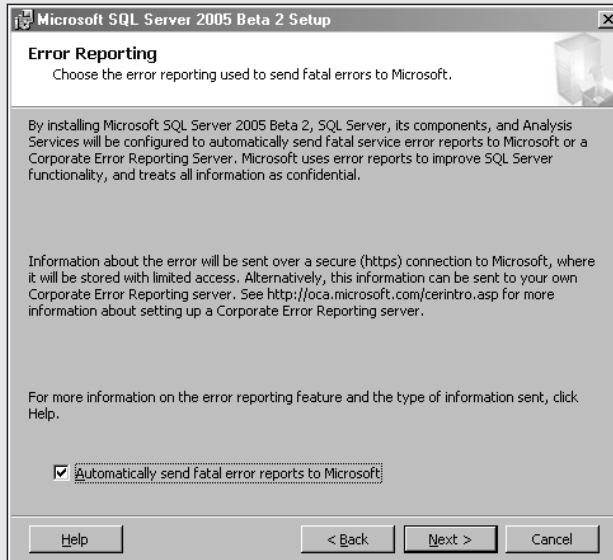
- 14 La pantalla **Collation Settings** (Configuración del conjunto de caracteres) le permitirá seleccionar el conjunto de caracteres del servidor (que asignará por omisión a todas las bases de datos). La opción **SQL Collations** le permitirá seleccionar los conjuntos de caracteres compatibles con versiones anteriores de SQL Server. Seleccione también los modos de ordenamiento por medio de las opciones: **Binary**, **Case-sensitive**, **Accent-sensitive**, **Kana-sensitive**, **Width-sensitive**. Presione el botón **Next**.



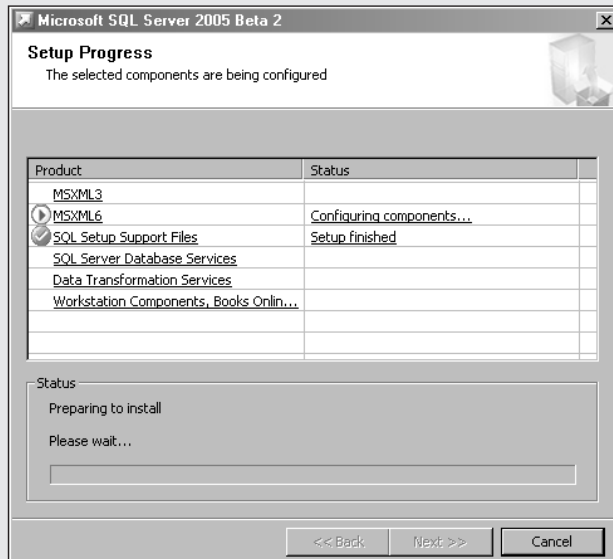
- 15 La pantalla **Error Reporting** le solicita la habilitación de reporte de errores fatales a Microsoft. Si desea habilitarla, seleccione la opción **Automatically send fatal error report to Microsoft**. Presione el botón **Next**.
- 16 La ventana titulada **Ready to Install** le brindará la posibilidad de visualizar un resumen de las opciones seleccionadas en los pasos anteriores. Presione el botón **Install** para continuar.

III USO DE HTTP

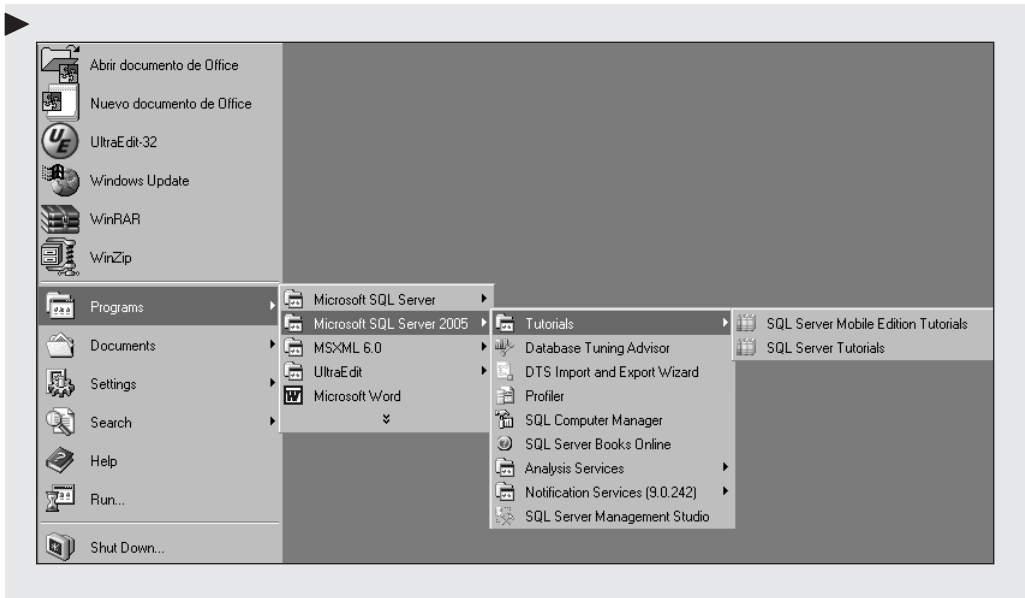
Microsoft Windows Server™ 2003 y SQL Server 2005 responden de manera innata a los requerimientos http, sin la intervención del Web Server IIS. Esto facilita implementar Web Services basados en una base de datos SQL Server.



- 17 La pantalla contigua, denominada **Setup progress**, le mostrará el avance de la instalación listando los ítems que se encuentran instalados y los ítems que se hallan en progreso.



- 18 En último término, una vez finalizada la instalación, podrá visualizar los elementos instalados. Para ello, diríjase al menú **Start** (Inicio) / **Programs** (Programas) / **Microsoft SQL Server 2005**.



PRESTACIONES DE SQL SERVER PARA SISTEMAS OLTP

SQL Server 2005 es un sistema de Gestión de Bases de Datos Relacionales cliente-servidor de alto rendimiento y escalabilidad, diseñado para soportar aplicaciones con grandes volúmenes de transacciones. Tal es el caso de las aplicaciones **OLTP** (*Online Transaction Processing*) y las aplicaciones de análisis de información **OLAP** (*Online Analytical Processing*, de minería y repositorios de datos).

Proporciona también herramientas cliente, interfaces para trabajo en red con otros sistemas operativos de Microsoft y una arquitectura abierta que permite a otros sistemas operativos (como UNIX) interactuar con él.

El motor de SQL Server y T-SQL

Consiste en el componente diseñado para soportar aplicaciones **OLTP** y **OLAP** altamente exigentes basándose en **T-SQL**, el lenguaje estructurado de consultas detrás del cual operan los principales componentes del servidor.

T-SQL, en su versión de Microsoft, es la implementación de un súper conjunto de **SQL** (*Structured Query Language*) muy potente y flexible, que permite acceder a los datos almacenados por el motor.

El **Instituto Americano de normalización** (*ANSI*) ha certificado que **T-SQL** cumple la norma SQL-92, por lo cual el lenguaje dispone del conjunto de instrucciones estándares del lenguaje estructurado de consultas y de un conjunto expandido de instrucciones que permiten obtener mejores resultados en las consultas a las bases de datos. Este conjunto expandido de instrucciones permite disponer de sentencias **SELECT** incluidas en la lista de campos de un **SELECT** inicial, es decir, obtener datos derivados de consultas a otras tablas.

Proporciona también operaciones matemáticas, por ejemplo, la raíz cuadrada y la utilización de sentencias como **CUBE** y **ROLLAP**, que permiten obtener análisis multidimensionales (similares a cortes de control) directamente desde el motor. Por otra parte, como opción a instrucciones **SELECT** complejas, se dispone de la instrucción **CASE** para la sustitución de valores condicionales.

La utilidad **sqlcmd** extiende las características del **osql** de la versión 2000, incluyendo la posibilidad de usar variables, conectarse a servidores dinámicamente para consultar información y enviar información de error al entorno al que está llamando. También provee una Conexión Dedicada de Administrador (DAC), que le permite al administrador de una base de datos conectarse al servidor y resolver problemas.

El optimizador de consultas

Es uno de los componentes más útiles del motor, dado que permite que el desarrollador se concentre en la codificación de una solución, delegando al **Optimizador de Consultas** la búsqueda de la mejor manera de devolver los resultados.

El **Optimizador** determinará cuál es el camino más conveniente para acceder a los datos, basándose en costes. Para ello, mantiene estadísticas sobre el volumen y la dispersión de los datos con el fin de crear el plan que resulte en menos operaciones de E/S (entrada/salida) para acceder a los datos y cumplir con la operación solicitada. Puesto que los optimizadores basados en costes operan con probabilidades surgidas de esas estadísticas, es posible omitir, en casos muy particulares, el plan de ejecución hecho por el motor.

La posibilidad de utilizar las **recomendaciones del optimizador** (*optimizer hints*) –indicando, por ejemplo, qué índices debe utilizar una consulta– cambiarán el plan de ejecución seleccionado por el Optimizador.

Por otra parte, T-SQL proporciona sentencias de control de flujo como **IF-THEN-ELSE**, **WHILE**, bucles y variables. Éstas ayudan a simplificar el desarrollo de aplicaciones y permiten, de esta forma, entregar al motor de SQL Server la lógica de

aplicación exclusiva sobre datos, con lo que mejora de manera ostensible el rendimiento de las aplicaciones cliente/servidor.

Esta característica resulta de suma utilidad al implementar aplicaciones **multicapa**, donde cada capa se ocupa de su especialización. En este tipo de aplicaciones, tienen un rol muy importante los **procedimientos almacenados** que, como conjuntos de instrucciones programados en **T-SQL**, permiten separar el comportamiento de la capa de datos del de la capa de negocios.

Tendríamos, por ejemplo, un procedimiento almacenado que devuelve el cálculo del monto de un impuesto a partir de la recepción del monto base y el porcentaje –como parámetros–. En este caso, el porcentaje que se quiere aplicar se almacenaría en la capa de negocios y sólo allí deberían introducirse modificaciones a ese porcentaje, y, en el procedimiento almacenado en el motor, permanecerían invariables la lógica matemática y el rango de datos sobre los que operan los cálculos.

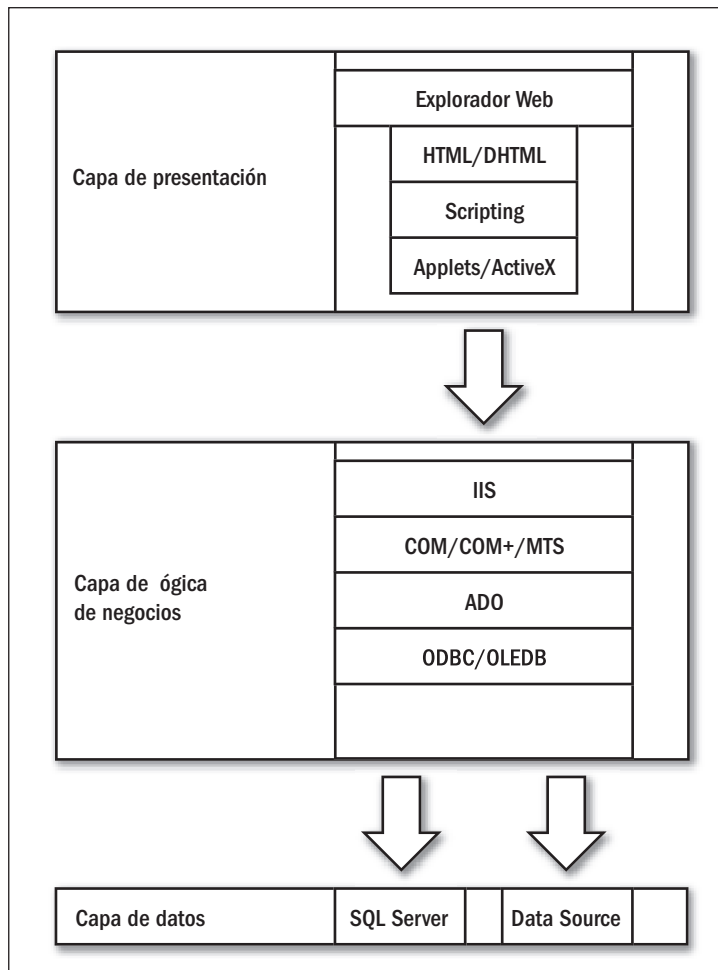


Figura 2. Esquema de una implementación multicapa por medio de una aplicación web.

Procedimientos almacenados extendidos

Son recursos exclusivos de SQL Server que permiten ampliar su funcionalidad y acceder a recursos ajenos a él como mensajería, escritura de archivos **HTML** y seguridad. Es posible crear procedimientos almacenados extendidos, bajo la forma de bibliotecas de vínculos dinámicos o **DLL** (*Dynamic Link Library*)

Integridad de datos

El motor de SQL Server permite asegurar que los datos almacenados son válidos mediante tres tipos de integridad. Mediante la **Integridad de Dominio**, utilizando las restricciones **PRIMARY KEY**, se impone la asignación de una clave primaria inviolable para cada tabla. También permite la composición de claves alternativas mediante las restricciones **UNIQUE**, garantizando de esta manera la unicidad de una clave compuesta por varios campos.

Para los casos en los que no es posible determinar la unicidad de un campo por cada fila de una tabla, SQL Server dispone de un campo autonumérico exclusivo que se implementa mediante la propiedad **identity** (*identidad*).

SQL Server implementa el segundo tipo de integridad, la **Integridad Referencial**, mediante las restricciones **FOREIGN KEY** (*clave externa*), por la cual una columna de una tabla hace referencia a un campo de tipo **PRIMARY KEY** de otra. Luego de definir la relación, el motor de SQL Server es el encargado de verificar el cumplimiento de la relación establecida. Las características de **update** y **delete** en cascada refuerzan el mantenimiento de la integridad referencial, que en versiones anteriores era necesario realizar mediante **Triggers**.

El tercer tipo de integridad, la **Integridad de dominio de Datos**, se ocupa de verificar que los datos introducidos en un campo se ajusten al rango y al tipo definidos para ese campo. SQL Server implementa esta integridad mediante la verificación de aceptación de valores nulos y restricciones de nivel campo.

Triggers (desencadenadores)

SQL Server 2005 incluye, a los ya conocidos desencadenadores de la versión 2000, los desencadenadores DDL.

Al contrario de los desencadenadores ordinarios que están asociados con tablas, un desencadenador DDL se ejecuta cuando se emite una sentencia particular, sin importar el destino de ésta. Por ejemplo, un desencadenador DDL **DROP_TABLE** se disparará cuando se elimine cualquier tabla. Algunos desencadenadores DDL operan a nivel de base de datos y sólo se disparan cuando el evento correspondiente ocurre en una base de datos específica.

La utilidad sqlcmd

No es siempre posible o deseable usar una herramienta gráfica como SQL Server Management Studio para ejecutar sentencias de Transact-SQL (por ejemplo, una serie de trabajos agendados para su ejecución desatendida). SQL Server 2005 incluye una utilidad llamada sqlcmd para facilitar este requerimiento, que permite ejecutar sentencias y scripts Transact-SQL desde la línea de comando y agendar trabajos agrupados mediante OLE DB.

Indexado

El motor de indexado de SQL Server posibilita el escaneo y ordenamiento paralelo de índices, característica de mayor provecho en servidores multiprocesador. Se han mejorado, respecto de la versión 7.0 de SQL Server, las tareas de indexado, al agregar, para la versión 2005, la opción **`SORT_IN_TEMPDB`** que utiliza la base **tempdb** para almacenar los resultados del ordenamiento durante la creación de índices (esta característica es de mayor utilidad cuando la base **tempdb** está en otro **filegroup** (Grupo de Archivos)).

Procesamiento de transacciones

El procesamiento de transacciones permite garantizar que las bases de datos permanezcan siempre en un estado coherente y consistente. Está conformado por un conjunto de sentencias T-SQL incluidos entre dos sentencias, **`BEGIN TRANSACTION`** y **`COMMIT TRANSACTION`**, con una tercera sentencia, **`ROLLBACK TRANSACTION`**, que da la opción de deshacer todas las operaciones sobre los datos ante cualquier situación de error.

SQL Server garantiza las propiedades **ACID** (*Atomicity, Consistency, Isolation, Durability*) de las transacciones de procesamiento de datos.

Se entiende por **Atomicidad** (*Atomicity*) la acción de tratar las operaciones incluidas en una transacción como si fueran una sola, donde se confirman todas las operaciones o se cancelan todas (aun aquellas donde no hubo error).

Mediante la **Consistencia** (*Consistency*) se asegura por redundancia que el sistema no quede en un estado inconsistente, es decir, con transacciones *pendientes* de ser cerradas en el momento de un fallo.

El **Aislamiento** (*Isolation*) es una propiedad que garantiza que, mientras se está realizando una transacción, se adquiere el nivel de bloqueo necesario para no producir distorsión en los datos que está visualizando un usuario. El bloqueo adquirido durante una transacción reduce la concurrencia de usuarios sobre los datos, pero asegura que durante el curso de una transacción no se alteren las filas involucradas en ella. La propiedad **Durabilidad** (*Durability*) de una transacción asegura que, una vez confirmados los cambios de una transacción, éstos persistan aun ante fallas de hardware.

Mejoras en la seguridad de SQL Server 2005

SQL Server 2005 implementa mejoras en la arquitectura de seguridad que usa un sistema de jerarquías de objetos asegurables.

Al igual que en las versiones anteriores, puede proveer accesos al sistema SQL Server 2005 a través de Microsoft Windows® logins o de SQL Server logins, pero en esta versión de SQL Server también puede aplicar políticas de seguridad a los SQL Server logins. Estas políticas actúan de forma similar a las del sistema operativo Windows cuando aplica restricciones mediante principales, asegurables y permisos.

Alta disponibilidad

Para aplicaciones de misión crítica, SQL Server proporciona opciones de alta disponibilidad mediante copias de seguridad en línea, mantenimiento en línea, recuperación automática e instalación de servidores en *clusters* de discos.

Por medio de **SQL Server Agent** se pueden programar copias de seguridad automática, sin que intervenga un administrador. También se pueden realizar tareas de cambios sobre el diseño de las bases, de mantenimiento y administración, sin apagar el motor. Asimismo, en la edición Corporativa, SQL Server provee **conmutación por fallos** (*failover*) sobre instalaciones basadas en clusters.

Vistas de metadatos estáticos y dinámicos

Puede usar las vistas de catálogo de SQL Server 2005 para consultar metadatos estáticos. Éstas proveen información acerca de los objetos en la base de datos de SQL Server, como tablas, procesos almacenados y restricciones. También puede usar vistas de administración y funciones de administración dinámicas para consultar metadatos dinámicos en SQL Server 2005, que proveen información del estado actual del SQL Server, tal como los bloqueos adquiridos dentro de una base de datos.

Replicación

Es el procedimiento por el cual un servidor SQL Server propaga copias de datos a otros servidores en emplazamientos remotos, manteniendo y asegurando la integridad de los datos. La replicación se implementa por actualizaciones de una base que se envían al destino un tiempo más tarde permitiendo trabajar en ambas bases y fusionando luego los cambios; por copia de una vista entera de datos hacia una base de datos en otro destino, sobrescribiendo la copia anterior sin verificar actualizaciones y tal como están los datos a un determinado momento; y por envío de un conjunto inicial de datos al destino con la replicación de las transacciones **INSERT**, **UPDATE** y **DELETE**.

Información de sesión compartida

SQL Server permite almacenar hasta 128 Kb de información binaria acerca del contexto de la conexión, asociada a la variable **@@SPID**, permitiendo el control de flujo basado en la actividad anterior de la conexión.

Propiedades extendidas

Permiten al desarrollador definir comentarios adicionales para los objetos (como columnas e índices) mediante algunos procedimientos almacenados extendidos. Estas propiedades pueden ser obtenidas por las aplicaciones cliente para entender el objeto.

Servicios de metadatos

Los servicios de metadatos extienden el Motor de Repositorio Microsoft para la administración de metadatos (palabra utilizada para aquellos elementos que guardan información acerca de otros objetos).

Estos servicios soportan el modelo **OIM** (*Open Information Model*), que consiste en un conjunto de modelos de objetos que pueden ser usados para intercambiar metadatos en formatos comunes con otros sistemas que pueden procesar y utilizar metadatos.

Soporte XML

SQL Server ofrece soporte nativo para leer, escribir, almacenar, entregar y actualizar documentos **XML** (*Extensible Markup Language*), en forma de **tags** (*marcas*) y datos. Provee los procedimientos almacenados para el armado de documentos.

Recuperación ante desastres en SQL Server 2005

SQL Server 2005 continúa soportando clustering, backup y restauración, y log shipping como mecanismo de recuperación ante desastres. Además, agrega un número de mejoras como snapshots de base de datos (copia de sólo lectura de la base de datos hecha en un momento en particular), operaciones online, mirroring, verificación improvisada y checksums para verificar backups cuando los datos son restaurados.

RESUMEN

La versión revisada del primer informe del Dr. Codd, publicada en 1970 en “Communications of the ACM”, desató una verdadera “fiebre” de proyectos de investigación. En aquel tiempo, la persistencia de información se realizaba mediante el manejo de archivos en *mainframes*, cuyos elevados costos en equipos y tiempo de procesamiento abrió el mercado al desarrollo del paradigma cliente/servidor y a las redes de computadoras.

Las alianzas empresariales, la fiabilidad de los productos y el uso de un lenguaje de consultas común y estandarizado permitió el progreso de estos sistemas, ampliando sus funcionalidades iniciales. Microsoft fue el participante más activo y nos brinda su última versión: SQL Server 2005.



TEST DE AUTOEVALUACIÓN

- 1** ¿Cuándo apareció por primera vez el estudio del doctor Codd respecto del modelo relacional?

- 2** ¿Qué es un Sistema de Gestión de Bases de Datos Relacionales?

- 3** ¿Cuáles son los objetivos de un Sistema de Gestión de Bases de Datos Relacionales?

- 4** ¿Cuándo enunció el doctor Codd las trece reglas para evaluar un Sistema de Gestión de Bases de Datos Relacionales?

- 5** ¿Cuándo se presentó el modelo semántico denominado “Diagrama de Entidad/Relación” y quién fue su autor?

- 6** ¿Cuál fue el primer proyecto dedicado al estudio de la Teoría del Modelo Relacional de Codd?

- 7** ¿Cuál fue el producto desarrollado por Microsoft asociado con Sybase?

- 8** ¿Cuál es la edición de SQL Server 2005 que permite implementar Conmutación ante fallos (failover)?

- 9** ¿Cuál es la edición de SQL Server 2005 que se utiliza para implementar servidores de desarrollo?

- 10** ¿Cuál es la ventaja de poder instalar varias instancias de SQL Server? ¿Cuánta memoria RAM mínima necesita una instalación de SQL Server?

- 11** ¿Por medio de qué mecanismo permite SQL Server extraer, manipular y guardar información de otras fuentes de datos?
