



© JMA 2016. All rights reserved

---

## INTRODUCCIÓN

---

© JMA 2016. All rights reserved

## Características

- Microsoft SQL Server es un sistema para la gestión de bases de datos producido por Microsoft basado en el modelo relacional cuya principal función es la de almacenar y consultar datos solicitados por otras aplicaciones, sin importar si están en el mismo equipo, si están conectadas a una red local o si están conectadas a través de internet.
- SQL Server es algo mas que un SGBDR, es una plataforma de datos de misión critica, ofreciendo desarrollo dinámico e inteligencia de negocio, y va mas allá de la estructura de datos relacional.

© JMA 2016. All rights reserved

## Modelo relacional

- Basado en la teoría conjuntos y el álgebra relacional
- Visión de la organización de los datos muy sencilla
  - Filas independientes
  - Filas únicas
  - Columnas independientes
  - Valores de las columnas unitarios
- Los usuarios no necesitan tener un conocimiento de la posición física de los datos, la navegación por la base de datos se realiza de forma automática
- Lenguaje de acceso a los datos muy sencillo (SQL)
- Sencillez del lenguaje, similar al lenguaje natural

© JMA 2016. All rights reserved

## Soporte a otros modelos

- Jerárquico
  - La posibilidad de almacenar datos en formato XML le permite trabajar con datos semiestructurados siguiendo un modelo jerárquico.
  - El uso de esquema XSD impone unas restricciones tan fuertes como las del modelo relacionas.
  - Los datos se pueden consultar mediante XPath y Xquery.
- Objetos
  - La posibilidad de crear nuevos tipos de datos CLR permite el almacenamiento de objetos
- Documental
  - La búsqueda de texto completo permite el tratamiento de los campos de texto muy largo como si fueran documentos

© JMA 2016. All rights reserved

## Características habituales

- Almacenamiento, extracción y actualización de datos
- Catálogo accesible por el usuario
- Servicios para mejorar la independencia de datos (metadatos)
- Soporte para la tramitación de datos (comunicaciones)
- Soporte multiusuario y servicios de autorización
- Servicios de control de concurrencia
- Soporte de transacciones
- Servicios de integridad
- Servicios de recuperación (y restauración)
- Servicios de utilidad: Importación/Exportación, Monitorización y Análisis de rendimientos, Automatización

© JMA 2016. All rights reserved

## Características

- Fusiona el procesamiento transaccional en línea (OLTP) y el de procesamiento analítico en línea (OLAP)
- Facilidad de uso
  - SQL Server Management Studio
  - SQL Management Objects (SMO)
- Disponibilidad
  - Reflejo de base de datos (1 BD espejo)
  - Clústeres de conmutación por error
  - Instantáneas de base de datos
  - Recuperación rápida
  - Conexión del administrador no compartida
  - Operaciones en línea (Índices, Operaciones y Restauración)
  - Réplica (incluida entre iguales)

© JMA 2016. All rights reserved

## Características

- Escalabilidad
  - Partición de tablas e índices
  - Aislamiento de instantáneas
  - Supervisor de réplicas
  - Compatible con Itanium 2 de 64 bits y x64
- Seguridad
  - Autenticación
  - Autorización
  - Cifrado nativo y Cifrado de base de datos transparente
  - SQL Server y Trustworthy Computing (Seguro por diseño, inicio e implementación)

© JMA 2016. All rights reserved

## Características

- Rendimiento
  - Compresión de datos y de copia de seguridad
  - Partición de Tablas e Índices
  - Índices filtrados, Columnas dispersas
  - Regulador de recursos
  - Copias de seguridad reflejadas
- Administración
  - Gestión de la administración basada en directivas
  - Servidores de administración central
  - Recopilador de datos y Seguimiento de cambios
  - Auditoría

© JMA 2016. All rights reserved

## Características

- Extensibilidad
  - Business Intelligence Development Studio
  - Integración CLR/.NET Framework
  - Tipos y agregados definidos por el usuario
  - Servicios Web y XML
  - Service Broker

© JMA 2016. All rights reserved

## Inteligencia empresarial

- La plataforma de datos SQL Server incluye las siguientes herramientas:
  - Motor de base de datos
  - Analysis Services.
  - Reporting Services.
  - Integration Services
  - Servicios de réplica.
  - Master Data Services
  - Service Broker

© JMA 2016. All rights reserved

## Motor de base de datos

- El Motor de base de datos es el servicio principal para almacenar, procesar y proteger los datos.
- El Motor de base de datos proporciona acceso controlado y procesamiento rápido de transacciones para cumplir los requisitos de las aplicaciones consumidoras de datos más exigentes de su empresa.
- El Motor de base de datos también proporciona una completa compatibilidad para mantener una gran disponibilidad.

© JMA 2016. All rights reserved

## Analysis Services

- Datos multidimensionales:
  - Analysis Services admite OLAP al permitir al usuario diseñar, crear y administrar estructuras multidimensionales que contienen datos agregados de otros orígenes tales como bases de datos relacionales.
- Minería de datos:
  - Analysis Services permite al usuario diseñar, crear y visualizar modelos de minería de datos.
  - Estos modelos de minería de datos se pueden construir a partir de otros orígenes de datos empleando una amplia variedad de algoritmos de minería de datos estándar.

© JMA 2016. All rights reserved

## Integration Services

- Integration Services es una plataforma para generar soluciones de integración de datos de alto rendimiento, lo que incluye paquetes que proporcionan procesamiento de extracción, transformación y carga (ETL) para almacenamiento de datos.

© JMA 2016. All rights reserved

## Reporting Services

- Reporting Services ofrece funcionalidad empresarial de informes habilitados para Web con el fin de poder crear informes que extraigan contenido a partir de una variedad de orígenes de datos, publicar informes con distintos formatos y administrar centralmente la seguridad y las suscripciones.
- Una solución global para crear, administrar y proporcionar tanto informes tradicionales orientados al papel como informes interactivos basados en la Web.

© JMA 2016. All rights reserved

## Replicación

- La replicación es un conjunto de tecnologías destinadas a la copia y distribución de datos y objetos de base de datos de una base de datos a otra, para luego sincronizar ambas bases de datos con el fin de mantener su coherencia.
- La replicación permite distribuir datos a diferentes ubicaciones y a usuarios remotos o móviles mediante redes de área local y de área extensa, conexiones de acceso telefónico, conexiones inalámbricas e Internet.

© JMA 2016. All rights reserved



## Master Data Services

- Master Data Services es el origen de datos maestros para la organización.
- Mediante la integración de diferentes sistemas de análisis y operaciones con Master Data Services, puede asegurarse de que todas las aplicaciones de la organización dependen de una fuente de información central y precisa.
- Con Master Data Services, se crea un origen único de datos maestros y se mantiene un registro de los datos, que cambian en el tiempo y que se puede controlar.

© JMA 2016. All rights reserved

## Service Broker

- Service Broker ayuda a los desarrolladores de software a crear aplicaciones de base de datos escalables y seguras.
- Esta nueva tecnología de Motor de base de datos proporciona una plataforma de comunicación basada en mensajes que permite a los componentes de aplicación independientes trabajar como un conjunto funcional.
- Service Broker incluye infraestructura para programación asíncrona que se puede utilizar para aplicaciones en una base de datos única o instancia única, y también para aplicaciones distribuidas.

© JMA 2016. All rights reserved

## Herramientas de administración

- SQL Server incluye herramientas integradas de administración para administración y optimización avanzadas de bases de datos, así como también integración directa con herramientas tales como Microsoft Operations Manager (MOM) y Microsoft Systems Management Server (SMS).
- Los protocolos de acceso de datos estándar reducen drásticamente el tiempo que demanda integrar los datos en SQL Server con los sistemas existentes.
- Asimismo, el soporte del servicio Web nativo está incorporado en SQL Server para garantizar la interoperabilidad con otras aplicaciones y plataformas.

© JMA 2016. All rights reserved

## Herramientas de desarrollo

- SQL Server ofrece herramientas integradas de desarrollo para el motor de base de datos, extracción, transformación y carga de datos, minería de datos, OLAP e informes que están directamente integrados con Microsoft Visual Studio para ofrecer capacidades de desarrollo de aplicación de extremo a extremo.
- Cada subsistema principal en SQL Server se entrega con su propio modelo de objeto y conjunto de interfaces del programa de aplicación (API) para ampliar el sistema de datos en cualquier dirección que sea específica de su negocio.

© JMA 2016. All rights reserved

## SQL Server Management Studio

- SQL Server Management Studio es un entorno integrado para obtener acceso a todos los componentes de SQL Server, configurarlos, administrarlos y desarrollarlos.
- SQL Server Management Studio combina un amplio grupo de herramientas gráficas con una serie de editores de script enriquecidos para ofrecer acceso a SQL Server a desarrolladores y administradores de todos los niveles de especialización.
- SQL Server Management Studio combina las características del Administrador corporativo, el Analizador de consultas y Analysis Manager, herramientas incluidas en versiones anteriores de SQL Server, en un único entorno.
- Además, SQL Server Management Studio funciona con todos los componentes de SQL Server, como Reporting Services, Integration Services y SQL Server Compact 3.5 SP2.
- Los administradores de bases de datos obtienen una única herramienta completa que combina herramientas gráficas fáciles de usar con funcionalidad de scripting enriquecida.

© JMA 2016. All rights reserved

## TRANSACT-SQL

© JMA 2016. All rights reserved

# Lenguaje SQL

- Definición
  - Lenguaje de comandos que se utiliza en los sistemas de gestión de bases de datos relacionales
- Características
  - Unificado para las bases de datos relacionales
  - Unificado para todo tipo de usuarios (dba, usuarios finales, programadores, etc.)
  - Parecido al idioma inglés
  - No procedimental
- Funcionalidad
  - CREAR tablas, vistas y sinónimos
  - ALMACENAR información en las tablas
  - MODIFICAR la información ya almacenada
  - CONSULTAR la base de datos
  - MANTENER la propia base de datos

© JMA 2016. All rights reserved

## Estructura del Lenguaje SQL

- El lenguaje de Consultas Estructurado SQL se puede dividir en:
  - Lenguaje DQL (Lenguaje de Consulta de Datos)
    - Sentencia SELECT
  - Lenguaje DML (Lenguaje de Manipulación de Datos)
    - Sentencias INSERT, DELETE, UPDATE y MERGE
  - Lenguaje DDL (Lenguaje de Definición de Datos)
    - Sentencias CREATE, DROP y ALTER
  - Lenguaje DCL (Lenguaje de Control de Datos)
    - Sentencias GRANT, REVOKE y DENY
  - Lenguaje DVL (Lenguaje de Validación de Datos)
    - Sentencias COMMIT, ROLLBACK y SET TRANSACTION

© JMA 2016. All rights reserved

# Transact-SQL

- Es el lenguaje que se utiliza para administrar instancias del SQL Server Database Engine, para crear y administrar objetos de base de datos, y para insertar, recuperar, modificar y eliminar datos.
- Transact-SQL es una extensión (supraconjunto) del lenguaje definido en los estándares de SQL publicados por International Standards Organization (ISO) y American National Standards Institute (ANSI).
- Cuenta con:
  - Instrucciones SQL estándar ampliadas
  - Instrucciones procedurales (declaración de variables, expresiones, control, excepciones, ...)
  - Instrucciones para administración, monitorización y optimización
  - Tablas, vistas, procedimientos y constantes del sistema

© JMA 2016. All rights reserved

## Introducción a la sintaxis

- No es sensible a mayúsculas y minúsculas salvo las cadenas en cuyo caso dependen de la intercalación de la columna.
- Varias líneas por instrucción y una instrucción por línea o usar ; de finalización de instrucción.
- Constantes:
  - Valores numéricos: Conjuntos de dígitos con el punto como separador decimal
  - Valores binarios: Tienen el prefijo 0x y son cadenas de números hexadecimales
  - Valores no numéricos y cadenas: '...', cadenas Unicode: N'...' [COLLATE intercalación]
    - Comillas dentro de una cadena: '..."', el . como separador de decimales
  - Valores uniqueidentifier: 32 dígitos hexadecimales en formato cadena (8-4-4-12) o en formato hexadecimal:
    - '6F9619FF-8B86-D011-B42D-00C04FC964FF' = 0x6F9619FF8B86D011B42D00C04FC964FF
- Comentarios:
  - 
  - /\* \*/
- Bloques:
  - BEGIN
  - ...
  - END

© JMA 2016. All rights reserved

# NULL y UNKNOWN

- NULL indica que el valor es desconocido.
- Un valor NULL no es lo mismo que un valor cero o vacío.
- No hay dos valores NULL que sean iguales.
- La comparación entre dos valores NULL, o entre un valor NULL y cualquier otro valor, tiene un resultado desconocido porque el valor de cada NULL es desconocido.
- Normalmente, los valores NULL indican que los datos son desconocidos, no aplicables o que se van a agregar posteriormente.
- Hay que tener en cuenta lo siguiente sobre los valores NULL:
  - Para comprobar si hay valores NULL en una consulta hay que usar IS NULL o IS NOT NULL en la cláusula WHERE.
  - Los valores NULL se pueden insertar en una columna si se indica explícitamente NULL en una instrucción INSERT o UPDATE, o si se deja fuera una columna de una instrucción INSERT.
  - Los valores NULL no se pueden usar como la información necesaria para distinguir una fila de una tabla de otra fila, como, por ejemplo, las claves principales, o bien para la información que se usa para la distribución de filas, como las claves de distribución.
- Cuando hay valores NULL en los datos, los operadores lógicos y de comparación pueden devolver un tercer resultado UNKNOWN (desconocido) en lugar de simplemente TRUE (verdadero) o FALSE (falso).

© JMA 2016. All rights reserved

# Identificadores

- **Identificadores estándar**
  - El primer carácter debe ser un carácter alfabético
  - De uno a 128 caracteres.
  - Otros caracteres pueden incluir letras, números o símbolos: @, \$, # o \_.
  - Los identificadores que comienzan con un símbolo tienen usos especiales: @, # o ##.
- **Identificadores delimitados**
  - Se utilizan cuando los nombres contienen espacios incrustados
  - Se utilizan cuando partes de los nombres incluyen palabras reservadas
  - Deben encerrarse entre corchetes ([ ])
  - o dobles comillas (" ") si SET QUOTED\_IDENTIFIER ON
- **Directrices de denominación para los identificadores**
  - Poner nombres cortos
  - Utilizar nombres significativos cuando sea posible
  - Utilizar una convención de denominación clara y sencilla
  - Utilizar un identificador que distinga el tipo de objeto (Vistas o Procedimientos almacenados)
  - Hacer que los nombres de los objetos y de los usuarios sean únicos
- **Identificador completo**
  - [Servidor.][Base de datos.][Esquema.]Objeto[.Subelemento]

© JMA 2016. All rights reserved

# Bases de Datos

- Bases de Datos de sistema
  - Almacenan información necesaria para el funcionamiento del servidor SQL Server
  - SQL Server utiliza estas BBDD para manejar y gestionar el sistema
- Bases de datos de usuarios
  - Bases de Datos donde se implementan los modelos de datos de los usuarios (OLTP)
- Almacén de datos
  - Organizar grandes cantidades de datos estables para facilitar el análisis y la recuperación con OLAP

© JMA 2016. All rights reserved

## Objetos de la BBDD

- Diagramas de Bases de Datos
  - Herramienta visual que permite diseñar y ver una base de datos completa o parcialmente
- Tablas
  - Colección de datos organizados en filas y columnas
- Índices
  - Estructura de datos que proporciona una manera rápida de acceder a la información
- Restricciones
  - Sirven para definir reglas relativas a los valores permitidos en las columnas
  - Es el mecanismo utilizado para reforzar la integridad referencial en las Bases de Datos que SQL Server exige automáticamente
  - Las restricciones establecen los valores aceptados por las columnas: nulos, rangos de valores, duplicados, claves principales o ajenas.

© JMA 2016. All rights reserved

## Objetos de la BBDD

- Estadísticas
  - Índices de distribución de valores que utiliza el optimizador de consultas para determinar el plan de consultas óptimo.
- Vistas
  - Proporciona una forma de ver los datos provenientes de una o más tablas
  - Sirven para facilitar el acceso a los datos, y garantizar la seguridad de los mismos
- Sinónimos
  - Es un nombre alternativo para un objeto de ámbito de esquema
- Service Broker
  - Tipos y contratos, colas, servicios, rutas, enlaces.
- Almacenamiento
  - Catálogos e índices de texto
    - Permiten las búsquedas de texto complejas en datos de cadenas de caracteres.
  - Esquemas de partición
  - Funciones de partición

© JMA 2016. All rights reserved

## Objetos para programación

- Procedimientos Almacenados
  - Conjunto de sentencias de Transact-SQL precompiladas con nombre que se ejecutan como una unidad
  - Mejoran el rendimiento del sistema
  - Se pueden implementar en .NET CLR
- Funciones (SQL definidas por el usuario)
  - Funciones definidas por el usuario
  - Conjunto de sentencias de Transact-SQL o CLR que suelen utilizarse para encapsular una transformación lógica
  - Pueden devolver un valor escalar o una tabla
- Desencadenadores
  - También llamados triggers o disparadores
  - Procedimiento almacenado de ejecución automática ante una operación DML o DDL
  - Se crean sobre una tabla
  - Ayudan a mantener la integridad

© JMA 2016. All rights reserved



## Objetos para programación

- **Ensamblados**
  - Son archivos DLL que se utilizan en una instancia de SQL Server para implementar funciones, procedimientos almacenados, desencadenadores, agregados definidos por el usuario y tipos definidos por el usuario que están escritos en uno de los lenguajes de código administrado que se alojan en Microsoft .NET Framework.
- **Tipos de Datos**
  - Definen los valores permitidos para una columna o variable
  - SQL Server suministra un conjunto de datos definidos por el sistema, aunque el usuario puede crearse tipos de datos propios
- **Reglas**
  - Contiene información que define los valores permitidos que serán almacenados en una columna o para un tipo de dato determinado
- **Valores predeterminados**
  - Valor por defecto para cuando no se suministre valor en una columna determinada

© JMA 2016. All rights reserved

## Objetos para la seguridad

- **Inicios de sesión**
  - Cuentas que controlan el acceso al sistema SQL Server.
- **Usuarios**
  - Identificador que asocia un inicio de sesión a un usuario dentro de una base de datos para otorgar los permisos.
- **Esquemas**
  - Es una colección de entidades de base de datos que forma un solo espacio de nombres
- **Funciones (Roles)**
  - Agrupan los permisos necesarios para los diferentes tipos de usuarios.
- **Claves simétricas y asimétricas**
  - Encriptación y desencriptación, autenticaciones, ..
- **Certificados**
  - Autenticación ante terceros (comunicaciones seguras) y cifrado
- **Credenciales**
  - Una credencial es un registro que contiene la información de autenticación necesaria para conectarse a un recurso fuera de SQL Server. La mayoría de las credenciales constan de un inicio de sesión de Windows y una contraseña.

© JMA 2016. All rights reserved

## Otros objetos

- Propiedades extendidas
  - Propiedades definidas por los usuarios para almacenar información adicional o específica sobre los objetos de una base de datos.
- Intercalaciones
  - Controlan el juego de caracteres utilizado en el almacenamiento físico de las cadenas, especificando las combinaciones de bits que representa cada carácter y las reglas por las que los caracteres se ordenan y comparan.

© JMA 2016. All rights reserved

## Tipos de datos

- Numéricos exactos
- Numéricos aproximados
- Fecha y hora
- Cadenas de caracteres
- Cadenas de caracteres Unicode
- Cadenas binarias
- Otros tipos de datos

© JMA 2016. All rights reserved

## Tipos de datos

Tipo	Bytes	Descripción
bigint	8	Datos enteros (números enteros) comprendidos entre $-2^{63}$ (-9223372036854775808) y $2^{63}-1$ (9223372036854775807).
int	4	Datos enteros (números enteros) comprendidos entre $-2^{31}$ (-2.147.483.648) y $2^{31}-1$ (2.147.483.647).
smallint	2	Datos enteros comprendidos entre 215 (-32.768) y 215 - 1 (32.767).
tinyint	1	Datos enteros comprendidos 0 y 255.
bit	1*	Datos enteros con valor 1 ó 0.
decimal	5, 9, 13, 17	Datos de precisión y escala numérica fijas con 38 dígitos significativos (escala + precisión). [vardecimal con SP2]
numeric		Funcionalmente equivalente a decimal.

© JMA 2016. All rights reserved

## Tipos de datos

Tipo	Bytes	Descripción
money	8	Valores de moneda comprendidos entre -922.337.203.685.477,5808 y +922.337.203.685.477,5807, con una precisión de una diezmilésima de la unidad monetaria. (escala 4)
smallmoney	4	Valores de moneda comprendidos entre -214.748,3648 y +214.748,3647, con una precisión de una diezmilésima de la unidad monetaria.
Numéricos aproximados		
float	4, 8	Números con precisión de coma flotante comprendidos entre $-1,79E+308$ y $1,79E+308$ .
real	4	Números con precisión de coma flotante comprendidos entre $-3,40E+38$ y $3,40E+38$ . float(24)

© JMA 2016. All rights reserved

## Tipos de datos

Tipo	Bytes	Descripción
datetime	8	Datos de fecha y hora comprendidos entre el 1 de enero de 1753 y el 31 de diciembre de 9999, con una precisión de 3,33 milisegundos.
smalldatetime	4	Datos de fecha y hora comprendidos entre el 1 de enero de 1900 y el 6 de junio de 2079, con una precisión de un minuto.
datetime2	6,7,8	Datos de fecha y hora comprendidos entre el Del 1 de enero del año 1 después de Cristo al 31 de diciembre de 9999. De 00:00:00 a 23:59:59.9999999
date	3	De 0001-01-01 a 9999-12-31
time	3 a 5	De 00:00:00.0000000 a 23:59:59.9999999
datetimeoffset	8..10	De 0001-01-01 00:00:00.0000000 a 9999-12-31 23:59:59.9999999 (en UTC)

© JMA 2016. All rights reserved

## Tipos de datos

Tipo	Bytes	Descripción
char	1	Datos de caracteres ANSI de longitud fija con una longitud máxima de 8.000 caracteres.
varchar	1...	Datos ANSI de longitud variable con un máximo de 8.000 caracteres.
<del>text</del> varchar(max)	8K...	Datos ANSI de longitud variable con una longitud máxima de 231 - 1 (2.147.483.647) caracteres.
nchar	2	Datos Unicode de longitud variable con una longitud máxima de 4.000 caracteres.
nvarchar	2...	Datos Unicode de longitud variable con una longitud máxima de 4.000 caracteres. sysname es el tipo de datos suministrado por el sistema y definido por el usuario que es funcionalmente equivalente a nvarchar(128) y que se utiliza para hacer referencia a nombres de objetos de bases de datos.
<del>ntext</del> nvarchar(max)	8K...	Datos Unicode de longitud variable con una longitud máxima de 230 - 1 (1.073.741.823) caracteres.

© JMA 2016. All rights reserved

## Tipos de datos

Tipo	Bytes	Descripción
binary	1...	Datos binarios de longitud fija con una longitud máxima de 8.000 bytes.
varbinary	1...	Datos binarios de longitud variable con una longitud máxima de 8.000 bytes.
image varbinary(max)	8K...	Datos binarios de longitud variable con una longitud máxima de 231 - 1 (2.147.483.647) bytes.
xml	8K...	Datos XML de longitud variable con una longitud máxima de 2 Gb (2.147.483.647) caracteres.
sql_variant	1...	Un tipo de datos que almacena valores de varios tipos de datos aceptados en SQL Server, excepto text, ntext, timestamp y sql_variant.
timestamp rowversion	8	Un número único para toda la base de datos que se actualiza cada vez que se actualiza una fila.
uniqueidentifier	16	Un identificador exclusivo global (GUID).

© JMA 2016. All rights reserved

## Tipos de datos

Tipo	Bytes	Descripción
hierarchyid	1..5	Dato utilizado para representar la posición en una jerarquía
geography	2...	Datos elípticos (globo), como las coordenadas de latitud y longitud del sistema GPS. Pueden contener uno o mas puntos.
geometry	2...	Datos en un sistema de coordenadas euclídeo (plano). Pueden contener uno o mas puntos.
Tipos programáticos		
cursor		Una referencia a un cursor.
table		Un tipo de datos especial que se utiliza para almacenar un conjunto de resultados para un proceso posterior.

© JMA 2016. All rights reserved

## Tipos de Datos definidos por el usuario

- Los tipos de datos definidos por el usuario están basados en un tipo integrado y permiten establecer su longitud, si acepta nulos, qué regla restringe sus valores y cuál es su valor por defecto.
- Para crear un nuevo tipo es necesario asignar:
  - Nombre del nuevo tipo
  - Tipo de datos del sistema en el que se basa el nuevo tipo de datos
  - Longitud (si el tipo de datos lo permite)
  - Aceptación de valores NULL
- Opcionalmente se le pueden asignar:
  - Regla que definen los valores permitidos
  - Valor predeterminado
- Una vez creados solo se puede modificar la regla y el valor predeterminado.

© JMA 2016. All rights reserved

## Directivas

- Indica a las utilidades de SQL Server el final de un lote de instrucciones Transact-SQL.
  - GO
- Cambia el contexto de la base de datos al de la base de datos especificada o a la instantánea en SQL Server.
  - USE { database }
- Muestra un mensaje en el área de mensajes
  - PRINT msg\_str | @local\_variable | string\_expr
- Ejecuta una cadena de comandos o una cadena de caracteres dentro de un lote de Transact-SQL
  - [[EXEC[UTE]]] { [@estadoDevuelto =] { nombreProcedimiento [;número] | @nombreProcedimientoVar } [ [ @parámetro = ] { valor | @variable [ OUTPUT ] | [ DEFAULT ] } [ ,...n ] [ WITH RECOMPILE ] }
- Fijar opciones que cambian el tratamiento de información específica por parte de la sesión actual.
  - SET opción valor

© JMA 2016. All rights reserved

## Opciones mas comunes

- **SET DATEFIRST 1**
  - Establece el primer día de la semana en un número del 1 (Lunes) al 7 (Domingo)
- **SET DATEFORMAT dmy**
  - Determina el orden de los componentes de la fecha (mes/día/año) para escribir datos de tipo datetime o smalldatetime.
- **SET LANGUAGE Español;**
  - Especifica el entorno de idioma de la sesión. El idioma de la sesión determina los formatos de datetime y los mensajes del sistema
- **SET IDENTITY\_INSERT { ON | OFF }**
  - Permite insertar valores explícitos en la columna identidad de una tabla
- **SET NOCOUNT { ON | OFF }**
  - Hace que deje de devolverse como parte de los resultados el mensaje que muestra el número de filas afectado por una instrucción Transact-SQL
- **SET NOEXEC { ON | OFF }**
  - Compila cada consulta, pero no la ejecuta
- **SET ROWCOUNT { ON | OFF }**
  - Hace que SQL Server detenga el procesamiento de la consulta una vez que se han devuelto las filas especificadas.
- **SET IMPLICIT\_TRANSACTIONS { ON | OFF }**
  - Establece el modo de transacción implícita para la conexión
- **SET TRANSACTION ISOLATION LEVEL**
  - Controla el comportamiento del bloqueo y de las versiones de fila de las instrucciones Transact-SQL emitidas por una conexión a SQL Server
- **SET XACT\_ABORT { ON | OFF }**
  - Especifica si SQL Server revierte automáticamente la transacción actual cuando una instrucción Transact-SQL genera un error en tiempo de ejecución

© JMA 2016. All rights reserved

Lenguaje de Consulta de Datos

# DQL

© JMA 2016. All rights reserved

## Lenguaje de Consulta de Datos (DQL)

- Se pueden consultar tablas y vistas
- La sentencia utilizada para la elaboración de consultas es la SELECT
- La sentencia SELECT permite realizar las siguientes operaciones:
  - Recuperar toda la información asociada a una tabla
  - Recuperar todas las columnas de una tabla
  - Recuperar sólo las columnas que se especifiquen
  - Recuperar una selección de filas
  - Controlar el orden de salida de las filas en las consultas
  - Obtener en el momento de la consulta columnas productos de operaciones con otras columnas, constantes, etc.
- Las consultas se realizan por asociación de valores, no por la localización de los mismos

© JMA 2016. All rights reserved

## Sintaxis general de la sentencia SELECT

SELECT columna, ...	el qué
FROM tabla, ...	de dónde
WHERE condiciones	bajo qué condiciones
GROUP BY valores	condición de agrupamiento
HAVING condiciones	filtra los grupos
ORDER BY columna	ordena la salida

- Solo son obligatorias las cláusulas SELECT y FROM

© JMA 2016. All rights reserved



## Sintaxis general de la sentencia SELECT

- SELECT
  - Se indican las columnas a consultar. Si se quiere consultar todas las columnas de una tabla se pone un '\*'
- FROM
  - Se indica la tabla o tablas de donde se va a realizar la consulta
- WHERE
  - Se indican las condiciones para la realización de la consulta
- GROUP BY
  - Agrupaciones por determinadas columnas
- HAVING
  - Se indican las condiciones para la realización de las agrupaciones.
- ORDER BY
  - Se indican las columnas por las cuales se va a realizar la ordenación de la consulta

© JMA 2016. All rights reserved

## Orden de interpretación

1. FROM
2. WHERE
3. GROUP BY
4. WITH CUBE o WITH ROLLUP
5. HAVING
6. SELECT
7. DISTINCT
8. ORDER BY
9. TOP

© JMA 2016. All rights reserved

## Cláusula SELECT

- SELECT sirve para seleccionar ciertas columnas, o valores derivados de ellas, y puede incluir:
  - Constantes numéricas, alfanuméricas y variables
  - Valores de columnas con el formato [nombre de tabla.] nombre de columna
  - Expresiones con nombres de columnas y constantes
  - Todo tipo de funciones aplicadas a columnas de una tabla y/o variables
  - Permite la especificación del título de la columna que aparecerá en la cabecera
- FROM sirve para definir las tablas donde se va a ir a buscar las columnas especificadas en la sentencia SELECT. Se puede asociar un nuevo nombre a las tablas para facilitar la consulta (sinónimo temporal)

© JMA 2016. All rights reserved

## Columnas

- \*
- Todas las columnas
- { table\_name | view\_name | table\_alias }.\*
- Todas las columnas de una tabla o vista
- [ { table\_name | view\_name | table\_alias }. ]column\_name
- Columna de una tabla o vista
- expresión
- Resultado de evaluar una expresión o constante.
- \$IDENTITY
- Devuelve la columna de identidad.
- \$ROWGUID
- Devuelve la columna GUID de fila.
- [ AS ] column\_alias
- Asigna un nombre alternativo a la columna de resultado

© JMA 2016. All rights reserved

## DISTINCT / ALL / TOP

- La función que tiene DISTINCT es la de eliminar las filas repetidas que se produzcan en una sentencia SELECT, por defecto es ALL: mostrar también las filas duplicadas.

```
SELECT DISTINCT COD_CAT
FROM EMP;
```

- TOP (expression ) [ PERCENT ] [ WITH TIES ]

– Indica que el conjunto de resultados de la consulta solamente devolverá un primer conjunto o porcentaje de filas especificado. expression puede ser un número o un porcentaje de las filas.

```
SELECT TOP(10) COD_CAT
FROM EMP;
```

© JMA 2016. All rights reserved

## Cláusula WHERE

- Si la cláusula WHERE está presente en una consulta, sirve para especificar qué filas de las tablas se desean obtener como resultado de la consulta.
- Para determinar las condiciones de selección de filas intervienen tres elementos:
  - Nombre de la columna
  - Operador de comparación
  - Nombre de columna, constante, lista de valores
- SINTAXIS
  - SELECT columnas
  - FROM tabla,...
  - WHERE condición (o condiciones) de selección de filas

© JMA 2016. All rights reserved

# Expresiones

- Se trata de una combinación de símbolos y operadores que Motor de base de datos de SQL Server evalúa para obtener un único valor de datos.
- Las expresiones simples pueden ser una sola constante, variable, columna o función escalar.
- Los operadores se pueden usar para combinar dos o más expresiones simples y formar una expresión compleja.
- Dos expresiones se pueden combinar mediante un operador si ambas tienen tipos de datos admitidos por el operador y se cumple al menos una de estas condiciones:
  - Las expresiones tienen el mismo tipo de datos.
  - El tipo de datos de menor prioridad se puede convertir implícitamente al tipo de datos de mayor prioridad.
- Si las expresiones no cumplen estas condiciones, se pueden usar las funciones CAST o CONVERT para convertir explícitamente el tipo de datos de menor prioridad al tipo de datos de mayor prioridad o a un tipo de datos intermedio que se puede convertir implícitamente al tipo de datos de mayor prioridad.

© JMA 2016. All rights reserved

## Operadores Expresión

- **ARITMÉTICOS**
  - Utilizables para formar expresiones con constantes, valores de columnas y funciones de valores de columnas

+	Suma	-	Resta
*	Multiplica	/	Divide
%	Módulo (Resto de la división entera)		
- **BINARIOS**
  - Aplicables a int, smallint, tinyint, binary o varbinary

&	AND bit a bit		OR bit a bit
^	OR exclusivo bit a bit	~	NOT bit a bit
- **PARA CADENAS DE CARACTERES**
  - Existe el operador de concatenación
    - cadena\_1 + cadena\_2

© JMA 2016. All rights reserved

## Operadores

- **GENERALES DE COMPARACIÓN**

=	IGUAL	>= !<	MAYOR ó IGUAL
!= <>	NO IGUAL	<	MENOR QUE
>	MAYOR QUE	<= !>	MENOR ó IGUAL

- **DE CADENAS DE CARACTERES**

- [NOT] LIKE permite los siguientes caracteres especiales en las cadenas de comparación:
  - '%' Cualquier cadena de cero o más caracteres
  - '\_' Cualquier carácter
  - '[a-z]' Coincidencia con alguno de los caracteres del intervalo
  - '[^abc]' No coincidencia con alguno de los caracteres del intervalo
- En función a la intercalación, las mayúsculas y minúsculas pueden ser significativas y las tildes se pueden ignorar.
- Usar ESCAPE para buscar un carácter especial  
... Where nombre LIKE '%AF\$\_E%' ESCAPE '\$'

© JMA 2016. All rights reserved

## Operadores Lógicos

- [NOT] BETWEEN valor\_1 AND valor\_2
  - Si tiene un valor comprendido entre los valores la fila es seleccionada.
- [NOT] IN (valor\_1,valor\_2,...,valor\_n)
  - Si la fila tiene un valor igual alguno del conjunto de valores la fila es seleccionada
- IS [NOT] NULL
  - Si la fila tiene valor null la fila es seleccionada
- comparación ANY, comparación SOME
  - Sí se cumple la comparación con alguno de los valores del conjunto
- comparación ALL
  - Sí se cumple la comparación con todos los valores del conjunto

© JMA 2016. All rights reserved

## Operadores Lógicos

- [NOT] EXISTS
  - si una subconsulta contiene cualquiera de las filas.
- AND
  - si ambas expresiones booleanas se cumplen.
- OR
  - si cualquiera de las dos expresiones booleanas se cumple.
- NOT
  - Invierte el valor de cualquier otro operador lógico.

© JMA 2016. All rights reserved

## Operador Condicional

- Evalúa una lista de condiciones y devuelve una de las expresiones de resultado posibles.
- La expresión CASE tiene dos formatos:
  - La expresión CASE genera un valor que se compara con los posibles resultados.
 

```
CASE input_expression
  WHEN value1 THEN result_expression
  WHEN value2 THEN result_expression [ ...n ]
  [ ELSE else_result_expression ]
END
```
  - Múltiples condiciones que obtiene el resultado de la primera que se cumpla.
 

```
CASE
  WHEN Boolean_expression1 THEN result_expression
  WHEN Boolean_expression2 THEN result_expression [ ...n ]
  [ ELSE else_result_expression ]
END
```
- Ambos formatos admiten un argumento ELSE opcional que suministra el resultado si no se cumple ninguna de las anteriores.

© JMA 2016. All rights reserved

## Cláusula ORDER BY

- Sirve para fijar el orden de salida de las filas seleccionadas en una sentencia SELECT
- Permite ordenar las filas utilizando los criterios siguientes:
  - Orden ascendente (ASC valor por defecto)
  - Orden descendente (DESC)
  - Por múltiples columnas (la columna más a la izquierda es por la que primero se clasifica)
  - Con valores nulos
- Debe ser la última cláusula de la sentencia SELECT
- Cuando no se usa, el orden de salida de las filas no está definido
- NOTA: Cuando se quiere ordenar por columnas calculadas se indica en la orden ORDER BY el alias de las columna en la selección.  
 Select nombre, Salario / 12 SalarioAnual  
 ....  
 Order by SalarioAnual

© JMA 2016. All rights reserved

## Ordenación Condicional

- La clausula ORDER BY puede contener un expresión según la que se ordenará el conjunto de resultados de la consulta.
- Usando el operador CASE se puede establecer una ordenación condicional que dependa de otras columnas, variables o parámetros.

```
ORDER BY CASE CountryRegionName
  WHEN 'United States' THEN TerritoryName
  ELSE CountryRegionName
END
```

© JMA 2016. All rights reserved

## Ordenación con paginación

- OFFSET-FETCH es una extensión de la cláusula ORDER BY que permite filtrar un rango de filas seleccionado de forma similar al TOP
- Proporciona un mecanismo para paginar a través de los resultados, especificando el número de filas a omitir (saltar) y el número de filas a recuperar:
 

```
OFFSET { offset_rows } { ROW | ROWS }
[ FETCH { FIRST | NEXT } { fetch_row_count } { ROW | ROWS } ONLY ]
```
- Es obligatorio indicar el valor de offset\_rows, pero puede ser cero si no se requiere saltar (primera página)
- La cláusula FETCH es opcional, si se omite se devuelven todas las filas que siguen al valor OFFSET
- Tanto OFFSET como FETCH pueden ser constantes, expresiones, variables y parámetros
- Con el enfoque del lenguaje natural al código, ROW | ROW, por un lado, y FIRST | NEXT, por otro, son intercambiables para obtener expresiones mas legibles.
 

```
OFFSET 0 ROWS FETCH FIRST 20 ROWS ONLY
OFFSET 20 ROWS FETCH NEXT 20 ROWS ONLY
```

© JMA 2016. All rights reserved

## Ejercicios

- Ejercicios propuestos:
  - Seleccionar todos los productos cuyo peso esté comprendido entre 10 y 100.
  - Obtener un listado de todos los colores diferentes que hay en los productos de estilo femenino.
  - Obtener una lista con los productos ordenada por peso y dentro de él por colores.
  - Obtener los productos rojos, azules y negros
  - Se pide una lista que recupere el peso de todos los productos cuyo número de producto empiece por 'BK'.
  - Obtener el Número de producto, Nombre y Precio de venta de los productos de Carretera y Montaña
  - Obtener los 5 productos mas caros de la línea estándar y estilo universal.
  - Obtener los productos de gama alta plateados y los de gama baja negros.

© JMA 2016. All rights reserved



# Funciones

- Escalares
  - Opera con elementos de una sola fila como entradas y devuelve un valor único (escalar) como salida
  - Se puede utilizar como una expresión en consultas.
  - Puede ser determinista o no determinista.
  - La intercalación depende del valor de entrada o la intercalación predeterminada de la base de datos
- Agregado Agrupado
  - Toma uno o más valores de una o mas filas pero devuelve un solo valor resumen
  - Requieren grupos (GROUP BY)
- Ventana
  - Opera en una ventana (conjunto) de filas pero devuelve un solo valor resumen
  - Incluye funciones de clasificación, desplazamiento, agregado y distribución.
- Conjunto de filas
  - Devuelve una tabla virtual que se pueden usar posteriormente en sustitución de tablas.

© JMA 2016. All rights reserved

# Funciones Aritméticas

- ABS(n)
  - Valor absoluto de n
- CEILING (n)
  - Entero inmediatamente superior a n (n no entero)
- FLOOR (n)
  - Entero inmediatamente inferior a n (n no entero)
- ROUND(valor[,precisión])
  - Redondea valor con la precisión indicada
- POWER(valor,exponente)
  - Eleva valor al exponente indicado
- SQRT (n)
  - Raíz cuadrada de n, si n < 0 NULL
- SIGN (n)
  - Si n < 0, -1; Si n = 0, 0; Si n > 0, 1

© JMA 2016. All rights reserved

## Funciones de Cadena de Caracteres

- ASCII(s)
  - Valor ASCII del primer carácter de la cadena "n"
- CHR(n)
  - Devuelve el carácter cuyo valor ASCII es el numero "n"
- LOWER(s)
  - Cadena "s" con todas sus letras convertidas en minúsculas
- UPPER (s)
  - Cadena "s" con todas sus letras en mayúsculas
- LEN(s)
  - El numero de caracteres de "s"
- SPACE (n)
  - Devuelve una cadena de n espacios repetidos.

© JMA 2016. All rights reserved

## Funciones de Cadena de Caracteres

- REPLICATE (s, n)
  - Devuelve s repetida n veces.
- RTRIM (s)
  - Suprime los blancos que contenga la cadena "s" a la derecha
- LTRIM (s)
  - Suprime los blancos que contenga la cadena "s" a la izquierda
- TRIM (s)
  - Suprime los blancos de la cadena "s" al principio y al final
- LEFT (s , n)
  - Devuelve los n primeros caracteres de la cadena
- RIGHT (s , n)
  - Devuelve los n últimos caracteres de la cadena

© JMA 2016. All rights reserved

## Funciones de Cadena de Caracteres

- SUBSTRING ( s, m [,n])
  - Devuelve la subcadena de "s" que empieza con el carácter número "m" y tiene "n" caracteres de longitud.
- CHARINDEX ( s , buscar [ , start\_location ] )
  - Busca una cadena dentro de otra cadena y devuelve la posición inicial de la primera coincidencia.
- PATINDEX ( '%pattern%', expression )
  - Devuelve la posición inicial de la primera repetición de un patrón en la expresión especificada o cero si el patrón no se encuentra
- REPLACE ( s , buscada, sustituto )
  - Reemplaza todas las instancias de subcadena por otra subcadena.
- TRANSLATE (cad1, cad2, cad3)
  - Cambia los caracteres de la cadena cad1 por los de la cadena cad3 en función de los caracteres de la cadena cad2

© JMA 2016. All rights reserved

## Funciones de Conversión

- STR ( float\_expression [ , length [ , decimal ] ] )
  - Convierte una valor numérico en cadena, se puede indicar número total de dígito y cuantos son decimales
- CAST ( expression AS data\_type [ ( length ) ] )
  - Convierte una expresión de un tipo de datos a otro
- CONVERT ( data\_type [ ( length ) ] , expression [ , style ] )
  - Convierte una expresión de un tipo de datos a otro aplicándole formato
- PARSE ( string\_value AS data\_type [ USING culture ] )
  - Convierte una cadena a tipos de fecha y hora y de número siguiendo la referencia cultural
- FORMAT ( value, format [ , culture ] )
  - Devuelve un valor con el formato y la cultura indicada

© JMA 2016. All rights reserved

## Funciones de Fechas

- Funciones de fecha y hora del sistema  
GETDATE()
- Funciones que devuelven partes de fecha y hora  
DATENAME(datepart, date), DATEPART(datepart, date), DAY(date),  
MONTH(date), YEAR(date)
- Funciones que devuelven fecha y hora a partir de sus partes  
DATETIMEFROMPARTS(year, month, day, hour, minute, seconds,  
milliseconds), DATEFROMPARTS(year, month, day),  
TIMEFROMPARTS(hour, minute, seconds, fractions, precision)
- Funciones de calculo con fecha y hora  
DATEADD(datepart, number, date), DATEDIFF(datepart, startdate,  
enddate), EOMONTH (start\_date [, month\_to\_add ] )

© JMA 2016. All rights reserved

## Funciones Lógicas

- COALESCE(exp1, exp2, exp3,...)  
– Devuelve el valor de la primera expresión que sea distinta de NULL
- CHOOSE ( index, val\_1, val\_2 [, val\_n ] )  
– Devuelve el elemento en el índice especificado
- IIF ( boolean\_expression, true\_value, false\_value )  
– Devuelve uno de dos valores, dependiendo de la condición
- NULLIF(exp1, exp2)  
– Devuelve NULL si exp1=exp2 sino devuelve exp1
- ISNULL ( check\_expression , replacement\_value )  
– Sustituye el valor NULL por el valor especificado.
- ISNUMERIC ( expression )  
– Determina si una expresión es un tipo numérico válido.

© JMA 2016. All rights reserved

## Ejercicios

- Ejercicios propuestos:
  - Calcular el porcentaje de beneficio  $[(P. \text{ Venta} - P. \text{ Coste}) / P. \text{ Venta}]$  de los productos.
  - Calcular la edad de los empleados
  - Calcular la nueva lista de precios donde los artículos de estilo masculino suban un 20% y los femeninos un 10%, el nombre del producto se debe mostrar en mayúsculas.
  - Calcular un nuevo código de producto con el formato (utilizar la X como carácter sustitutivo de los valores desconocidos):
    - X: Línea de producto
    - X: Gama
    - X: Estilo
    - XXX: 3 primeras letras del nombre en mayúsculas
    - X: Inicial del color en mayúsculas
    - 999999: 6 dígitos con el identificador de producto

© JMA 2016. All rights reserved

DQL: Lenguaje de Consulta de Datos

## AGRUPAMIENTO

© JMA 2016. All rights reserved

## Cláusula GROUP BY

- Sirve para crear grupos de filas (resúmenes) y muestra una sólo fila por cada grupo de la SELECT
- Debería aparecer el mismo número de columnas en la cláusula GROUP BY que en la sentencia SELECT sin tener en cuenta las columnas de funciones agregadas.
  - SELECT A, B, AVG(SAL)
  - .....
  - GROUP BY A,B
- RESTRICCIONES
  - No se puede agrupar por columnas calculadas.
  - El NULL lo toma como si fuese un valor mas.
  - No se puede introducir dentro de una SUBSELECT.
  - Las columnas del ORDER BY debería coincidir con las del GROUP BY.

© JMA 2016. All rights reserved

## CLÁUSULA HAVING

- La cláusula HAVING, similar al WHERE, se emplea para FILTRAR los conjuntos agrupados por la cláusula GROUP BY que se visualizan
- La evaluación de las cláusulas de la sentencia SELECT en tiempo de ejecución se efectúa en el siguiente orden:
  1. FROM
  2. WHERE
  3. GROUP BY
  4. HAVING
  5. SELECT
  6. DISTINCT
  7. ORDER BY
  8. TOP

© JMA 2016. All rights reserved

## Funciones de agrupamiento

- AVG ( [ ALL | DISTINCT ] expression )
  - Valor medio de expression (ignorando los valores nulos)
  - Sólo columnas numéricas.
- SUM ( [ ALL | DISTINCT ] expression )
  - Suma todos los valores o solo de los valores DISTINCT de la expresión (ignorando los valores nulos)
  - Sólo columnas numéricas.
- MAX (expression)
  - Máximo valor de expression
- MIN (expression)
  - Mínimo valor de expression
- COUNT ( { [ ALL | DISTINCT ] expression } | \* }, COUNT\_BIG(expr)
  - Número de veces que "expression" evalúa dato con valor no nulo.
- APPROX\_COUNT\_DISTINCT(expression)
  - Número aproximado de valores no nulos únicos de un grupo.

© JMA 2016. All rights reserved

## Funciones de agrupamiento

- STDEV ( [ ALL | DISTINCT ] expression )
  - Desviación típica estadística de todos los valores de la expresión especificada.
- STDEVP ( [ ALL | DISTINCT ] expression )
  - Desviación estadística estándar para la población de todos los valores de la expresión especificada.
- VAR ( [ ALL | DISTINCT ] expression )
  - Varianza estadística de todos los valores de la expresión especificada.
- VARP ( [ ALL | DISTINCT ] expression )
  - Varianza estadística de la población para todos los valores de la expresión especificada.

© JMA 2016. All rights reserved

## Particularidades de las funciones de agrupamiento

- Los valores nulos no participan en el cálculo de las funciones de conjuntos
  - `SELECT AVG(SALARIO)` y `SELECT SUM(SALARIO) / COUNT(*)` no siempre tendrán el mismo resultado
- No se pueden combinar resultados agrupados con resultados no agrupados salvo con la cláusula `OVER`.
  - `<AGREGADO> OVER ( [ PARTITION BY value, ... [ n ] ] )`
- En la cláusula `SELECT` sólo pueden aparecer funciones de conjuntos que afecten a todas las filas si la consulta no esta agrupada, una única fila para la consulta.
- En la cláusula `SELECT` sólo pueden aparecer funciones de conjuntos y las columnas incluidas en la cláusula `GROUP BY` si la consulta esta agrupada, una fila por grupo.

© JMA 2016. All rights reserved

## Ejercicios

- Ejercicios propuestos:
  - Calcular el precio medio de los pedidos
  - Calcular el total de las ventas trimestrales, el numero de pedidos, el pedido máximo y el mínimo
  - Mostrar el total de las ventas mensuales de los pedidos enviados de los meses que hayan superado el millar de pedidos
  - Clasificar a los vendedores por el volumen de ventas
  - Resumir los productos por subcategorías indicando en cada fila cuantos productos hay de cada una de las gamas.

© JMA 2016. All rights reserved



## Ventanas

- La cláusula OVER define una ventana o un conjunto de filas definido por el usuario en un conjunto de resultados de la consulta.
- Determina las particiones (agrupamiento) y el orden de un conjunto de filas antes de que se aplique la función de ventana asociada.
- Una función de ventana calcula por fila de resultados un valor único para cada ventana.  

```
<FUNC> () OVER (
    [ <PARTITION BY clause> ]
    [ <ORDER BY clause> ]
)
```
- Puede utilizar la cláusula OVER con funciones de agregado, estadísticas, categoría y analíticas.
- El uso de ventanas evita la necesidad de combinaciones con consultas de resumen.

© JMA 2016. All rights reserved

## Funciones de categoría

- RANK()
  - Devuelve la clasificación de cada fila en la partición de un conjunto de resultados
- DENSE\_RANK ()
  - Devuelve la clasificación de filas dentro de la partición de un conjunto de resultados, sin ningún espacio en los valores repetidos.
- NTILE ()
  - Distribuye las filas de una partición ordenada en un número especificado de grupos
- ROW\_NUMBER ()
  - Devuelve el número secuencial de una fila de una partición de un conjunto de resultados, comenzando con 1 para la primera fila de cada partición

© JMA 2016. All rights reserved

## Ejercicios

- Ejercicios propuestos:
  - Calcular la diferencia del precio de venta del producto respecto a la media de su subcategoría, el mas caro y el mas barato.
  - Crear un ranking de los productos por su precio de coste.
  - Clasificar los productos por su precio de venta: Caros, Normales, Baratos.

© JMA 2016. All rights reserved

## GROUP BY ROLLUP

- Crea un grupo para cada combinación de expresiones de columna. Además, "acumula" los resultados en subtotales y totales generales.
- Para ello, se mueve de derecha a izquierda reduciendo el número de expresiones de columna para las que crea grupos y agregaciones.
- El orden de las columnas influye en la salida de ROLLUP y también puede afectar al número de filas del conjunto de resultados.
 

```
SELECT Country, Region, Product, SUM(Sales) AS TotalSales
FROM Sales
GROUP BY ROLLUP (Country, Region, Product);
```
- En el ejemplo anterior crea filas adicionales grupos para cada combinación de expresiones de columna en las listas siguientes.
  - col1, col2, col3, rslt (filas normales)
  - col1, col2, NULL, rslt (subtotal por región del país, una fila por país+región)
  - col1, NULL, NULL, rslt (este es el subtotal por país, una fila por país)
  - NULL, NULL, NULL, rslt (este es el total general, una sola fila)

© JMA 2016. All rights reserved

## GROUP BY CUBE

- GROUP BY CUBE es similar a GROUP BY ROLLUP pero crea subtotales para todas las combinaciones posibles de columnas.
- El orden de las columnas influye en la salida de ROLLUP y también puede afectar al número de filas del conjunto de resultados.  

```
SELECT Country, Region, Product, SUM(Sales) AS TotalSales
FROM Sales
GROUP BY CUBE (Country, Region, Product)
```
- En el ejemplo anterior crea las siguientes filas adicionales:
  - col1, col2, col3, rslt (filas normales)
  - NULL, col2, col3, rslt (subtotal por producto en una región)
  - NULL, NULL, col3, rslt (subtotal por producto)
  - col1, NULL, col3, rslt (subtotal por producto en un país)
  - col1, col2, NULL, rslt (subtotal por región del país)
  - NULL, col2, NULL, rslt (subtotal por región)
  - col1, NULL, NULL, rslt (este es el subtotal por país)
  - NULL, NULL, NULL, rslt (este es el total general, una sola fila)

© JMA 2016. All rights reserved

## GROUP BY GROUPING SETS

- La opción GROUPING SETS permite combinar varias cláusulas GROUP BY en una cláusula GROUP BY.
- Los resultados son equivalentes a usar la instrucción UNION ALL en los grupos especificados.  

```
SELECT Country, Region, Product, SUM(Sales) AS TotalSales
FROM Sales
GROUP BY GROUPING SETS (Country)
```
- En el ejemplo anterior solo añade los subtotales por país.
- Con el grupo vacío ( ) se especifica que se genere el total general.  

```
SELECT Country, Region, Product, SUM(Sales) AS TotalSales
FROM Sales
GROUP BY GROUPING SETS (Country, ( ))
```
- Ahora se añade los subtotales por país y el total general.

© JMA 2016. All rights reserved

## GROUPING y GROUPING\_ID

- Dado que las filas de totales muestran a NULL las columnas totalizadas, si la columna a totalizar puede contener nulos será complicado decidir, en las filas de resultados, si el NULL es de la columna o de la totalización.
- La función GROUPING indica con un 1 si una expresión de columna especificada en una lista GROUP BY es agregada o no con un 0.  

```
SELECT Country, Region, Product, GROUPING(Product), SUM(Sales) AS TotalSales
FROM Sales
GROUP BY ROLLUP (Country, Region, Product);
```
- En el ejemplo anterior crea filas adicionales grupos para cada combinación de expresiones de columna en las listas siguientes.
  - col1, col2, NULL, 0, rslt (producto con NULL)
  - col1, col2, NULL, 1, rslt (subtotal por región del país)
- La función GROUPING\_ID calcula el nivel de agrupación y devuelve una máscara de bits indicando las columnas que participan en la totalización (con 1 el bit de cada columna NULL por totalización) o un 0 si no es una fila de totalización.
- Para GROUPING\_ID(Country, Region, Product)
 

0 → 000 → col1, col2, col3, rslt (filas normales)	4 → 100 → NULL, col2, col3, rslt (subtotal por producto en una región)
6 → 110 → NULL, NULL, col3, rslt (subtotal por producto)	2 → 010 → col1, NULL, col3, rslt (subtotal por producto en un país)
1 → 001 → col1, col2, NULL, rslt (subtotal por región del país)	5 → 101 → NULL, col2, NULL, rslt (subtotal por región)
3 → 011 → col1, NULL, NULL, rslt (este es el subtotal por país)	7 → 111 → NULL, NULL, NULL, rslt (este es el total general, una sola fila)

© JMA 2016. All rights reserved

## Ejercicios

- Ejercicios propuestos:
  - Crear un informe trimestral de las ventas clasificadas por zona y vendedor.
  - Totalizar para cada uno de los grupos
  - Mostrar una columna con los literales apropiado para cada uno de los totales.

© JMA 2016. All rights reserved

---

DQL: Lenguaje de Consulta de Datos

## CONSULTAS MULTITABLA

---

© JMA 2016. All rights reserved

### Múltiples Tablas

---

- **CONJUNTOS (UNION, INTERSECT, EXCEPT)**
    - Son utilizado para combinar los resultados de dos o más consultas en un solo conjunto de resultados.
  - **COMBINACIONES (JOINS)**
    - Son utilizadas para recuperar datos de varias tablas mediante el producto cartesiano
  - **SUBCONSULTAS (SUB-QUERIES)**
    - Son utilizadas para generar preguntas subsidiarias necesitadas por la consulta principal
- 

© JMA 2016. All rights reserved

# Conjuntos

- Existen tres tipos de relaciones
  - UNION
  - INTERSECT
  - EXCEPT
- UNION [ALL]
  - Combinación de todas las filas del primer conjunto con todas las filas del segundo. Con ALL salen las duplicadas.
- INTERSECT
  - Combinación de las filas de los dos conjuntos, cuando la fila exista en ambos conjuntos
- EXCEPT
  - Combinación de las filas del primer conjunto que no estén en el segundo

© JMA 2016. All rights reserved

## Reglas para el uso de conjuntos

- Pueden ser encadenados en cualquier combinación  
Select ..... union select ..... intersect....
- Los conjuntos son evaluados de izquierda a derecha
- No existe jerarquía de precedencia en el uso de estos operadores, pero puede ser forzada mediante paréntesis
- Los operadores de conjuntos pueden emplearse con conjuntos de diferentes tablas siempre que se apliquen las reglas siguientes:
  - Las columnas son relacionadas en orden, de izquierda a derecha
  - Los nombres de las columnas son irrelevantes
  - Los tipos de datos deben coincidir
  - Los nombres de las columnas del primer conjunto se utilizaran en el conjunto resultante

© JMA 2016. All rights reserved


## Cláusula FROM

- La cláusula FROM determina las tablas de origen que se utilizarán en la instrucción SELECT
- La cláusula FROM puede contener tablas y operadores
- El conjunto de resultados de la cláusula FROM es una tabla virtual
- Las operaciones lógicas posteriores en la instrucción SELECT consumen esta tabla virtual
- La cláusula FROM puede establecer alias de tablas para su uso en fases posteriores de la consulta
- En la cláusula FROM pueden participar tablas, vistas, tablas derivadas, funciones de tipo tabla virtual, funciones rowset, pivoted/unpivoted y variables de tipo tabla
- Salvo las tablas y vistas, el resto requiere asociarles un alias  
FROM T [AS] Alias

© JMA 2016. All rights reserved

## Características de un producto cartesiano

- Se realizan por pares de conjuntos y el resultado con el siguiente conjunto.
- Combinación de todas las filas y columnas de un conjunto con todas las filas y columnas del otro conjunto.
- El resultado tiene tantas filas como el producto del número de filas de todos los participantes y tantas columnas como el sumatorio del número de columnas de todos los participantes.
- Una vez obtenido el conjunto resultante se eliminan las filas que no cumplan la cláusula WHERE y se eliminan las columnas que no aparezcan en la cláusula SELECT

Id	Name		Product
1	Davis		Alice Mutton
2	Funk		Crab Meat
3	King		Ipoh Coffee

Id	Name	Product
1	Davis	Alice Mutton
1	Davis	Crab Meat
1	Davis	Ipoh Coffee
2	Funk	Alice Mutton
2	Funk	Crab Meat
2	Funk	Ipoh Coffee
3	King	Alice Mutton
3	King	Crab Meat
3	King	Ipoh Coffee

© JMA 2016. All rights reserved

## Reglas para las combinaciones (JOINS)

- Se pueden obtener la combinación de tantas tablas como se quiera
- Las combinaciones se evalúan 2 a 2 de izquierda a derecha.
- Puede usarse más de una pareja de columnas para especificar una condición de combinación entre dos tablas
- Se pueden seleccionar columnas de todas las tablas implicadas en la combinación
- Si tenemos columnas con idéntico nombre en dos tablas de la combinación se deberán calificar las columnas con el nombre de la tabla al que pertenecen
- Como regla general, la combinación tendrá como mínimo tantas condiciones en la cláusula WHERE como número de tablas de la cláusula FROM menos uno

© JMA 2016. All rights reserved

## Tipos de sintaxis de Combinaciones

- ANSI SQL-92
  - Las tablas se combinan y establecen las combinaciones con el operador JOIN en la cláusula FROM
  - Separación lógica entre el filtrado con el fin de combinar y el filtrado de los resultados en WHERE
  - Es la sintaxis recomendada.
 

```
SELECT ...
FROM Table1 JOIN Table2
ON <on_predicate>
```
- ANSI SQL-89
  - Las tablas se combinan en la cláusula FROM separándolas por comas y se filtran en la cláusula WHERE
  - Sintaxis no recomendada al complicar mucho el WHERE.
 

```
SELECT ...
FROM Table1, Table2
WHERE <where_predicate>
```

© JMA 2016. All rights reserved



## Tipos de Combinaciones

- **CROSS**
  - Combina todas las filas en ambas tablas (crea un producto cartesiano)  
FROM t1 CROSS JOIN t2
- **INNER**
  - Se inicia con producto cartesiano; aplica el filtro para hacer coincidir las filas entre tablas basadas en el predicado  
FROM t1 [INNER] JOIN t2 ON t1.column = t2.column [AND ...]
- **OUTER**
  - Se inicia con producto cartesiano; se conservan todas las filas de la tabla designada mostrando las coincidencias con las filas de otra tabla recuperada y nulos como marcadores de posición cuando no hay coincidencias  
FROM t1 { LEFT | RIGHT | FULL } [ OUTER ] JOIN t2 ON t1.column = t2.column [AND ...]
- **SELF**
  - Para las relaciones reflexivas, el producto cartesiano se realiza con la propia tabla, las tablas requieren un alias  
FROM t [AS] t1 [INNER] JOIN t [AS] t2 ON t1.columnPK = t2.columnFK [AND ...]

© JMA 2016. All rights reserved

## Ejercicios

- **Ejercicios propuestos:**
  - Completar los ejercicios anteriores sustituyendo los identificadores migrados (FK) por información significativa de la entidad (nombre, nombre y apellidos, ...)
  - Crear un listado completo de pedidos sustituyendo los identificadores por sus descripciones.
  - Elaborar un listado de categorías, sub categorías y productos.
  - Comprobar que los totales de los pedidos son correctos usando las líneas como referencias.
  - Listar los productos que ha adquirido cada uno de los clientes.

© JMA 2016. All rights reserved

## Tablas Derivadas

- Se denomina tabla derivada a una consulta SELECT que participa en la cláusula FROM
- Deben ir entre paréntesis y contar con un alias, tener nombres o alias únicos para todas las columnas y no utilizar la cláusula ORDER BY.  
FROM (SELECT ... ) [AS] Alias
- Se pueden anidar tablas derivadas en la definición de otras tablas derivadas.
- Se puede usar la característica de constructor con valores de tabla de Transact-SQL para especificar varias filas.

```
SELECT * FROM (VALUES (1, 'UNO'), (2, 'DOS'), (3, 'TRES')) AS
MyTable(Id, Name);
```

© JMA 2016. All rights reserved

## CTE

- Especifica un conjunto de resultados temporal con nombre, conocido como expresión de tabla común (CTE).
- Se deriva de una consulta simple y se define en el ámbito de ejecución de una sola instrucción SELECT, INSERT, UPDATE, DELETE o MERGE.
- Una expresión de tabla común puede incluir referencias a ella misma (conocida como expresión de tabla común recursiva) y acepta múltiples referencias.

```
WITH Alias_CTE(column_name [ ,...n ] ) AS (
    SELECT ...
)
SELECT ...
FROM ... Alias_CTE
...
```

© JMA 2016. All rights reserved

# Funciones

- Funciones tabla virtual (TVF)
  - Funciones que devuelven un conjunto de resultados en formato tabla  
`CREATE FUNCTION MyFunc( ... )`  
`RETURNS TABLE ...`  
`FROM MyFunc(...) [AS] f`
- Funciones rowset
  - OPENQUERY
    - Ejecuta la consulta de paso a través especificada en el servidor vinculado especificado. (OLEDB)
    - OPENQUERY ( linked\_server , 'query' )
  - OPENDATASOURCE
    - Proporciona información de conexión ad hoc como parte de un nombre de objeto de cuatro partes sin utilizar un nombre de servidor vinculado.
    - OPENDATASOURCE ( provider\_name, init\_string )
  - OPENROWSET
    - Contiene toda la información de conexión necesaria para tener acceso a datos remotos desde un origen de datos OLE DB.
    - OPENROWSET('provider\_name' , 'datasource' ; 'user\_id' ; 'password', 'query' )
  - OPENXML
    - Proporciona una vista de un conjunto de filas en un documento XML.
    - OPENXML( idoc int [ in ] , rowpattern nvarchar [ in ] , [ flags byte [ in ] ] )
  - FROM OPENQUERY(...) [AS] rs

© JMA 2016. All rights reserved

# APPLY

- El operador APPLY permite combinar un conjunto con el resultado de la ejecución de una función de tipo tabla que se ejecuta para cada fila del conjunto tomando como argumento una columna de dicha fila.  
`source { CROSS | OUTER } APPLY`  
`table_function(source.colum[, ...]) [AS] alias`
- Si se especifica CROSS, no se genera ninguna fila cuando la función devuelve un conjunto de resultados vacío.
- Si se especifica OUTER, se genera una fila para cada fila de source, incluso si la función devuelve un conjunto de resultados vacío.

© JMA 2016. All rights reserved

## PIVOT

- La cláusula PIVOT permite girar las filas en columnas en la salida de una consulta y aplicar una función de agregación
- Permite la generación de resultados en formato de tablas cruzadas.

FROM (<SELECT query that produces the data>)

AS <alias for the source query>

PIVOT (

<aggregation function>(<column being aggregated>)

FOR [<column that contains the values that will become column headers>]

IN ( [first pivoted column], [second pivoted column], ... [last pivoted column])

) AS <alias for the pivot table>

© JMA 2016. All rights reserved

## PIVOT

```
SELECT dept, sexo, sum(salar)
```

```
FROM templa
```

```
GROUP BY dept, sexo
```

```
ORDER BY dept, sexo
```

```
SELECT *
```

```
FROM (
```

```
    SELECT dept, sexo, salar FROM templa
```

```
) origen PIVOT (
```

```
    sum(salar)
```

```
    FOR sexo
```

```
    IN ('H','M')
```

```
) cruzada
```

```
ORDER BY dept
```

DEP	S	SUM<SALAR>
A00	H	7575000
A00	M	5275000
B01	H	4125000
C01	M	9047000
D11	H	11642000
D11	M	8343000
D11		3225000
D21	H	7012000
D21	M	8080000
E01	H	4017500
E11	H	3155600
DEP	S	SUM<SALAR>
E11	M	7118000
E21	H	9531000

DEP	'H'	'M'
A00	7575000	5275000
B01	4125000	
C01		9047000
D11	11642000	8343000
D21	7012000	8080000
E01	4017500	
E11	3155600	7118000
E21	9531000	

© JMA 2016. All rights reserved

# UNPIVOT

- También tiene una cláusula UNPIVOT que gira columnas en filas en el resultado de una consulta.
  - UNPIVOT es el inverso de PIVOT, pero sin la posibilidad de desagregar los datos agregados.
  - La cláusula UNPIVOT permite separar una fila con columnas en tantas filas de una sola columna como columnas se indique, fundamentalmente para poder realizar operaciones de agregaciones con varias columnas
- ```
FROM table_source UNPIVOT(
    value_column FOR pivot_column IN ( column_name [,...n ] )
) [ [ AS ] table_alias ]
```
- Donde value\_column es el nombre de la columna de salida que contendrá los valores de las columnas y pivot\_column es el nombre de la columna de salida que contendrá los nombres de las columnas origen de los valores

© JMA 2016. All rights reserved

# UNPIVOT

```
SELECT *
FROM ParidadSalarial

SELECT *
FROM ParidadSalarial
UNPIVOT(
    sumas
    FOR sexo
    IN (HOMBRES, MUJERES)
)
```

| DEP | HOMBRES  | MUJERES |
|-----|----------|---------|
| A00 | 7575000  | 5275000 |
| B01 | 4125000  |         |
| C01 |          | 9047000 |
| D11 | 11642000 | 8343000 |
| D21 | 7012000  | 8080000 |
| E01 | 4017500  |         |
| E11 | 3155600  | 7118000 |
| E21 | 9531000  |         |

| DEP | SEXO    | SUMAS    |
|-----|---------|----------|
| A00 | HOMBRES | 7575000  |
| A00 | MUJERES | 5275000  |
| B01 | HOMBRES | 4125000  |
| C01 | MUJERES | 9047000  |
| D11 | HOMBRES | 11642000 |
| D11 | MUJERES | 8343000  |
| D21 | HOMBRES | 7012000  |
| D21 | MUJERES | 8080000  |
| E01 | HOMBRES | 4017500  |
| E11 | HOMBRES | 3155600  |
| E11 | MUJERES | 7118000  |

© JMA 2016. All rights reserved

## Ejercicios

- Ejercicios propuestos:
  - Informe trimestral de ventas con una columna para cada trimestre.

© JMA 2016. All rights reserved

## Subconsultas

- En muchos casos para filtrar una consulta es necesario realizar una consulta para obtener los datos de filtrado.
- Las subconsultas son consultas anidadas: consultas dentro de consultas.
- Los resultados de la consulta interna se pasan a la consulta externa.
- La consulta interna actúa como una expresión desde la perspectiva de la consulta externa.
- Las subconsultas pueden ser escalares o multivaluadas.
- Las subconsultas pueden ser autónomas o correlacionadas
  - Las subconsultas autónomas no tienen dependencia en la consulta externa
  - Las subconsultas correlacionadas dependen de los valores de la consulta externa

© JMA 2016. All rights reserved

## Subconsultas escalares

- La subconsulta escalar devuelve un solo valor a la consulta externa
- Se puede utilizar en cualquier lugar donde se use una expresión de un solo valor: SELECT, WHERE, etc.
- Si la consulta interna devuelve un conjunto vacío, el resultado se convierte a NULL  

```
SELECT productid, name, unitprice
FROM Sales.Products
WHERE unitprice > (
    SELECT MAX(unitprice)
    FROM Sales.Products)
```
- La construcción de la consulta externa determina si la consulta interna debe devolver un solo valor

© JMA 2016. All rights reserved

## Subconsultas multivaluadas

- La subconsulta multivalor devuelve múltiples valores como una única columna establecida en la consulta externa
- Los predicados deben utilizar operadores que soporten múltiples valores:
  - IN
  - { = | != | <> | > | >= | < | <= } { ANY | SOME | ALL }

```
SELECT productid, name, unitprice
FROM Sales.Products
WHERE productid in (
    SELECT productid
    FROM Sales.OrderDetails
    WHERE qty > 10)
```
- También se pueden expresar como una combinación.

© JMA 2016. All rights reserved

## Subconsultas correlacionadas

- Una subconsulta correlacionada se filtra con elementos de las tablas usadas en la consulta externa.
- Dependen de la consulta externa, no se puede ejecutar por separado
- Son más difíciles de probar que las subconsultas autocontenidas.
- Pueden ser, a su vez, escalares o multivaluadas.  

```
SELECT productid, name,
       (SELECT Name FROM Category WHERE id = categoryID) category
FROM Sales.Products p
WHERE EXISTS (
  SELECT *
  FROM Sales.OrderDetails od
  WHERE od.Productid = p.productid)
```
- Se comporta como si la consulta interna se ejecutara una vez por cada fila externa.
- También se pueden expresar como una combinación.

© JMA 2016. All rights reserved

## Ejercicios

- Ejercicios propuestos:
  - Listar los productos de gama alta cuyo precio sea menor que alguno de los productos de gama baja
  - Listar todos los clientes que hayan comprado productos blancos
  - Listar todas la mujeres de los departamentos de producción y ventas
  - Calcular el salario anual de los empleados
  - Calcular la brecha salarial entre hombre y mujeres, casados y solteros

© JMA 2016. All rights reserved



---

## BÚSQUEDA DE TEXTO COMPLETO

---

© JMA 2016. All rights reserved

### Búsqueda de texto completo

---

- Proporciona una forma de realizar consultas en lenguaje natural al buscar, en columnas con datos basados en caracteres, valores que coincidan con el significado en lugar de con las palabras exactas o frases, a una cierta distancia las unas de las otras y coincidencias ponderadas.
  - Para poder ejecutar consultas búsqueda de texto completo debe crear primero un catálogo de texto completo y un índice de texto completo en las vistas indexadas o tablas en las que quiere realizar una búsqueda.
  - Para hacer las consultas de búsquedas de texto completo:
    - WHERE: CONTAINS y FREETEXT
    - FROM: CONTAINSTABLE y FREETEXTTABLE (KEY y RANK)
- 

© JMA 2016. All rights reserved

## FREETEXT

- Es un predicado que se usa en la cláusula WHERE de una instrucción SELECT para realizar una búsqueda de texto completo simple en las columnas indexadas que contienen tipos de datos basados en caracteres. Este predicado busca valores que coincidan con el significado y no solo con la redacción exacta de las palabras en la condición de búsqueda. Cuando se usa FREETEXT, el motor de consultas de texto completo realiza internamente las siguientes acciones en cadena `freetext`, asigna a cada uno de los términos una ponderación y busca las coincidencias:
    - Separa la cadena en palabras individuales basándose en límites de palabras (separación de palabras).
    - Genera formas no flexionadas de las palabras (lematización).
    - Identifica una lista de expansiones o reemplazos de los términos basándose en coincidencias en el diccionario de sinónimos.
- ```
FREETEXT ( { column_name | (column_list) | * } , 'freetext_string' [ ,
LANGUAGE language_term ] )
```

© JMA 2016. All rights reserved

## CONTAINS

- Busca coincidencias precisas o aproximadas (menos precisas) de palabras o frases, palabras que se encuentran a cierta distancia de otra o coincidencias ponderadas.
  - CONTAINS es un predicado que se usa en la cláusula WHERE de una instrucción SELECT para realizar una búsqueda de texto completo en las columnas indexadas que contienen tipos de datos basados en caracteres.
  - CONTAINS puede buscar:
    - Una palabra o una frase, o una combinación de ellas que estén presentes o se excluyan.
    - El prefijo de una palabra o una frase.
    - Una palabra cerca de otra palabra o no.
    - Una palabra que sea una inflexión de otra (por ejemplo, las palabras controles, controladores, controlando y controlado son inflexiones de control).
    - Una palabra que sea un sinónimo de otra mediante un diccionario de sinónimos.
- ```
CONTAINS (
{
column_name | ( column_list ) | * | PROPERTY ( { column_name }, 'property_name' )
}
, '<contains_search_condition>'
[ , LANGUAGE language_term ]
)
```

© JMA 2016. All rights reserved

## FREETEXTTABLE y CONTAINSTABLE

- Las funciones FREETEXTTABLE y CONTAINSTABLE son las versiones tabla de resultados de las correspondientes funciones escalares FREETEXT y CONTAINS.
- Devuelven una tabla de cero, uno o más filas para las columnas que contienen valores seleccionados por la función escalar.
- Cada fila solo cuenta con dos columnas:
  - KEY: clave primaria (PK) de la fila que contiene la columna o columnas que satisfacen la condición de búsqueda de texto completo para realizar el JOIN.
  - RANK: un valor (de 0 a 1000) de clasificación por relevancia, solo indican un orden relativo de relevancia de las filas en el conjunto de resultados, cuanto mas alto mayor relevancia. Permite las ordenaciones.
- Con top\_n\_by\_rank se limita el número de resultados:

```
FREETEXTTABLE (table , { column_name | (column_list) | * }, 'freetext_string'
[ , LANGUAGE language_term ]
[ , top_n_by_rank ] )
```

```
CONTAINSTABLE ( table , { column_name | ( column_list ) | * }, '<contains_search_condition>'
[ , LANGUAGE language_term ]
[ , top_n_by_rank ]
)
```

© JMA 2016. All rights reserved

Lenguaje de Manipulación de Datos

## DML

© JMA 2016. All rights reserved  
Tabla5.sql

# Lenguaje de Manipulación de Datos (DML)

- Cuenta con las instrucciones para añadir, modificar y borrar las filas de las tablas.
- Solo pueden afectar a una única tabla, salvo las automatizaciones de las integridad referencial.
- Pueden usarse con vistas para manipular los datos de las tablas subyacentes bajo ciertas condiciones:
  - La modificación afecte sólo a una de las tablas subyacentes.
  - En la inserción, deben aparecer todas las columnas requeridas que no dispongan de valor predeterminado.
  - Las columnas a modificar en la vista deben hacer referencia directa a los datos subyacentes de las columnas de la tabla (no puede ser resultado de una función de agregado o cálculo).
  - Las columnas que se modifican no pueden verse afectadas por cláusulas GROUP BY, HAVING o DISTINCT.
  - La cláusula WITH CHECK OPTION exige que todas las instrucciones de modificación de datos ejecutadas en la vista se ajusten a los criterios especificados en la instrucción SELECT que define la vista.

© JMA 2016. All rights reserved

## INSERT

- Agrega una o varias filas a una tabla o una vista.  

```
INSERT table_or_view_name [ ( column_list ) ]  
VALUES ( { DEFAULT | NULL | expression } [ ,...n ] ) [ ,...n ]
```
- Las columnas se identifican por su nombre.
- La asociación de columna y su valor es posicional.
- La lista de columnas es opcional si los valores se encuentran en el mismo orden que las columnas de la tabla.
- Los valores deben cumplir con las restricciones de la columna.
- Los valores constantes de tipo carácter o fecha deben ir encerrados entre comillas simples ( ' ' ).
- DEFAULT representa el valor por defecto de la columna y NULL la ausencia de valor.
- Deben aparecer todas las columnas requeridas que no dispongan de valor predeterminado.

© JMA 2016. All rights reserved

## INSERT

- Se pueden insertar en la misma instrucción varios conjuntos de valores separando los ( ) por comas.
- INSERT ... SELECT se usa para insertar el conjunto de resultados de una consulta en una tabla existente  

```
INSERT table_or_view_name [ ( column_list ) ]
SELECT ...
```
- INSERT ... EXEC se utiliza para insertar el resultado de un procedimiento almacenado o una expresión de SQL dinámico en una tabla existente  

```
INSERT table_or_view_name [ ( column_list ) ]
EXEC ...
```
- BULK INSERT importa un archivo de datos en una tabla o vista con un formato especificado por el usuario.

© JMA 2016. All rights reserved

## SELECT INTO

- SELECT ... INTO es similar a INSERT ... SELECT pero SELECT ... INTO crea una nueva tabla cada vez que se ejecuta la instrucción  

```
SELECT { column_list | * }
INTO table_name
FROM ...
```
- Para crear la tabla copia los nombres de columnas y deduce los tipos de datos y aceptación de nulos usando el conjunto de resultados.
- No copia restricciones ni índices.
- En caso de existir la tabla de destino dará error, es necesario borrar manualmente la tabla antes de volver a ejecutar la consulta.
- Una vez creada la tabla realiza un INSERT ... SELECT del conjunto de resultados en la tabla recién creada.

© JMA 2016. All rights reserved

# UPDATE

- La modificación de los datos ya insertados en una tabla se realiza con el comando UPDATE  

```
UPDATE [ TOP ( expression ) [ PERCENT ] ] table_or_view_name
SET column_name { = | += | -= | *= | /= | %= | &= | ^= | |= } expression [,n]
[FROM table_or_view_name alias JOIN ...]
[WHERE condition]
```
- SET
  - Indica la columna a modificar y la expresión que generara el nuevo valor que tomará la columna.
  - Se pueden modificar varias o todas las columnas, pero al menos debe haber una.
  - El valor puede ser DEFAULT o NULL
- FROM
  - Se puede combinar la tabla con otras para obtener los valores que participaran en las expresiones que obtienen los nuevos valores.
- WHERE
  - Indica la fila o filas en las que se va a realizar la modificación
  - Si no se cumple la condición no se modificará ninguna fila
  - Si se omite, la modificación se realiza en todas las filas de la tabla
  - TOP puede limitar el número de filas afectadas

© JMA 2016. All rights reserved

# DELETE

- Utilizada para borrar filas de una tabla  

```
DELETE [ TOP ( expression ) [ PERCENT ] ] table_or_view_name
[FROM table_or_view_name alias JOIN ...]
[WHERE condition]
```
- FROM
  - Se puede combinar la tabla con otras para obtener la condición de borrado.
- WHERE
  - Indica la fila o filas en las que se va a realizar el borrado
  - Si no se cumple la condición no se borrará ninguna fila
  - Si se omite, se borrarán todas las filas de la tabla
  - TOP puede limitar el número de filas afectadas
- TRUNCATE TABLE es similar a la instrucción DELETE sin una cláusula WHERE; no obstante, TRUNCATE TABLE es más rápida y utiliza menos recursos de registros de transacciones y de sistema al desasociar el almacenamiento físico de toda la tabla.  

```
TRUNCATE TABLE table_name
```

© JMA 2016. All rights reserved

# MERGE

- Ejecuta operaciones de inserción, actualización o eliminación en una tabla de destino a partir de los resultados de una combinación con una tabla de origen.
- Resume en una única instrucción lo que anteriormente requería múltiples instrucciones.

```
MERGE [ TOP ( expression ) [ PERCENT ] ]
[ INTO ] <target_table>
USING <table_source>
ON <merge_search_condition>
[ WHEN MATCHED [ AND <clause_apply_condition> ]
  THEN <merge_matched> ] [ ...n ]
[ WHEN NOT MATCHED [ BY TARGET ] [ AND <clause_apply_condition> ]
  THEN <merge_not_matched> ]
[ WHEN NOT MATCHED BY SOURCE [ AND <clause_apply_condition> ]
  THEN <merge_matched> ] [ ...n ]
```

- **INTO**
  - Fija la tabla a actualizar por lo que se omite en las instrucciones INSERT INTO, UPDATE y DELETE.
- **USING**
  - tablas, vistas, CTE, tablas derivadas, funciones de tipo tabla virtual y variables de tipo tabla

© JMA 2016. All rights reserved

# MERGE

- **ON**
  - Condición que establece las coincidencias o no coincidencias entre el origen y el destino, como las operaciones se realizan a nivel de fila no es necesario el WHERE en UPDATE y DELETE.
- **WHEN MATCHED**
  - Coincidencia en los dos que requiere una modificación o borrado:  
{ UPDATE SET <set\_clause> | DELETE }
- **WHEN NOT MATCHED [ BY TARGET ]**
  - No existe en el destino por lo que habrá que insertarla  
INSERT [ ( column\_list ) ] VALUES ( values\_list )
- **WHEN NOT MATCHED BY SOURCE**
  - Existe en el destino pero no en el origen, en algunos casos desencadena el borrado en el destino.  
DELETE
- **AND**
  - Permite disponer de múltiples cláusulas MATCHED o NOT MATCHED que se aplicaran solo si se cumple la condición asociada con el AND.
- **TOP**
  - Limita el número de filas afectadas

© JMA 2016. All rights reserved

## Cláusula OUTPUT

- Devuelve información de las filas afectadas por una instrucción INSERT, UPDATE, DELETE o MERGE, o expresiones basadas en esas filas.
- Estos resultados se pueden devolver a la aplicación de procesamiento para que los utilice en mensajes de confirmación, archivado y otros requisitos similares de una aplicación.
- Los resultados también se pueden insertar en una tabla o variable de tabla.
- Además, puede capturar los resultados de una cláusula OUTPUT en una instrucción anidada INSERT, UPDATE, DELETE o MERGE, e insertar los resultados en una tabla de destino o vista.

```
OUTPUT <dml_select_list> [
    INTO { @table_variable | output_table } [
        ( { { DELETED | INSERTED | from_table_name } . { * | column_name }
          | $action | scalar_expression } [ [AS] column_alias_identifier ] [ ,...n ] )
    ] ]
```

© JMA 2016. All rights reserved

## Columnas especiales

- Columnas de numeración automática (Autonúmericas)
  - Identidad, una sola columna por tabla
  - Inicialización de identidad: valor de inicial de una columna de identidad
  - Incremento de identidad: valor en que se incrementa la columna de identidad
  - Vista sys.identity\_columns columna last\_value
- Columnas de identificadores exclusivos globales (GUID)
  - Es RowGuid: Muestra si SQL Server utiliza la columna como una columna ROWGUID. Este valor se puede establecer como Sí sólo para una columna de identidad
  - Valor predeterminado: newid()
- Columnas calculadas
  - Fórmula: Muestra la fórmula para una columna calculada
  - Ignora tipo y longitud

© JMA 2016. All rights reserved



## Columnas automáticas

- Columnas de solo lectura, se omiten en los INSERT
- IDENTITY
  - Columna que genera números secuenciales automáticamente para la inserción en una tabla. Se puede especificar valores de semilla e incremento opcionales. Solo una columna en la tabla puede tener la propiedad IDENTITY definida, destinada a ser la clave primaria.
  - Las Secuencias son objetos que surgen como alternativa a IDENTITY para generar claves primarias que se utilizan en varias tablas: `SELECT NEXT VALUE FOR dbo.MySeq`
- ROWGUID
  - Columna con un identificador exclusivo global (GUID), destinada a ser la clave primaria distribuida. Valor predeterminado: `newid()`
- ROWVERSION (timestamp)
  - Un contador único para toda la base de datos que se actualiza cada vez que se actualiza una fila destinado al control de cambios en las filas.
- Columnas calculadas
  - Obtienen el valor de una fórmula de calculo, ignora el tipo y longitud

© JMA 2016. All rights reserved

## Ejercicios

- Ejercicios propuestos:
  - Añadir un nuevo pedido con al menos tres productos
  - Subir un 10% el precio de venta de los productos de la categoría Accesorios si son de gama alta.

© JMA 2016. All rights reserved

---

## ARQUITECTURA FÍSICA

---

© JMA 2016. All rights reserved

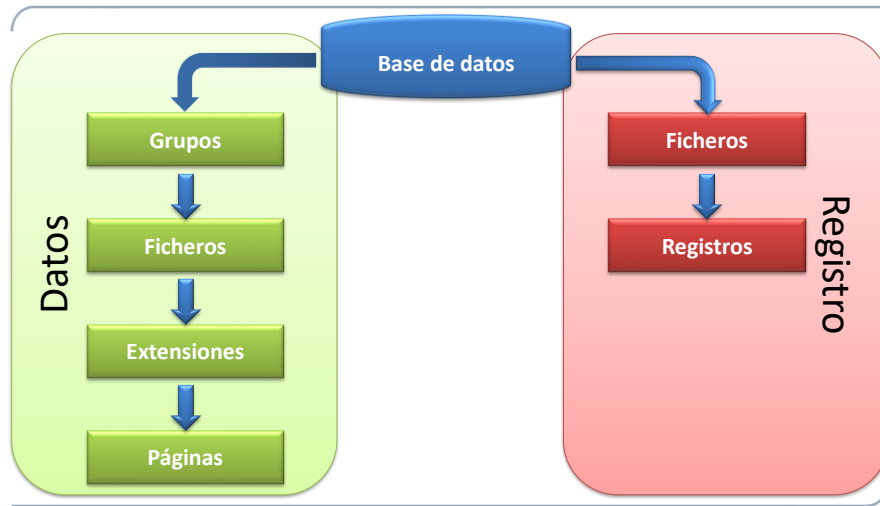
### Composición de un BBDD

---

- Desde el punto de vista lógico
    - Compuesta por objetos
  - Desde el punto de vista físico
    - Páginas, extensiones y ficheros
  - Dos tipos de ficheros
    - Datos
      - Contiene los objetos de la Base de Datos
    - Registro
      - Histórico de modificaciones realizadas en la Base de Datos
      - Sirve para garantizar la integridad de la Base de Datos
- 

© JMA 2016. All rights reserved

## Estructura Física



© JMA 2016. All rights reserved

## Páginas

- La página es la unidad fundamental de asignación de almacenamiento de datos en SQL Server.
- El tamaño de página es de 8 KB (8192 bytes) y no se puede cambiar. Esto significa que las bases de datos de SQL Server tienen 128 páginas por megabyte.
- Cada página empieza con un encabezado de 96 bytes, que se utiliza para almacenar la información del sistema acerca de la página.
- Esta información incluye el número de página, el tipo de página, el espacio disponible en la página y el Id. de unidad de asignación del objeto propietario de la página.

© JMA 2016. All rights reserved

## Tipos de Páginas

- **Datos:** Filas de datos con todos los datos excepto los datos text, ntext e image
- **Índice:** Entradas de índices
- **Texto o imagen (LOB):** Tipos de datos de objetos grandes (LOB): Datos text, ntext e image, opción text in row
- **Espacio libre en páginas:** Información acerca del espacio libre disponible en las páginas
- **Mapa de asignación de índices:** Información acerca de las extensiones utilizadas por una tabla o un índice
- **Mapa de asignación global, Mapa de asignación global secundaria:** Información acerca de las extensiones asignadas
- **Mapa cambiado masivamente:** Información acerca de las extensiones modificadas por operaciones masivas desde la última instrucción BACKUP LOG
- **Mapa cambiado diferencial:** Información acerca de las extensiones que han cambiado desde la última instrucción BACKUP DATABASE

© JMA 2016. All rights reserved

## Tipos de longitudes de datos

- Longitud predefinida
  - El dato siempre ocupa lo mismo en función al tipo de dato elegido.
- Longitud fija
  - El dato siempre ocupa lo mismo en función al tipo de dato elegido y la longitud definida.
- Longitud variable
  - La ocupación viene determinada por la longitud del dato y el tipo, se almacena en paginas de datos.
- Longitud variable – LOB
  - La ocupación viene determinada por la longitud del dato y el tipo, se almacena en paginas LOB.
- Longitud variable – MAX
  - La ocupación viene determinada por la longitud del dato y el tipo, se almacena en paginas de datos o paginas LOB en función del espacio libre.

© JMA 2016. All rights reserved

## Paginas de Datos

- La página está vinculada a un único objeto.
- La página puede contener varias filas.
- Una fila tiene que estar contenida en una página (más páginas adicionales para las columnas LOB con un puntero de texto de 16 bytes).
- Las filas de datos son colocadas en posiciones consecutivas, de tal forma que una nueva fila pueda ocupar todo el espacio libre de la página.
- De cada página se pueden aprovechar 8060 bytes para almacenar datos.
- Al final de la página, comienza una tabla de desplazamiento de fila que contiene una entrada para cada fila de la página y registra la distancia del primer byte de la fila desde el inicio de la página. Las entradas están en orden inverso a la secuencia de las filas de la página y ocupan 2 bytes.

© JMA 2016. All rights reserved

## Paginas LOB

- Los datos LOB se guardan en una cadena enlazada de páginas LOB.
- El enlazamiento determina un máximo de  $2^{18}$  paginas por lo el tamaño máximo de un dato LOB es de 2 Gb.
- Se requiere una cadena LOB para cada celda, columna de la fila, no nula.
- Una página LOB solo contiene datos de una única celda.
- Los datos LOB no nulos ocupan la menos una página LOB, mas el puntero de 16 bytes almacenado a nivel de fila en la página de datos.

© JMA 2016. All rights reserved

## Optimización del espacio

- text in row:
  - Al activarlo, los datos LOB se guardan en la página de datos si tienen espacio.
  - Se puede fijar el límite entre 24 hasta 7000.
  - Puede provocar problemas de rendimiento.
- large value types out of row:
  - Activa el desbordamiento de página, cuando la fila no cabe en la página trata la columnas de longitud variable como si fueran LOB.
  - Mueve dinámicamente una o más columnas de longitud variable a páginas de la unidad de asignación ROW\_OVERFLOW\_DATA, empezando por la columna con el mayor ancho.
  - La longitud máxima de columna sigue siendo 8000, es la combinación de columnas la que la puede superar.
  - Puede empeorar el rendimiento de consultas y ordenaciones, suele ser preferible partición vertical.
- Estas opciones se configuran con el procedimiento almacenado: `sp_tableoption`

© JMA 2016. All rights reserved

## Páginas de Índices

- Un índice es una estructura de disco asociada con una tabla o una vista que acelera la recuperación de filas de la tabla o de la vista.
- Un índice contiene claves generadas a partir de una o varias columnas ordenadas ascendente o descendentemente de la tabla o la vista.
- Dichas claves están almacenadas en una estructura (árbol b) que permite la búsqueda de forma rápida y eficiente de la fila o filas asociadas a los valores de cada clave.
- Es una estructura que maximiza la anchura y minimiza la profundidad.

© JMA 2016. All rights reserved

## Paginas de Índices

- Están compuestas por registros de indexación, uno por fila de la tabla.
- Cada Registro de indexación contiene:
  - Clave de indexación:
    - compuesta por los valores de las columnas para la fila
    - como máximo de 900 bytes y 16 columnas
  - Localizador de fila:
    - Sin índice agrupado: RowID, el localizador es un puntero que se genera a partir del identificador (Id.) de archivo, el número de página y el número de la fila dentro de la página (8 bytes).
    - Con índice agrupado: es el valor de la clave del índice agrupado para la fila.
  - Columnas Incluidas.

© JMA 2016. All rights reserved

## Tipos de índices: Agrupado

- Los índices agrupados ordenan y almacenan las filas de los datos de la tabla o vista de acuerdo con los valores de la clave del índice.
- Son columnas incluidas en la definición del índice.
- Sólo puede haber un índice clúster por cada tabla, porque las filas de datos sólo pueden estar ordenadas de una forma.
- La única ocasión en la que las filas de datos de una tabla están ordenadas es cuando la tabla contiene un índice clúster.
- Cuando una tabla tiene un índice clúster, la tabla se denomina tabla agrupada.
- Si una tabla no tiene un índice clúster, sus filas de datos están almacenadas en una estructura sin ordenar denominada montón.
- Puede ser único, garantiza que la clave de índice no contenga valores duplicados y, por tanto, cada fila de la tabla o vista es en cierta forma única.

© JMA 2016. All rights reserved

## Tipos de índices: No agrupado

- Los índices no agrupados tienen una estructura separada de las filas de datos. Se pueden establecer como únicos.
- Un índice no agrupado contiene los valores de clave de índice no agrupado y cada entrada de valor de clave tiene un puntero a la fila de datos que contiene el valor clave.
- El puntero de una fila de índice no agrupado hacia una fila de datos se denomina localizador de fila.
- La estructura del localizador de filas depende de si las páginas de datos están almacenadas en un montón o en una tabla agrupada.
- Si están en un montón, el localizador de filas es un puntero hacia la fila.
- Si están en una tabla agrupada, el localizador de fila es la clave de índice agrupada.
- Se puede incluir columnas sin clave al nivel hoja de un índice no agrupado con el fin de eludir los límites existentes para las claves de índice, 900 bytes y columnas de 16 claves, así como para ejecutar consultas indizadas y totalmente cubiertas.

© JMA 2016. All rights reserved

## Filtrado de índices

- Un índice filtrado es un índice no clúster optimizado, especialmente indicado para atender consultas que realizan selecciones a partir un subconjunto bien definido de datos.
- Utiliza un predicado de filtro para indizar una parte de las filas de la tabla.
- Un índice filtrado bien diseñado puede, en relación con los índices de tabla completa:
  - mejorar el rendimiento de las consultas y mayor calidad del plan
  - reducir los costos de mantenimiento
  - reducir el coste de almacenamiento del índice.
- Para diseñar índices filtrados efectivos, es importante entender qué consultas utiliza la aplicación y cómo se relacionan con los subconjuntos de datos.

© JMA 2016. All rights reserved



## Otros tipos de índices

- Espacial
  - Un índice espacial proporciona la capacidad de realizar de forma más eficaz determinadas operaciones en objetos espaciales (datos espaciales) en una columna del tipo de datos geometry.
  - El índice espacial reduce el número de objetos a los que es necesario aplicar las operaciones espaciales, que son relativamente costosas.
- XML
  - Representación dividida y permanente de los objetos XML binarios grandes (BLOB) de la columna de tipo de datos xml.
- Texto completo
  - Tipo especial de índice funcional basado en símbolos (token) que crea y mantiene el motor de texto completo de Microsoft para SQL Server.
  - Proporciona la compatibilidad adecuada para búsquedas de texto complejas en datos de cadenas de caracteres.
  - Se almacenan fuera de los ficheros de la base de datos.

© JMA 2016. All rights reserved

## Factor de relleno y fragmentación

- El factor de relleno determina la densidad de ocupación de páginas por parte del índice y fija de llenado de las paginas al crear el índice.
- La fragmentación de índices ocurre cuando los índices tienen páginas en las que la ordenación lógica, basada en el valor de clave, no coincide con la ordenación física dentro del archivo de datos.
- La fragmentación de índices provoca un desequilibrio en el llenado de las paginas de índice.
- Los índices muy fragmentados pueden reducir el rendimiento de la consulta y ralentizar la respuesta de la aplicación.
- Con la función del sistema `sys.dm_db_index_physical_stats` se podrá detectar la fragmentación de un índice específico, de todos los índices de una tabla o vista indizada, de todos los índices de una base de datos o de todos los índices de todas las bases de datos.
- Se puede solucionar la fragmentación del índice reorganizándolo o volviéndolo a generar.

© JMA 2016. All rights reserved

## Número de Índices

- El número de índices afecta al rendimiento y ocupación de la base de datos.
- Es necesario mantener el número de índices al mínimo estrictamente necesario.
- Las restricciones PRIMARY KEY crean índices de forma automática de tipo UNIQUE. Por defecto como agrupados.
- Las restricciones UNIQUE crean índices UNIQUE de forma automática.
- Como máximo se puede tener:
  - Un índice agrupado por tabla.
  - 249 índices no agrupados (incluidos los creados por restricciones PRIMARY KEY o UNIQUE).

© JMA 2016. All rights reserved

## Recomendaciones

- Indexar:
  - Tablas con muchas páginas (no necesariamente filas)
  - Columnas con un amplio rango de valores
  - Columnas que se usan: en consultas, agrupaciones, funciones de agregación y ordenaciones.
  - Columnas usadas en uniones
- NO Indexar:
  - Tablas que ocupen poco (pocas páginas)
  - Tablas con muchas modificaciones pero pocas consultas.
  - Columnas de gran tamaño
- El Asistente para optimización de índices suministra recomendaciones en función a la monitorización del sistema.

© JMA 2016. All rights reserved

## Realizar operaciones de índice en línea

- Por defecto, las operaciones generar o volver a generar un índice de mantienen bloqueos exclusivos de los datos subyacentes y los índices asociados.
- Gracias a la opción ONLINE, es posible que usuarios simultáneos obtengan acceso a los datos de la tabla subyacente o del índice, así como a los índices no clúster asociados durante las operaciones de desfragmentación del índice.
- Para volver a generar un índice en línea:
  - Seleccionar Índice>Propiedades>Opciones:
    - Permitir procesamiento DML en línea = True
  - ALTER INDEX ... REBUILD WITH (ONLINE = ON);

© JMA 2016. All rights reserved

## Directrices para operaciones en línea

- Se recomienda realizar operaciones de índices en línea en entornos empresariales que funcionan 24 horas al día, siete días a la semana, y en los que resulta fundamental la actividad simultánea de los usuarios durante las operaciones de índices.
- La tabla subyacente no se puede modificar (DDL), truncar o quitar mientras se está llevando a cabo una operación de índice en línea.
- Normalmente, las operaciones de índice en línea son más lentas que las operaciones de índice sin conexión equivalentes, independientemente del nivel de actividad de actualización simultánea.
- Como las estructuras de origen y de destino se mantienen durante la operación de índice en línea, el uso de recursos para insertar, actualizar y eliminar transacciones aumenta, potencialmente hasta el doble.
- Las operaciones de índice a gran escala pueden generar grandes cargas de datos que pueden hacer que el registro de transacciones se llene rápidamente.
- La recompilación de índices en línea puede provocar más fragmentación.

© JMA 2016. All rights reserved

# Extensiones

- Unidad básica de almacenamiento de datos (E/S) que consta de 8 páginas contiguas físicamente. El SQL Server lee y escribe extensiones completas.
- Las extensiones están compuestas por 64 KB contiguos de espacio en disco, es decir, 16 extensiones por megabyte.
- Para hacer que la asignación de espacio sea eficaz, no se asigna extensiones completas a tablas con pequeñas cantidades de datos.
- Dos tipos de extensiones:
  - Mixtas
    - Pueden contener datos de hasta 8 objetos de la BBDD, uno por página
  - Uniformes
    - Contienen datos de un único objeto de la BBDD
- Cuando se crea un objeto se sitúa en una extensión mixta
- Según se va llenando de datos, pasan a ocupar una extensión uniforme

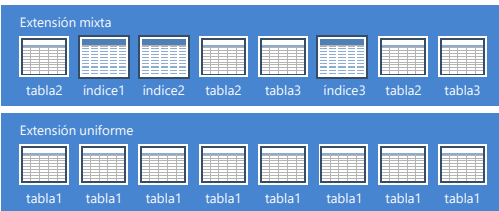
© JMA 2016. All rights reserved

## Extensiones y páginas

Extensión de cabecera de fichero



Extensiones



GAMs y SGAMs

| Current use of extent                 | GAM bit setting | SGAM bit setting |
|---------------------------------------|-----------------|------------------|
| Free, not being used                  | 1               | 0                |
| Uniform, extent, or full mixed extent | 0               | 0                |
| Current use of extent                 | 0               | 1                |

© JMA 2016. All rights reserved

## Ficheros de Datos

- De dos tipos
  - Primario (.mdf): Un fichero de datos primario por BBDD (uno y solo uno). Almacena la información del catálogo y las tablas del sistema de la Base de Datos
  - Secundario (.ndf): Ficheros adicionales que pueden contener datos. Utilizados para balancear cargas y mejorar rendimiento
- Los archivos tienen dos nombres:
  - logical\_file\_name es el nombre que se utiliza para hacer referencia al archivo en las instrucciones Transact-SQL.
  - os\_file\_name es el nombre del archivo físico en disco.
- Se dimensionan en Megabytes (16 extensiones por Megabyte, 128 páginas por Megabyte)
  - Los ficheros pueden crecer automáticamente: porcentualmente o valor absoluto
  - Se puede impedir o limitar el crecimiento a un número máximo de megas (Máximo 16 TB)
  - Balancean la ocupación

© JMA 2016. All rights reserved

## Grupos de Ficheros

- Permiten una gestión más óptima del espacio de una Base de Datos
- Como mínimo cuentan con uno, el PRIMARY
  - No puede ser borrado de la Base de Datos
  - Contiene el fichero Primario (.mdf):.
- El sistema balancea la carga de los datos proporcionalmente a los ficheros de datos de cada grupo
  - Permite un mejor rendimiento en la E/S
- Cada tabla se puede almacenar en grupos distintos, un grupo para las filas, otro para los LOB y cada índice en un grupo.
- Uno de los grupos de ficheros debe ser especificado como soporte de datos por defecto, se le asignaran los elemento que no indiquen lo contrario.
- Los grupos pueden ser marcados de solo lectura, con lo que se impide escribir en los ficheros del grupo.
- Permiten balancean la carga de rendimientos

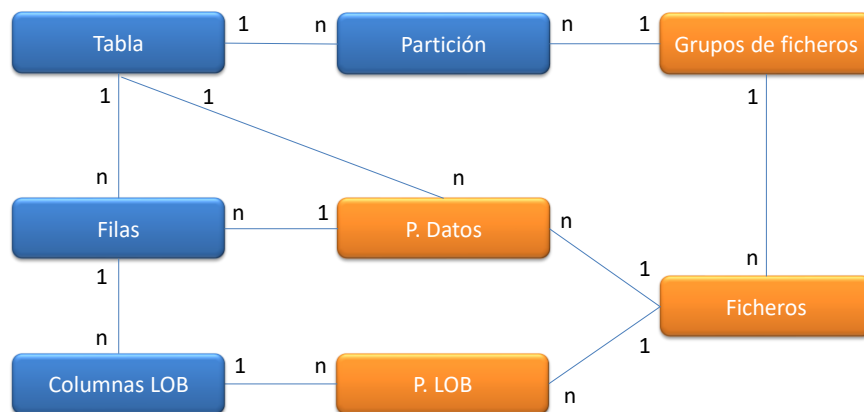
© JMA 2016. All rights reserved

## Tablas e índices con particiones

- Sólo se admiten en las ediciones Enterprise.
- La partición facilita el uso de tablas e índices grandes, ya que permite administrar y tener acceso a subconjuntos de datos de forma rápida y eficaz, a la vez que mantiene la integridad de la recopilación de datos.
- Los datos se dividen en sentido horizontal y se reparten por más de un grupo de archivos dentro de la misma base de datos, puede tener un máximo de 1.000 particiones.
- La tabla o el índice se tratarán como una sola entidad lógica cuando se realicen consultas o actualizaciones en los datos.
- Útiles para tablas muy grandes con una división lógica establecida y mejorar el aprovechamiento de recursos.
- Una **función de partición** define la forma de asignar las filas de una tabla o un índice a un conjunto de particiones a partir de los valores de determinadas columnas, denominadas columnas de partición.
- Un **esquema de particiones** asigna cada partición especificada con la función de partición a un grupo de archivos.

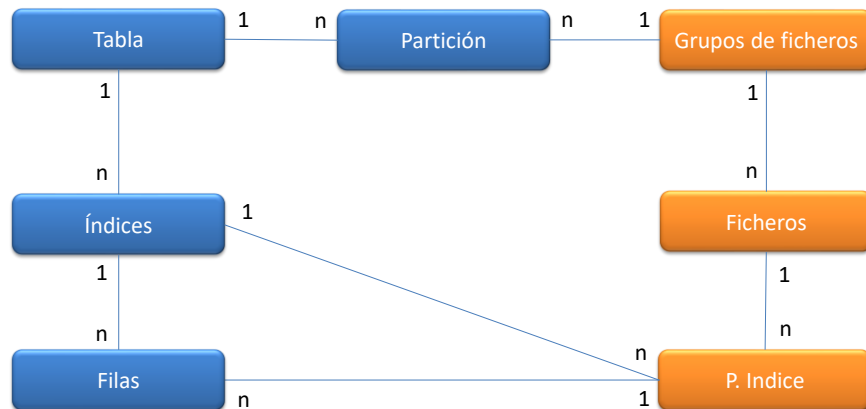
© JMA 2016. All rights reserved

## Tabla: Almacenamiento de Datos



© JMA 2016. All rights reserved

## Tabla: Almacenamiento de Índices



© JMA 2016. All rights reserved

## Operaciones de Partición

- Crear función de partición  
`CREATE PARTITION FUNCTION PF (int) AS RANGE RIGHT FOR VALUES (2014, 2015);`
- Crear esquema de partición  
`CREATE PARTITION SCHEME PS AS PARTITION PF TO (FG_HIST, FG_ACT, FG_FUT);`
- Quitar una partición y mezclar sus valores en la partición anterior  
`ALTER PARTITION FUNCTION PF() MERGE RANGE(2015);`
- Añadir una nueva partición  
`ALTER PARTITION SCHEME PS NEXT USED FG2016;`  
`ALTER PARTITION FUNCTION PF() SPLIT RANGE(2016);`

© JMA 2016. All rights reserved

## Compresión de datos

- SQL Server admite la compresión de fila y de página para las tablas e índices.
- Ayuda para reducir el tamaño de la base de datos.
- Puede contribuir a mejorar el rendimiento de las cargas de trabajo que hacen un uso intensivo de las operaciones de E/S porque los datos se almacenan en menos páginas y las consultas deben leer menos páginas del disco.
- Requieren recursos de CPU adicionales en el servidor de base de datos para comprimir y descomprimir los datos, mientras los datos se intercambian con la aplicación.
- Para las tablas e índices con particiones, la opción de compresión se puede configurar para cada partición una configuración de compresión diferente.

© JMA 2016. All rights reserved

## Nivel de compresión

- El tipo de compresión se puede establecer en ROW, PAGE o NONE
  - ROW: Se realiza a nivel de fila. Todas las columnas de longitud fija se tratan como si fueran de longitud variable. Ocupan lo que el dato contenido. No afecta a las columnas LOB.
  - PAGE: Se realiza a nivel de página completa, empieza con la compresión ROW, a la que se le aplica los algoritmos de compresión por prefijos y patrones (diccionarios).
- La compresión conlleva una sobrecarga de datos que puede superar la ocupación sin comprimir.

```
ALTER TABLE Tabla REBUILD WITH (DATA_COMPRESSION = PAGE);
ALTER TABLE TablaParticionada REBUILD PARTITION = 2 WITH
(DATA_COMPRESSION = ROW) ;
```

© JMA 2016. All rights reserved



## Estimación de la ocupación

- Para calcular el espacio necesario para almacenar datos en una tabla:
  1. Calcular el espacio necesario para las filas:
    1. Calcular el numero de paginas de datos.
    2. Calcular el numero de páginas LOB por cada columna, sumar todas las columnas y multiplicar por el número de filas.
  2. Calcular el espacio necesario para los índices:
    1. Por cada índice, calcular el numero de paginas de índice.
    2. Sumar el numero de páginas de cada índice.
  3. Sumar el espacio necesario para las filas con el espacio necesario para los índices.
- El resultado viene expresado en páginas, para obtenerlo en megas, dividirlo entre 128.

© JMA 2016. All rights reserved

## Gestión automatizada

- Se puede automatizar la gestión del espacio asignado a una base de datos, fundamentalmente para sistemas no administrados.
- Crecimiento (solo a nivel de ficheros):
  - Se puede permitir que los ficheros crezcan automáticamente.
  - El crecimiento de ficheros provoca fragmentación en el disco duro con la consecuente reducción de rendimiento.
- Decrecimiento (solo a nivel de base de datos):
  - La opción de base de datos AUTO\_SHRINK marca que cuando la base de datos tenga un 75% libre, reduzca todos los ficheros dejando un 25% de espacio libre a cada uno.
- Los automatismos degradan el rendimiento al entrar en periodos de máxima actividad de la base de datos.
- Los cambios de tamaño conllevan bloqueos a nivel de fichero y de base de datos.

© JMA 2016. All rights reserved

## Registro de Transacciones

- Usados por el motor para grabar los cambios generados durante el uso de la Base de Datos
  - Registra tanto transacciones de usuarios como transacciones implícitas del sistema
  - El registro de transacciones es un archivo de registro circular.
- Utilizados para mantener la integridad de la Base de Datos y repararla en caso de fallo del sistema
- Toda base de datos debe tener al menos uno como mínimo.
- La extensión del archivo transaction log es .ldf
- Al igual que los ficheros de datos:
  - Se dimensionan en megas.
  - Pueden crecer automáticamente: porcentualmente o valor absoluto
  - Se puede impedir o limitar el crecimiento a un número máximo de megas (Máximo 12 TB)
- La falta de espacio puede impedir operar con la base de datos.

© JMA 2016. All rights reserved

## Transacciones

- Una transacción es una secuencia de operaciones realizadas como una sola unidad lógica de trabajo.
- Una unidad lógica de trabajo debe exhibir cuatro propiedades, conocidas como propiedades ACID (atomicidad, coherencia, aislamiento y durabilidad), para ser calificada como transacción:
  - Atomicidad: Una transacción debe ser una unidad atómica de trabajo, tanto si se realizan todas sus modificaciones en los datos, como si no se realiza ninguna de ellas.
  - Coherencia: Cuando finaliza, una transacción debe dejar todos los datos en un estado coherente, es decir, se deben cumplir todas las reglas de integridad de todos los datos.
  - aislamiento: Las modificaciones realizadas por transacciones simultáneas se deben aislar de las modificaciones llevadas a cabo por otras transacciones simultáneas.
  - Durabilidad: Una vez concluida una transacción, sus efectos son permanentes en el sistema. Las modificaciones persisten aún en el caso de producirse un error del sistema.

© JMA 2016. All rights reserved

## Especificar y exigir transacciones

- Los programadores de SQL son los responsables de iniciar y finalizar las transacciones en los puntos que exijan la coherencia lógica de los datos.
- El programador debe asegurar la atomicidad de las transacciones.
- SQL Server proporciona:
  - Servicios de bloqueo que preservan el aislamiento de la transacción.
  - Servicios de registro que aseguran la durabilidad de la transacción:
    - Aún en el caso de que falle el hardware del servidor, el sistema operativo o el propio SQL Server, SQL Server utiliza registros de transacciones, al reinicio, para deshacer automáticamente las transacciones incompletas en el momento en que se produjo el error en el sistema.
  - Características de administración de transacciones que exigen la atomicidad y coherencia de la transacción:
    - Una vez iniciada una transacción, debe concluirse correctamente o SQL Server deshacerá todas las modificaciones de datos realizadas desde que se inició la transacción.

© JMA 2016. All rights reserved

## Tipos de transacciones

- Transacciones de confirmación automática
  - Éste es el modo predeterminado de SQL Server. Cada instrucción individual de Transact-SQL se confirma cuando termina. No tiene que especificar instrucciones para controlar las transacciones.
- Transacciones explícitas
  - Se inicia con BEGIN TRANSACTION, se confirma con COMMIT y se descarta con ROLLBACK.
- Transacciones implícitas
  - Establezca el modo de transacción implícita a través de una función de la API o la instrucción SET IMPLICIT\_TRANSACTIONS ON de Transact-SQL. La siguiente instrucción inicia automáticamente una nueva transacción. Cuando se concluye la transacción, la instrucción de Transact-SQL siguiente inicia una nueva transacción.
- Transacciones distribuidas
  - Implican a varias instancias y se dividen en Fases de preparación y de confirmación.

© JMA 2016. All rights reserved

## Información registrada

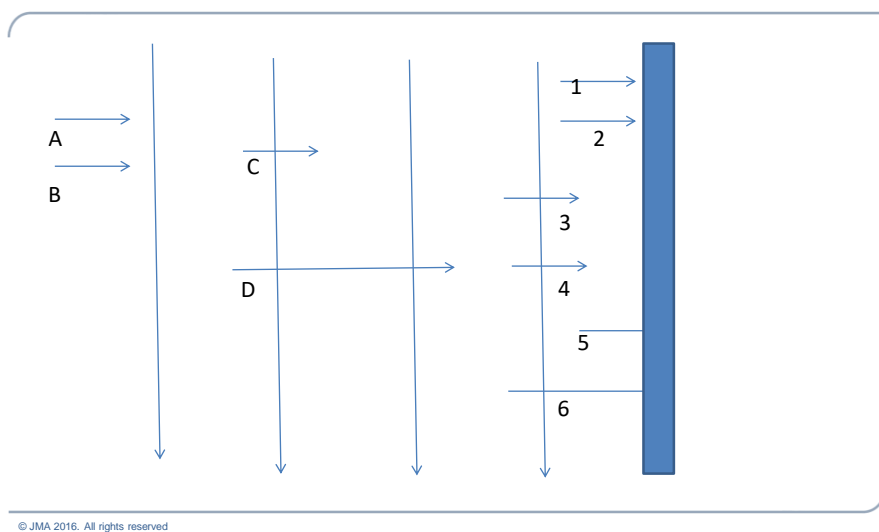
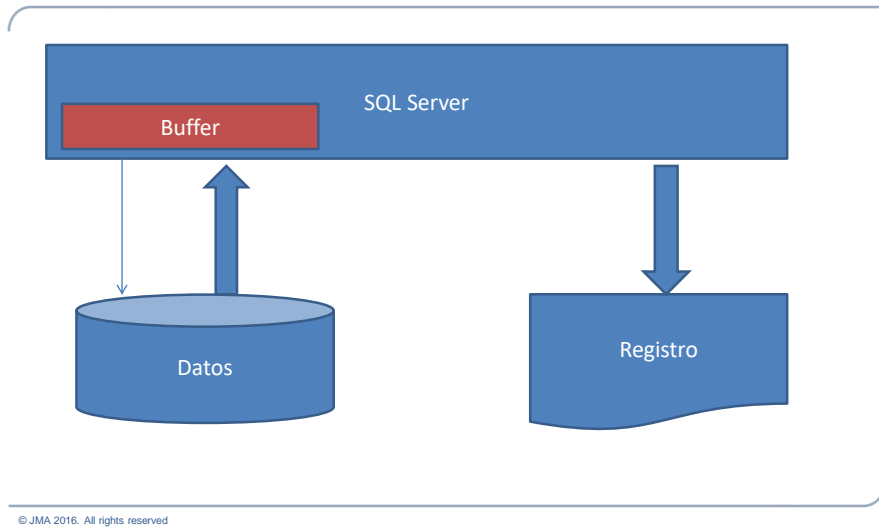
- Registro de la operación lógica (SQL).
  - Para confirmar la operación lógica, se vuelve a ejecutar.
  - Para deshacer la operación lógica, se ejecuta la operación lógica inversa.
- Registro de las imágenes (páginas) anterior y posterior.
  - Para confirmar la operación, se aplica la imagen posterior.
  - Para deshacer la operación, se aplica la imagen anterior.
- En el registro de transacciones se registran muchos tipos de operaciones, que incluyen:
  - El inicio y el final de cada transacción.
  - Todas las modificaciones de los datos (inserción, actualización y eliminación). Esto incluye las modificaciones de las tablas del sistema hechas por procedimientos almacenados del sistema o por instrucciones del lenguaje de definición de datos (DDL).
  - Las asignaciones o cancelaciones de asignación de extensiones.
  - La creación o eliminación de una tabla o un índice.

© JMA 2016. All rights reserved

## Operativa

- Toda operación se registra en el registro de transacciones en el momento de producirse.
- Los datos se cargan en memoria, en caso de no estar ya, y modifican en memoria, pero no se graban en los ficheros de datos una vez terminadas las modificaciones.
- Los datos se graban periódicamente en los ficheros de datos en los denominados puntos de comprobación (CHECKPOINT).
  - Se marca el inicio del punto de comprobación en el registro de transacciones.
  - Se graban las páginas desfasadas en los ficheros de datos.
  - Se marca el final del punto de comprobación en el registro de transacciones.
- Se reserva el cauce de lectura para la carga de páginas de datos y el de escritura para la grabación en el fichero de transacciones.

© JMA 2016. All rights reserved



## Modelos de recuperación

- Sencillo:
  - Uso mínimo del registro de transacciones, se eliminan las transacciones una vez superado su punto de comprobación.
- Completo:
  - Uso extensivo del registro de transacciones, se eliminan las transacciones cuando se aseguran mediante una copia de seguridad.
- Carga masiva:
  - Modelo completo pero con registro mínimo de las operaciones de carga masiva. Truncan la secuencia de restauración.
- Cuando se produce la recuperación:
  - Al arrancar la instancia
  - Al restaurar la base de datos

© JMA 2016. All rights reserved

## Puntos de comprobación

- Se ejecutan de forma fija:
  - Cuando se ejecuta una instrucción CHECKPOINT. Se insertan puntos de comprobación en la base de datos actual de la conexión.
  - Cuando se utiliza ALTER DATABASE para cambiar una opción de base de datos. ALTER DATABASE inserta puntos de comprobación en la base de datos cuando se cambian las opciones de la base de datos.
  - Cuando se detiene una instancia de SQL Server por: Ejecutar una instrucción SHUTDOWN o se para el servicio.
- Se genera periódicamente de manera automática en cada base de datos para reducir el tiempo que tardaría la instancia en recuperar la base de datos.
  - Se genera cuando el número de registros alcanza el número que estima que puede procesar durante el tiempo especificado en la opción intervalo de recuperación.
  - En el modelo de recuperación sencilla, se genera puntos adicionales cuando el registro está ocupado en un 70 por ciento.

© JMA 2016. All rights reserved

## Proceso de recuperación

- Relanza las transacciones iniciadas y confirmadas después del último punto de comprobación.
- Ignora las transacciones iniciadas y canceladas después del último punto de comprobación.
- En las transacciones iniciadas antes del último punto de comprobación y confirmadas después del último punto de comprobación, sobrescribe los ficheros de datos con la imagen posterior almacenada en el registro de transacciones.
- En las transacciones iniciadas antes del último punto de comprobación y descartadas después del último punto de comprobación, sobrescribe los ficheros de datos con la imagen anterior almacenada en el registro de transacciones.
- En las transacciones iniciadas antes del último punto de comprobación y no completadas después del último punto de comprobación, sobrescribe los ficheros de datos con la imagen anterior almacenada en el registro de transacciones.

© JMA 2016. All rights reserved

## Bloqueos

- Balanceo: simultaneidad frente a coherencia
- Estrategias: optimista y pesimista
- Problemas de simultaneidad
- Grados de aislamiento
- Modos de interbloqueo
- Granularidad y jerarquías de bloqueo
- Interbloqueos

© JMA 2016. All rights reserved

## Problemas de simultaneidad

- Actualizaciones perdidas:
  - Este problema surge cuando dos o más transacciones seleccionan la misma fila y, a continuación, la actualizan de acuerdo con el valor seleccionado originalmente.
  - Ninguna transacción es consciente de las otras transacciones.
  - La última actualización sobrescribe las actualizaciones realizadas por las otras transacciones y, en consecuencia, se pierden datos.
- Dependencia no confirmada (lectura no actualizada o desfasada):
  - Este problema se produce cuando una transacción selecciona una fila que está siendo actualizada por otra transacción.
  - La transacción que llega en segundo lugar lee datos que todavía no han sido confirmados y que la transacción que actualiza la fila puede modificar.

© JMA 2016. All rights reserved

## Problemas de simultaneidad

- Análisis incoherente (lectura irrepensible):
  - Este problema se produce cuando una transacción obtiene acceso a la misma fila varias veces y en cada ocasión lee datos diferentes.
  - El análisis incoherente es similar a la dependencia no confirmada en tanto que una transacción está modificando los datos que está leyendo una segunda transacción.
  - Sin embargo, en el caso del análisis incoherente, los datos que lee la segunda transacción están confirmados por la transacción que realizó el cambio.
  - Además, el análisis incoherente comprende varias lecturas (dos o más) de la misma fila, y cada vez hay otra transacción que modifica la información; de ahí el término lectura irrepensible.

© JMA 2016. All rights reserved



## Problemas de simultaneidad

- **Lecturas fantasmas o irrepetibles:**
  - Este problema se produce cuando se realiza una acción de insertar o eliminar en una fila y ésta pertenece a un intervalo de filas que está leyendo una transacción.
  - La primera lectura que hizo la transacción en el intervalo de filas muestra una fila que ya no existe en la segunda lectura o en lecturas sucesivas porque otra transacción la ha eliminado.
  - De forma similar, como consecuencia de una inserción realizada por otra transacción, la segunda lectura o las lecturas sucesivas de la transacción muestran una fila que no existía en la primera lectura.

© JMA 2016. All rights reserved

## Niveles de aislamiento

- **READ UNCOMMITTED**

Implementa las lecturas no confirmadas o el bloqueo de nivel de aislamiento 0, lo que significa que no hay bloqueos compartidos y que los bloqueos exclusivos no están garantizados. Cuando se establece esta opción, es posible leer datos no confirmados, los valores pueden cambiar y pueden aparecer y desaparecer filas en el conjunto de datos antes del final de la transacción. Esta opción tiene el mismo efecto que establecer NOLOCK en todas las tablas y en todas las instrucciones SELECT de la transacción. Se trata del menos restrictivo de los cuatro niveles de aislamiento.
- **READ COMMITTED (predeterminada)**

Especifica que se mantengan los bloqueos compartidos mientras se leen datos para evitar lecturas no actualizadas, pero se pueden modificar los datos antes del final de la transacción, lo que provoca lecturas no repetibles o datos fantasmas. Esta opción es la predeterminada en SQL Server.

© JMA 2016. All rights reserved

## Niveles de aislamiento

- **REPEATABLE READ**

Se establecen bloqueos para todos los datos utilizados en la consulta, lo que impide que otros usuarios los actualicen, aunque es posible insertar nuevas filas fantasmas en los datos que otro usuario establezca, de modo que se incluyan en lecturas posteriores de la misma transacción. Como la simultaneidad es inferior que el nivel de aislamiento predeterminado, sólo se debe usar esta opción cuando sea necesario.

- **SERIALIZABLE**

Se establece un bloqueo de intervalo en el conjunto de datos, lo que impide que otros usuarios actualicen o inserten filas en el conjunto de datos hasta que finalice la transacción. Es el más restrictivo de los cuatro niveles de aislamiento. Al ser menor la simultaneidad, sólo se debe utilizar esta opción cuando sea necesario. Esta opción tiene el mismo efecto que establecer **HOLDLOCK** en todas las tablas y en todas las instrucciones **SELECT** de la transacción.

© JMA 2016. All rights reserved

## Niveles de aislamiento

- **SNAPSHOT**

Especifica que los datos leídos por cualquier instrucción de una transacción sean la versión coherente, desde el punto de vista transaccional, de los datos existentes al comienzo de la transacción.

La transacción únicamente puede reconocer las modificaciones de datos confirmadas antes del comienzo de la misma.

Las instrucciones que se ejecuten en la transacción actual no verán las modificaciones de datos efectuadas por otras transacciones después del inicio de la transacción actual.

El efecto es el mismo que se obtendría si las instrucciones de una transacción obtuviesen una instantánea de los datos confirmados tal como se encontraban al comienzo de la transacción.

© JMA 2016. All rights reserved

## Niveles de aislamiento

| Nivel de aislamiento                     | Actualización perdidas | Lectura no actualizada | Lectura no repetible | Lectura Fantasma |
|------------------------------------------|------------------------|------------------------|----------------------|------------------|
| Lectura no confirmada (READ UNCOMMITTED) | Sí                     | Sí                     | Sí                   | Sí               |
| Lectura confirmadas (READ COMMITTED)     | Sí                     | No                     | Sí                   | Sí               |
| Lectura repetible (REPEATABLE READ)      | No                     | No                     | No                   | Sí               |
| Instantánea (SNAPSHOT)                   | No                     | No                     | No                   | No               |
| Serializable (SERIALIZABLE)              | No                     | No                     | No                   | No               |

© JMA 2016. All rights reserved

## Modos de bloqueo

- Los modos de bloqueo determinan el modo en que las transacciones simultáneas pueden tener acceso a los recursos.
  - Compartido (S)**: Se utiliza para operaciones de lectura que no cambian ni actualizan datos, como la instrucción SELECT.
  - Actualizar (U)**: Se utiliza en recursos que se pueden actualizar. Evita una forma común de interbloqueo que se produce cuando varias sesiones leen, bloquean y actualizan recursos.
  - Exclusivo (X)**: Se utiliza para operaciones de modificación de datos, como INSERT, UPDATE o DELETE. Garantiza que no puedan realizarse varias actualizaciones simultáneamente en el mismo recurso.
  - Intención**: Se utiliza para establecer una jerarquía de bloqueos. Los tipos de bloqueo de intención son: **intención compartido (IS)**, **intención exclusivo (IX)** y **compartido con intención exclusivo (SIX)**.
  - Esquema**: Se utiliza cuando se ejecuta una operación que depende del esquema de una tabla. Hay dos tipos de bloqueo de esquema: **modificación del esquema (Sch-M)** y **modificación de estabilidad (Sch-S)**.
  - Actualización masiva (BU)**: Se utiliza cuando se copian datos de forma masiva en una tabla y se especifica la sugerencia TABLOCK.
  - Intervalo de claves**: Protege el intervalo de filas que lee una consulta cuando se utiliza el nivel de aislamiento de transacciones serializables. Garantiza que otras transacciones no puedan insertar filas que podrían incluirse como respuesta de las consultas de la transacción serializable si las consultas se volvieran a ejecutar.

© JMA 2016. All rights reserved

## Granularidad y jerarquías de bloqueo

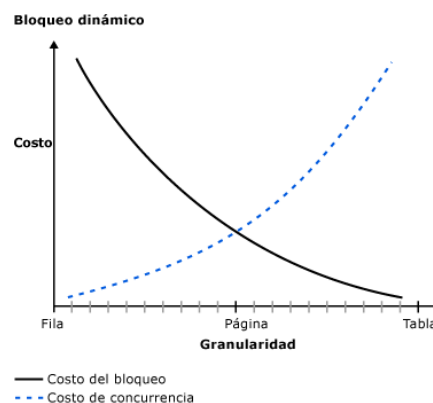
La extensión de bloqueo es el proceso de convertir muchos bloqueos concretos en menos bloqueos más generales, lo que reduce la sobrecarga del sistema al tiempo que aumenta la probabilidad de contención de simultaneidad.

- **RID**: Identificador de fila que se utiliza para bloquear una sola fila de un montón.
- **KEY**: Bloqueo de fila dentro de un índice que se utiliza para proteger intervalos de claves en transacciones serializables.
- **PAGE**: Página de 8 kilobytes (KB) de una base de datos, como páginas de datos o de índices.
- **EXTENT**: Grupo contiguo de ocho páginas, como páginas de datos o de índices.
- **HoBT**: Montón o árbol b. Bloqueo que protege un árbol B (índice) o las páginas de datos del montón en una tabla que no posee un índice agrupado.
- **TABLE**: Tabla completa, con todos los datos e índices.
- **FILE**: Archivos de la base de datos.
- **APPLICATION**: Recurso especificado por la aplicación.
- **METADATA**: Bloqueos de metadatos.
- **ALLOCATION\_UNIT**: Unidad de asignación.
- **DATABASE**: Base de datos completa.

© JMA 2016. All rights reserved

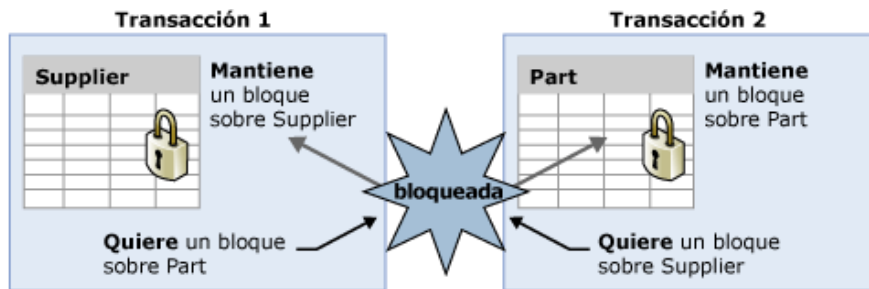
## Bloqueo dinámico

- La utilización de bloqueos de bajo nivel, como los de fila, aumenta la simultaneidad reduciendo la probabilidad de que dos transacciones soliciten bloqueos de los mismos datos al mismo tiempo. También aumenta el número de bloqueos y los recursos necesarios para administrarlos.
- Los bloqueos de alto nivel de tabla o página producen una sobrecarga menor, pero a costa de reducir la simultaneidad.



© JMA 2016. All rights reserved

# Interbloqueos



© JMA 2016. All rights reserved

# Interbloqueos

- Un interbloqueo se produce cuando hay una dependencia cíclica entre dos o más subprocesos para algún conjunto de resultados.
- Detección de interbloqueos: un subproceso de supervisión de bloqueos que periódicamente inicia una búsqueda por todas las tareas de una instancia
  - El intervalo predeterminado es de 5 segundos
  - Baja hasta sólo 100 milisegundos
  - Segundo intento, solución de interbloqueo
- Solución de interbloqueo
  - Prevención mediante el Motor de consultas y plan de ejecución
  - Expiración por tiempos de espera
    - sesión, transacción, SET LOCK\_TIMEOUT, ...
  - Finalización de subproceso (error 1205)
    - Menor nivel (SET DEADLOCK\_PRIORITY)
    - reversión menos costosa (deshacer)
    - a igualdad: aleatoriamente

© JMA 2016. All rights reserved

## Interbloqueos

- Para evitar los interbloqueos:
  1. No abra una transacción mientras examina los datos si es posible.
  2. Las transacciones deben ser atómicas.
  3. Haga la transacción lo más corta posible.
  4. La transacción no debería depender de interfaz de usuario.
  5. Haga un uso inteligente de los niveles más bajos de aislamiento de las transacciones.
  6. Maneje siempre que sea posible los conjuntos de datos en el mismo orden

© JMA 2016. All rights reserved

## Información de bloqueo

- Los bloqueos pueden reducir el rendimiento de la instancia.
- SQL Server ofrece varias formas de obtener información acerca de la actividad de bloqueo actual en una instancia:
  - Se puede realizar una consulta de la vista de administración dinámica sys.dm\_tran\_locks para obtener información acerca del estado actual de bloqueo de una instancia del Database Engine (Motor de base de datos).
  - Mediante el SQL Server Profiler, se puede especificar la categoría de eventos de los bloqueos para capturar información acerca de los bloqueos de eventos en una traza.
  - En el Monitor de sistema, se pueden especificar contadores del objeto de bloqueos para supervisar el nivel de bloqueo de una instancia del Motor de base de datos.

© JMA 2016. All rights reserved

## Ajustes de instancia

- Management Studio - Propiedades:
  - Memoria (máxima y mínima)
  - Procesador (afinidad)
  - Seguridad (modo y auditorias)
  - Conexiones (número y valores por defecto)
  - Configuración de base de datos (por defecto)
  - Opciones Avanzadas (paralelismo, red, ...)
- Transact SQL:
  - `sp_configure 'show advanced options', 1;`
  - `RECONFIGURE;`
- Especificaciones de producto para SQL Server

© JMA 2016. All rights reserved

## Conexión de administrador dedicada (DAC)

- SQL Server proporciona una conexión de diagnóstico especial para los administradores cuando no son posibles las conexiones estándar con el servidor.
- Está disponible a través de la utilidad del símbolo del sistema `sqlcmd` mediante un modificador de administrador especial (-A).
- También se puede conectar agregando el prefijo **admin:** al nombre de la instancia:
  - `sqlcmd -Sadmin:instance_name`
  - SSMS: Nombre del servidor: **admin:instance\_name**
- Desactivada por defecto, para activarla:
 

```
sp_configure 'remote admin connections', 1;
GO
RECONFIGURE;
GO
```

© JMA 2016. All rights reserved

## Propiedades de la base de datos

- Management Studio :
  - Nombre, Propietario, Modelo de recuperación, Nivel de compatibilidad y Tipo de contención.
  - Grupos de archivos (Filas, FILESTREAM, Datos con optimización para memoria) y Archivos.
  - Automatismos, contención, estado, recuperación y valores por defecto.
- Transact SQL:
  - CREATE DATABASE y ALTER DATABASE
  - ALTER DATABASE SET
  - Función: DATABASEPROPERTYEX

© JMA 2016. All rights reserved

## Dimensionado de la B.D. tempdb

- La base de datos del sistema **tempdb** es un recurso global disponible para todos los usuarios de la instancia de SQL Server, utilizada para:
  - Objetos temporales de usuario que se hayan creado explícitamente, como: tablas y procedimientos temporales, variables de tabla o cursores.
  - Objetos internos creados por Motor de base de datos de SQL Server, como: tablas de trabajo para almacenar resultados intermedios para colas u ordenación.
  - Versiones de fila generadas por las transacciones de modificación de datos en una base de datos que utiliza transacciones de lectura confirmada que usan transacciones de aislamiento de versiones de fila o de aislamiento de instantáneas.
  - Versiones de fila que se hayan generado mediante transacciones de modificación de datos para características, como: operaciones de índice en línea, conjuntos de resultados activos múltiples (MARS) y desencadenadores AFTER.

© JMA 2016. All rights reserved



## Consideraciones sobre tempdb

- tempdb se vuelve a crear cada vez que se inicia SQL Server, de forma que el sistema siempre se inicia con una copia limpia de la base de datos.
- El tamaño de tempdb puede afectar al rendimiento de un sistema, si el tamaño es demasiado pequeño, el procesamiento del sistema podría estar demasiado ocupado con el crecimiento automático de la base de datos y, por tanto, no podría satisfacer las necesidades de carga de trabajo cada vez que se inicia SQL Server.
- La ausencia de espacio en tempdb puede acarrear el bloqueo de la instancia, los ficheros deben crecer automáticamente.

© JMA 2016. All rights reserved

## Operaciones no disponibles sobre tempdb

- Agregar grupos de archivos, quitar el grupo de archivos principal, el archivo de datos principal o el archivo de registro.
- Activar reducir o cerrar automáticamente.
- Cambiar el nombre de la base de datos o del grupo de archivos principal.
- Cambiar la intercalación o el propietario de la base de datos (es propiedad de **sa**).
- Establecer la base de datos en OFFLINE o READ\_ONLY.
- Crear una instantánea, eliminar la base de datos o eliminar el usuario **guest**.
- Realizar una copia de seguridad o restaurar la base de datos.
- Habilitar el mecanismo de captura de cambios en los datos.
- Participar en el reflejo de la base de datos.
- Ejecutar DBCC CHECKALLOC o DBCC CHECKCATALOG.

© JMA 2016. All rights reserved

---

Lenguaje de Validación de Datos

**DVL**

---

© JMA 2016. All rights reserved

## Transacciones

- Una transacción es una secuencia de operaciones realizadas como una sola unidad lógica de trabajo.
- Una unidad lógica de trabajo debe exhibir cuatro propiedades, conocidas como propiedades ACID (atomicidad, coherencia, aislamiento y durabilidad), para ser calificada como transacción:
  - Atomicidad: Una transacción debe ser una unidad atómica de trabajo, tanto si se realizan todas sus modificaciones en los datos, como si no se realiza ninguna de ellas.
  - Coherencia: Cuando finaliza, una transacción debe dejar todos los datos en un estado coherente, es decir, se deben cumplir todas las reglas de integridad de todos los datos.
  - aislamiento: Las modificaciones realizadas por transacciones simultáneas se deben aislar de las modificaciones llevadas a cabo por otras transacciones simultáneas.
  - Durabilidad: Una vez concluida una transacción, sus efectos son permanentes en el sistema. Las modificaciones persisten aún en el caso de producirse un error del sistema.

---

© JMA 2016. All rights reserved

## Especificar y exigir transacciones

- Los programadores de SQL son los responsables de iniciar y finalizar las transacciones en los puntos que exijan la coherencia lógica de los datos.
- El programador debe asegurar la atomicidad de las transacciones.
- SQL Server proporciona:
  - Servicios de bloqueo que preservan el aislamiento de la transacción.
  - Servicios de registro que aseguran la durabilidad de la transacción:
    - Aún en el caso de que falle el hardware del servidor, el sistema operativo o el propio SQL Server, SQL Server utiliza registros de transacciones, al reinicio, para deshacer automáticamente las transacciones incompletas en el momento en que se produjo el error en el sistema.
  - Características de administración de transacciones que exigen la atomicidad y coherencia de la transacción:
    - Una vez iniciada una transacción, debe concluirse correctamente o SQL Server deshacerá todas las modificaciones de datos realizadas desde que se inició la transacción.

© JMA 2016. All rights reserved

## Tipos de transacciones

- Transacciones de confirmación automática
  - Éste es el modo predeterminado de SQL Server. Cada instrucción individual de Transact-SQL se confirma cuando termina. No tiene que especificar instrucciones para controlar las transacciones.
- Transacciones explícitas
  - Se inicia con BEGIN TRANSACTION, se confirma con COMMIT y se descarta con ROLLBACK.
- Transacciones implícitas
  - Establezca el modo de transacción implícita a través de una función de la API o la instrucción SET IMPLICIT\_TRANSACTIONS ON de Transact-SQL. La siguiente instrucción inicia automáticamente una nueva transacción. Cuando se concluye la transacción, la instrucción de Transact-SQL siguiente inicia una nueva transacción.
- Transacciones distribuidas
  - Implican a varias instancias y se dividen en Fases de preparación y de confirmación.

© JMA 2016. All rights reserved

## Lenguaje de Validación de Datos

- Para fijar el nivel de aislamiento que determina el uso de los bloqueos  
`SET TRANSACTION ISOLATION LEVEL {READ UNCOMMITTED | READ COMMITTED  
 | REPEATABLE READ | SNAPSHOT | SERIALIZABLE};`
- Para indicar que las transacciones se abran automáticamente:  
`SET IMPLICIT_TRANSACTIONS ON`
- Para abrir explícitamente una transacción  
`BEGIN { TRAN | TRANSACTION }  
 [ { transaction_name | @tran_name_variable }  
 [ WITH MARK [ 'description' ] ]  
 ][ ; ]`
- Se pueden anidar transacciones para confirmar o descartar parcialmente partes de la transacciones, pendientes de la confirmación o descarte de la transacción principal.
- Se pueden marcar las transacciones en el registro de transacción para facilitar la restauración de la base de datos hasta el momento de iniciar la transacción o de su confirmación.

© JMA 2016. All rights reserved

## Lenguaje de Validación de Datos

- Para abrir una transacción distribuida:  
`BEGIN DISTRIBUTED { TRAN | TRANSACTION } [ name | @variable ]`
- Para confirmar la transacción como correcta y finalizarla:  
`COMMIT { TRAN | TRANSACTION } [ name | @variable ] ]  
COMMIT [ WORK ]`
- Para establecer un punto de retorno dentro de una transacción:  
`SAVE { TRAN | TRANSACTION } { savepoint_name | @savepoint_variable }`
- Para descartar los cambios realizados en la transacción y finalizarla:  
`ROLLBACK {TRAN | TRANSACTION} [trans_name | @tran_variable |  
 savepoint_name ]  
ROLLBACK [ WORK ]`
- **@@TRANCOUNT:**
  - Devuelve el número de transacciones activas de la conexión actual.

© JMA 2016. All rights reserved

---

Lenguaje de Definición de Datos

## DDL

---

© JMA 2016. All rights reserved

## Lenguaje de Definición de Datos DDL

- SQL permite crear, modificar y borrar la estructura física de los objetos de la base de datos
  - CREATE
  - ALTER
  - DROP
  - RENAME
  - ENABLE TRIGGER
  - DISABLE TRIGGER
  - UPDATE STATISTICS
- Las instrucciones DDL no permiten rollback de la acción realizada
- Deben ejecutarse independientemente, con una directiva GO después de cada una.
- SQL Server dispone de un interfaz gráfico que hace innecesario el uso de las instrucciones DDL en la mayoría de los casos.

---

© JMA 2016. All rights reserved

## Bases de datos

- Creación de una BBDD
  - Mediante el SQL Server Management Studio
  - Mediante Transact-SQL
    - Con la instrucción CREATE DATABASE
    - Existe una plantilla para la creación en el SSMS
    - Se puede generar el script de creación en el SSMS, y ejecutarlo posteriormente como sentencia SQL
- Las Bases de Datos heredan las características de la Base de Datos model
  - Algunas propiedades pueden ser determinadas durante la creación de la BBDD
- Se pueden modificar posteriormente
  - Mediante el SSMS
  - Mediante Transact-SQL
    - Exec sp\_dboption nombre\_bd, opcion, TRUE | FALSE
    - Mediante el comando ALTER DATABASE
- Para ver las propiedades actuales de una Base de Datos
  - Exec sp\_dboption nombre\_bd

© JMA 2016. All rights reserved

## Valores predeterminados

- Los valores predeterminados especifican el valor por defecto de una columna si no se especifica un valor al insertar las filas
- Para crear un nuevo valor predeterminado es necesario asignar:
  - Nombre del valor predeterminado
  - Valor que puede ser cualquier expresión cuyo resultado sea una constante: constantes, funciones integradas o expresiones matemáticas

Hoy: getdate()

Mañana: DATEADD(day, 1, getdate())

Cero: 0

© JMA 2016. All rights reserved

## Reglas

- Las reglas contienen la condición que define los valores permitidos para una columna o para un tipo de dato determinado (por compatibilidad con versiones anteriores)
- Para crear una nueva regla es necesario asignar:
  - Nombre de la regla
  - Expresión condicional, válida en una cláusula WHERE, que incluye una variable local que representa el valor a comprobar (@valor)
- Una regla se puede utilizar en tantos objetos como sea necesario.
- Se vinculan mediante Transact-SQL

```
CREATE RULE id_chk AS @id BETWEEN 0 and 10000
sp_bindrule id_chk, 'cust_sample.cust_id'
```

© JMA 2016. All rights reserved

## Tipos de Datos definidos por el usuario

- Los tipos de datos definidos por el usuario están basados en un tipo integrado y permiten establecer su longitud, si acepta nulos, qué regla restringe sus valores y cuál es su valor por defecto.
- Para crear un nuevo tipo es necesario asignar:
  - Nombre del nuevo tipo
  - Tipo de datos del sistema en el que se basa el nuevo tipo de datos
  - Longitud (si el tipo de datos lo permite)
  - Aceptación de valores NULL
- Opcionalmente se le pueden asignar:
  - Regla que definen los valores permitidos
  - Valor predeterminado
- Una vez creados solo se puede modificar la regla y el valor predeterminado.

© JMA 2016. All rights reserved

## Números de secuencia (2012)

- Una secuencia es un objeto enlazado a un esquema definido por el usuario que genera una secuencia de valores numéricos según la especificación con la que se creó la secuencia.
- A diferencia de los valores de columnas de identidad que se generan cuando se insertan filas, una aplicación puede obtener el número de secuencia siguiente sin insertar la fila llamando a la función NEXT VALUE FOR .
- Se asigna el número de secuencia cuando se llama a NEXT VALUE FOR aun cuando el número nunca se inserta en una tabla.
- La función NEXT VALUE FOR se puede utilizar como valor predeterminado para una columna en una definición de tabla.
- Se utiliza sp\_sequence\_get\_range para obtener un intervalo de varios números de secuencia enseguida.

© JMA 2016. All rights reserved

## Tipos de Tablas

- Definición:
  - Nombre y ubicación
  - Columnas
  - Claves y restricciones
- Tipos de Tablas
  - Tablas del sistema
  - Tablas de datos
  - Tablas temporales
    - Locales (#) y Globales (##)
  - Tablas anchas (1.024 → 30.000 columnas)
  - Tablas internas (variables)

© JMA 2016. All rights reserved



## Definición de las columnas

- Nombre de columna
- Tipo de datos
- Longitud de los datos
  - Sólo puede cambiar la longitud de los tipo de datos sin longitud fija
  - Para los tipos decimal y numéricos
    - Precisión: Muestra el número máximo de dígitos
    - Escala: Muestra el número máximo de dígitos decimales
- Aceptación de valores NULL (Restricción)
- Descripción
- Valor predeterminado
- Intercalación (sólo para columnas de texto)

© JMA 2016. All rights reserved

## Columnas especiales

- Columnas de numeración automática (Autonúmericas)
  - Identidad, una sola columna por tabla
  - Inicialización de identidad: valor de inicial de una columna de identidad
  - Incremento de identidad: valor en que se incrementa la columna de identidad
  - Vista sys.identity\_columns columna last\_value
- Columnas de identificadores exclusivos globales (GUID)
  - Es RowGuid: Muestra si SQL Server utiliza la columna como una columna ROWGUID. Este valor se puede establecer como Sí sólo para una columna de identidad
  - Valor predeterminado: newid()
- Columnas calculadas
  - Fórmula: Muestra la fórmula para una columna calculada
  - Ignora tipo y longitud

© JMA 2016. All rights reserved

## Restricciones

- Sirven para definir reglas relativas a los valores permitidos en las columnas
- Es el mecanismo utilizado para reforzar la integridad en las Bases de Datos que SQL Server exige automáticamente
- Las restricciones establecen los valores aceptados por las columnas: nulos, rangos de valores, duplicados, claves principales o ajenas.
- Se dividen en restricciones de tabla y de columna.
- Admite cinco clases de restricciones.
  - NOT NULL especifica que la columna no acepta valores NULL (Obligatorio).
  - Las restricciones CHECK exigen la integridad del dominio mediante la limitación de los valores que se pueden asignar a una columna.
  - Las restricciones UNIQUE exigen la unicidad de los valores de un conjunto de columnas (conlleva un índice).
  - Las restricciones PRIMARY KEY identifican la columna o el conjunto de columnas cuyos valores identifican de forma unívoca cada una de las filas de una tabla (NOT NULL + UNIQUE) (conlleva un índice).
  - Las restricciones FOREIGN KEY identifican las relaciones entre las tablas (REFERENCES).

© JMA 2016. All rights reserved

## Relaciones

- Se definen a nivel de tabla.
- Puede crear una relación entre las tablas en un diagrama de base de datos para mostrar cómo se vinculan las columnas de una tabla a las columnas de otra tabla.
- Permiten exigir la integridad referencial:
  - No se puede especificar un valor en la columna de clave externa de la tabla relacionada si ese valor no existe en la clave principal de la tabla relacionada (o NULL si la columna lo acepta).
  - No se puede eliminar una fila de una tabla de claves principales si existen filas que coinciden con ella en una tabla relacionada.
  - No se puede cambiar un valor de clave principal en la tabla de clave principal si esa fila tiene filas relacionadas.
- Pueden automatizar el proceso en cascada.

© JMA 2016. All rights reserved

## Seguimiento de cambios de datos

- A partir del SQL Server 2008 se proporciona dos características que realizan el seguimiento de los cambios en los datos de una base de datos: captura de datos modificados y seguimiento de cambios.
- Estas características permiten a las aplicaciones determinar los cambios de DML (operaciones de inserción, actualización y eliminación) que se realizaron en las tablas de usuario de una base de datos.
- La captura de datos modificados y el seguimiento de cambios pueden habilitarse en la misma base de datos; no se requiere ninguna consideración especial.

© JMA 2016. All rights reserved

## Diagramas de Bases de Datos

- Herramienta visual que permite diseñar y ver una base de datos completa o parcialmente
- Utiliza como repositorio la tabla de sistema: `dbo.sysdiagrams`.

© JMA 2016. All rights reserved

## Vistas

- Una vista es una tabla virtual cuyo contenido está definido por una consulta.
  - Combinar columnas de varias tablas de forma que parezcan una sola tabla.
  - Agregar información agrupada o calculada en lugar de presentar los detalles.
  - Unir información de varios resultados
  - Restringir el acceso del usuario a filas concretas de una tabla.
  - Restringir el acceso del usuario a columnas específicas.
- Tipos:
  - Vistas estándar
  - Vistas indexadas o materializada
  - Vistas con particiones o distribuidas
  - Vistas parametrizadas (funciones de tipo tabla)

© JMA 2016. All rights reserved

## Vistas

- Se puede modificar los datos mediante una vista.
  - Son actualizables (UPDATE, DELETE o INSERT) mientras la modificación afecte sólo a una de las tablas base.
  - Las columnas a modificar en la vista deben hacer referencia directa a los datos subyacentes de las columnas de la tabla (no puede ser resultado de una función de agregado o cálculo).
  - Deben aparecer todas las columnas requeridas que no dispongan de valor predeterminado.
  - Las columnas que se modifican no pueden verse afectadas por cláusulas GROUP BY, HAVING o DISTINCT.
  - Permite el uso de desencadenadores INSTEAD OF
  - La cláusula WITH CHECK OPTION exige que todas las instrucciones de modificación de datos ejecutadas en la vista se ajusten a los criterios especificados en la instrucción SELECT que define la vista.

© JMA 2016. All rights reserved

## Vistas

- Una vista distribuida combina los datos procedentes de un conjunto de tablas miembro en uno o más servidores, y hace que los datos parezcan proceder todos de una sola tabla que combinado con desencadenadores `INSTEAD OF` simulan tablas particionadas o distribuidas.
- La cláusula `WITH ENCRYPTION` convertirá el texto original en un formato ofuscado.
- Las vistas, las tablas o las funciones que participan en una vista creada con la cláusula `SCHEMABINDING` no se pueden quitar, a menos que se quite o cambie esa vista de forma que deje de tener un enlace de esquema.
- Además, las instrucciones `ALTER TABLE` sobre tablas que participan en vistas que tienen enlaces de esquemas provocarán un error si estas instrucciones afectan a la definición de la vista.

© JMA 2016. All rights reserved

## Índices

- Compuestos por una o varias columnas ordenadas ascendente o descendentemente
- Guardan clave indexación, clave en índice agrupado o marcador fila real y columnas incluidas.
- Se pueden crear sobre Tablas, Vistas y columnas calculadas.
- Los dos tipos:
  - Agrupado
  - No agrupado
- Un índice único garantiza que la columna indexada no contiene valores no nulos duplicados.
- Índices y Restricciones:
  - Las restricciones `PRIMARY KEY` crean índices agrupados de forma automática de tipo `UNIQUE`.
  - Las restricciones `UNIQUE` crean índices `UNIQUE` de forma automática.
  - Las restricciones `FOREIGN KEY` no crean índices, pero suelen ser convenientes.
- Los no agrupados permiten columnas `INCLUIDAS`

© JMA 2016. All rights reserved

# Índices

- El factor de relleno determina la densidad de ocupación de páginas por parte del índice.
- El número de índices afecta al rendimiento y ocupación de la base de datos.
- El Asistente para optimización de índices suministra recomendaciones en función a la monitorización del sistema.
- Limitaciones
  - Sólo el propietario de la tabla puede crear índices en la misma tabla.
  - Un índice agrupado por tabla.
  - 249 índices como máximo (incluidos los creados por restricciones PRIMARY KEY o UNIQUE).
  - 16 columnas como máximo.
  - El tamaño máximo de la clave de indexación es 900 bytes.
- Índices especiales:
  - Texto completo, XML y Espaciales

© JMA 2016. All rights reserved

# Estadísticas

- SQL Server (2005...) permite crear información estadística acerca de la distribución de valores en una columna.
- El optimizador de consultas utiliza esta información estadística para determinar el plan de consultas óptimo realizando una estimación del costo de usar un índice para evaluar la consulta.
- Se almacenan con la misma estructura que los índices, sustituyendo el localizador por un contador.
- Se pueden crear automáticamente o manualmente. Es necesario actualizarlas periódicamente.

© JMA 2016. All rights reserved

## Procedimientos almacenados

- Un procedimiento almacenado es un grupo de instrucciones Transact-SQL compiladas en un único plan de ejecución, se almacena en `sys.sql_modules` y en cache.
- Los procedimientos almacenados:
  - Permiten una programación modular.
  - Permiten una ejecución más rápida.
  - Pueden reducir el tráfico de red.
  - Pueden utilizarse como mecanismo de seguridad.
  - Un máximo de 2.100 parámetros.

© JMA 2016. All rights reserved

## Procedimientos almacenados

- Tipos:
  - Procedimientos almacenados del sistema (`sp_`)
  - Procedimientos almacenados definidos por el usuario
    - Transact-SQL y CLR
  - Procedimientos almacenados extendidos (`xp_`)
  - Procedimientos almacenados temporales
    - Locales (`#`) y Globales (`##`)
- Los procedimientos almacenados extendidos permiten crear rutinas externas propias, en un lenguaje de programación como C.
- Se pueden crear procedimientos almacenados temporales (agregando el prefijo `#` o `##`) que dejan de existir cuando se cierra SQL Server.
- Los procedimientos de inicio deben estar en la base de datos master y no pueden contener parámetros de entrada (INPUT) ni salida (OUTPUT). La ejecución de los procedimientos almacenados se inicia cuando se recupera la base de datos master en el inicio.
  - `sp_procoption 'procedure', 'startup', 'on'`

© JMA 2016. All rights reserved

## Procedimientos almacenados

- Devuelven datos de cuatro formas distintas:
  - Parámetros de salida: datos o cursores.
  - Códigos de retorno, que siempre son un valor entero.
  - Un conjunto de resultados por una o varias instrucciones SELECT.
  - Un cursor global al que se puede hacer referencia desde fuera del procedimiento almacenado.
- Cláusulas:
  - WITH RECOMPILE (en el procedimiento o en el EXEC)
  - WITH ENCRYPTION (convertirá el texto original en un formato ofuscado)
  - { EXEC | EXECUTE } AS { CALLER | SELF | OWNER | 'user\_name' }

© JMA 2016. All rights reserved

## Funciones definidas por el usuario

- Las funciones son subrutinas formadas por una o varias instrucciones Transact-SQL o CLR que se pueden utilizar para encapsular un código con el fin de utilizarlo de nuevo posteriormente.
- Son subrutinas que se utilizan para encapsular lógica que se ejecuta frecuentemente que no modifique la base de datos.
- Dos tipos de funciones: integradas y definidas por el usuario.
- Las funciones definidas por el usuario pueden tener parámetros de entrada y devuelven un único valor (dato o tabla interna)
- SQL Server admite cuatro tipos de funciones definidas por el usuario:
  - Funciones escalares
  - Funciones de valores de tabla en línea
  - Funciones de valores de tabla de múltiples instrucciones
  - Funciones de agregado (solo CLR)
- Todas las funciones son deterministas o no deterministas:
  - Las funciones deterministas siempre devuelven el mismo resultado cada vez que son llamadas con un conjunto específico de valores de entrada.
  - Las funciones no deterministas podrían devolver resultados diferentes cada vez que son llamadas con un conjunto específico de valores de entrada.

© JMA 2016. All rights reserved



## Desencadenadores DML

- Un desencadenador es un tipo especial de procedimiento almacenado que entra en vigor cuando se modifican datos en una tabla especificada utilizando una o más operaciones de modificación de datos: UPDATE (actualización), INSERT (inserción) o DELETE (eliminación).
- Los desencadenadores pueden exigir restricciones más complejas que las definidas con restricciones CHECK y permiten implementar las reglas de negocio.
- Pueden realizar cambios en cascada a través de tablas relacionadas de la base de datos.
- Los desencadenadores son automáticos: se activan inmediatamente después de que se efectúen modificaciones en los datos de la tabla, como una entrada manual o una acción de la aplicación.
  - Se pueden habilitar y deshabilitar.
  - Se pueden definir varios por tabla, cada uno para una o varias operaciones.
  - Se puede fijar cuál es el primero y el último en ejecutarse: `sp_settriggerorder` 'First|None|Last'

© JMA 2016. All rights reserved

## Desencadenadores DML

- Se pueden implementar en Transact-SQL o CLR
- Los desencadenadores también pueden evaluar el estado de una tabla antes y después de realizar una modificación de datos y actuar en función de la diferencia.
- Seudotablas:
  - INSERTED: nuevos valores
  - DELETED: valores antiguos
- Funciones:
  - UPDATE( column ) y COLUMNS\_UPDATED( )
- Momentos:
  - AFTER
  - INSTEAD OF

© JMA 2016. All rights reserved

## Desencadenadores DDL/Logon

- Pueden utilizarse para tareas administrativas como auditar y regular las operaciones de base de datos: CREATE, ALTER y DROP.
  - ámbito del servidor
  - ámbito de la base de datos
- La información acerca de un evento que activa un desencadenador DDL se captura mediante la función EVENTDATA (formato XML)

© JMA 2016. All rights reserved

## Sinónimos

- Un sinónimo es un nombre alternativo para un objeto (local o remoto) de ámbito de esquema.
- Proporciona una capa de abstracción que protege una aplicación cliente de cambios hechos en el nombre o la ubicación del objeto base.
- Se pueden proporcionar sinónimos a:
  - Tablas, Vistas, Procedimientos Almacenados y Funciones.

© JMA 2016. All rights reserved

---

# PROGRAMACIÓN CON T-SQL

---

© JMA 2016. All rights reserved

## Programación

---

- No todas las operaciones con la base de datos se pueden realizar directamente con las instrucciones SQL.
  - Para ello Transact-SQL incorpora instrucciones procedurales como la declaración de variables, flujo de control, excepciones, ... para poder programar las acciones.
  - Los programas se pueden almacenar en:
    - Ficheros: procesos por lotes
    - Base de datos: Procedimientos almacenados, funciones y desencadenadores.
- 

© JMA 2016. All rights reserved

## Procesos por lotes

- Los lotes T-SQL son colecciones de una o más declaraciones T-SQL enviadas al SQL Server como una unidad para análisis, optimización y ejecución.
- Los lotes se terminan con GO por defecto por lo que los ficheros pueden contener varios lotes
- Los lotes fijan los límites para el alcance variable.
- Las instrucciones DDL no pueden combinarse con otras en el mismo lote
- Los lotes se analizan para la sintaxis como una unidad y los errores de sintaxis hacen que todo el lote sea rechazado
- Los errores de tiempo de ejecución pueden permitir que el lote continúe después de la falla, de manera predeterminada
- Los lotes pueden contener código de manejo de errores.

© JMA 2016. All rights reserved

## Variables

- Las variables se declaran en el cuerpo de un proceso por lotes o un procedimiento con la instrucción DECLARE, y se les asignan valores con una instrucción SET o SELECT.
- Después de la declaración, todas las variables se inicializan como NULL, a menos que se proporcione un valor como parte de la declaración.
- Se declaran antes de ser usados en cualquier punto.  
`DECLARE @NOMBRE TIPO (LONGITUD) = VALOR,...`
- Ambito.:
  - Variables locales: @proceso
  - Variables globales: @@conexión
- Asignación:
  - `SET @NOMBRE = Valor`
  - `SELECT @NOMBRE = Valor [ | column FROM ...]`
- Mostrar valor:
  - `PRINT @NOMBRE`
  - `SELECT @NOMBRE`

© JMA 2016. All rights reserved

## IF...ELSE

- Impone condiciones en la ejecución de una instrucción de Transact-SQL.
- La instrucción Transact-SQL que sigue a una palabra clave IF y a su condición se ejecuta si la condición se cumple: la expresión booleana devuelve TRUE.
- La palabra clave opcional ELSE introduce otra instrucción Transact-SQL que se ejecuta cuando la condición IF no se cumple: la expresión booleana devuelve FALSE.
- Ejecutan una sola instrucción o un bloque de instrucciones marcado el inicio con BEGIN y el final con END.  

```
IF Boolean_expression
    { sql_statement | statement_block }
[ ELSE
    { sql_statement | statement_block } ]
```

© JMA 2016. All rights reserved

## WHILE

- Establece una condición para la ejecución repetida de una instrucción o bloque de instrucciones SQL.
- Las instrucciones se ejecutan repetidamente siempre que la condición especificada sea verdadera.
- Se puede controlar la ejecución de instrucciones en el bucle WHILE con las palabras clave BREAK y CONTINUE.  

```
WHILE Boolean_expression
    { sql_statement | statement_block }
```
- Ejecutan una sola instrucción o un bloque de instrucciones marcado el inicio con BEGIN y el final con END.
- Hay que vigilar la condición para no incurrir en un bloque infinito.
- BREAK produce la salida del bucle WHILE más interno y continua por la siguiente instrucción.
- CONTINUE salta a evaluar la condición que determina si sigue en el bucle o se sale.
- BREAK y CONTINUE deben estar dentro de un IF.

© JMA 2016. All rights reserved

## Saltos y Esperas

- Saltos:
  - NOMBRE\_ETIQUETA:
  - GOTO NOMBRE\_ETIQUETA
  - CONTINUE
  - BREAK
  - RETURN [ integer\_expression ]
- Espera:
  - WAITFOR DELAY 'período de tiempo que hay que esperar'
  - WAITFOR TIME 'hora a la que termina la espera'
  - WAITFOR (RECEIVE receive\_statement ) [ , TIMEOUT timeout ]

© JMA 2016. All rights reserved

## Tratamiento de excepciones

- Implementa un mecanismo de control de errores, se puede incluir un grupo de instrucciones en un bloque TRY y, si se produce un error, el control se transfiere a otro grupo de instrucciones que está incluido en un bloque CATCH.
 

```
BEGIN TRY
    { sql_statement | statement_block }
END TRY
BEGIN CATCH
    { sql_statement | statement_block }
END CATCH
```
- Funciones
  - @@ERROR: devuelve el número de error de la última instrucción ejecutada.
  - ERROR\_NUMBER() devuelve el número del error.
  - ERROR\_SEVERITY() devuelve la gravedad.
  - ERROR\_STATE() devuelve el número de estado del error.
  - ERROR\_PROCEDURE() devuelve el nombre del procedimiento almacenado o desencadenador donde se produjo el error.
  - ERROR\_LINE() devuelve el número de línea de la rutina que provocó el error.
  - ERROR\_MESSAGE() devuelve el texto completo del mensaje de error. Este texto incluye los valores suministrados para los parámetros reemplazables, como longitudes, nombres de objetos u horas.

© JMA 2016. All rights reserved

## Tratamiento de excepciones

- **RAISERROR**
  - Genera un mensaje de error e inicia el procesamiento de errores de la sesión.
  - Puede hacer referencia a un mensaje definido por el usuario almacenado en la vista de catálogo sys.messages o puede generar un mensaje dinámicamente.
  - El mensaje se devuelve como un mensaje de error de servidor a la aplicación que realiza la llamada o a un bloque CATCH asociado de una construcción TRY...CATCH.

```
RAISERROR ( { msg_id | msg_str | @local_variable }
           { ,severity ,state }
           [ ,argument [ ,...n ] ] )
           [ WITH option [ ,...n ] ]
```
- **THROW**
  - Genera una excepción y transfiere la ejecución a un bloque CATCH de una construcción TRY...CATCH

```
THROW [ { error_number | @local_variable },
       { message | @local_variable },
       { state | @local_variable } ]
```

© JMA 2016. All rights reserved

## Cursores

- Si una consulta produce una única fila se pueden asignar directamente los valores de las columnas a variables.
- Si produce un conjunto completo de resultados hay que procesarlo de fila en fila mediante un cursor.
- Declarar el cursor:
  - SQL 92 Syntax
 

```
DECLARE cursor_name [ INSENSITIVE ] [ SCROLL ] CURSOR FOR
select_statement [ FOR { READ ONLY | UPDATE [ OF column_name [ ,...n ] ] } ][:]
```
  - Transact-SQL Extended Syntax
 

```
DECLARE cursor_name CURSOR[ LOCAL | GLOBAL ][ FORWARD_ONLY |
SCROLL ][ STATIC | KEYSET | DYNAMIC | FAST_FORWARD ][ READ_ONLY
| SCROLL_LOCKS | OPTIMISTIC ][ TYPE_WARNING ]FOR
select_statement[ FOR UPDATE [ OF column_name [ ,...n ] ] ][:]
```

© JMA 2016. All rights reserved

## Cursores

- Abrir el cursor, se ejecuta la consulta:  
`OPEN { { [ GLOBAL ] cursor_name } | cursor_variable_name }`
- La función `@@CURSOR_ROWS` devuelve el número de filas certificadas que se encuentran en el último cursor abierto en la conexión
- Posicionar el cursor y traspasar los valores de las columnas de la fila actual a las variables:  
`FETCH [ [ NEXT | PRIOR | FIRST | LAST | ABSOLUTE { n | @nvar } |  
RELATIVE { n | @nvar } ]  
[ FROM ] { { [ GLOBAL ] cursor_name } | @cursor_variable_name }  
[ INTO @variable_name [ ,...n ] ]`
- La función `@@FETCH_STATUS` devuelve el estado de la última instrucción `FETCH` de cursor emitida para cualquier cursor abierto en ese momento por la conexión.

© JMA 2016. All rights reserved

## Cursores

- Cerrar el cursor y liberar la memoria:  
`CLOSE { { [ GLOBAL ] cursor_name } | cursor_variable_name }  
DEALLOCATE { { [ GLOBAL ] cursor_name } | @cursor_variable_name }`
- Algoritmo tipo:  
`DECLARE EICursor CURSOR FOR SELECT ...  
OPEN EICursor;  
FETCH NEXT FROM EICursor INTO ...;  
WHILE @@FETCH_STATUS = 0  
BEGIN  
... Tratamiento de los datos recuperados ...  
FETCH NEXT FROM EICursor INTO ...;  
END;  
CLOSE EICursor;  
DEALLOCATE EICursor;`

© JMA 2016. All rights reserved



---

# AUTOMATIZACIÓN DE TAREAS ADMINISTRATIVAS

---

© JMA 2016. All rights reserved

## Tareas habituales del administrador

---

- **Mantenimiento preventivo**
    - Operaciones rutinarias:
      - Copias de seguridad
      - Importación y exportación de datos
      - Replicaciones
      - Mantenimiento de datos (paso a históricos, consolidaciones, ...)
      - Mantenimiento de estructuras (reindexación, liberación de espacio, ...)
    - Supervisión del sistema y acciones correctivas:
      - Espacio en la base de datos y del registro
      - Rendimientos
  - **Mantenimiento reactivo**
    - Resolución de incidencias
- 

© JMA 2016. All rights reserved

## Agente de SQL Server

- SQL Server Agent es el servicio responsable de automatizar las tareas administrativas de SQL Server
- Se debe configurar el servicio SQL Server Agent:
  - para inicio automático
  - para reiniciarlos de forma automática si se detienen de forma inesperada
  - que la cuenta de inicio de sesión esté asignada a la función sysadmin
- La cuenta de inicio de sesión DEBERÍA:
  - Usar el modo de autenticación de Windows
  - Utilizar una cuenta de usuario de dominio de Windows
  - Asignar esta cuenta al grupo local Administradores

© JMA 2016. All rights reserved

## Base de Datos msdb

- La Base de Datos msdb se encarga de almacenar la información referente al servicio SQL Server Agent, y a la planificación de trabajos.
- Entre la información que almacena se encuentra la información relativa a:
  - Trabajos
  - Alertas
  - Operadores
  - Programaciones
  - Planes de Mantenimiento
  - Historial
- Para utilizar el Agente SQL Server, los usuarios deben ser miembros de una o más de las siguientes funciones fijas de base de datos:
  - SQLAgentUserRole
  - SQLAgentReaderRole
  - SQLAgentOperatorRole

© JMA 2016. All rights reserved

## Correo electrónico de base de datos

- El Correo electrónico de base de datos permite a las aplicaciones de base de datos enviar mensajes de correo electrónico a los usuarios.
- Los mensajes enviados pueden incluir resultados de consultas y archivos de cualquier recurso de la red.
- Utiliza el protocolo estándar SMTP
- Está organizado en Perfiles: Públicos y Privados
- Cada perfil puede incluir varias cuentas de conmutación por error.
- Se encuentra desactivado de forma predeterminada.
- La base de datos msdb es la base de datos host del Correo electrónico de base de datos.
- Para enviar correo debe pertenecer a la función de BD DatabaseMailUserRole en msdb.
- Características
  - Seguridad de los perfiles, Regulador del tamaño y Extensiones de archivo prohibidas.
  - Configuración integrada, Registro, Auditoría y Compatibilidad con HTML
  - Procedimiento sp\_send\_dbmail, vistas y procedimientos sysmail\_

© JMA 2016. All rights reserved

## Operadores

- Un operador es la personas responsable del mantenimiento, total o parcial, de una o varias bases de datos o instancias de SQL Server
- Métodos de envío de Notificaciones:
  - Correo electrónico
  - Localizador (por correo electrónico) (SMS, "pager" o busca)
  - Envío de red (net send)
- Se puede fijar los horarios de uso del localizador (Programación del localizador)
- Un operador a prueba de errores recibe la notificación cuando:
  - no se pueden enviar mensajes al localizador de ninguno de los operadores responsables de la alerta
  - no puede tener acceso a las tablas del sistema en la base de datos msdb

© JMA 2016. All rights reserved

## Trabajos

- Un trabajo es un conjunto de pasos secuenciales
- La definición de un paso puede usar:
  - Programas ejecutables y comandos del sistema operativo.
  - Instrucciones Transact-SQL, incluidos los procedimientos almacenados y los procedimientos almacenados extendidos.
  - Secuencias de comandos Microsoft ActiveX
  - Tareas de réplica.
  - Tareas de Analysis Services.
  - Paquetes de Integration Services.
- El flujo del trabajo se controla indicando el siguiente paso o deteniéndolo cuando se ejecute correctamente o cause un error, así como los reintentos después de dar error.
- El propietario del trabajo debe tener los permisos adecuados para ejecutar todos los pasos del trabajo.
- Se les pueden asignar Operadores a los que se notifican el resultado de las ejecuciones.
- Los trabajos se pueden organizar en categorías.

© JMA 2016. All rights reserved

## Programaciones

- Un trabajo se puede ejecutar manualmente, según su programación, en respuesta a una alerta o dentro de un plan de mantenimiento.
- Se puede programar trabajos para que se inicien automáticamente:
  - Al iniciarse SQL Server Agent.
  - A una hora específica (sólo una vez).
  - De forma periódica (diaria, semanal o mensualmente).
  - Cuando la CPU esté inactiva (configurable a nivel de servicio).
- Cada trabajo acepta varias programaciones (programación múltiple).
- Los trabajos se pueden desactivar, en cuyo caso no se iniciarán según la programación

© JMA 2016. All rights reserved

## Historial de trabajos

- Los trabajos son automáticos pero no deben quedar desatendidos
- Se recomienda el uso de listas de control (CHECKLIST) para su supervisión.
- Revisión del historial de un trabajo individual:
  - Resultado de cada paso del trabajo: correcto o erróneo
  - Duración de ejecución
  - Errores y mensajes
- Se puede supervisar la actividad actual mediante el Monitor de actividad de trabajo.
- Configuración del tamaño del historial de trabajos:
  - Conservar información acerca de cada trabajo
  - Sobrescribir historial cuando se alcance el tamaño máximo

© JMA 2016. All rights reserved

## Alertas

- Una alerta indica que ha ocurrido un suceso. Una alerta responde a un tipo de evento específico, pueden ser de los siguientes tipos :
  - Eventos de SQL Server
    - un error específico (Número de error)
    - un error de la gravedad específica (Nivel de gravedad)
    - en una base de datos específica
    - con una la cadena de texto especifica en el mensaje del suceso
  - Condiciones de rendimiento de SQL Server, dependiente de un objeto, contador e instancia.
  - Eventos de WMI, debe definir el Espacio de nombres y la Consulta
- En respuesta a la alerta se puede:
  - Ejecutar un trabajo
  - Mandar una notificación a uno o varios operadores

© JMA 2016. All rights reserved

## Planes de mantenimiento

- Permiten configurar y agrupar las principales tareas de mantenimiento necesarias para garantizar el buen funcionamiento de la base de datos.
- El plan puede contar con una o varias de las siguientes tareas:
  - Reorganizar los datos de las páginas de datos y de índices mediante una nueva generación de los índices con un nuevo factor de relleno
  - Comprimir archivos de datos mediante la eliminación de las páginas de la base de datos que estén vacías
  - Actualizar las estadísticas de los índices para el optimizador de consultas.
  - Realizar comprobaciones de coherencia interna de los datos y de las páginas de datos de la base de datos
  - Realizar copias de seguridad de la base de datos y de los archivos de registro de transacciones
  - Configurar el trasvase de registros (sincronizar bases de datos)

© JMA 2016. All rights reserved

## Herramientas del administrador

- Vistas del sistema
- Procedimientos y funciones del sistema
- Informes del SSMS
- Utilidades del sistema operativo
- Directivas
- DBCC: Data Base Command Console

© JMA 2016. All rights reserved

## Administración basada en directivas

- La administración basada en directivas es un sistema para administrar una o varias instancias de SQL Server similar a la existente en el Windows.
- Las Facetas son un conjunto de propiedades lógicas que modelan el comportamiento o las características de ciertos aspectos administrados. Las facetas vienen predefinidos.
- Una Condición es una expresión booleana que especifica un conjunto de estados permitidos para una o mas facetas.
- Una Directiva de cuando, como y sobre que se aplica una condición.
- Modos de evaluación
  - A petición. Este modo evalúa la directiva cuando lo especifica el usuario directamente.
  - Al cambiar: impedir. Este modo automatizado utiliza desencadenadores DDL para evitar las infracciones de las directivas.
  - Al cambiar: sólo registrar. Este modo automatizado utiliza la notificación de eventos para evaluar una directiva cuando se realiza un cambio relevante.
  - Al programar. Este modo automatizado utiliza un trabajo del Agente SQL Server para evaluar una directiva periódicamente.

© JMA 2016. All rights reserved

## DBCC: Instrucciones de mantenimiento

| Comando        | Descripción                                                                                                                                                                           |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DBREINDEX      | Regenera uno o varios índices de una tabla de la base de datos especificada.                                                                                                          |
| DBREPAIR       | Quita una base de datos dañada (obsoleta). utilizar DROP DATABASE                                                                                                                     |
| INDEXDEFRAG    | Desfragmenta los índices agrupados y secundarios de la tabla o la vista especificada.                                                                                                 |
| SHRINKDATABASE | Reduce el tamaño de los archivos de datos de la base de datos especificada.                                                                                                           |
| SHRINKFILE     | Reduce el tamaño del archivo de datos o de registro especificado de la base de datos relacionada.                                                                                     |
| CLEANTABLE     | Recupera espacio correspondiente a columnas de longitud variable y a columnas de texto que se han quitado.                                                                            |
| UPDATEUSAGE    | Informa y corrige las imprecisiones de la tabla sysindexes, que pueden provocar que el procedimiento almacenado del sistema sp_spaceused informe incorrectamente del uso del espacio. |

© JMA 2016. All rights reserved

## DBCC: Instrucciones de validación

| Comando          | Descripción                                                                                                                                 |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------|
| CHECKALLOC       | Comprueba la coherencia de las estructuras de asignación de espacio en disco para una base de datos determinada.                            |
| CHECKCATALOG     | Comprueba la coherencia de las tablas del sistema y de éstas entre sí, en la base de datos especificada.                                    |
| CHECKCONSTRAINTS | Comprueba la integridad de una restricción especificada o de todas las restricciones de una tabla determinada.                              |
| CHECKDB          | Comprueba la asignación y la integridad estructural de todos los objetos de la base de datos especificada.                                  |
| CHECKFILEGROUP   | Comprueba la asignación y la integridad estructural de todas las tablas (de la base de datos actual) del grupo de archivos especificado.    |
| CHECKIDENT       | Comprueba el valor de identidad actual de la tabla especificada y, si fuera necesario, corrige el valor de identidad.                       |
| CHECKTABLE       | Comprueba la integridad de las páginas de datos, de índices y de columnas text, ntext e image de la tabla especificada o la vista indexada. |
| NEWALLOC         | Idéntica a DBCC CHECKALLOC                                                                                                                  |

© JMA 2016. All rights reserved

## SEGURIDAD LÓGICA

© JMA 2016. All rights reserved



## Seguridad del entorno

- Seguridad física
  - Limitar de forma estricta el acceso al servidor físico y a los componentes de hardware.
- Seguridad del sistema operativo
  - Service Packs, actualizaciones y firewalls
  - Seguridad de los archivos del sistema operativo de SQL Server
- Seguridad de aplicaciones
  - Utilización de aplicaciones cliente seguras
  - Antivirus

© JMA 2016. All rights reserved

## Conceptos previos

- Autenticación
  - Proceso mediante el cual se obtiene y verifica la identidad del que accede.
  - En SQL Server (al igual que en Windows) se denomina inicio de sesión o login.
- Autorización
  - Proceso por el cual se autoriza o deniega a la identidad autenticada el acceso a determinados recursos (Permisos).
- Impersonalización:
  - Proceso mediante el cual la identidad autenticada asume una identidad estándar sobre la que se ha realizado la autorización.
  - Simplifica el proceso de autorización pero impide la trazabilidad de las acciones.

© JMA 2016. All rights reserved

## Modos de Autenticación

- Hay dos modos posibles:
  - Modo de autenticación de Windows
    - Habilita la autenticación de Windows y deshabilita la autenticación de SQL Server.
  - Modo mixto
    - El modo mixto habilita tanto la autenticación de Windows como la de SQL Server.
- La autenticación de Windows está disponible siempre y no se puede deshabilitar.

© JMA 2016. All rights reserved

## Autenticación de Windows

- Cuando un usuario se conecta a través de una cuenta de usuario de Microsoft Windows, SQL Server valida el nombre de cuenta y la contraseña con el token de la entidad de seguridad de Windows del sistema operativo.
- Esto significa que Windows confirma la identidad del usuario y SQL Server no pide la contraseña y no realiza la validación de identidad.
- La autenticación de Windows es el modo de autenticación predeterminado y es mucho más seguro que la autenticación de SQL Server.
- La autenticación de Windows usa el protocolo de seguridad de Kerberos, proporciona la aplicación de directivas de contraseñas en cuanto a la validación de la complejidad de las contraseñas seguras, ofrece compatibilidad para el bloqueo de cuentas y admite la expiración de las contraseñas.
- La autenticación de Windows se denomina a veces autenticación integrada porque se integra la autenticación del SQL Server con la Windows.

© JMA 2016. All rights reserved

## Autenticación de SQL Server

- El SQL Server asume la responsabilidad de la autenticación.
- El nombre de usuario y la contraseña se crean utilizando SQL Server y se almacenan en SQL Server.
- Los usuarios que se conectan utilizando la autenticación de SQL Server deben proporcionar sus credenciales (inicio de sesión y contraseña) cada vez que se conectan.
- Al utilizar la autenticación de SQL Server, debe establecer contraseñas seguras para todas las cuentas de SQL Server.
- Permite que SQL Server admita entornos con sistemas operativos mixtos, en los que un dominio de Windows no autentica a todos los usuarios.
- Es menos robusta y completa que la autenticación Windows.

© JMA 2016. All rights reserved

## Implementación del modo de autenticación

- Configurar el modo de autenticación en SQL Server con el SSMS
- Parar y arrancar el servicio MSSQLServer
- Crear grupos y usuarios en Windows
- Crear inicios de sesión asociados a los grupos y usuarios de Windows
- Crear inicios de sesión para usuarios que se conecten con conexiones sin confianza

© JMA 2016. All rights reserved

## Entidades de seguridad

- Windows
  - Inicio de sesión local o del dominio de Windows
  - Grupo local o del dominio de Windows
- SQL Server
  - Inicio de sesión de SQL Server
  - Inicios de sesión de SQL Server basados en certificados y claves asimétricas
    - Permiten a servicios y aplicaciones conectarse directamente a SQL Server
- El inicio de sesión sa de SQL Server es una entidad de seguridad del servidor
  - Representa al SuperAdministrador.
  - Se crea de forma predeterminada cuando se instala una instancia.
  - No se puede eliminar al ser propietario de las bases de datos de sistema.
  - Se debe desactivar para evitar problemas de seguridad.

© JMA 2016. All rights reserved

## Inicios de Sesión

- Autenticación de Windows
  - Nombre de inicio de sesión: Nombre de inicio de sesión o grupo de Windows
- Autenticación de SQL Server
  - Nombre de inicio de sesión
  - Contraseña
  - Exigir directivas de contraseñas
  - Exigir expiración de contraseña
  - Cambiar la contraseña en el siguiente inicio de sesión
- Tipo: Concesión o denegación
- Estado: Habilitado, deshabilitado, bloqueado
- Opcionalmente se puede establecer:
  - Credenciales
  - La base de datos a la que se conecta por defecto
  - El idioma

© JMA 2016. All rights reserved

## Credenciales

- Una credencial es un registro que contiene la información de autenticación (credenciales) necesaria para conectarse a un recurso situado fuera de SQL Server.
- Esta información la utiliza SQL Server internamente.
- La mayoría de las credenciales incluyen un nombre de usuario y una contraseña de Windows.
- La información almacenada en una credencial permite al usuario que se ha conectado a SQL Server mediante la autenticación de SQL Server obtener acceso a recursos situados fuera de la instancia del servidor.
- Se puede asignar una única credencial a varios inicios de sesión de SQL Server. Sin embargo, un inicio de sesión de SQL Server sólo se puede asignar a una credencial.

© JMA 2016. All rights reserved

## Usuarios

- El usuario es una entidad de seguridad de la base de datos.
- El usuario esta asociado a un inicio de sesión.
- Pertenece al proceso de autorización, concede o deniega el permiso de conexión a la base de datos.
- Usuarios Predefinidos (no se puede borrar):
  - **dbo**: propietario de la base de datos., cuenta con todos los permisos.
  - **guest**: invitados, asociado a todos los inicios de sesión que no tienen su correspondiente usuario. Si se le deniega el permiso de conexión la base de datos no contara con invitados.
  - **INFORMATION\_SCHEMA** : Propietario de las vistas de esquema de información.
  - **sys**: propietario de las vistas y funciones del sistema.

© JMA 2016. All rights reserved

## Funciones (Roles o Grupos)

- Son las entidades de seguridad que agrupan a otras entidades de seguridad y otorgan un juego de permisos común.
- Niveles:
  - Servidor: Fijan los permisos sobre los objetos del servidor. Asociados a inicios de sesión.
  - Base de datos: Fijan los permisos sobre los objetos de la base de datos. Asociados a usuarios y otras funciones.
- Tipos:
  - Fijos: Los permisos están preestablecidos y no se pueden cambiar, solo se pueden añadir o quitar inicios de sesión o usuarios.
  - Públicos: Todos los inicios de sesión y usuarios están asociados, solo se definen los permisos.
  - Definidos(Base de datos): Se definen los permisos y se asocian a usuarios y otras funciones.
  - Aplicación (Base de datos): Se definen los permisos pero no están asociados a nadie.

© JMA 2016. All rights reserved

## Roles fijos de nivel de servidor

| Función       | Descripción                                                                                                            |
|---------------|------------------------------------------------------------------------------------------------------------------------|
| sysadmin      | pueden realizar cualquier actividad en el servidor.                                                                    |
| serveradmin   | pueden cambiar las opciones de configuración en el servidor y apagarlo.                                                |
| securityadmin | administran los inicios de sesión y pueden administrar los permisos de servidor y de base de datos (si tienen acceso). |
| processadmin  | pueden finalizar los procesos que se ejecuten en una instancia de SQL Server.                                          |
| setupadmin    | pueden agregar y quitar servidores vinculados.                                                                         |
| bulkadmin     | pueden ejecutar la instrucción BULK INSERT.                                                                            |
| diskadmin     | se usa para administrar archivos de disco.                                                                             |
| dbcreator     | pueden crear, modificar, quitar y restaurar cualquier base de datos.                                                   |
| public        | cada inicio de sesión de SQL Server pertenece al rol.                                                                  |

© JMA 2016. All rights reserved

## Roles fijos de base de datos

| Función           | Descripción                                                                                                           |
|-------------------|-----------------------------------------------------------------------------------------------------------------------|
| db_owner          | Tiene todos los permisos de la base de datos.                                                                         |
| db_accessadmin    | Puede agregar o eliminar usuarios.                                                                                    |
| db_securityadmin  | Puede administrar todos los permisos, todas las propiedades de objetos, las funciones y las pertenencias a funciones. |
| db_ddladmin       | Puede emitir instrucciones ALL DDL, pero no GRANT, REVOKE ni DENY.                                                    |
| db_backupoperator | Puede emitir instrucciones DBCC, CHECKPOINT y BACKUP.                                                                 |
| db_datareader     | Puede seleccionar todos los datos de cualquier tabla de usuario de la base de datos.                                  |
| db_datawriter     | Puede modificar todos los datos de cualquier tabla de usuario de la base de datos.                                    |
| db_denydatareader | No puede seleccionar ningún dato de ninguna tabla de usuario de la base de datos.                                     |
| db_denydatawriter | No puede modificar ningún dato de ninguna tabla de usuario de la base de datos.                                       |

© JMA 2016. All rights reserved

## Asegurables

- Los asegurables son los recursos cuyo acceso es regulado por el sistema de autorización.
- Ámbitos y elementos asegurables:
  - Servidor:
    - Extremo, Inicio de sesión, Base de datos
  - Base de datos:
    - Usuario, Función, Función de aplicación, Ensamblado, Tipo del mensaje, Ruta, Servicio, Enlace de servicio remoto, Catálogo de texto, Certificado, Clave asimétrica, Clave simétrica, Contrato, Esquema
  - Esquema:
    - Tipo, Colección de esquemas XML, Agregado, Restricción, Función, Procedimiento, Cola, Estadística, Sinónimo, Tabla, Vista

© JMA 2016. All rights reserved

## Esquemas

- Los esquemas ya no son equivalentes a los usuarios de la base de datos; cada esquema ahora es un espacio de nombres distinto que existe de forma independientemente del usuario de base de datos que lo creó.
- Es decir, un esquema simplemente es un contenedor de objetos.
- Todos los objetos deben pertenecer a su correspondiente esquema.
- Los esquemas cuentan con un propietario (usuario o función) que dispone de todos los permisos sobre los objetos del esquema.
- Cualquier usuario puede ser propietario de un esquema, y esta propiedad es transferible.

© JMA 2016. All rights reserved

## Permisos

- Especifican qué usuarios pueden acceder, y de qué forma, a qué objetos
- También determinan qué acciones pueden realizar los usuarios
- Tres tipos:
  - Permisos de sentencia o instrucción: Son permisos de sistema
  - Permisos sobre objetos: Incluso columnas en el caso de SELECT, UPDATE o REFERENCES
    - SELECT, INSERT, UPDATE y DELETE, que pueden aplicarse a toda la tabla o vista.
    - SELECT y UPDATE, que pueden aplicarse selectivamente a columnas individuales de una tabla o vista.
    - EXECUTE, que afecta a funciones (escalares) y procedimientos almacenados.
    - SELECT, que se puede aplicar a funciones (no escalares) definidas por el usuario.
    - INSERT y DELETE, que afectan a toda una fila y, por ello, sólo pueden aplicarse a la tabla o vista, y no a columnas individuales.
    - DRI, que concede, revoca o deniega permisos declarativos de integridad referencial para este objeto.
  - Permisos implícitos: controlan las actividades que sólo pueden realizar los miembros de las funciones de sistema predefinidas o los propietarios de objetos de base de datos.

© JMA 2016. All rights reserved



## Permisos de instrucción

- **CONTROL:** Todos los permisos salvo el cambio de propietario.
- **ALTER:** Confiere la posibilidad de cambiar las propiedades, excepto al propietario.
- **ALTER ANY <Server Securable>**, donde Server Securable puede ser cualquier asegurable de servidor.
- **ALTER ANY <Database Securable>**, donde Database Securable puede ser cualquier asegurable a nivel de la base de datos.
- **ALTER ANY SCHEMA:** confiere la posibilidad de crear, modificar o quitar cualquier esquema en la base de datos.
- **TAKE OWNERSHIP:** Permite al receptor del permiso tomar propiedad del asegurable para el que se concede este permiso.
- **IMPERSONATE <Login>**: Permite al receptor suplantar el inicio de sesión.
- **IMPERSONATE <User>**: Permite al receptor suplantar al usuario.
- **CREATE <Server Securable>**: Confiere al receptor la posibilidad de crear el asegurable de servidor.
- **CREATE <Database Securable>**: Confiere al receptor la posibilidad de crear el asegurable de base de datos.
- **CREATE <Schema-contained Securable>**: Confiere la posibilidad de crear el asegurable contenido en el esquema. No obstante, para crear el asegurable en un esquema concreto se requiere el permiso ALTER en el esquema.
- **VIEW DEFINITION:** Permite al receptor obtener acceso a los metadatos.
- **BACKUP y DUMP** (son sinónimos): Copias de seguridad
- **RESTORE y LOAD** (son sinónimos): Restaurar copias de seguridad.

© JMA 2016. All rights reserved

## Permisos sobre objetos

| Permiso         | Descripción                                                           |
|-----------------|-----------------------------------------------------------------------|
| SELECT          | Consultar datos y ejecutar funciones de tipo tabla                    |
| UPDATE          | Modificar datos                                                       |
| REFERENCES      | Seguir una restricción FOREIGN KEY que hace referencia a otras tablas |
| INSERT          | Insertar datos                                                        |
| DELETE          | Borrar datos                                                          |
| EXECUTE         | Ejecutar procedimientos y funciones escalares.<br>Sinónimos           |
| RECEIVE         | Colas de Service Broker                                               |
| VIEW DEFINITION | Ver la definición                                                     |
| ALTER           | Modificar la definición o el código                                   |
| TAKE OWNERSHIP  | Asumir la propiedad                                                   |
| CONTROL         | Todos los permisos anteriores salvo el TAKE OWNERSHIP                 |

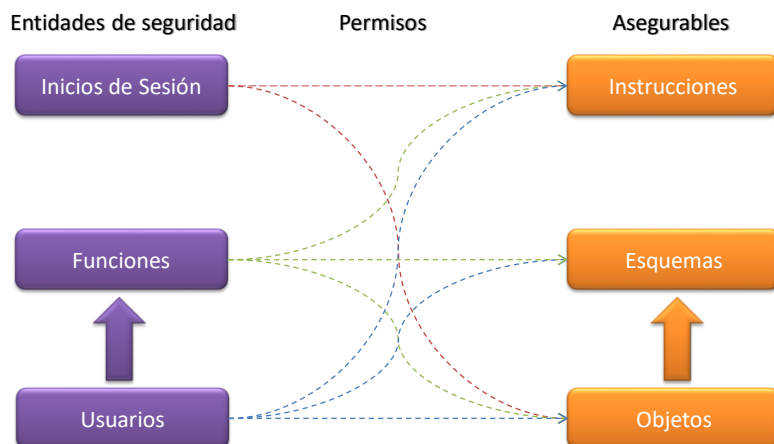
© JMA 2016. All rights reserved

## Cambiar permisos

- Pueden cambiar permisos el propietario de los objetos, los miembros de sysadmin, db\_owner y db\_securityadmin
- Cambio de permisos:
  - GRANT: Concede un permiso
  - WITH GRANT: Cadenas de propiedad, conceder permiso de fijar permiso.
  - REVOKE: Quita la definición de un permiso
  - DENY: Deniega absolutamente el permiso
- Evaluación de permisos:
  - Se reúnen todas las definiciones de permisos.
  - Si hay definida a menos un DENY, se deniega.
  - Si hay algún GRANT, se permite.
  - Por defecto, se deniega.

© JMA 2016. All rights reserved

## Definición de permisos



© JMA 2016. All rights reserved

## Definición de permisos

Objetivo: Definir los permisos el número mínimo de veces.

1. Función public (todos los permisos por defecto)
  1. Instrucciones
  2. Esquemas (todos los permisos comunes para sus objetos)
  3. Objetos (permisos específicos)
2. Funciones definidas (todos los permisos comunes para su miembros)
  1. Instrucciones
  2. Esquemas
  3. Objetos
3. Usuario (permisos específicos, aunque es preferible crear nuevas funciones y no definir permisos de usuario)
  1. Instrucciones
  2. Esquemas
  3. Objetos

© JMA 2016. All rights reserved

## Funciones de aplicación

- Una función de aplicación es una entidad de seguridad de base de datos que permite que una aplicación se ejecute con sus propios permisos de usuario.
- Permiten que los usuarios cuenten con juegos de permisos diferentes dependiendo de si se conectan directamente a la base de datos (permisos mas restrictivos) o lo hacen a través de una aplicación (controla la integridad de la información).
- Las funciones de aplicación no contienen miembros y están inactivas de manera predeterminada.
- Las funciones de aplicación se activan desde las aplicaciones empleando el procedimiento **sp\_setapprole** mediante una contraseña.
- Si el nombre y la contraseña de la función de aplicación son válidos, la función de aplicación se habilita y la conexión pierde los permisos del usuario y asume los permisos de la función de aplicación.
- Los permisos adquiridos durante la función de aplicación se mantienen mientras dura la conexión.
- Debido a que las funciones de aplicación son una entidad de seguridad de la base de datos, sólo pueden obtener acceso a otras bases de datos mediante los permisos que se conceden en dichas bases de datos a guest.

© JMA 2016. All rights reserved

## Complementar la seguridad

- Mediante vistas
  - Los permisos permiten la granularidad a nivel de objetos y columnas.
  - Las vistas permiten la granularidad a nivel de filas y columnas.
- Mediante procedimientos almacenados y desencadenadores
  - Cuando un usuario tiene privilegio para ejecutar un procedimiento almacenado, hereda los privilegios del creador del procedimiento desde el punto de acceso a la información.
  - Permiten establecer e imponer restricciones, validaciones y reglas de negocios.
- Mediante funciones de aplicación
  - Permiten establecer permisos específicos para las aplicaciones

© JMA 2016. All rights reserved

## Bases de datos parcialmente independientes

- Una base de datos totalmente independiente incluye todos los metadatos y opciones de configuración necesarios para definirla y no tiene dependencias de configuración en la instancia de Motor de base de datos de SQL Server donde esté instalada.
- La característica de base de datos independiente solo está disponible actualmente en un estado contenido de forma parcial.
- Una base de datos parcialmente independiente es una base de datos independiente que permite el uso de características no contenidas, funciones que residen fuera de la base de datos .

© JMA 2016. All rights reserved

## Usuario contenido

- Las entidades de usuario que residen completamente dentro de la base de datos se consideran independientes.
- Hay dos tipos de usuarios para las bases de datos independientes.
  - Usuario de la base de datos independiente con contraseña
    - La base de datos autentica a los usuarios de la base de datos independiente con contraseñas.
  - Entidades de seguridad de Windows
    - Los usuarios de Windows autorizados y los miembros de grupos de Windows autorizados pueden conectarse directamente a la base de datos y no requieren inicios de sesión en la base de datos maestra. La base de datos confía en la autenticación de Windows.
- La base de datos parcialmente independientes delega el control sobre el acceso a la instancia de SQL Server en los propietarios de la base de datos.

© JMA 2016. All rights reserved

## Características

- Ventajas:
  - Facilita el movimiento de bases de datos entre instancias.
  - Simplifica la configuración de Grupos de disponibilidad AlwaysOn.
  - Posibilita el desarrollo inicial de la base de datos independiente de la implementación.
  - Permite a cada propietario de la base de datos tener un mayor control sobre ella.
- Limitaciones:
  - No pueden utilizar la replicación, la captura de datos modificados ni el seguimiento de cambios.
  - Objetos enlazados a esquema que dependen de funciones integradas con cambios de intercalación.

© JMA 2016. All rights reserved

## Vincular servidores

- Una configuración con servidores vinculados permite a SQL Server ejecutar comandos en orígenes de datos OLE DB situados en servidores remotos.
- Los servidores vinculados ofrecen las siguientes ventajas:
  - Acceso al servidor remoto.
  - Capacidad de ejecutar consultas distribuidas, actualizaciones, comandos y transacciones en orígenes de datos heterogéneos en toda la organización.
  - Capacidad de tratar diferentes orígenes de datos de manera similar.
- Permiten definir la seguridad mediante:
  - Mapeo de identidades
  - Propagación de identidades
  - Impersonalización
  - Sin contexto

© JMA 2016. All rights reserved

## SQL Server Audit

- La auditoría de una instancia de SQL Server o de una base de datos individual implica el seguimiento y registro de los eventos que se producen en Motor de base de datos.
- Puede registrar grupos de acciones de auditoría en el servidor por instancia, así como grupos de acciones o acciones de auditoría en la base de datos y por base de datos.
- Las auditorías de nivel de base de datos se limitan a las ediciones Enterprise.
- El evento de auditoría se producirá cada vez que se encuentre la acción auditable.
- Se pueden crear tantas auditorías como sean necesarias, compuestas por tantas especificaciones de auditorías como eventos a auditar.
- Tanto las auditorías como las especificaciones se pueden habilitar y deshabilitar de forma individual.

© JMA 2016. All rights reserved

## Auditoría

- Destino de auditoría:
  - El registro de aplicación del Windows.
  - El registro de seguridad de Windows.
  - Archivos binario: Ruta del archivo, Límite máximo del archivo de auditoría: Sustitución incremental o Número máximo de archivos (ilimitado), Tamaño máximo del archivo (ilimitado), Reservar espacio en disco
- Si hay un error de registro de auditoría:
  - **Continuar:** Las operaciones de SQL Server continúan y los registros de auditoría no se conservan.
  - **Apagar el servidor:** Fuerza el apagado del servidor cuando la instancia de servidor que escribe en el destino no puede escribir datos en el destino de la auditoría.
  - **Error en la operación:** En los casos en que no puede escribir en el registro de auditoría, las acciones de base de datos que generan lo eventos auditados producen un error

© JMA 2016. All rights reserved

## Especificación de auditoría

- Las auditorías pueden tener las siguientes categorías de acciones:
  - Nivel de servidor. Estas acciones incluyen las operaciones del servidor, como cambios de administración y operaciones de inicio y cierre de sesión.
  - Nivel de base de datos. Estas acciones comprenden operaciones de lenguaje de manipulación de datos (DML) y de lenguaje de definición de datos (DDL).
  - Nivel de auditoría. Son las acciones relacionadas con el proceso de auditoría.
- Tipo de acción de auditoría:
  - Grupos de eventos
  - Eventos individuales
    - Esquema y Nombre de objeto, Nombre de la entidad

© JMA 2016. All rights reserved

## Ver registros de auditoria

- Con el Visor del archivo de registros, todos los destinos.
- Con el Visor de eventos de Windows, para destinos de registro de aplicación o seguridad de Windows.
- Se puede leer la información en el archivo de auditoría directamente desde Transact-SQL con la función `sys.fn_get_audit_file` (`file_pattern`, `initial_file_name`, `audit_record_offset`) donde:
  - `<path>\*`: recopila todos los archivos de auditoría de la ubicación especificada.
  - `<path>\LoginsAudit_{GUID}`: recopila todos los archivos de auditoría con el par de nombre y GUID especificado.
  - `<path>\LoginsAudit_{GUID}_00_29384.sqlaudit`: recopila un solo archivo de auditoría determinado.

```
SELECT * FROM sys.fn_get_audit_file(  
    '\\serverName\Audit\demo.sqlaudit', default, default);
```

© JMA 2016. All rights reserved

Lenguaje de Control de Datos

## DCL

© JMA 2016. All rights reserved



## GRANT

- Concede permisos sobre un elemento protegible a una entidad de seguridad.
- El concepto general es GRANT <un permiso> ON <un objeto> TO <un usuario, inicio de sesión o grupo>.

```
GRANT { ALL [ PRIVILEGES ] }
      | permission [ ( column [ ,...n ] ) ] [ ,...n ]
      [ ON [ class :: ] securable ] TO principal [ ,...n ]
      [ WITH GRANT OPTION ] [ AS principal ]
```

© JMA 2016. All rights reserved

## DENY

- Deniega un permiso a una entidad de seguridad.
- Evita que la entidad de seguridad herede permisos por su pertenencia a grupos o roles.
- DENY tiene prioridad sobre todos los permisos, aunque DENY no se aplica a los propietarios de objetos o miembros del rol fijo de servidor sysadmin.

```
DENY { ALL [ PRIVILEGES ] }
      | <permission> [ ( column [ ,...n ] ) ] [ ,...n ]
      [ ON [ <class> :: ] securable ]
      TO principal [ ,...n ]
      [ CASCADE ] [ AS principal ]
```

© JMA 2016. All rights reserved

## REVOKE

- Quita la concesión o denegación previamente establecida.
- Revocar una concesión no equivale a una denegación, dado que una entidad de seguridad puede establecer concesiones a múltiple niveles, se conserva el permiso mientras se mantenga al menos una concesión y no haya ninguna denegación.

```

REVOKE [ GRANT OPTION FOR ]
{
  [ ALL [ PRIVILEGES ] ]
  |
  permission [ ( column [ ,...n ] ) ] [ ,...n ]
}
[ ON [ class :: ] securable ]
{ TO | FROM } principal [ ,...n ]
[ CASCADE ] [ AS principal ]

```

© JMA 2016. All rights reserved

Copias de Seguridad y restauración

## SEGURIDAD FÍSICA

© JMA 2016. All rights reserved

## Cuestiones de disponibilidad

- ¿Cuáles son sus requisitos de disponibilidad? ¿Qué parte de cada día debe estar en línea la base de datos?
- ¿Qué costo financiero supone para el negocio el tiempo de inactividad?
- En caso de error de uno de los medios, por ejemplo, una unidad de disco, ¿cuál sería el tiempo de inactividad aceptable?
- En caso de desastre, como la pérdida de un servidor en un incendio, ¿cuál sería el tiempo de inactividad aceptable?
- ¿Qué importancia tiene no perder nunca cambios?
- ¿Con qué facilidad podrían volverse a crear los datos perdidos?
- ¿La organización dispone de administradores de sistema o de bases de datos?
- ¿Quién será el responsable de realizar las operaciones de copia de seguridad y recuperación y cómo se les formará?

© JMA 2016. All rights reserved

## Cuestiones técnicas

- ¿Qué tamaño tiene cada base de datos?
- ¿Con qué frecuencia se modifican los datos de cada base de datos?
- ¿Se modifican algunas tablas más a menudo que otras?
- ¿Cuáles son los periodos de producción de la base de datos más importantes?
- ¿Cuándo se utiliza mucho la base de datos y se obtienen como resultado inserciones y actualizaciones frecuentes?
- ¿Podría suponer un problema que el registro de transacciones consuma espacio debido a una gran actividad de actualización?
- ¿La base de datos se somete periódicamente a cargas masivas de datos?
- ¿La base de datos está sujeta a errores de aplicación o actualizaciones arriesgadas que podrían no detectarse inmediatamente?
- ¿El servidor de base de datos forma parte de un clúster de conmutación por error de Windows para aumentar la disponibilidad?
- ¿La base de datos está en un entorno multiservidor con administración centralizada?

© JMA 2016. All rights reserved

## Plan de contingencias

- Errores Lógicos
  - errores de aplicación
  - actualizaciones arriesgadas que no detectan inmediatamente
  - errores de usuarios
- Errores Físicos
  - Cuantificables
    - Caducidad de los elementos físicos
  - Catastróficos
    - Incendios, inundaciones, robos, ...
- Cualquier solución requiere que se hayan hecho copias de seguridad regularmente

© JMA 2016. All rights reserved

## Objetivos

- RPO (Recovery Point Objective): hace referencia al volumen de datos en riesgo de pérdida que la organización considera tolerable. ¿Las transacciones de cuánto tiempo estamos dispuestos a perder, o a tener que reintroducir al sistema?
- RTO (Recovery Time Objective): Expresa el tiempo durante el cual una organización pueda tolerar la falta de funcionamiento de sus aplicaciones y la caída de nivel de servicio asociada, sin afectar a la continuidad del negocio
- Objetivos:
  - Minimizar la pérdida de datos
  - Recuperar los datos perdidos
  - Restaurar con un coste mínimo para el sistema de producción

© JMA 2016. All rights reserved

## Planificación

### Planes de contingencia

- Tipo de incidencia
- Tiempos medios:
  - TMF (entre fallos)
  - TMR (de recuperación)
  - TMI (de inactividad)
- Recursos
- Responsables
- Pasos a seguir en la recuperación

### Planes de mantenimiento

- Operación
- Frecuencia
- Responsables
- Descripción

© JMA 2016. All rights reserved

## Copias de seguridad

- SQL Server Backup permite
  - Realizar backups en caliente
  - Realizar backups mientras los usuarios están activos
  - Realizar backups de los ficheros, grabando dónde están localizados
  - Capturar las actividades de la BD que ocurren durante el backup
    - Marca un checkpoint en BD y graba el LSN (Log Sequence Number) del último log
    - Escribe todas las páginas al dispositivo de backup.
    - Escribe todos los registros del transaction log escritos durante el backup.
- Actividades restringidas durante el backup
  - Crear o modificar la BD
  - Crear índices
  - Operaciones que no hacen log (select into, ...)
  - Compresión de la BD

© JMA 2016. All rights reserved

## Copias de seguridad

- Quién puede realizar copias de seguridad
  - sys admin, db\_owner, db\_backup operator
- Dónde se pueden realizar copias de seguridad
  - Disco duro, Cinta, Una localización identificada por un Named Pipe (NAS)
- Tipos:
  - Copias completas de datos
  - Copias diferenciales de datos
  - Copias de transacciones
- Granularidad de datos:
  - Completa (toda la base de datos)
  - Parciales (grupo principal y todos los grupos de lectura/escritura)
  - Ficheros o grupo de ficheros

© JMA 2016. All rights reserved

## Copias de seguridad completas

- Copia de seguridad de toda la base de datos, incluida parte del registro de transacciones, de modo que se pueda recuperar la copia de seguridad completa.
- Las copias de seguridad completas son independientes; representan la totalidad de la base de datos en el momento en que terminó la copia de seguridad.
- Son la base de trabajo para recuperar el sistema hasta el instante en que termino la copia de seguridad
- Al hacerlo, SQL Server:
  - Hace backups de cada actividad que ocurra durante el backup
  - Hace backups de cualquier transacción no validada en el log

© JMA 2016. All rights reserved

## Copias de seguridad diferenciales completas

- Una copia de seguridad de todos los archivos de la base de datos; sólo contiene las extensiones de datos modificadas desde la copia de seguridad completa de cada archivo más reciente.
- Una copia de seguridad diferencial resume todos los cambios desde que se hizo la copia completa.
- No confundir con el backup incremental del sistema operativo. Una copia diferencial incluye todos los cambios de las copias diferenciales anteriores.
- Hace backup de cada actividad, así como de las transacciones no completadas.
- Se realizan sobre BBDD que se modifican frecuentemente.
- Permite conjuntos de backups más pequeños y permiten una recuperación más rápida al no tener que aplicar todas las transacciones.
- Requiere un backup completo previo.

© JMA 2016. All rights reserved

## Copias de seguridad parciales y diferenciales parciales

- Una copia de seguridad parcial contiene todos los datos del grupo de archivos principal y todos los grupos de archivos de lectura/escritura, así como cualquier archivo que se haya especificado.
- Son útiles cuando una base de datos contiene uno o varios grupos de archivos de sólo lectura han sido de sólo lectura desde la última copia de seguridad completa.
- Indicada para grandes bases de datos cuyas copias de seguridad consumen muchos recursos y tiempo.
- Requiere una copia de seguridad completa

© JMA 2016. All rights reserved

## Copia de seguridad de archivos y diferencial de archivos

- Copia de seguridad de todos los datos de uno o varios archivos (o grupos de archivos).
- Una copia de seguridad de uno o varios archivos; contiene las extensiones de datos modificadas desde la copia de seguridad completa de cada archivo más reciente.
- La recuperación de errores de dispositivos aislados es más rápida.
- Indicada en grandes bases de datos que contienen datos con características de actualización diferentes.

© JMA 2016. All rights reserved

## Copia de seguridad del registro de transacciones

- Realiza una copia de todas las transacciones del registro de transacciones, al terminar, si no se indica lo contrario, trunca el log al principio de la parte activa y descarta la información de la inactiva borra todas las transacciones copiadas del fichero (mueve los punteros permitiendo la sobrescritura).
- Las copias de seguridad administran el espacio de los ficheros de transacciones, por lo que requiere copias frecuentes para mantener el tamaño controlado.
- Son similares al backup incremental del sistema operativo.
- Al ser una copia de transacciones requiere copias de datos previas.
- Sólo está disponible para las bases de datos que usan los modelos de recuperación completa u optimizados para cargas masivas de registros.
- Las opciones NO\_TRUNCATE y CONTINUE\_AFTER\_ERROR no eliminan las transacciones copiadas.
- La instrucción BACKUP LOG con las opciones TRUNCATE\_ONLY o NO LOG, truncan el registro de transacciones sin hacer copia de seguridad.

© JMA 2016. All rights reserved



## Backups de las BBDD del Sistema

- master
  - Manualmente: Cada vez que se realiza un conjunto de operaciones que modifican la base de datos:
    - Cambios en los inicios de sesión,
    - Crear, adjuntar o separar bases de datos
    - Modificaciones en los servidores vinculados, servicios y otros objetos de servidor.
  - Periódicamente: Copias de refresco en prevención de fallos en los medios de copia.
- model
  - Manualmente: Cada vez que se realizan cambios la base de datos.
  - Periódicamente: Copias de refresco.
- msdb
  - Manualmente: Cada vez que se realiza un conjunto de operaciones los elementos del Agente de SQL Server:
    - alertas, trabajos, programaciones y operadores.
  - Periódicamente: Mantiene el histórico de la ejecución de los trabajos por lo que requiere copias frecuentes.

© JMA 2016. All rights reserved

## Backup de las BD de usuarios

- Manualmente:
  - Después de la creación de la BD o cambios estructurales.
  - Después de la creación de índices
    - Si se hace backup sólo de log, se tendrá que reconstruir el índice.
  - Después de limpiar el log
    - Backup log with truncate\_only / no\_log
  - Después de operaciones que no hacen log
    - Select ... Into
    - Los de backup anteriores
- Periódicamente (según volumen)
  - Copia completa semanal.
  - Copia diferencial diaria.
  - Copia del registro varias veces al día.

© JMA 2016. All rights reserved

## Secuencia de copia de seguridad

- Copia completa
  - Copia del registro de transacciones
  - Copia del registro de transacciones
  - Copia del registro de transacciones
  - Copia diferencial
    - Copia del registro de transacciones
    - Copia del registro de transacciones
  - Copia diferencial
    - Copia del registro de transacciones
    - Copia del registro de transacciones
    - Copia del registro de transacciones
  - Copia diferencial
    - Copia del registro de transacciones
    - Copia del registro de transacciones
  - ...

© JMA 2016. All rights reserved

## Restauración de la BBDD

- La restauración es el proceso de restaurar datos de una o más copias de seguridad después de un problema y, a continuación, recuperar la base de datos (aplicar las transacciones pendientes).
- SQL Server admite la restauración de datos en los siguientes niveles:
  - La base de datos (una restauración completa de la base de datos)
  - El archivo de datos (una restauración de archivos)
  - La página de datos (una restauración de páginas)
- SQL Server permite recuperar la base de datos hasta:
  - El instante mismo en que se produjo el error (con copia final).
  - Un instante determinado: marca temporal o transacción.
  - El momento que termino la copia de seguridad.

© JMA 2016. All rights reserved

## Actividades durante la Restauración

- Al ejecutar la sentencia RESTORE DATABASE, SQL Server realiza un chequeo seguro del sistema
- Este mecanismo previene de una sobrescritura accidental
- SQL Server no restaura la BBDD si:
  - El nombre de la Base de Datos ya existe en el servidor
  - El conjunto de ficheros de la BBDD en el servidor difieren del conjunto de ficheros contenidos en la copia de seguridad
  - No se le suministran todos los ficheros necesarios para restaurar la copia
- Cuando se restaura la BBDD desde una copia de seguridad completa, SQL Server recrea los ficheros de la BBDD original y los sitúa en los directorios desde donde se grabaron al hacer la copia de seguridad
- Todos los objetos de la BBDD se recrean automáticamente

© JMA 2016. All rights reserved

## Verificación de las Copias de Seguridad

- Para asegurar que las copias de seguridad son válidas, se puede utilizar el comando RESTORE con opciones
  - RESTORE HEADERONLY
    - Devuelve la información de la cabecera del conjunto de ficheros de backup
    - Si el fichero de backup contiene más de un conjunto de copias de seguridad, devuelve la información de todas las copias
  - RESTORE FILELISTONLY
    - Muestra la información de los ficheros de datos originales y de los ficheros de transaction log contenidos en la copia de seguridad
  - RESTORE LABELONLY
    - Devuelve información sobre el dispositivo de backup
  - RESTORE VERIFYONLY
    - Sirve para comprobar que el conjunto de ficheros que componen la copia de seguridad están completos y que los ficheros individuales son legibles
    - No verifica la estructura de los datos contenidos dentro del backup

© JMA 2016. All rights reserved

## Tareas previas a la Restauración

- Restringir el acceso a la Base de Datos
  - Limitar el acceso a los miembros de db\_owner, dbcreator o sysadmin
- En caso de no haber perdido los dispositivos de soporte de los ficheros del registro de transacciones, realizar una copia del registro de transacciones con la opción “Realizar copia de seguridad del final del registro y dejar la base de datos en estado de restauración”
  - Realiza una copia de seguridad del final del registro y deja la base de datos en estado de restauración.
  - Esta opción crea una copia del final del registro, que realiza una copia de seguridad de los registros de los que todavía no se ha realizado ninguna (registro activo), generalmente, para la preparación de la restauración de una base de datos.
  - La base de datos no estará disponible para los usuarios hasta que haya finalizado su restauración.

© JMA 2016. All rights reserved

## Fases de la restauración

- La fase de copia de datos implica copiar todos los datos, el registro y las páginas de índice desde el medio de copia de seguridad de una base de datos a los archivos de la base de datos.
- La fase de rehacer aplica las transacciones registradas a los datos copiados desde la copia de seguridad para poner al día esos datos hasta el punto de recuperación.
  - Normalmente, en este punto una base de datos tiene transacciones no confirmadas y se encuentra en un estado inutilizable.
  - En ese caso, se requiere una fase de deshacer como parte de la recuperación de la base de datos.
- La fase de deshacer, que es la primera parte de la recuperación, revierte cualquier transacción no confirmada y hace que la base de datos esté disponible para los usuarios.
  - Después de la fase de reversión, no se pueden restaurar las copias de seguridad subsiguientes.

© JMA 2016. All rights reserved

## Restaurando la copia completa

- Cuándo hacerlo
  - Cuando esté dañado físicamente el disco
  - Cuando la base de datos entera esté corrupta, dañada o haya sido borrada
  - Para mantener una copia idéntica de la base de datos en otra instancia de SQL Server
- Opciones de recuperación
  - Utilizar NORECOVERY si se piensa restaurar posteriormente una copia diferencial o una copia del registro de transacciones
  - Sobrescribir la base de datos existente, cuando se restauren copias para errores lógicos.
  - Restringir el acceso a la base de datos restaurada, en caso de tener que verificar el resultado de la restauración.
- Inicia la secuencia de restauración con la última copia disponible (error físico) o la última copia estimada antes del error (lógico).

© JMA 2016. All rights reserved

## Restaurando la copia diferencial

- Restaura sólo la parte de la base de datos que cambio desde la última copia de seguridad completa
- Devuelve la base de datos al estado en el que estaba cuando se realizó el copia de seguridad diferencial
- Suele tardar mucho menos que la aplicación de los registros de transacciones.
- Consideraciones
  - Hay que restaurar la copia de seguridad completa ANTES de restaurar la copia de seguridad diferencial
  - Hay que especificar la opción NORECOVERY cuando se quieran aplicar las transacciones pendientes contenidas en las copias del registro de transacciones.
- Se utiliza la última copia disponible (error físico) o la última copia estimada antes del error (lógico).

© JMA 2016. All rights reserved

## Restaurando las copias del registro

- Cuando se restaura una copia de seguridad del registro de transacciones se vuelven a realizar las transacciones contenidas en la copia, rehaciendo todos los cambios ocurridos desde la última copia completa o diferencial.
- Es necesario haber restaurado la copia completa y, opcionalmente, la copia diferencial.
- Hay que aplicar ordenadamente las copias del registro desde la última copia restaurada (completa o diferencial).
- Opciones:
  - RESTORE WITH RECOVERY: Deja la base de datos lista para su uso revirtiendo las transacciones no confirmadas. No se pueden restaurar registros de transacciones adicionales.
  - RESTORE WITH NORECOVERY: Deja la base de datos no operativa y sin revertir transacciones no confirmadas. Pueden restaurarse registros de transacciones adicionales.
  - RESTORE WITH STANDBY: Deja la base de datos en modo de solo lectura. Permite revertir el deshacer las transacciones sin confirmar, guardando las acciones de deshacer en un archivo para anular los efectos de recuperación.

© JMA 2016. All rights reserved

## Marcas en las transacciones

- Para operaciones de alto riesgo, se pueden incluir marcas en las transacciones y así poder recuperar el sistema antes de estas marcas
- Para insertar marcas con nombre en un registro de transacciones
  - BEGIN TRANSACTION y la cláusula WITH MARK [description]
  - Debido a que el nombre de la marca es el mismo que el de la transacción, es necesario un nombre de transacción
  - La descripción [description] opcional es un texto descriptivo de la marca
  - Por cada transacción marcada que se confirma, se inserta una fila en la tabla **logmarkhistory** de **msdb**
- Hay dos maneras de recuperar hasta una marca del registro
  - RESTORE LOG y la cláusula WITH STOPATMARK='mark\_name' para confirmar hasta la marca e incluir la transacción que contiene la marca
  - RESTORE LOG y la cláusula WITH STOPBEFOREMARK='mark\_name' para confirmar hasta la marca y excluir la transacción que contiene la marca

© JMA 2016. All rights reserved

## Secuencia de restauración

### Cerrar secuencia de copia de seguridad

- Copia completa
  - Copia del registro de transacciones
  - Copia del registro de transacciones
  - Copia del registro de transacciones
  - Copia diferencial
    - Copia del registro de transacciones
    - Copia del registro de transacciones
  - Copia diferencial
    - Copia del registro de transacciones
    - Copia del registro de transacciones
    - Copia del registro de transacciones
  - Copia diferencial
    - Copia del registro de transacciones
    - Copia del registro de transacciones
    - ...
    - **Copia de seguridad del final del registro**

### Iniciar secuencia de restauración

- Última Copia completa
- Última Copia diferencial
- Siguiendo Copia del registro de transacciones
- Siguiendo Copia del registro de transacciones
- ...
- Última Copia de seguridad del registro (WITH RECOVERY)
- Recuperar la base de datos
  - Restaurar la última copia WITH RECOVERY.

© JMA 2016. All rights reserved

## OPTIMIZACIÓN DE SENTENCIAS

© JMA 2016. All rights reserved

# Introducción

- El SQL, al ser un lenguaje de alto nivel, requiere ser compilado antes de poder ejecutarse.
- El resultado de la compilación se denomina plan de ejecución y se almacenan en una parte de la memoria de SQL Server denominada caché del plan.
- Fases de la ejecución:
  1. Búsqueda en la cache
  2. Validación sintáctica
  3. Conversión y homogeneización (parametrización)
  4. Optimización
  5. Almacenamiento en cache
  6. Establecer contexto
  7. Ejecutar plan

© JMA 2016. All rights reserved

## Pasos básicos para procesar una instrucción

1. El analizador examina la instrucción y la divide en unidades lógicas como palabras clave, expresiones, operadores e identificadores.
2. Se genera un árbol de la consulta, a veces denominado árbol de secuencia, que describe los pasos lógicos que se requieren para transformar los datos de origen en el formato que necesita el conjunto de resultados.
3. El optimizador de consultas analiza diferentes formas de acceso a las tablas de origen y a continuación:
  - Selecciona la serie de pasos que devuelve los resultados de la forma más rápida utilizando el menor número posible de recursos.
  - El árbol de la consulta se actualiza para registrar esta serie exacta de pasos.
  - La versión final y optimizada del árbol de la consulta se denomina plan de ejecución.
4. El motor relacional comienza a ejecutar el plan de ejecución. A medida que se procesan los pasos que necesitan datos de las tablas base, el motor relacional solicita al motor de almacenamiento que pase los datos de los conjuntos de filas solicitados desde el motor relacional.
5. El motor relacional procesa los datos que devuelve el motor de almacenamiento en el formato definido para el conjunto de resultados y devuelve el conjunto de resultados al cliente.

© JMA 2016. All rights reserved



## Optimizador de consultas

- El optimizador de consultas de SQL Server es un optimizador basado en el costo.
- Cada plan de ejecución posible tiene asociado un costo en términos de la cantidad de recursos del equipo que se utilizan.
- El optimizador de consultas debe analizar los planes posibles y elegir el de menor costo estimado.
- Algunas instrucciones complejas tienen miles de planes de ejecución posibles. En estos casos, el optimizador de consultas no analiza todas las combinaciones posibles. En lugar de esto, utiliza algoritmos complejos para encontrar un plan de ejecución que tenga un costo razonablemente cercano al mínimo posible.
- El optimizador elige, además de por el costo de recursos mínimo, el plan que devuelve resultados al usuario a la mayor brevedad posible con un costo razonable de recursos.

© JMA 2016. All rights reserved

## Optimizador de consultas

- El optimizador utilizará un plan de ejecución en paralelo para devolver resultados si esto no afecta negativamente a la carga del servidor.
- El optimizador confía en las estadísticas de distribución y en los índices disponibles cuando calcula los costos de recursos de métodos diferentes para extraer información de una tabla o un índice.
- Las estadísticas de distribución se mantienen para las columnas y los índices, indicando la posibilidad de seleccionar los valores de un índice o de una columna determinados.
- Si las estadísticas no están actualizadas, puede que el optimizador de consultas no realice la mejor elección para el estado actual de la tabla.
- El optimizador de consultas permite que el servidor se ajuste dinámicamente a las condiciones cambiantes de la base de datos

© JMA 2016. All rights reserved

## Causas de ejecución lenta

- Comunicaciones de red lentas.
- Memoria inadecuada en el equipo servidor o falta de memoria disponible para SQL Server.
- Bloqueos e interbloqueos.
- Falta de estadísticas útiles
- Falta de índices útiles.
- Falta de vistas indizadas útiles.
- Falta de particiones útiles.
- Inadecuada redacción de la consulta.
- Inadecuada estructura lógica o física de la base de datos.

© JMA 2016. All rights reserved

## Planes de ejecución gráficos

- Los planes de ejecución gráficos deben leerse de arriba a abajo y de derecha a izquierda.
- Cada nodo de la estructura en árbol se representa como un icono que especifica el operador lógico y físico utilizado para ejecutar esa parte de la consulta o instrucción.
- Cada nodo está relacionado con un nodo principal.
- Los nodos secundarios que tienen el mismo nodo principal se dibujan en la misma columna. Sin embargo, no todos los nodos de la misma columna tienen necesariamente el mismo nodo principal.
- Cada nodo se conecta a su nodo principal mediante reglas con puntas de flecha.
- Los operadores se muestran como símbolos relacionados con un nodo principal específico.
- El ancho de la flecha es proporcional al número de filas.
- Se utiliza el número real de filas cuando está disponible. Si no, se utiliza el número estimado de filas.
- Cuando la consulta contiene varias instrucciones, se dibujan varios planes de ejecución de la consulta.
- Las partes de las estructuras en árbol se determinan por el tipo de instrucción ejecutada.
- Si se coloca el puntero sobre uno de los iconos o una de las flechas muestra la información ampliada sobre el mismo
- Se pueden incluir y comparar las estadísticas de cliente, para obtener resultados fiables es conveniente forzar un punto de comprobación y borrar el buffer cache para que vuelva a cargar las páginas de disco:  
CHECKPOINT  
DBCC DROPCLEANBUFFERS

© JMA 2016. All rights reserved

# Operadores

- Los operadores describen cómo SQL Server ejecuta una consulta o una instrucción DML (Lenguaje de manipulación de datos).
- El optimizador de consultas utiliza operadores para generar un plan de consulta con el fin de crear el resultado especificado en la consulta o para realizar la operación especificada en la instrucción DML.
- El plan de consulta es un árbol que consta de operadores físicos.
- Los operadores se clasifican como:
  - Operadores lógicos: describen una operación de procesamiento de consulta relacional a nivel conceptual.
  - Operadores físicos: implementan realmente la operación definida por un operador lógico utilizando un método o algoritmo concreto. Los operadores físicos se inicializan, recopilan datos y se cierran.
- Algunos operadores son a la vez lógicos y físicos.

© JMA 2016. All rights reserved

## Recuperación de páginas

- **Scan:** recupera todas las páginas aunque solo se devuelven las filas que cumplan el predicado.



Table Scan



Clustered Index Scan



Index Scan

- **Seek:** usa la capacidad de búsqueda de los índices para recuperar solo las páginas que contengan las filas que cumplan el predicado.



Clustered Index Seek



Index Seek

- **Spool:** guarda un resultado de consulta intermedio en la base de datos tempdb.



Spool






Table Spool



Index Spool



© JMA 2016. All rights reserved

## Búsquedas

-  **Bookmark Lookup:** usa un marcador (identificador de fila o clave de agrupación en clústeres) para buscar la fila correspondiente en la tabla o índice clúster.  
Sustituido por los operadores Clustered Index Seek y RID Lookup.
-  **Key Lookup:** es una búsqueda de marcadores en un tabla con un índice clúster.
-  **RID Lookup:** es una búsqueda de marcadores en un montón que usa un identificador de fila suministrado (RID).

© JMA 2016. All rights reserved

## Combinaciones

-  **Nested Loops:** realiza las operaciones lógicas de combinación interna, combinación externa izquierda, semicombinación izquierda y anti semicombinación. Las combinaciones de bucles anidados buscan en la tabla interna cada fila de la tabla externa, normalmente mediante un índice. El procesador de consultas decide, en función de los costos anticipados, si debe ordenar o no la entrada externa para mejorar la ubicación de las búsquedas en el índice de la entrada interna. Se devuelven las filas que cumplen el predicado.
-  **Merge Join:** requiere dos entradas ordenadas por sus respectivas columnas, que se pueden realizar mediante la inserción de operaciones de ordenación explícitas en el plan de consulta. Es especialmente eficaz si no se necesita un orden explícito, por ejemplo, si hay un índice idóneo de árbol b en la base de datos o si el orden se puede aprovechar en varias operaciones, como en una combinación de mezcla y una agrupación con acumulación.

© JMA 2016. All rights reserved

# Combinaciones



**Hash Match:** genera una tabla hash y calcula un valor hash para cada fila de su entrada de compilación. A continuación, por cada fila de sondeo (como corresponda), calcula un valor hash (con la misma función hash) y busca las coincidencias en la tabla hash. El comportamiento depende de la operación lógica que se esté realizando:

- Para cualquier combinación, utilice la primera entrada (superior) para generar la tabla hash y la segunda entrada (inferior) para sondear la tabla hash. Obtendrá como resultado las coincidencias (o las no coincidencias) que indique el tipo de combinación. Si varias combinaciones utilizan la misma columna de combinación, estas operaciones se agrupan en un equipo hash.
- Para los operadores Distinct o Aggregate, utilice la entrada para generar la tabla hash (para ello, quite los duplicados y calcule las expresiones de agregado). Cuando se haya generado la tabla hash, recorra la tabla y presente todas las entradas.
- En el caso del operador Union, utilice la primera entrada para generar la tabla hash (para ello, quite los duplicados). Use la segunda entrada (que no debe tener duplicados) para sondear la tabla hash, devolver todas las filas que no tengan coincidencias y, a continuación, recorrer la tabla hash para devolver todas las entradas.



© JMA 2016. All rights reserved

## Tipos de combinación

- **Cross Join:** combina cada fila de la primera entrada (superior) con cada fila de la segunda entrada (inferior).
- **Inner Join:** devuelve todas las filas que cumplen la combinación de la primera entrada (superior) con la segunda entrada (inferior).
- **Full Outer Join:** devuelve cada fila que cumple el predicado de combinación de la primera entrada (superior) combinada con cada fila de la segunda entrada (inferior). También devuelve filas de la primera entrada que no tenga coincidencias en la segunda entrada y de la segunda entrada que no tenga coincidencias en la primera entrada. La entrada que no contiene valores coincidentes se devuelve como un valor nulo.
- **Left Outer Join:** devuelve cada fila que cumple la combinación de la primera entrada (superior) con la segunda entrada (inferior). También devuelve las filas de la primera entrada que no tienen filas coincidentes en la segunda entrada. Las filas que no coinciden en la segunda entrada se devuelven como valores NULL. Si no hay ningún predicado de combinación, cada una de las filas es coincidente.
- **Left Semi Join:** devuelve todas las filas de la primera entrada (superior) cuando hay una fila coincidente en la segunda entrada (inferior). Si no hay ningún predicado de combinación, cada fila es una fila coincidente.
- **Left Anti Semi Join:** devuelve todas las filas de la primera entrada (superior) cuando no hay ninguna fila coincidente en la segunda entrada (inferior). Si no hay ningún predicado de combinación, cada fila es una fila coincidente.
- **Right Outer Join, Right Semi Join, Right Anti Semi Join:** Similares a los Left empezando con la segunda entrada.
- **Union:** recorre varias entradas, obtiene cada fila recorrida y quita los duplicados.






© JMA 2016. All rights reserved

## Filtrados

-  **Filter:** recorre la entrada y solo devuelve las filas que cumplen la expresión del filtro (predicado).
-  **Bitmap:** usado para implementar filtros de mapas de bits en planes de consulta paralelos. Los filtros de mapas de bits agiliza la ejecución de la consulta al eliminar las filas con valores de clave que no pueden generar ningún registro de combinación antes de pasar las filas a través de otro operador, por ejemplo, el operador Parallelism. Un filtro de mapas de bits usa una representación compacta de un conjunto de valores de una tabla en una parte del árbol de operadores para filtrar filas de una segunda tabla en otra parte del árbol. Al quitar las filas innecesarias en las primeras fases de la consulta, los operadores subsiguientes tienen que trabajar en menos filas y el rendimiento total de la consulta mejora. El optimizador determina cuándo un mapa de bits es suficientemente selectivo para resultar útil y en qué operadores se aplica el filtro. Bitmap es un operador físico.






© JMA 2016. All rights reserved

## Otros operadores

-  **Sort:** ordena todas las filas entrantes. Las columnas llevan como prefijo el valor ASC, si el orden de las columnas es ascendente, o el valor DESC, si es descendente.
-  **Stream Aggregate:** agrupa las filas por una o varias columnas y, a continuación, calcula una o varias expresiones agregadas devueltas por la consulta. El resultado de este operador puede ser utilizado por operadores posteriores de la consulta, devuelto al cliente, o ambas cosas. Requiere una entrada ordenada por las columnas dentro de sus grupos.
-  **Compute Scalar:** evalúa una expresión para generar un valor escalar calculado que se puede devolver al usuario, hacer referencia a él en cualquier otra parte de la consulta o ambas cosas a la vez.
-  **UDX:** implementan una de las múltiples operaciones XQuery y XPath. Los operadores extendidos (UDX): FOR XML, XML SERIALIZER, XML FRAGMENT SERIALIZER, XQUERY STRING, XQUERY LIST DECOMPOSER, XQUERY DATA, XQUERY CONTAINS, UPDATE XML NODE.
-  **Top:** recorre la entrada y solo devuelve el primer número o porcentaje especificado de filas, basándose en un criterio de ordenación si es posible.

© JMA 2016. All rights reserved

## Otros operadores

-  **Assert:** comprueba una condición y si la expresión da como resultado un valor distinto a NULL se generará el error apropiado (validaciones).
-  **Concatenation:** explora varias entradas y devuelve cada fila explorada, se utiliza para implementar la construcción UNION ALL, tiene dos o más entradas y una salida. Concatenation copia filas del primer flujo de entrada en el flujo de salida y, a continuación, repite esta operación con cada flujo de entrada adicional.
-  **Segment:** divide el conjunto de entrada en segmentos basados en el valor de una o varias columnas.
-  **Split:** se utiliza para optimizar el procesamiento de actualizaciones. Divide cada operación de actualización en una operación de eliminación y una operación de inserción.
-  **Table-valued Function:** evalúa una función con valores de tabla y almacena las filas resultantes en la base de datos tempdb. Cuando los iteradores principales solicitan las filas, la función con valores de tabla devuelve las filas desde tempdb.

© JMA 2016. All rights reserved

## Detalles del nodo

- **Operación física:** Operador físico utilizado. Los operadores físicos presentados en color rojo indican que el optimizador de consultas ha emitido una advertencia, por ejemplo, la falta de estadísticas de columna o de predicados de combinación. Esto puede causar que el optimizador de consultas elija un plan de consulta menos eficaz que el esperado.
- **Operación lógica:** Operador lógico que coincide con el operador físico, como el operador Inner Join.
- **Tamaño de fila estimado:** Tamaño estimado de la fila creada por el operador (bytes).
- **Costo de E/S estimado:** Costo estimado de toda la actividad de E/S para la operación. Este valor *debe ser lo más pequeño posible*.
- **Costo de CPU estimado:** Costo estimado de toda la actividad de la CPU para la operación.
- **Costo de operador estimado:** Costo del optimizador de consultas para ejecutar esta operación. El costo de esta operación como porcentaje del costo total de la consulta se muestra entre paréntesis. Debido a que el motor de consultas selecciona la operación más eficaz para realizar la consulta o ejecutar la instrucción, este valor *debe ser el menor posible*.
- **Costo de subárbol estimado:** Costo total del optimizador de consultas para ejecutar esta operación y todas las operaciones del mismo subárbol anteriores a ésta.
- **Número de filas o Número de filas estimado:** Número de filas que produce el operador.
- En función del nodo: Objeto de salida, Lista de salida (columnas), Número de ejecuciones, Número de filas, Tamaño, Ordenación
- En algunos casos se mostraran advertencias o sugerencias sobre el nodo.

© JMA 2016. All rights reserved

## Sugerencias de consulta

- Las sugerencias de consulta (hint) invalidan el comportamiento predeterminado del optimizador de consultas mientras dura la instrucción de consulta.
- Son sugerencias por que el optimizador no está obligado a seguirlas.
- Se pueden usar para especificar un método de bloqueo en las tablas afectadas, uno o varios índices, una operación de procesamiento de la consulta como un recorrido de tabla o una búsqueda de índice, u otras opciones.
- Las sugerencias de consulta se especifican en la cláusula OPTION y se aplican a toda la consulta.
- Como el optimizador de consultas de SQL Server suele seleccionar el mejor plan de ejecución para las consultas, se recomienda que solo se hagan como último recurso.

© JMA 2016. All rights reserved

## Sugerencias de consulta

- |                                                                               |                                                                       |
|-------------------------------------------------------------------------------|-----------------------------------------------------------------------|
| • { HASH   ORDER } GROUP                                                      | • PARAMETERIZATION { SIMPLE   FORCED }                                |
| • { CONCAT   HASH   MERGE } UNION                                             | • RECOMPILE                                                           |
| • { LOOP   MERGE   HASH } JOIN                                                | • ROBUST PLAN                                                         |
| • FAST number_rows                                                            | • KEEP PLAN                                                           |
| • FORCE ORDER                                                                 | • KEEPFIXED PLAN                                                      |
| • MAXDOP                                                                      | • EXPAND VIEWS                                                        |
| number_of_processors                                                          | • MAXRECURSION number                                                 |
| • OPTIMIZE FOR ( @variable_name { UNKNOWN   = literal_constant } [ , ...n ] ) | • USE PLAN N'xml_plan'                                                |
| • OPTIMIZE FOR UNKNOWN                                                        | • TABLE HINT ( exposed_object_name [ , <table_hint> [ [ , ]...n ] ] ) |

© JMA 2016. All rights reserved



## Sugerencias de tabla

- Las sugerencias de tabla invalidan el comportamiento predeterminado del optimizador de consultas mientras dura la instrucción de lenguaje de manipulación de datos (DML), especificando un método de bloqueo, uno o varios índices, una operación de procesamiento de la consulta como, por ejemplo, un examen de tabla o una búsqueda de índice, u otras opciones.
- Las sugerencias de tabla se especifican en la cláusula FROM ... WITH de la instrucción DML y solo afectan a la tabla o a la vista a la que se hace referencia en esa cláusula.

© JMA 2016. All rights reserved

## SELECT

- |                                                              |                     |
|--------------------------------------------------------------|---------------------|
| • INDEX ( index_value [ ,...n ] )   INDEX = ( index_value )  | • READCOMMITTED     |
| • FASTFIRSTROW                                               | • READCOMMITTEDLOCK |
| • FORCESEEK [( index_value ( index_column_name [ ,... ] ) )] | • READPAST          |
| • FORCESCAN                                                  | • READUNCOMMITTED   |
| • HOLDLOCK                                                   | • REPEATABLEREAD    |
| • NOLOCK                                                     | • ROWLOCK           |
| • NOWAIT                                                     | • SERIALIZABLE      |
| • PAGLOCK                                                    | • TABLOCK           |
|                                                              | • TABLOCKX          |
|                                                              | • UPDLOCK           |
|                                                              | • XLOCK             |

© JMA 2016. All rights reserved

## INSERT, UPDATE, DELETE

- KEEPIDENTITY
- KEEPDEFAULTS
- FASTFIRSTROW
- HOLDLOCK
- IGNORE\_CONSTRAINTS
- IGNORE\_TRIGGERS
- NOLOCK
- NOWAIT
- PAGLOCK
- READCOMMITTED
- READCOMMITTEDLOCK
- READPAST
- REPEATABLEREAD
- ROWLOCK
- SERIALIZABLE
- TABLOCK
- TABLOCKX
- UPDLOCK
- XLOCK

© JMA 2016. All rights reserved

## Guías de plan

- Las guías de plan se pueden utilizar para optimizar el rendimiento de las consultas cuando no pueda o no desee cambiar directamente el texto de dichas consultas.
- Las guías de plan pueden ser de gran utilidad cuando el rendimiento de un pequeño subconjunto de consultas de una aplicación de base de datos implementada por otro proveedor no es el esperado.
- Las guías de plan influyen en la optimización de las consultas adjuntando sugerencias de consulta o un plan de consulta fijo para ellas.
- En la guía de plan, se especifica la instrucción de Transact-SQL que se desea optimizar y además una cláusula OPTION que incluye las sugerencias de consulta que se desean utilizar o un plan de consulta específico con el que desea optimizar la consulta.
- Cuando la consulta se ejecuta, el SQL Server hace coincidir la instrucción de Transact-SQL con la guía de plan y además asocia en tiempo de ejecución la cláusula OPTION a la consulta o utiliza el plan de consultas especificado.

© JMA 2016. All rights reserved

## Tipos de guías de plan

- **OBJECT:** compara las consultas que se ejecutan en el contexto de procedimientos almacenados de Transact-SQL, funciones escalares definidas por el usuario, funciones definidas por el usuario con valores de tabla de múltiples instrucciones y desencadenadores DML.
- **SQL:** compara las consultas que se ejecutan en el contexto de instrucciones independientes de Transact-SQL y lotes que no forman parte de un objeto de base de datos.
  - Las guías de plan basadas en SQL también se pueden usar para comparar consultas que se parametrizan en un formulario especificado.
  - Las guías de plan de SQL se aplican a las instrucciones y lotes independientes de Transact-SQL.
  - Con frecuencia, las aplicaciones envían esas instrucciones utilizando el procedimiento almacenado del sistema `sp_executesql`.
- **TEMPLATE:** compara consultas independientes que se parametrizan en un formulario especificado.
  - Estas guías de plan se usan para reemplazar la opción `PARAMETERIZATION` actual de una base de datos para una clase de consultas por medio de `SET`.

© JMA 2016. All rights reserved

## Ejemplos de guías de plan

```
sp_create_plan_guide @name = N'Guide1',
    @type = N'OBJECT',
    @module_or_batch = N'Sales.GetSalesOrderByCountry',
    @stmt = N'SELECT * FROM Sales.SalesOrderHeader AS h,
        Sales.Customer AS c,
        Sales.SalesTerritory AS t
    WHERE h.CustomerID = c.CustomerID
        AND c.TerritoryID = t.TerritoryID
        AND CountryRegionCode = @Country_region',
    @params = NULL,
    @hints = N'OPTION (OPTIMIZE FOR (@Country_region = N"US"))';

sp_create_plan_guide @name = N'Guide2',
    @type = N'SQL',
    @module_or_batch = NULL,
    @stmt = N'SELECT TOP 1 * FROM Sales.SalesOrderHeader ORDER BY OrderDate DESC',
    @params = NULL,
    @hints = N'OPTION (MAXDOP 1)';
```

© JMA 2016. All rights reserved

## Limitaciones para las guías

- El número total de guías de plan que se pueden crear solo está limitado por los recursos de los que disponga el sistema.
- Las guías de plan deberían limitarse a aquellas consultas de gran importancia cuyo rendimiento se desea mejorar o estabilizar.
- No se deben usar las guías de plan para influenciar la mayor parte de la carga de la consulta de una aplicación implementada.
- Se debe volver a evaluar y probar las definiciones de guías de plan al actualizar la aplicación a una nueva versión de SQL Server.
  - Los requisitos de optimización del rendimiento y el comportamiento de la coincidencia de las guías de plan pueden cambiar.
  - Aunque una guía de plan no válida no hará que una consulta provoque un error, el plan se compilara sin utilizar la guía de plan y posiblemente no sea la mejor opción
- No están disponibles en las ediciones Express.

© JMA 2016. All rights reserved

## Planes de ejecución forzados

- `xml_plan` es un literal de cadena derivado del plan de consulta con formato XML que se genera para una consulta.
- Un `xml_plan` se puede redactar manualmente u obtenerse mediante “Guardar plan de ejecución”, consultando la columna `query_plan` de la función `sys.dm_exec_query_plan` o eventos de traza: Showplan XML, Showplan XML Statistics Profile y Showplan XML For Query Compile.
- No todos los elementos de los planes de consulta con formato XML se fuerzan con la sugerencia `USE PLAN`. Los elementos que calculan expresiones escalares se omiten, así como algunas expresiones relacionales.
- El plan de consulta con formato XML especificado en `xml_plan` debe validarse con el esquema XSD `Showplanxml.xsd`.
- La sugerencia de consulta `USE PLAN` se puede especificar como una sugerencia de consulta en una instrucción SQL independiente, o especificarse en el parámetro `@hints` de una guía de plan.
- Se debe indicar siempre `xml_plan` como literal Unicode especificando el prefijo `N`, como en `N'xml_plan'`, con lo que se garantiza que no se pierda ninguno de los caracteres del plan específico del estándar Unicode cuando SQL Server Database Engine (Motor de base de datos de SQL Server) interprete la cadena.

© JMA 2016. All rights reserved

## Limitaciones para Planes forzados

- Los cambios que se produzcan en la base de datos, como la eliminación de índices, pueden invalidar un plan de consulta forzado.
- Un plan de consulta puede quedarse obsoleto aunque no se haga directamente referencia en el plan a un objeto eliminado.
- Instalar un Service Pack o una nueva versión de SQL Server puede impedir que fuerce un plan generado en una versión anterior, por lo que deberán probarse todos los planes forzados siempre que se actualice el servidor.
- El uso de planes forzados en una consulta reemplaza todas las sugerencias de combinación y de índice utilizadas en la misma consulta.
- Los únicos planes de consulta que pueden forzarse son los que pueden encontrarse mediante la estrategia de búsqueda típica del optimizador de consultas.