



Implementación de Base de Datos con Microsoft SQL Server 2000

Ing. Martín Manuel Leiva Castillo
Ing. Jorge Martín Rodríguez Castro

Visión general de SQL Server

Contenido

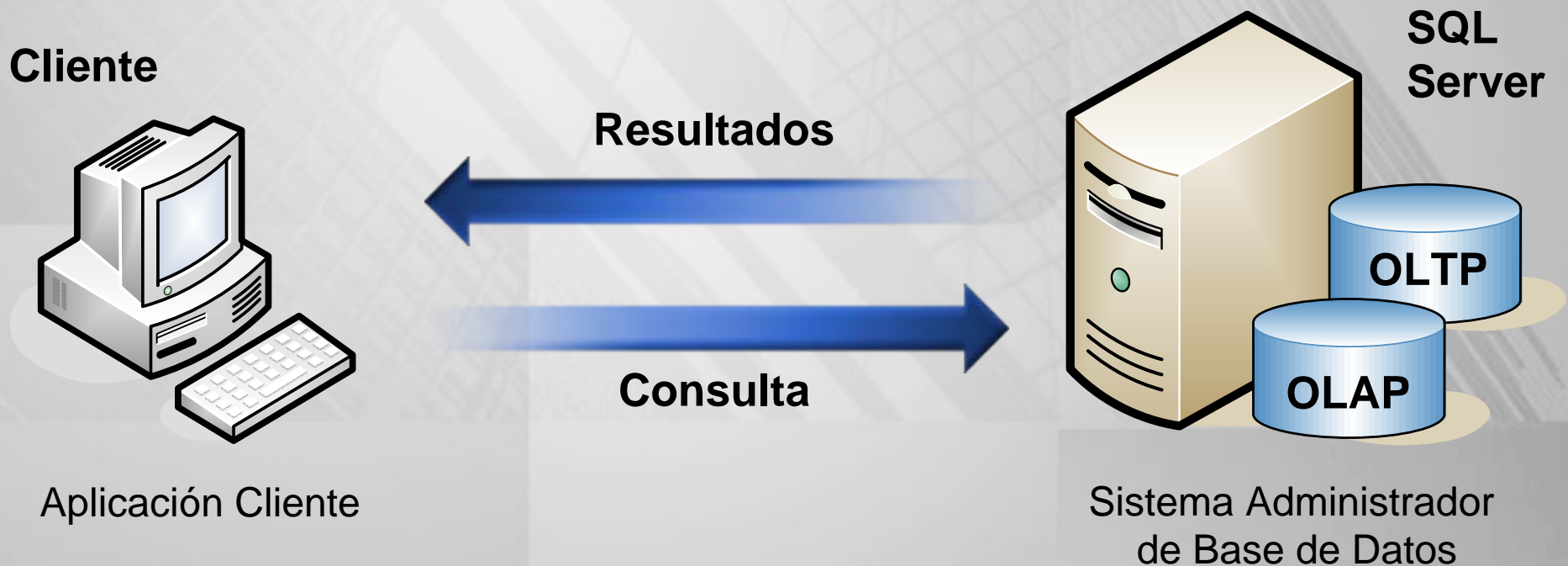
1. ¿Qué es SQL Server 2000?
2. Base de datos SQL Server
3. Trabajando con SQL Server

1. ¿Qué es SQL Server?

- Introducción a SQL Server
- Características de SQL Server
- Componentes Cliente-Servidor
- Comunicaciones Cliente-Servidor
- Servicios SQL Server

Introducción a SQL Server

SQL Server es un sistema administrador de base de datos relacional (RDBMS) cliente-servidor que usa Transact-SQL (T-SQL) para enviar respuestas entre un cliente y SQL Server.



Características de SQL Server

- Integración con Internet.
- Admite de forma nativa el Lenguaje de marcado extensible (XML) a través de HTTP, ADO u OLE DB.
- Escalabilidad y disponibilidad. El mismo motor de base de datos trabaja sobre cualquier versión de Windows.
- Características de base de datos corporativa.
- Facilidad de instalación, distribución y utilización.
- Almacenamiento de datos.

Servicios SQL Server



Servidor

MSSQLServer Service

- Administración de Datos
- Procesamiento de Transacciones y Consultas
- Integridad de Datos

SQLServerAgent Service

- Trabajos
- Alertas
- Operadores

Microsoft Distributed Transaction Coordinator

- Administración de transacciones distribuidas

Microsoft Search

- Catálogos de Texto
- Índices de Texto

2. Base de Datos SQL Server

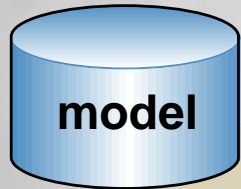
- Tipos de base de datos
- Objetos de base de datos
- Referencia a objetos SQLServer
- Tablas del Sistema

Tipos de Bases de Datos 1/2

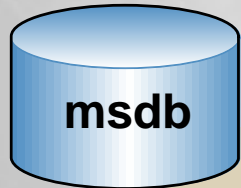
Bases de Datos del Sistema



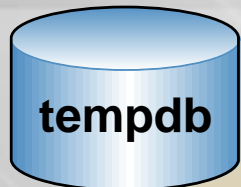
Base de datos principal. Controla las bases de datos de usuario y la operación de SQL Server de forma global al realizar el seguimiento de la información tal como las cuentas de usuario y los mensajes de error del sistema.



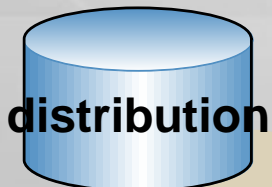
Base de datos plantilla. Proporciona una plantilla o prototipo para las nuevas bases de datos de usuario. Cada vez que se crea una nueva base de datos se hace una copia de *model* como base de la nueva base de datos.



Base de datos utilizada por el servicio SQL Server Agent. Almacena información sobre copias de seguridad, mantenimiento y duplicación de bases de datos.



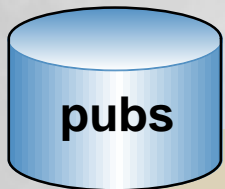
Base de datos temporal. Es un espacio de trabajo. Se crea cada vez que se inicia SQL Server. Utilizada para las tablas temporales creadas explícitamente, para las tablas que mantienen resultados intermedios y cursores estáticos.



Almacena los datos del historial y de las transacciones que se utilizan en la duplicación.

Tipos de Bases de Datos 2/2

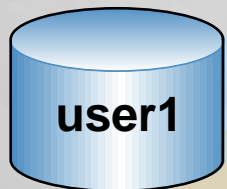
Bases de Datos de Usuario



Base de datos de ejemplo. Consume 2 MB y puede ser eliminada. Puede reconstruirse ejecutando el archivo de script *instpubs.sql* ubicado en el subdirectorio *Install* de SQL Server.



Base de datos de ejemplo. Consume 4 MB y puede ser eliminada. Es más compleja de *pubs*. Puede reconstruirse ejecutando el archivo de script *instnwnd.sql* ubicado en el subdirectorio *Install* de SQL Server.



Base de datos de usuario. Identifica una base de datos definida por el usuario.

Objetos de Bases de Datos



Referenciando a Objetos SQL Server

- **Nombres completamente cualificados**

Servidor.BaseDatos.Propietario.Objeto

Ejm.

```
SELECT * FROM SERVER05.Northwind.dbo.Customers
```

- **Nombres parcialmente cualificados**

- Servidor predeterminado es la instancia actual del servidor local

- Base de datos predeterminada es la actual base de datos

- Propietario predeterminado es el nombre de usuario en la base de datos

Ejm.

```
SELECT * FROM Northwind..Customers
```

```
--(Si Northwind no es la base de datos predeterminada)
```

```
CREATE TABLE Empleado
```

```
--(Si estamos en la base de datos predeterminada)
```

Tablas del sistema

- Almacenan información (Metadatos) acerca del Sistema y Objetos de Base de Datos
- El **Catálogo de la Base de Datos** almacena metadatos acerca de una base de datos específica
- El **Catálogo del Sistema** almacena Metadatos acerca de todo el Sistema y de todas las demás bases de datos
- Todas las tablas de sistema comienzan con el prefijo **sys**.



syslogins (master): contiene una fila por cada cuenta de inicio de sesión que puede conectar con SQL Server.



sysmessages (master): contiene una fila por cada error o advertencia del sistema que SQL Server pueda devolver



sysdatabases (master): contiene una fila por cada base de datos de un servidor SQL Server.

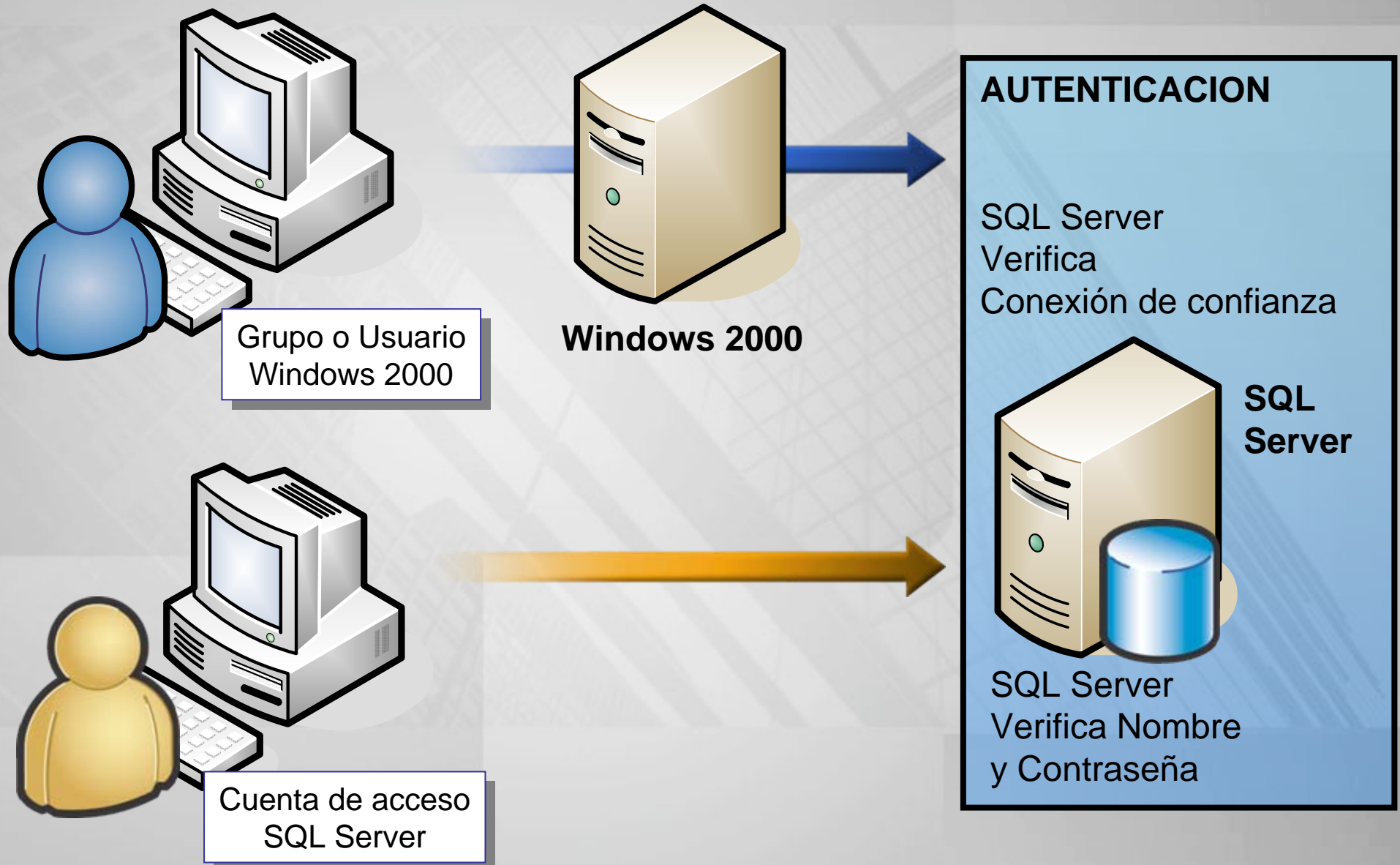


sysusers (todas): contiene una fila por cada usuario o función de SQL Server de una base de datos



sysobjects (todas): contiene una fila por cada objeto de base de datos.

Autenticación de acceso



3. Trabajando con SQL Server

- Administrando una base de datos SQL Server
- Implementando una base de datos SQL Server
- Seleccionando una arquitectura de aplicación para SQL Server

Administrando una base de datos SQL Server

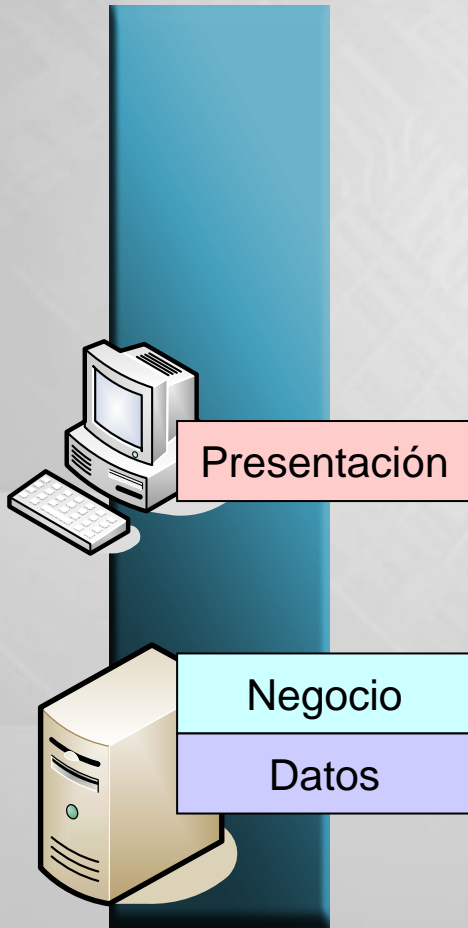
- Instalar SQL Server (Establecer la seguridad de la red. Configurar SQL Server)
- Generar base de datos (asignar espacio a la BD, transferir datos, definir e implementar la seguridad, automatizar trabajos, duplicar y publicar datos)
- Administrar actividades cotidianas (Importar y exportar datos. Hacer y restaurar copias de seguridad de la base de datos y el registro. Supervisar y ajustar la base de datos)
- Tareas administrativas comunes
- Administrador corporativo
- Herramientas administrativas y asistentes
- Herramientas de administración de línea de comandos
- Ayuda en línea y libros en pantalla.

Implementando una base de datos SQL Server

- Diseñando la base de datos
- Creando la base de datos y los objetos de base de datos
- Probando y afinando las aplicaciones y la base de datos
- Planificando el desarrollo

Seleccionando una arquitectura de aplicación para SQL Server

Servidor Inteligente (2-capas)



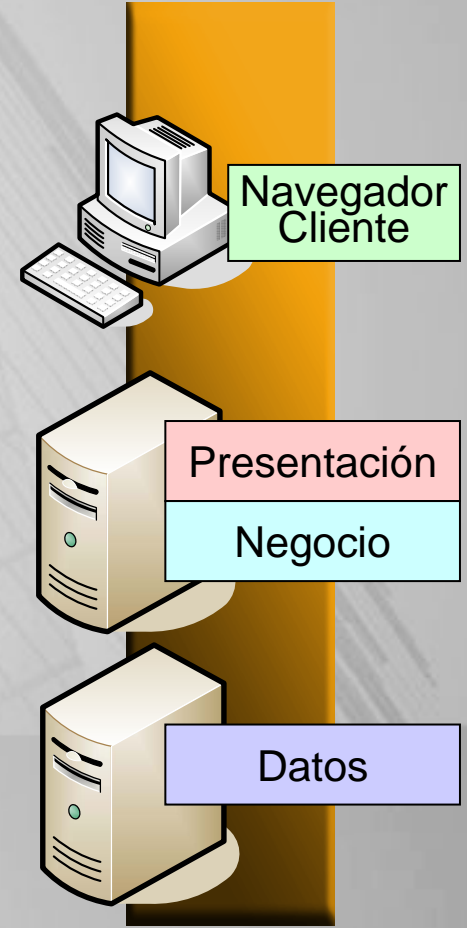
Cliente Inteligente (2-capas)



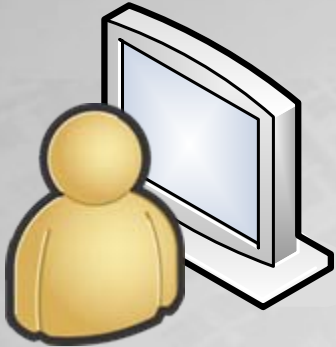
N-capas



Internet

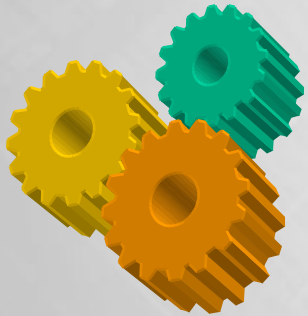


Capas lógicas



Presentación

Incluye la lógica para presentar datos y aplicaciones a los usuarios. Esta capa es casi siempre implementada en la computadora cliente.



Negocio

Incluye la lógica y reglas del negocio. SQL Server puede involucrarse con esta capa.



Datos

Incluye la definición de la base de datos, lógica de integridad de datos, procedimientos almacenados y otras operaciones que están muy asociadas directamente a los datos. SQL Server administra esta capa.

Introducción a Transact-SQL

1. Herramientas de programación
2. Lenguaje Transact-SQL (T-SQL)
3. Elementos de T-SQL

1. Herramientas de programación

- Analizador de consultas
 - Codificación automática de color de los elementos del sintaxis
 - Crear rápidamente los objetos de base de datos de uso más frecuente desde secuencias de comandos predefinidas. (Plantillas)
 - Ejecutar procedimientos almacenados sin conocer los parámetros. (Función de ejecución de procedimiento del Examinador de objetos)
 - Depurar problemas de rendimiento de consultas. (Mostrar plan de ejecución, Mostrar traza del servidor, Mostrar estadísticas del cliente, Asistente para optimización de índices)
 - Múltiples ventanas de consultas
 - Vistas personalizadas del conjunto de resultados
 - Planes de ejecución gráfica
 - Ejecuta partes de un script

2. Lenguaje Transact-SQL (T-SQL)

- Versión del Lenguaje de consulta estructurado (SQL, Structured Query Language),
- Implementación SQL Server del estándar ANSI SQL-92
- Lenguaje de consulta y programación de base de datos
- Puede correr en cualquier producto que cumple con el estándar
- Contiene funciones adicionales únicas

3. Elementos de T-SQL

- Sentencias de lenguaje de control de datos
- Sentencias de lenguaje de definición de datos
- Sentencias de lenguaje de manipulación de datos
- Elementos adicionales del lenguaje

Sentencias de control de datos

- Configuran y modifican permisos
 - GRANT
 - DENY
 - REVOKE

```
USE northwind
```

```
GRANT SELECT ON products TO public
```

- Por defecto solo los roles sysadmin, dbcreator, db_owner y db_securityadmin pueden ejecutarlos

Sentencias de definición de datos

- Definen los objetos de base de datos
 - CREATE *<tipoobjeto>* *<nombre>*
 - ALTER *<tipoobjeto>* *<nombre>*
 - DROP *<tipoobjeto>* *<nombre>*

```
CREATE DATABASE prueba
```

```
DROP DATABASE prueba
```

```
ALTER TABLE Cliente
```

```
DROP CONSTRAINT CK_EMPLEADO_SALARIO
```


Nombre de objetos SQL Server

- Identificadores estándar
 - Primer carácter debe de ser alfabético
 - Los otras caracteres pueden incluir letras, números, o símbolos
 - Los identificadores que empiezan con símbolos tienen especiales usos
- Identificadores delimitados
 - Cuando el nombre contiene espacios
 - Cuando palabras reservadas forman parte del nombre
 - Usar [] o " "

Reglas para nombrar objetos

- Mantener nombre cortos
- Usar nombre significativos cuando sea posible
- Use convenciones de nombres claros y simples
- Use un identificador que distinga los tipos de objetos
 - Vistas
 - Procedimientos almacenados
- Mantener los nombres de objetos y nombres de usuarios únicos
 - Tabla **venta** y rol **venta**

Sentencias de manipulación de datos

- Usarlos cuando se trabaja con datos de la base de datos

SELECT, INSERT, UPDATE, DELETE

```
SELECT Nombre, Apellido, Salario FROM  
EMPLEADO
```

```
INSERT INTO EMPLEADO (Nombre, Apellido,  
Salario) VALUES ('Juan', 'Quiroga', 250.45)
```

```
UPDATE EMPLEADO SET Salario=340.45 WHERE  
Nombre='Juan'
```

```
DELETE EMPLEADO
```

Elementos adicionales del lenguaje

- Variables locales
- Operadores
- Funciones
- Elementos de flujo de control
- Comentarios

Variables locales

- Definido por el usuario con la sentencia DECLARE
- Asigna valores con la sentencia SET o SELECT

```
DECLARE @nombre char(20)
SET @nombre = 'Alberto'
SELECT *
FROM Autores
WHERE Nombre = @nombre
```


Funciones

- Funciones del conjunto de filas

```
SELECT a.*  
FROM OPENROWSET('Microsoft.Jet.OLEDB.4.0',  
    'c:\Mis Documentos\Ventas.mdb';'admin';'', cliente)AS a
```

- Funciones agregadas

```
USE pubs  
SELECT COUNT(*) FROM authors
```

- Funciones escalares

```
SELECT DB_NAME() AS 'BaseDatos'
```

Elementos de flujo de control

- Nivel de sentencias
 - Bloques BEGIN ... END
 - Bloques IF ... ELSE
 - Construcción WHILE
- Nivel de fila
 - Expresión CASE

BEGIN ... END y IF ... ELSE

```
USE northwind
IF EXISTS (SELECT orderid FROM orders
           WHERE customerid = 'frank')
  PRINT 'El cliente no puede ser eliminado'
ELSE
  BEGIN
    DELETE customers WHERE customerid = 'frank'
    PRINT '*** Cliente eliminado ***'
  END
```

Construcción WHILE

```
DECLARE @num int
SET @num=1
WHILE @num<=10
    BEGIN
        SELECT @num
        SET @num = @num + 1
    END
```

```
DECLARE @num int
SET @num=1
WHILE @num<=100
    BEGIN
        SELECT @num
        SELECT @num = @num + 1
        IF @num=50
            BREAK
        ELSE
            CONTINUE
    END
```

Expresión CASE

- Sintaxis

CASE *expresión*

{WHEN *expresión* THEN *resultado*} [,...n]

[ELSE *resultado*]

END

```
SELECT au_fname, au_lname,  
        CASE state  
            WHEN 'CA' THEN 'California'  
            WHEN 'KS' THEN 'Kansas'  
            WHEN 'TN' THEN 'Tennessee'  
            WHEN 'OR' THEN 'Oregon'  
            ELSE 'No determinado'  
        END AS StateName  
FROM pubs.dbo.authors
```

Comentarios

- Comentarios en línea

```
SELECT * -- Lista todas las columnas de autores  
FROM authors
```

- Comentarios en bloque

```
/* Este código recupera todas las filas de  
autores de la base de datos pubs  
*/  
SELECT * FROM authors
```

Creación de base de datos

1. Como los datos son almacenados
2. Crear base de datos
3. Modificando base de datos
4. Tipos de datos
5. Creando tablas
6. Generando scripts

Cómo se almacenan los datos



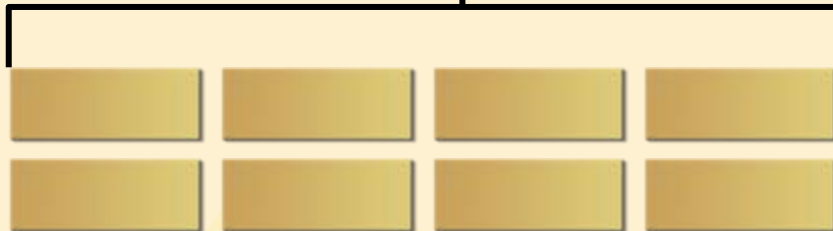
Base de Datos



Data (archivo)
.mdf o .ndf



Log (archivo)
.ldf



Tablas, índices

Extensión

(8 contiguos 8-KB página)



Datos

Página (8 KB)

Máximo tamaño de fila = 8060 bytes

2. Crear base de datos

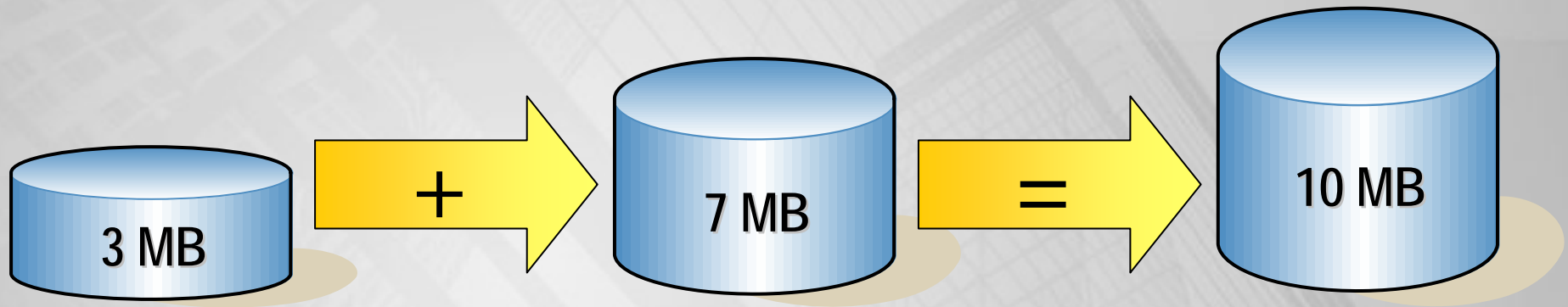
- Definir:
 - El nombre de la base de datos
 - El tamaño de la base de datos
 - Los archivos donde las base de datos se almacenarán

```
CREATE DATABASE compañía
ON PRIMARY
(NAME=compañía_data, FILENAME = 'c:\Archivos de programa\Microsoft
SQL Server\mssql\data\compañía_data.mdf',
SIZE=3 MB, MAXSIZE=5 MB, FILEGROWTH = 20%)
LOG ON
(NAME=compañía_log, FILENAME = 'c:\Archivos de
programa\Microsoft SQL Server\mssql\data\compañía.ldf',
SIZE=3 MB, MAXSIZE=5 MB, FILEGROWTH=1 MB)
```

3. Modificando base de datos

- Administrando el crecimiento de archivos de datos y de transacciones
- Expandiendo un archivo de transacciones
- Reduciendo el tamaño de una base de datos o un archivo
- Eliminando una base de datos

Administrando el crecimiento de los datos y el registro de transacciones



```
ALTER DATABASE compañía
MODIFY FILE (NAME = compañía_log, SIZE = 10 MB)
GO
ALTER DATABASE compañía
ADD FILE(NAME = compañía_data2, FILENAME = 'c:\Archivos de
programa\Microsoft SQL Server\mssql\data\compañía2.ndf',
SIZE = 10 MB, MAXSIZE = 15 MB)
```

Eliminando una base de datos

- Métodos para eliminar una base de datos
 - Administrador corporativo
 - Sentencia DROP DATABASE

```
DROP DATABASE compañía
```

- Nos es posible eliminar una base de datos cuando...
 - Mientras se este restaurando
 - Cuando un usuario esta conectado
 - Cuando se esta publicando como parte de una replicación

5. Tipos de datos

- **Categorías de tipos de datos del sistema:**
 - Binarios
 - Carácter
 - Carácter unicode
 - Fecha y hora
 - Numéricos exactos
 - Numéricos aproximados
- Identificador global
Entero
Monetario
Especiales
Texto e imagen
Texto unicode

Creando y eliminando tipos de datos

- Creando

```
EXEC sp_addtype isbn, 'smallint', 'NOT NULL'
```

```
EXEC sp_addtype dni, 'char(8)', 'NULL'
```

```
EXEC sp_addtype longsstring, 'varchar(63)'
```

Los tipos creados se almacenan en la tabla systypes de la base de datos master

- Eliminando

```
EXEC sp_droptype isbn
```

6. Creando tablas

- Creando y eliminando tablas
- Agregando y eliminando una columna
- Generando valores de columna

Creación y eliminando tablas

- Creando tablas

```
CREATE TABLE empleado
(
  Codigo      int           NOT NULL,
  Nombre      char(40)      NOT NULL,
  Dirección   varchar(40)   NULL,
  FechaNac   smalldatetime NOT NULL,
  Sexo        char(1)       NOT NULL,
  Dni         char(8)       NOT NULL,
  Salario     smallmoney    NOT NULL
)
```

- Eliminado tablas

```
DROP TABLE empleado
```

Adicionando y eliminando columnas

- Adicionar

```
ALTER TABLE empleado ADD Telefono char(6) NULL
```



Nss	Nombre	Direccion	FechaNac	Telefono

- Eliminar

```
ALTER TABLE empleado DROP COLUMN FechaNac
```



Generando valores de columna

- Usando la propiedad IDENTITY
 - Considera el valor inicial
 - Considera el valor de incremento

```
CREATE TABLE Producto
(
    IdProducto int IDENTITY(1,1),
    NombreProducto varchar(30) NOT NULL
)
```

- Usando la función NEWID y el tipo de dato UNIQUEIDENTIFIER
 - Asegurar valores únicos globales
 - Usarlo con la restricción DEFAULT.

7. Generar scripts

- Genera esquema como un script T-SQL
 - Mantener un script backup
 - Crear o actualizar el script de desarrollo de la base de datos
 - Crear un ambiente de desarrollo o pruebas
 - Entrenar nuevos empleado
- Cómo generar
 - Base de datos completa dentro de un solo archivo de script
 - Esquema de sólo tablas
 - Esquema de tablas e índices

Implementando integridad de datos

1. Tipos de integridad de datos
2. Forzando la integridad de datos
3. Definiendo restricciones
4. Usando valores predeterminados y reglas
5. Definiendo que método usar para forzar la integridad de datos

1. Tipos de integridad de datos

Integridad de dominio (columnas)



Blue	Blue	Blue
White	Blue	White
Blue	Blue	Blue
White	Blue	White

Integridad de entidad (filas)



Integridad referencial (entre tablas)

Teal	Teal	Teal
White	White	White
White	White	White

2. Forzando la integridad de datos

- Integridad de datos declarativa
 - Criterios definidos en la definición de objetos
 - Forzando automáticamente por SQL Server
 - Implementar usando restricciones, predeterminados y reglas
- Integridad de datos procedimental
 - Criterios definidos en scripts
 - Forzados por scripts
 - Implementar usando desencadenadores y procedimientos almacenados

3. Usando restricciones

Tipos de integridad	Tipos de restricciones
Dominio	DEFAULT
	CHECK
	FOREIGN KEY
Entidad	PRIMARY KEY
	UNIQUE
Referencial	FOREIGN KEY
	CHECK

Definiendo restricciones

- Use en CREATE TABLE o en ALTER TABLE
- Puede agregar restricciones a tablas con datos existentes
- Puede considerar una o múltiples columnas
 - Una columna, llamada restricción a nivel de columna
 - Múltiples columnas, llamada restricciones a nivel de tabla

Consideraciones para usar restricciones

- Pueden ser creadas sin volver a crear la tabla
- Requiere chequeo de error en las aplicaciones
- Verifica datos ya existentes en la tabla

Restricciones DEFAULT

- Se usa sólo en sentencias INSERT
- Sólo una restricción DEFAULT por columna
- No puede ser usada con la propiedad IDENTITY o tipo de datos *timestamp*
- Permite especificar algunos valores propios del sistema

```
CREATE TABLE Producto
(  IdProducto int NOT NULL,
   NombreProducto varchar(30) NOT NULL,
   FechaIngreso smalldatetime DEFAULT GETDATE()
)
```

```
CREATE TABLE Producto
(  IdProducto int NOT NULL,
   NombreProducto varchar(30) NOT NULL,
   FechaIngreso smalldatetime NOT NULL,
)
ALTER TABLE Producto
ADD DEFAULT GETDATE() FOR FechaIngreso
```

Restricciones CHECK

- Son usados con la sentencia INSERT y UPDATE
- Puede referenciar otras columnas en la misma tabla
- No puede:
 - Ser usado con la propiedad IDENTITY
 - Ser usado con los tipos de datos *timestamp* o *uniqueidentifier*

```
CREATE TABLE Cliente
(
  Dni char(8) NOT NULL,
  Nombre varchar(30) NOT NULL,
  Apellidos varchar(40) NOT NULL,
  Sexo char(1) NOT NULL,
  Salario smallmoney NOT NULL,
  CONSTRAINT CK_Cliente_Dni CHECK
    (Dni LIKE '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),
  CONSTRAINT CK_Cliente_Sexo CHECK (Sexo IN ('M', 'F')),
  CONSTRAINT CK_Cliente_Salario CHECK
    (Salario>0 And Salario<=2500)
)
```


Restricciones PRIMARY KEY

- Sólo una restricción PRIMARY KEY por tabla
- Los valores deben de ser únicos
- Los valores nulos no son almacenados
- Crea un índice único en columnas especificadas

```
CREATE TABLE Producto
(
    IdProducto int NOT NULL
        CONSTRAINT PK_Producto PRIMARY KEY,
    NombreProducto varchar(30) NOT NULL,
    FechaIngreso smalldatetime NOT NULL,
)
```

```
CREATE TABLE Producto
(
    IdProducto int NOT NULL,
    NombreProducto varchar(30) NOT NULL,
    FechaIngreso smalldatetime NOT NULL,
)
ALTER TABLE Producto
ADD CONSTRAINT PK_Producto PRIMARY KEY (IdProducto)
```

Restricciones UNIQUE

- Permite nulos
- Permite múltiples restricciones UNIQUE en una tabla
- Definido en una o más columnas
- Crea un índice único

```
CREATE TABLE Alumno
(
    Codigo char(6) NOT NULL PRIMARY KEY,
    Nombre varchar(20) NOT NULL
        CONSTRAINT UQ_Alumno_Nombre UNIQUE,
    Direccion varchar(30) NOT NULL,
)
```

```
CREATE TABLE Alumno
(
    Codigo char(6) NOT NULL PRIMARY KEY,
    Nombre varchar(20) NOT NULL,
    Direccion varchar(30) NOT NULL )
ALTER TABLE Alumno
ADD CONSTRAINT UQ_Alumno_Nombre UNIQUE(Nombre)
```

Restricciones FOREIGN KEY

- Debe referenciar a una restricción PRIMARY KEY o UNIQUE
- Una o múltiples columnas
- No crea automáticamente índices

```
ALTER TABLE empleado  
ADD CONSTRAINT FK_empleado_departamento  
FOREIGN KEY (Nd)  
REFERENCES departamento(NumeroD)
```

Acciones de integridad referencial

- Se aplica en sentencias DELETE y UPDATE
- Se tienen dos opciones: NO ACTION y CASCADE
- Por defecto es restrictiva NO ACTION

```
ALTER TABLE dependiente  
ADD  
CONSTRAINT FK_dependiente_empleado  
FOREIGN KEY (NssE)REFERENCES empleado(Nss)  
ON DELETE CASCADE ON UPDATE CASCADE
```

Definir que método usar para forzar la integridad de datos

Componentes de integridad de datos	Funcionalidad	Sobrecarga	Antes o después de la transacción
Restricciones	Medio	Bajo	Antes
Predeterminados y reglas	Bajo	Medio	Antes
Desencadenadores	Alto	Alto	Antes y en lugar de

Practicas recomendadas

- Usar las restricciones porque cumplen con el estándar ANSI
- Use desencadenadores cuando una funcionalidad adicional es requerida

Administrador Corporativo de SQL Server 2000

Servidor registrado

Categorías de objetos de base de datos

Instancia con nombre

Modificar diagrama 'AdventureWorks2000Diagram' en 'Adventur...

Nombre	Propietario	Tipo	Fecha de creación
Address	dbo	Usuario	16/07/200...
AddressType	dbo	Usuario	16/07/200...
AWBuildVersion	dbo	Usuario	16/07/200...
BillOfMaterials	dbo	Usuario	16/07/200...
Contact	dbo	Usuario	16/07/200...
ContactType	dbo	Usuario	16/07/200...
CountryCurrency	dbo	Usuario	16/07/200...
CountryRegion	dbo	Usuario	16/07/200...
CreditCard	dbo	Usuario	16/07/200...
Currency	dbo	Usuario	16/07/200...
CurrencyRate	dbo	Usuario	16/07/200...
Customer	dbo	Usuario	16/07/200...
Customer Address	dbo	Usuario	16/07/200...
Department	dbo	Usuario	16/07/200...
Document	dbo	Usuario	16/07/200...
dtproperties	dbo	Usuario	16/07/200...
Employee	dbo	Usuario	16/07/200...
EmployeeDepartment-hist	dbo	Usuario	16/07/200...
EmployeePayHistory	dbo	Usuario	16/07/200...
EmployeeReview	dbo	Usuario	16/07/200...
Illustration	dbo	Usuario	16/07/200...
Individual	dbo	Usuario	16/07/200...
JobCandidate	dbo	Usuario	16/07/200...
Locale	dbo	Usuario	16/07/200...
Location	dbo	Usuario	16/07/200...
Product	dbo	Usuario	16/07/200...
ProductCategory	dbo	Usuario	16/07/200...
ProductCostHistory	dbo	Usuario	16/07/200...
ProductDealerPriceHistory	dbo	Usuario	16/07/200...
ProductDescription	dbo	Usuario	16/07/200...
ProductDescriptionXLocale	dbo	Usuario	16/07/200...
ProductInventory	dbo	Usuario	16/07/200...
ProductListPriceHistory	dbo	Usuario	16/07/200...
ProductModel	dbo	Usuario	16/07/200...
ProductModelXProductDes	dbo	Usuario	16/07/200...
ProductPhoto	dbo	Usuario	16/07/200...

Nombre de columna	Tipo de datos	Longitud	mitr valores nvl
ContactID	int	4	
Salutation	nvarchar	8	✓
FirstName	Name (nvarchar)	50	✓
MiddleName	Name (nvarchar)	50	✓
LastName	Name (nvarchar)	50	✓
NameStyle	NameStyle (bit)	1	
Phone	Phone (nvarchar)	25	✓
Suffix	nvarchar	10	✓
EmailAddress	nvarchar	50	✓
Password	nvarchar	30	✓
EmailPromotion	int	4	✓
ModifiedDate	datetime	8	

Descripción

Valor predeterminado

Precisión 10

Escala 0

Identidad Sí (no disponible para replicación)

Inicialización de identidad 1

Incremento de identidad 1

Columna RowGuid No

Fórmula

Intercalación

Vista de detalle de objetos

Vista de diagrama de base de datos

Vista de diseño de tabla

Analizador de Consultas de SQL Server 2000

Base de datos activa

Editor

The screenshot shows the SQL Server Enterprise Manager interface. On the left is the 'Examinador de objetos' (Object Explorer) showing a tree view of the 'Academico' database. The main window is split into two panes: the top pane is the 'Editor' containing SQL code for altering tables and inserting data, and the bottom pane is the 'Panel de resultados' (Results Grid) displaying a table of customer data.

```
ALTER TABLE Periodo
ADD CONSTRAINT PK_Periodo PRIMARY KEY (codigo_Per)
go

-- Insertando datos en tabla Periodo
INSERT INTO Periodo (codigo_Per, nombre_Per) VALUES ('S', 'SEMESTRAL')
INSERT INTO Periodo (codigo_Per, nombre_Per) VALUES ('A', 'ANUAL')

-----
-- PASO 02
-- Alterando la estructura de la tabla SemestreFacultad (se convertira en SemestrePeriodoFacultad)

ALTER TABLE SemestreFacultad
DROP CONSTRAINT PK_semestreFacultad
GO
ALTER TABLE SemestreFacultad
DROP CONSTRAINT DF_SemestreFacultad_estado_Sfa
GO
ALTER TABLE SemestreFacultad
DROP COLUMN estado_Sfa
GO
ALTER TABLE SemestreFacultad
ADD codigo_Per char(1) NULL
```

	CustomerID	CompanyName	ContactName	ContactTitle	Address
1	ALFKI	Alfreds Futterkiste	Maria Anders	Sales Representative	Obere Str. 57
2	ANATR	Ana Trujillo Emparedados y helados	Ana Trujillo	Owner	Avda. de la Constitución 22
3	ANTON	Antonio Moreno Taqueria	Antonio Moreno	Owner	Mataderos 2312
4	AROUT	Around the Horn	Thomas Hardy	Sales Representative	120 Hanover Sq.
5	BERGS	Berglunds snabbköp	Christina Berglund	Order Administrator	Berguvsvägen 8
6	BLAUS	Blauer See Delikatessen	Hanna Moos	Sales Representative	Forsterstr. 57
7	BLONP	Blondesdél père et fils	Frédérique Citeaux	Marketing Manager	24, place Kléber
8	BOLID	Bólido Comidas preparadas	Martin Sommer	Owner	C/ Araquil, 67
9	BONAP	Bon app'	Laurence Lebihan	Owner	12, rue des Bouchers
10	BOTM	Bottom-Dollar Markets	Elizabeth Lincoln	Accounting Manager	23 Tsawassen Blvd.
11	BSBEV	B's Beverages	Victoria Ashworth	Sales Representative	Fauntleroy Circus
12	CACTU	Cactus Comidas para llevar	Patricio Simpson	Sales Agent	Cerrito 333
13	CENTC	Centro comercial Moctezuma	Francisco Chang	Marketing Manager	Sierras de Granada 9993

Examinador de objetos
(muestra las bases de datos y sus objetos)

Funciones (agrupadas en categorías)

Plantillas de código

Panel de mensajes

Panel de resultados

Analizador de SQL Server 2000

Información de los eventos de Servidor

Barra de herramientas desde la cual podemos administrar las trazas de Servidor

Instrucciones SQL procesadas por el servidor

The screenshot shows the SQL Profiler interface with the following table of events:

EventClass	TextData	ApplicationName	NTUserName	LoginName	CPU	Reads	Writes	Duration	ClientProcessID	SPID	StartTime
SQL:BatchCompleted	SELECT codigo_Fac, codigo_Sem, 'A' ...	Analizador ...	JorgeR...	SERVI...	0	55	0	0	3784	54	2005-08-
SQL:BatchCompleted	SELECT DISTINCT S.codigo_Sem, P.cod...	Analizador ...	JorgeR...	SERVI...	0	52	0	0	3784	54	2005-08-
SQL:BatchCompleted	SELECT codigo_Sem, codigo_Per, 1 F...	Analizador ...	JorgeR...	SERVI...	0	41	0	46	3784	54	2005-08-
Audit Login	-- network protocol: TCP/IP set quo...	Academico	ASPNET	SERVI...					2228	55	2005-08-
SQL:BatchCompleted	set implicit_transactions on SET NO...	Academico	ASPNET	SERVI...	0	0	0	0	2228	55	2005-08-
RPC:Completed	declare @P1 int set @P1=1 exec sp_p...	Academico	ASPNET	SERVI...	0	64	0	30	2228	55	2005-08-
RPC:Completed	exec sp_unprepare 1	Academico	ASPNET	SERVI...	0	0	0	0	2228	55	2005-08-
SQL:BatchCompleted	SET NO_BROWSETABLE OFF	Academico	ASPNET	SERVI...	0	0	0	0	2228	55	2005-08-
SQL:BatchCompleted	SELECT codigo_Fac,nombre_Fac,abrevi...	Academico	ASPNET	SERVI...	0	48	0	63	2228	55	2005-08-
SQL:BatchCompleted	IF @@TRANCOUNT > 0 COMMIT TRAN	Academico	ASPNET	SERVI...	0	0	0	0	2228	55	2005-08-
Audit Logout		Academico	ASPNET	SERVI...	0	172	0	576	2228	55	2005-08-
Audit Login	-- network protocol: TCP/IP set quo...	Academico	ASPNET	SERVI...					2228	55	2005-08-
SQL:BatchCompleted	set implicit_transactions on SET NO...	Academico	ASPNET	SERVI...	0	0	0	0	2228	55	2005-08-
RPC:Completed	declare @P1 int set @P1=1 exec sp_p...	Academico	ASPNET	SERVI...	15	192	0	46	2228	55	2005-08-
RPC:Completed	exec sp_unprepare 1	Academico	ASPNET	SERVI...	0	0	0	0	2228	55	2005-08-
SQL:BatchCompleted	SET NO_BROWSETABLE OFF	Academico	ASPNET	SERVI...	0	0	0	0	2228	55	2005-08-
SQL:BatchCompleted	SELECT e.codigo_Esc, e.nombre_Esc, ...	Academico	ASPNET	SERVI...	16	269	0	46	2228	55	2005-08-
SQL:BatchCompleted	IF @@TRANCOUNT > 0 COMMIT TRAN	Academico	ASPNET	SERVI...	0	0	0	0	2228	55	2005-08-
Audit Logout		Academico	ASPNET	SERVI...	31	521	0	93	2228	55	2005-08-

The SQL query being executed is:

```
declare @P1 int
set @P1=1
exec sp_prepare @P1 output, NULL, N'SELECT e.codigo_Esc, e.nombre_Esc, e.director_esc, e.abreviatura_esc, ''vigencia''=CASE e.vigencia_esc WHEN 1 THEN ''Vigente''
select @P1
```