



BASE DE DATOS AVANZADO I

Unidad 1: Base de Datos y Tablas

Tema 2: Manejo de Tablas y Esquemas





Tema 2: Manejo de Tablas y Esquemas

1.2. Tema 2: Manejo de Tablas y Esquemas

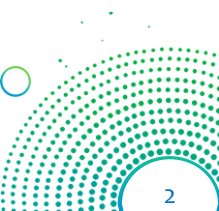
1.2.1. Creación de Esquemas

1.2.2. Creación de tipos de datos de usuario

1.2.3. Creación de una tabla

1.2.4. Modificación de la estructura de una tabla

1.2.5. Manejo de una tabla particionada





Capacidades

1. Identifica las opciones de base de datos, tablas e índices.
2. Diseña e implementa una base de datos optimizada, así como las tablas, esquemas e índices garantizando disponibilidad y rendimiento.





1. Manejo de esquemas

¿Qué es un esquema o schema?

- Un esquema es un namespace que permite organizar objetos.
- Contenedor de objetos.
- Permite gestionar seguridad.
- El schema predeterminado es DBO.





1. Manejo de esquemas

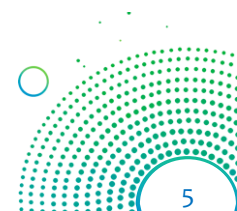
Creación de Schema

- Se utiliza el comando CREATE SCHEMA.
- Sintaxis: CREATE SCHEMA schema_name
- Ejemplo: Crear en la Base de Datos BDTravel, los schemas Ventas y RRHH.

```
Use BDTravel
go

Create Schema Ventas
go

Create Schema RRHH
go
```





1. Manejo de esquemas

Visualizar los Schemas creados en la Base de Datos

- Se utiliza el catálogo SYS.SCHEMAS.
- Ejemplo: Ver los schemas creados.

```
Select * from sys.schemas  
go
```





1. Manejo de esquemas

Modificar Schema: Transferir objetos de un esquema a otro

- Se utiliza el comando ALTER SCHEMA.
- Sintaxis: ALTER SCHEMA schema_name
 TRANSFER schema_name. Object_name
- Ejemplo: Transferir el objeto tabla dbo.TBPromocion hacia el esquema Ventas.

```
- Alter Schema Ventas
  Transfer dbo.TBPromocion
go
```



1. Manejo de esquemas

Eliminar un Schema

- Se utiliza el comando DROP SCHEMA.
- Sintaxis: DROP SCHEMA schema_name
- Ejemplo: Eliminar el Schema MKT, previamente creado.

```
Create Schema MKT  
go
```

```
Drop Schema MKT  
go
```



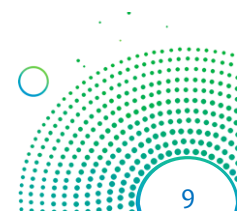
El schema a eliminar debe estar sin objetos.



2. Tipos de datos de usuario

Tipo de dato

- Antes de crear una tabla, debe definir los tipos de datos para la tabla.
- Los tipos de datos especifica el tipo de información (caracteres, números o fechas) que una columna puede contener, además de cómo se almacenan los datos.
- Microsoft SQL Server proporciona varios tipos de datos del sistema. SQL Server también permite que el usuario defina los tipos de datos que se basan en datos del sistema de tipos.





2. Tipos de datos de usuario

Tipo de dato

Numérico exacto

| Data type | Range | Storage |
|---------------|---|--------------|
| bigint | -2^{63} (-9,223,372,036,854,775,808) to $2^{63}-1$ (9,223,372,036,854,775,807) | 8 Bytes |
| int | -2^{31} (-2,147,483,648) to $2^{31}-1$ (2,147,483,647) | 4 Bytes |
| smallint | -2^{15} (-32,768) to $2^{15}-1$ (32,767) | 2 Bytes |
| tinyint | 0 to 255 | 1 Byte |
| money | -922,337,203,685,477.5808 to 922,337,203,685,477.5807 | 8 Bytes |
| smallmoney | -214,748.3648 to 214,748.3647 | 4 Bytes |
| Numeric (p,s) | (p = Precisión) Número total de dígitos. Máximo 38 (s = Scale) Número de decimales. Máximo es p-1 | 5 - 17 Bytes |
| Decimal (p,s) | (p = Precisión) Número total de dígitos. Máximo 38 (s = Scale) Número de decimales. Máximo es p-1 | 5 - 17 Bytes |
| Bit | Dígito binario | 1 Byte |



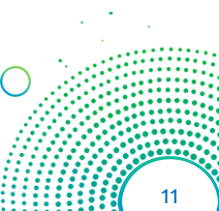


2. Tipos de datos de usuario

Tipo de dato

Numerico aproximado

| Data type | Range | Storage |
|-----------|---|----------------------|
| float | - 1.79E+308 to -2.23E-308, 0 and 2.23E-308 to 1.79E+308 | Depends on the value |
| real | - 3.40E + 38 to -1.18E - 38, 0 and 1.18E - 38 to 3.40E + 38 | 4 Bytes |





2. Tipos de datos de usuario

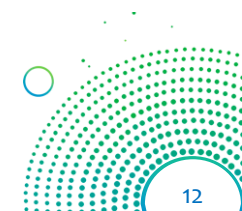
Tipo de dato

Fecha y Hora

| Data type | Range | Storage |
|-------------|---|-------------|
| Date | 1/1/0001 - 31/12/9999 | 3 Bytes |
| DateTime | January 1, 1753, through December 31, 9999 / 00:00:00 thr | 8 Bytes |
| DateTime2 | 0001-01-01 through 9999-12-31 / 00:00:00 through 23:59:5 | 6 - 8 Bytes |
| SmallDateTi | 1900-01-01 through 2079-06-06 / 00:00:00 through 23:59: | 4 Bytes |
| Time | 00:00:00.0000000 through 23:59:59.9999999 | 5 Bytes |

Datos Binarios

| Data type | Range | Storage |
|--------------|-------------------------------------|-----------|
| Binary (n) | Datos binarios de longitud fija | Hasta 2GB |
| VarBinary(n) | Datos binarios de longitud variable | Hasta 2GB |



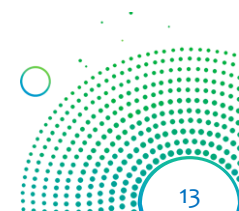


2. Tipos de datos de usuario

Tipo de dato

Cadena de Caracteres

| Data type | Range | Storage |
|-------------|---|-----------|
| Char(n) | (n=longitud fija de cadena no UNICODE) hasta 8000 | Hasta 2GB |
| Varchar(n) | (n=longitud variable de cadena no UNICODE) hasta 8000 | Hasta 2GB |
| NChar(n) | (n=longitud fija de cadena UNICODE) hasta 8000 | Hasta 2GB |
| NVarchar(n) | (n=longitud variable de cadena UNICODE) hasta 8000 | Hasta 2GB |
| Text | Texto amplio de caracteres NO UNICODE | Hasta 2GB |
| Ntext | Texto amplio de caracteres UNICODE | Hasta 2GB |

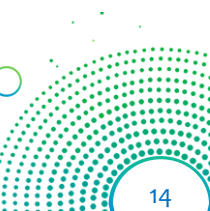




2. Tipos de datos de usuario

Tipo de dato definido por el usuario

- Permiten afinar aún más los tipos de datos para garantizar la coherencia cuando se trabaja con elementos de datos comunes en diferentes tablas o bases de datos.
- Los tipos de datos definidos por el usuario no permiten definir estructuras o tipos de datos complejos.

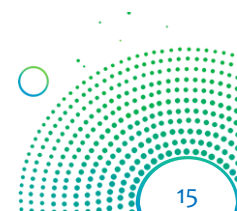




2. Tipos de datos de usuario

¿Cómo se crea un tipo de dato de usuario?

- Existen 02 formas de crear tipos de datos:
 - Usando el Stored Procedure del sistema SP_ADDTYPE
 - Usando el comando CREATE TYPE
- Utilizando CREATE TYPE, es mas directo, se actualiza automáticamente en el servidor, se crea a través de una clase de un ensamblado de Microsoft. NET Framework (CLR).





2. Tipos de datos de usuario

Usando SP_ADDTYPE

- Sintaxis:
sp_addtype <nombre>, tipo_dato [, 'NULL' | 'NOT NULL']
- Ejemplo: Creando tipo de dato FONO basado en el tipo varchar(15) y GENERO basado en el tipo char(1).

```
SP_ADDTYPE fono, 'varchar(15)', 'NULL'  
GO  
  
SP_ADDTYPE genero, 'char(1)', 'NOT NULL'  
GO
```




2. Tipos de datos de usuario

Usando comando CREATE TYPE

- Sintaxis:
`CREATE TYPE [schema_name.] type_name [NULL | NOT NULL]`
- Ejemplo: Creando tipo de dato TIEMPO basado en el en el tipo datetime.

```
Create Type tiempo from datetime NOT NULL  
go
```



2. Tipos de datos de usuario

¿Cómo eliminar tipo de dato de usuario?

- Para eliminar tipo de datos de usuario también existen 02 formas:
 - Usando el Stored Procedure SP_DROPTYPE
 - Usando el comando DROP TYPE
- Por ejemplo eliminemos el tipo genero usando SP_DROPTYPE

```
SP_DROPTYPE genero  
go
```

- Ahora eliminemos el tipo tiempo usando DROP TYPE

```
Drop Type tiempo  
go
```



3. Manejo de tablas

- *Tipo de dato Tabla*
- Siempre existió la necesidad de pasar información en forma batch, esto es, pasar múltiples registros de información a SQL como parámetro de un Stored Procedure, lo cual tendría un rendimiento superior al pasar de una sola vez una gran cantidad de información.
 - Una solución sería crear tipo de dato tabla, para luego implementar una instancia mediante una variable o parámetro.



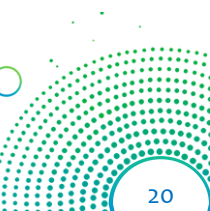


3. Manejo de tablas

Crear tipo de dato Tabla

- Utilizar comando CREATE TYPE
- Sintaxis:

```
CREATE TYPE [Nombre] AS TABLE  
(  
  <campo1> tipo_dato [NULL | NOT NULL],  
  <campo2> tipo_dato [NULL | NOT NULL],  
  <campo2> tipo_dato [NULL | NOT NULL],  
)
```





3. Manejo de tablas

Crear tipo de dato Tabla

- Por ejemplo, crear un tipo CLIENTE con los campos nombre, apellido, email.

```
CREATE TYPE TCliente AS TABLE
(
    nombre      varchar(100) NOT NULL,
    apellido    varchar(100) NOT NULL,
    Email       varchar(200) NOT NULL
)
go
```





3. Manejo de tablas

Utilizando Tipo de dato Tabla

- Ahora declaramos una variable de tipo TABLE creada recientemente, ingresamos registros a ella y lo mostramos.

```
DECLARE @v_TablaCli TCliente
INSERT @v_TablaCli
VALUES
    ('Juan', 'Alva', 'abc@yahoo.com'),
    ('Luis', 'Rios', 'lrrios@yahoo.com'),
    ('Carlos', 'Perez', 'cp@hotmail.com')
SELECT * FROM @v_tablaCli
GO
```



Ejercicio 1

- En la empresa Recobra, se decidió implementar schemas para una mejor organización de sus objetos , estos son: VENTAS, RRHH y FINANZAS.
- Solución:

```
Use BDRcobra  
go
```

```
Create Schema VENTAS  
go
```

```
Create Schema RRHH  
go
```

```
Create Schema FINANZAS  
go
```



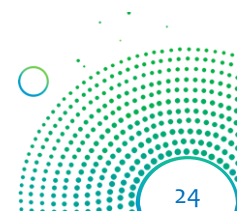
Ejercicio 2

- Por un tema de estándar, en la empresa Recobra, se implementará tipos de datos definidos por el usuario que serán:
TCadena basado en varchar(50).
TMoneda basado en smallmoney.
TTiempo basado en date.
- Solución:

```
Create Type TCadena From Varchar(50) not null  
go
```

```
Create Type TMoneda from Smallmoney not null  
go
```

```
Create Type TTiempo from Date not null  
go
```





1. Manejo de tablas

¿Qué es una TABLA?

- Colección de datos sobre una entidad (Persona, Lugar, Cosa) específica, que tiene un número discreto de atributos designados (por ejemplo cantidad o tipo).
- Las tablas están en el corazón de SQL Server y del modelo relacional en general.

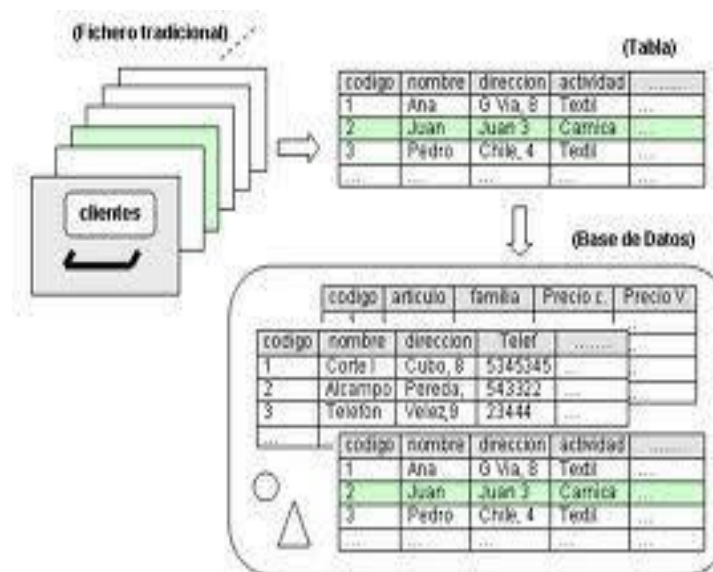


25

1. Manejo de tablas

Creación de una TABLA

- Para crear una tabla debemos especificar :
 - ☐ El nombre que le queremos asignar
 - ☐ El nombre de cada campo
 - ☐ El tipo de dato de cada campo
 - ☐ Características de cada campo
- Tener en cuenta el estándar para crear objetos





1. Manejo de tablas

Creación de una TABLA

- Se utiliza el comando CREATE TABLE.
- Sintaxis:

```
CREATE TABLE schema_name.table_name
(field1_name      dataType      Null | Not null,
 field2_name      dataType      Null | Not null,
 ....
 fieldn_name      dataType      Null | Not null
) On Filegroup_name
```





1. Manejo de tablas

Creación de una TABLA

- Por ejemplo, crear la tabla Empleado en la base de datos BDTravel, en el schema RRHH y en el filegroup FGAdministra.
- Solución:

```
Use BDTravel
go

Create Table RRHH.TBEmpleado
(
    codEmp      char(5)      not null,
    nomEmp      varchar(40)  not null,
    finEmp      tiempo       not null,
    sueEmp      smallmoney   not null,
) on FGAdministra
go
```





1. Manejo de tablas

Ver estructura de una TABLA

- Se utiliza el procedimiento SP_HELP
- Por ejemplo ver la estructura de la tabla Empleado.
- Solución:

```
sp_Help 'RRHH.TBEmpleado'  
go
```

- El resultado es:

Results Messages

| | Name | Owner | Type | Created_datetime | | | | | | |
|---|------------|-------|------------|-------------------------|--|--|--|--|--|--|
| 1 | TBEmpleado | dbo | user table | 2018-05-01 23:38:19.907 | | | | | | |

| | Column_name | Type | Computed | Length | Prec | Scale | Nullable | Trim TrailingBlanks | FixedLenNullInSource | Collation |
|---|-------------|------------|----------|--------|------|-------|----------|---------------------|----------------------|---------------------|
| 1 | codEmp | char | no | 5 | | | no | no | no | Modem_Spanish_CI_AS |
| 2 | nomEmp | varchar | no | 40 | | | no | no | no | Modem_Spanish_CI_AS |
| 3 | finEmp | Tiempo | no | 8 | | | no | (n/a) | (n/a) | NULL |
| 4 | sueEmp | smallmoney | no | 4 | 10 | 4 | no | (n/a) | (n/a) | NULL |

| Identity | | | Seed | Increment | Not For Replication |
|----------|-----------------------------|--|------|-----------|---------------------|
| 1 | No identity column defined. | | NULL | NULL | NULL |

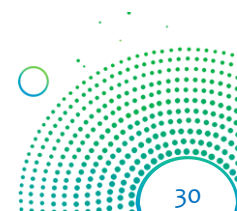
| RowGuidCol | |
|------------|-------------------------------|
| 1 | No rowguidcol column defined. |



2. Modificación de la estructura de una tabla

- Para agregar una o varias columnas a una tabla se ejecuta el comando ALTER TABLE.
- Por ejemplo, adicionar el campo teléfono a la tabla empleado.
- Solución:

```
Alter Table RRHH.TBEmpleado  
Add tlfEmp varchar(14) not null  
go
```





2. Modificación de la estructura de una tabla

- Para modificar las propiedades de un campo de una tabla se ejecuta el comando ALTER TABLE con la opción ALTER COLUMN.
- Por ejemplo, modificar el campo teléfono a la tabla empleado que será tipo de dato entero.
- Solución:

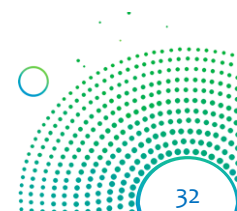
```
Alter Table RRHH.TBEmpleado  
Alter Column tlfEmp int not null  
go
```



2. Modificación de la estructura de una tabla

- Para eliminar un campo de una tabla se ejecuta el comando ALTER TABLE con la opción DROP COLUMN.
- Por ejemplo: Eliminar el campo teléfono de la tabla empleado.
- Solución:

```
Alter Table RRHH.TBEmpleado  
Drop Column tlfEmp  
go
```





2. Modificación de la estructura de una tabla

- Para adicionar campo calculado.
- Por ejemplo: Agregar un campo calculado llamado tiempo de servicio, que será igual a la diferencia de fechas entre la fecha de ingreso y la fecha actual expresado en años.
- Solución:

```
Alter Table RRHH.TBEmpleado  
Add añosServicio As DateDiff(YY, finEmp, Getdate())  
go
```





3. Manejo de una tabla particionada

¿Qué es el particionamiento de tablas ?

- Particionar es el proceso donde tablas muy grandes son divididas en múltiples partes más pequeñas.
- Al separar una tabla grande en tablas individuales más pequeñas, las consultas que acceden sólo a una fracción de los datos pueden correr más rápido porque hay menos datos que escanear.
- El objetivo principal de particionar es ayudar en el mantenimiento de tablas grandes y reducir el tiempo de respuesta general para leer y cargar datos para operaciones SQL particulares.

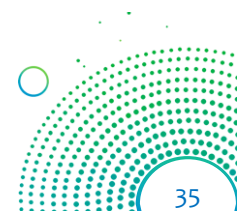
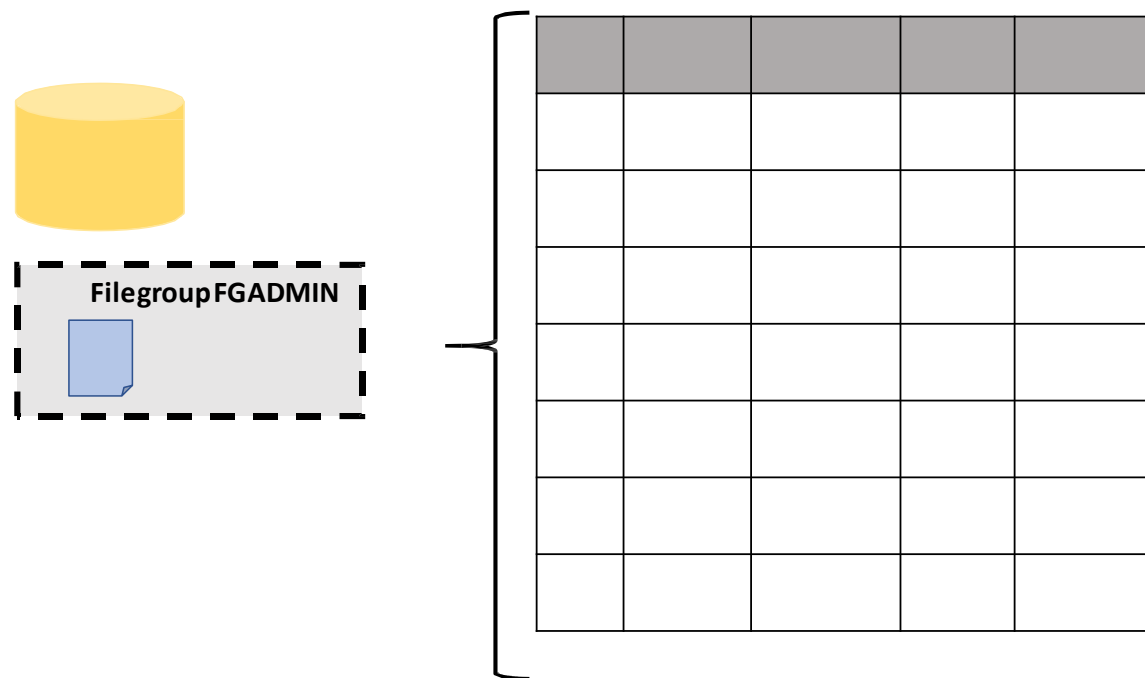




3. Manejo de una tabla particionada

Estructura de una tabla normal

- Este es una estructura normal de una tabla, al ser muy grande, va a demorar en responder las consultas.

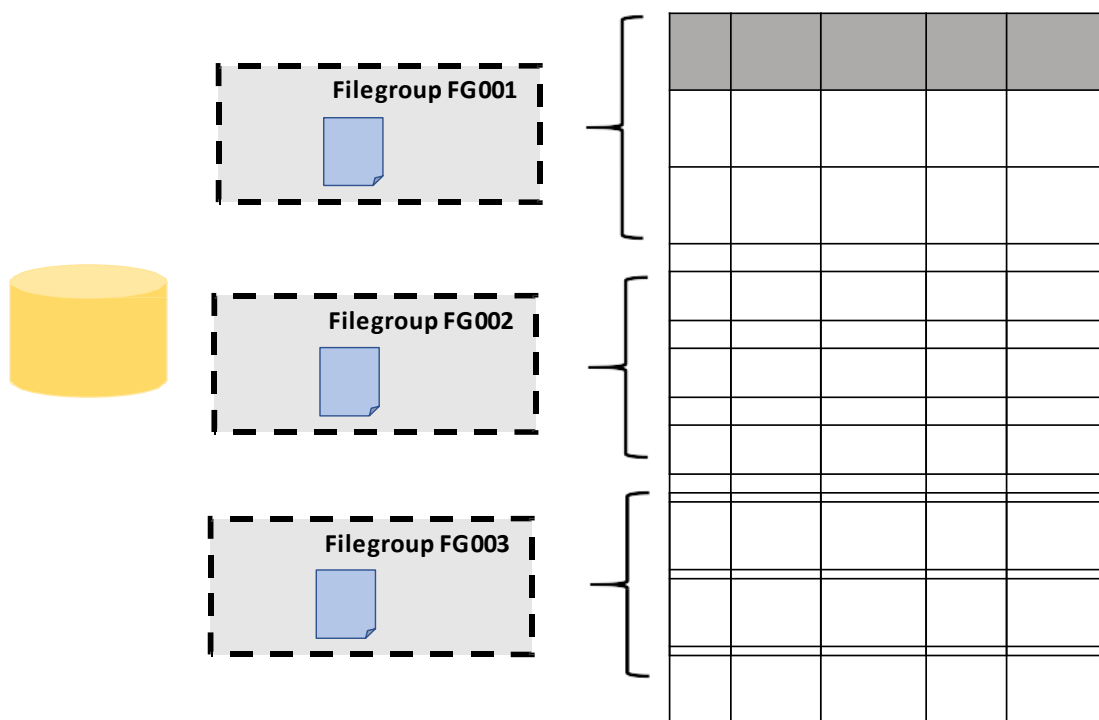




3. Manejo de una tabla particionada

Estructura de una tabla particionada

- Este es la estructura de una tabla particionada, logrando carga balanceada y respuesta rápida a las consultas.

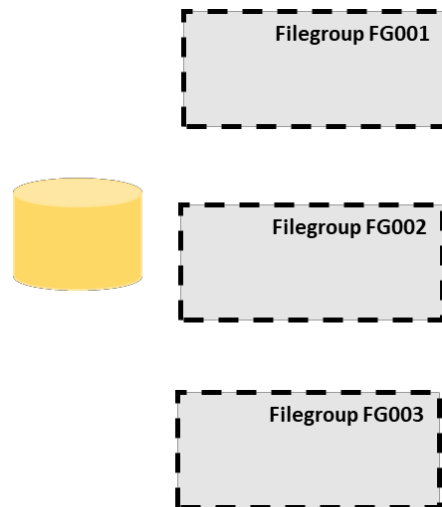




3. Manejo de una tabla particionada

Creación de una Tabla Particionada

- Para crear una tabla particionada, primero se debe crear Filegroups, según el número de particiones, en nuestro ejemplo serán 03.



```
Use master
```

```
go
```

```
[-] Alter Database BDTravel  
    [ Add Filegroup FG001  
        go
```

```
[-] Alter Database BDTravel  
    [ Add Filegroup FG002  
        go
```

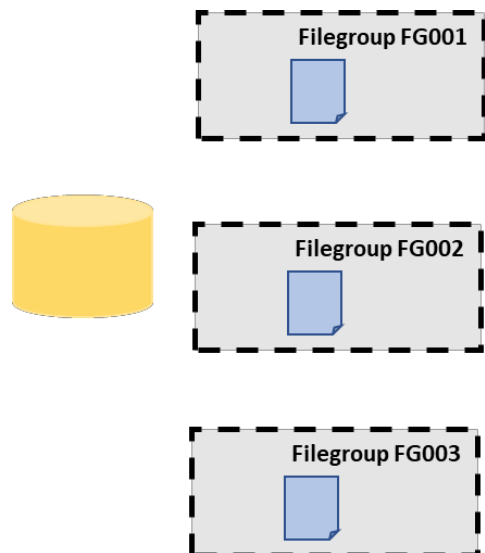
```
[-] Alter Database BDTravel  
    [ Add Filegroup FG003  
        go
```



3. Manejo de una tabla particionada

Creación de una Tabla Particionada

- A continuación adicionar a cada filegroup sus respectivo datafile.



```
Alter Database BDTravel
Add File
(name=DataP01, FileName='D:\Data\DataP01.NDF')
to Filegroup FG001
go

Alter Database BDTravel
Add File
(name=DataP02, FileName='D:\Data\DataP02.NDF')
to Filegroup FG002
go

Alter Database BDTravel
Add File
(name=DataP03, FileName='D:\Data\DataP03.NDF')
to Filegroup FG003
go
```



3. Manejo de una tabla particionada

Creación de una Tabla Particionada

- Luego en la base de datos se crea la función de partición, donde se define el tipo de datos del campo que será el criterio de partición y los rangos de cada partición con los límites Range Left o Range Right.

```
CREATE PARTITION FUNCTION Ejemplo(int)  
AS RANGE LEFT FOR VALUES(100,1000,10000);
```

```
CREATE PARTITION FUNCTION Ejemplo1(int)  
AS RANGE RIGHT FOR VALUES(100,1000,10000);
```

| Partición | Valores (LEFT) | Valores (RIGHT) |
|-----------|-------------------|-------------------|
| 1 | <=100 | <100 |
| 2 | >100 AND <=1000 | >=100 AND <1000 |
| 3 | >1000 AND <=10000 | >=1000 AND <10000 |
| 4 | >10000 | >=10000 |

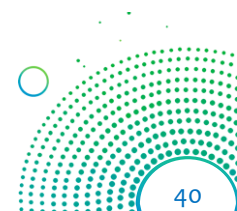


3. Manejo de una tabla particionada

- Creación de una Tabla Particionada
- En el ejemplo el tipo de datos es numérico entero, definiremos usando Range Left. Partición 1 es hasta 2000, Partición 2 hasta el 4000 y en Partición 3 los demás

```
Use BDTravel
go

Create Partition Function fnpNumerador(INT)
As Range Left
for values (2000, 4000)
go
```





3. Manejo de una tabla particionada

Creación de una Tabla Particionada

- A continuación se crea el esquema de partición, mapeando cada partición al Filegroup respectivo

```
[-] Create Partition Scheme scpNumerador  
    As Partition fnpNumerador  
    to ('FG001', 'FG002', 'FG003')  
go
```



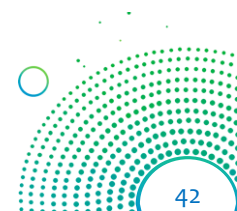


3. Manejo de una tabla particionada

Creación de una Tabla Particionada

- Por último, crear la tabla particionada.

```
CREATE TABLE Ventas.TBVenta
(
    idVenta      Int          not null,
    fecVenta     date         not null,
    valVenta     smallmoney   not null
) ON scpNumerador (idVenta)
go
```

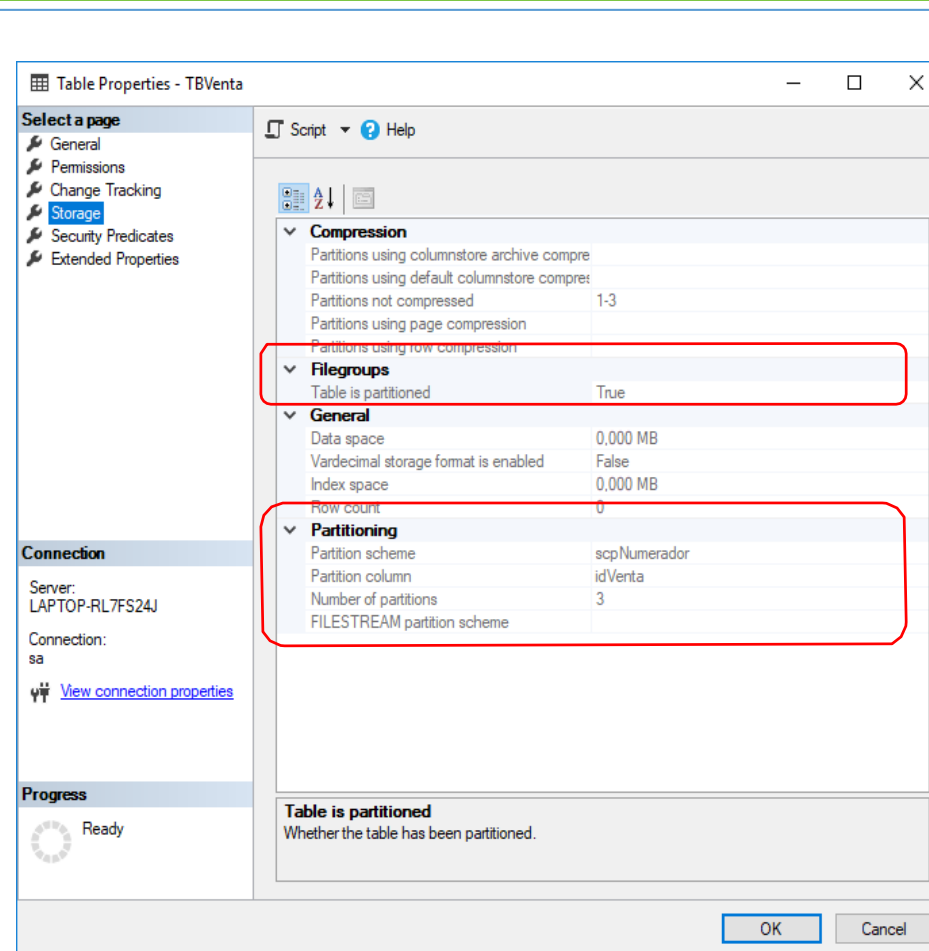




3. Manejo de una tabla particionada

Ver propiedades de Tabla Particionada

- Para verificarla creación de la tabla particionada, visualizamos sus propiedades. Click derecho en la tabla \ opción propiedades. Hacer click en la opción almacenamiento (Storage).



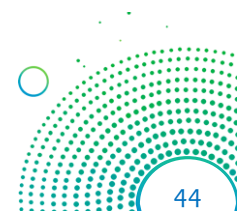


3. Manejo de una tabla particionada

Verificar funcionamiento de la Tabla Particionada

- Ingresaremos 3 registros con id de ventas 1001, 2010 y 4015.

```
Insert Ventas.TBVenta  
Values  
('1001', '10/12/2000', '5500'),  
('2010', '10/10/2010', '6500'),  
('4015', '21/02/2015', '3500')  
go
```





3. Manejo de una tabla particionada

Verificar funcionamiento de la Tabla Particionada

- Ahora visualizaremos en una consulta los registros y el número de partición donde se encuentra almacenado.

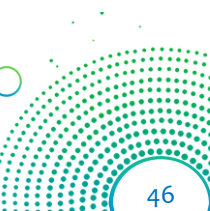
```
Select *,  
        $Partition.fnpNumerador(idVenta) as [Nro Partición]  
From Ventas.TBVenta  
go
```

| | idVenta | fecVenta | valVenta | Nro Partición |
|---|---------|------------|----------|---------------|
| 1 | 1001 | 2000-12-10 | 5500,00 | 1 |
| 2 | 2010 | 2010-10-10 | 6500,00 | 2 |
| 3 | 4015 | 2015-02-21 | 3500,00 | 3 |



Ejercicio

- En la empresa Recobra, se decidió implementar una Tabla particionada para almacenar sus solicitudes y poder optimizar las consultas por trimestre de cualquier año, por un tema de análisis comparativo.



Ejercicio: Solución

1. Crear los filegroups.

```
[-] Alter Database BDRcobra  
    | add filegroup FGTrimestre1  
    | go
```

```
[-] Alter Database BDRcobra  
    | add filegroup FGTrimestre2  
    | go
```

```
[-] Alter Database BDRcobra  
    | add filegroup FGTrimestre3  
    | go
```

```
[-] Alter Database BDRcobra  
    | add filegroup FGTrimestre4  
    | go
```

Ejercicio: Solución

- Adicionar los datafiles.

```
Alter Database BDRcobra  
add file  
(name=DataTrimestre1,filename='D:\Data\DataTrimestre1.NDF')  
to filegroup FGTrimestre1  
go
```

```
Alter Database BDRcobra  
add file  
(name=DataTrimestre2,filename='D:\Data\DataTrimestre2.NDF')  
to filegroup FGTrimestre2  
go
```

```
Alter Database BDRcobra  
add File  
(name=DataTrimestre3,filename='D:\Data\DataTrimestre3.NDF')  
to filegroup FGTrimestre3  
go
```

```
Alter Database BDRcobra  
add File  
(name=DataTrimestre4,filename='D:\Data\DataTrimestre4.NDF')  
to filegroup FGTrimestre4  
go
```


Ejercicio: Solución

- Crear la Función de Partición.

```
Use BDRcobra
```

```
go
```

```
[-] Create Partition Function fnpTrimestre (tinyint)
```

```
  | As Range Left for values ('3','6','9')
```

```
go
```



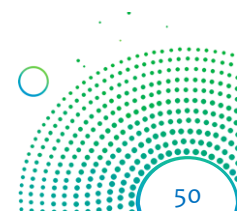
Ejercicio: Solución

- Crear el esquema de Partición.

```
[-] Create Partition scheme scpTrimestre
    As Partition fnpTrimestre
    to ('FGTrimestre1', 'FGTrimestre2', 'FGTrimestre3', 'FGTrimestre4')
    go
```

- Crear la tabla Particionada

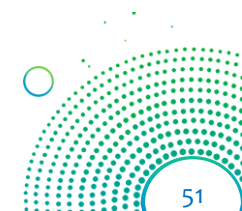
```
[-] Create Table dbo.TBSolicitud
(
    nroSolicitud      int          not null,
    titSolicitus      varchar(50) not null,
    fecSolicitud       date         not null,
    Trimestre          tinyint      not null
) On scpTrimestre (Trimestre)
go
```





Conclusiones

- Las tablas están en el core de SQL Server y del modelo relacional en general.
- Las tablas particionadas ayudan a optimizar el rendimiento
- obteniendo mejores tiempos de respuestas de las consultas.





Conclusiones

- Los esquemas nos permiten organizar objetos de una base de datos como tablas, vistas, procedimientos, etc. También nos ayuda a gestionar el tema de seguridad.
- Los tipos de datos definidos por el usuario nos permite estandarizar y evitar conflictos de datos en las aplicaciones dentro de una organización.





Bibliografía

- Microsoft (2017) Schemas. Recuperado de: <https://docs.microsoft.com/en-us/sql/t-sql/statements/create-schema-transact-sql?view=sql-server-2017>
- Microsoft (2017) Create type. Recuperado de: <https://docs.microsoft.com/en-us/sql/t-sql/statements/create-type-transact-sql?view=sql-server-2017>



GRACIAS

**SEDE LIMA CENTRO**

Av. Uruguay 514
Cercado – Lima
Teléfono: 419-2900

SEDE INDEPENDENCIA

Av. Carlos Izaguirre 233
Independencia – Lima
Teléfono: 633-5555

SEDE BREÑA

Av. Brasil 714 – 792
(CC La Rambla – Piso 3)
Breña – Lima
Teléfono: 633-5555

SEDE TRUJILLO

Calle Borgoño 361
Trujillo
Teléfono: (044) 60-2000

SEDE SAN JUAN DE LURIGANCHO

Av. Próceres de la Independencia 3023-3043
San Juan de Lurigancho – Lima
Teléfono: 633-5555

SEDE BELLAVISTA

Av. Mariscal Oscar R. Benavides 3866 – 4070
(CC Mall Aventura Plaza)
Bellavista – Callao
Teléfono: 633-5555

SEDE AREQUIPA

Av. Porongoche 500
(CC Mall Aventura Plaza)
Paucarpata - Arequipa
Teléfono: (054) 60-3535