

# **Beneficios y Ventajas del Particionado de Datos con SQL Server 2005 Enterprise Edition.**



# Características de gama alta (relacional)

- Clustering (para más de 2 nodos)
- Database Mirroring (espejado de BD)
  - Modo asíncrono, REDO multithread
  - Database Snapshot (instantáneas de BD)
- Método de la recuperación de la BD
  - Más rápida: Redo --> online --> undo
- Recuperación de páginas y ficheros
  - En línea, sin pérdida de disponibilidad
- Operaciones sobre índices
  - En paralelo, en línea
- Particionamiento de datos
  - Vistas particionadas y distribuidas actualizables
  - Particionado real de datos

# Particionado de datos



- ¿Usas vistas distribuidas particionadas?
- ¿"Implementas" particiones horizontales?
- ¿Necesitas hacer archivado de datos?
- ¿Necesitas dividir las tablas físicamente?



SOLID  
QUALITY  
MENTORS

# Particionado de datos en SQL2005 EE

- **Agenda**

- Qué es el particionado de datos
- Cómo se implementa
- Mantenimiento

# ¿Qué es el particionado de datos?

- Los registros de una tabla se dividen en diferentes trozos (como tablas individuales)
- Todas las filas siguen perteneciendo a la misma tabla lógica

Sin particionar



A, B, C, D, E, F, G, H, I, J, K, L

Particionada



A, B, C

D, E, F

G, H, I

J, K, L

# ¿Qué se puede particionar?

- Objetos que se pueden particionar
  - Tablas
  - Índices (agrupados o no agrupados)
  - Vistas indexadas
- La fila es la unidad más pequeña de particionado
- Únicamente particionado horizontal
  - Particionado vertical... ¿necesario?
  - Simulado con vistas sobre tablas con relación 1:1

# Tabla de particionado

**Tabla**

| <b>ID</b> | <b>c1</b> | <b>c2</b> | <b>c3</b> | <b>c4</b> |
|-----------|-----------|-----------|-----------|-----------|
| 1         | 3         | A         | ...       | ...       |
| 2         | 5         | B         | ...       | ...       |
| 3         | 2         | B         | ...       | ...       |
| 4         | 1         | L         | ...       | ...       |
| 5         | 5         | Y         | ...       | ...       |
| 6         | 5         | A         | ...       | ...       |
| 7         | 2         | F         | ...       | ...       |

**Función de  
particionado**



**3**



**5**



**2**



**1**



**5**



**5**



**2**

**Esquema de  
particionado**



Filegroup1



Filegroup2



Filegroup1



Filegroup4



Filegroup2



Filegroup2



Filegroup1

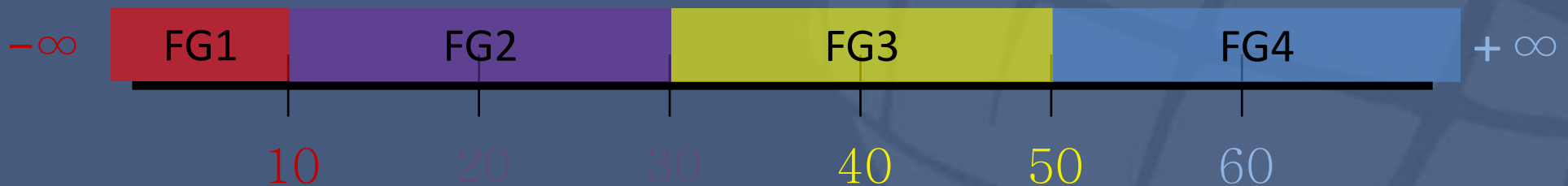


# Creación de una tabla particionada

```
CREATE PARTITION FUNCTION pfn (int)
AS RANGE LEFT FOR VALUES (10, 30, 50)
```

```
CREATE PARTITION SCHEME p_schema
AS PARTITION pfn TO
([FG1], [FG2], [FG3], [FG4])
```

```
CREATE TABLE Employees (
EmpId int, EmpName varchar(50)
) on p_schema (EmpID);
```





# Mantenimiento de particiones



# Añadir y borrar particiones

```
ALTER PARTITION FUNCTION pfr  
{SPLIT|MERGE}  
[RANGE (clave_particionado)]
```

- Los rangos se añaden especificando una nueva clave de particionado – divide (SPLITS) la partición existente
- La operación afecta a TODAS las tablas e índices que usan la función de particionado

**ALTER TABLE miTabla**

**SWITCH PARTITION 1 TO PARTITION 2**

- Permite mover datos como operaciones de metadatos
- Permite añadir una tabla con la misma estructura a la particionada o viceversa (SWITCH IN-OUT)
- Algunas restricciones... ☹
  - Índices alineados, no índices XML, no fulltext, no FK, no recomendable con replicación, etc.

# Particionar una tabla existente

- Podemos hacerlo de dos formas:
  - Crear un índice agrupado con particiones en la tabla mediante la instrucción CREATE INDEX
    - Si ya existía, utilizar DROP EXISTING = ON
  - Con ALTER TABLE SWITCH
    - Modificar los datos de la tabla por los de una tabla con particiones que sólo tenga una partición.
    - Modificar la función de partición correspondiente para que agregue nuevas particiones

# DEMO: PARTICIONADO DE DATOS

Creación, optimización de acceso, y mantenimiento (sliding window)