



**SOLID
QUALITY
MENTORS**

Acciones y medidas en servidores OLTP que se utilizan como servidores BI

Contenido

- Introducción a la problemática
- Indexación
- Particionamiento
- Desnormalización
 - Vistas indexadas
 - Tablas resumen
 - Diseño estrella/copo de nieve
- Escalabilidad horizontal
 - Log shipping
 - Replicación
 - Database Mirroring

Introducción

- Érase una vez... una pequeña base de datos que con el tiempo se convirtió en la «BASE DE DATOS» de toda la corporación.
- En sus inicios
 - Poca carga y poco volumen de datos
 - Departamental y control sobre las operaciones realizadas
 - Infra-indexación o supra-indexación
- Con el tiempo
 - Alta carga y gran volumen de datos sin historificar
 - Aparecen consultas de naturaleza analítica e informes
 - Intradepartamental y gran caos de operaciones realizadas
 - Problemas de rendimiento
 - Diseño, indexación, fragmentación, recursos insuficientes, bloqueos, etc.

OLTP

- Operaciones orientadas a transacciones de negocio
- Ejemplos:
 - Inserción de una factura
 - Edición de los datos de contacto de un cliente
 - Cancelación de un pedido de un cliente
- Alta concurrencia de operaciones por segundo
- Aislamiento de transacciones en datos críticos
- Respuesta rápida y con baja desviación típica
 - Media < 100ms y desviación inferior a 25 ms
- Operaciones adecuadas para a un diseño OLTP normalizado e indexado.

Consultas estilo «BI»

- Orientadas a la toma de decisiones informada en el negocio
- Ejemplos:
 - Impacto de una promoción sobre el volumen de ventas
 - Evolución de las ventas de varios artículos en el tiempo
 - Margen de beneficio mensual por provincia y oficina
- Número de operaciones por segundo reducido comparado con OLTP
- En ocasiones es tolerable obtener datos aproximados o no actualizados al segundo
- Respuesta razonable y predecible
 - Media < 5s y desviación inferior a 1s
- **Operaciones costosas y «antinaturales» para una base de datos típica OLTP**

Indexación

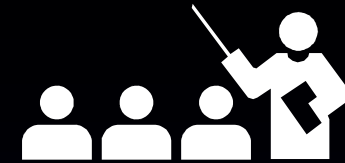
- Una aproximación sencilla para mejorar el rendimiento consiste en añadir índices útiles para las consultas estilo BI
- Un mayor número de índices implica mayor coste en inserciones, modificaciones y borrados.
- Gran cantidad de espacio «desperdiciado»
- Posibles refinamientos
 - Compresión de índices
 - Filtrado de índices
 - Creación y destrucción de índices adhoc
- Seguimos pagando un coste de CPU elevado por los cruces de tablas y los filtrados.

Demo

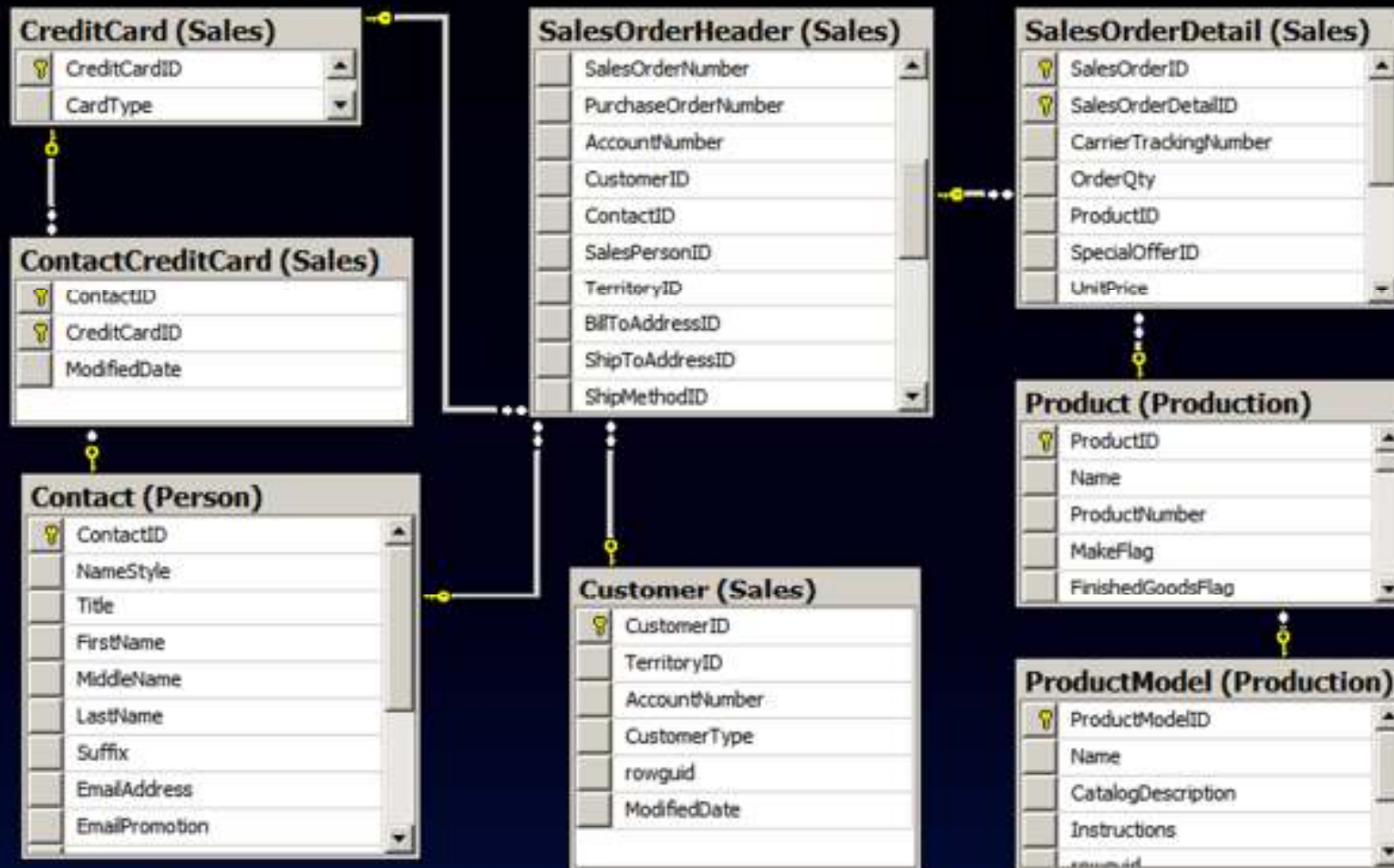


- Concurrencia de operaciones
- Carga OLTP
 - 40% consultas
 - 40% inserciones
 - 10% updates
 - 10% borrados
- Carga BI
 - 100% consultas

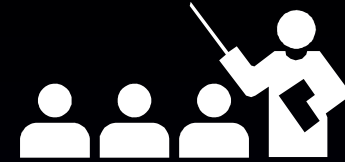
Demo



- Base de datos AdventureWorks



Demo



- OLTP
- BI
- OLTP+BI
- OLTP+BI + Índice recomendado

Resultados

Test	Media por operación	Operaciones por segundo
OLTP	37.3 ms	27.02
BI	7130 ms	0.14
BI2 (INDICE)	6130 ms	0.16

Test	Media por operación	Operaciones por segundo
OLTP (+BI)	158ms	6.33
BI2 (+OLTP)	11801 ms	0.08

- El índice recomendado por el optimizador no reduce considerablemente el cuello de botella que es la CPU

Particionamiento

- Solo se soporta nativamente particionado horizontal
 - Vertical «simulado» con índices o N tablas + vista
- Reducción de lecturas por descarte de particiones
 - Filtros por fechas
 - Filtros por país, provincia, o similar
- Mejora la granularidad del escalado a partición
- Permite almacenar distintas particiones en medios de distintas características
 - La partición OLTP → SSD, 15K RAID10
 - Las particiones históricas → 10K RAID5
- Es posible que interese particionar únicamente ciertos índices

Desnormalización

- Aumentando la redundancia de datos reducimos el número de cruces necesarios para obtener los datos.
- Los cruces de tablas requieren de más CPU cuando el volumen de datos es grande comparado con un recorrido de tabla sencillo.
 - Hash joins
 - Merge joins
- Debemos considerar desnormalizar cuando con la indexación apropiada u otras técnicas no es posible obtener un rendimiento adecuado
- Distintas técnicas de desnormalización

Vistas indexadas

- Una vista indexada es una vista sobre la que se crea un índice cluster único
- El índice cluster materializa la vista en disco por lo que la información se duplica
- No requiere modificar nuestras tablas normalizadas → schemabinding
- En versión enterprise, no necesitamos modificar las consultas
 - Sin enterprise deberemos consultar la tabla directamente
 - En ocasiones necesitaremos utilizar el hint NO EXPAND pues se considera un índice más simplemente.
- Coste de mantenimiento ante cambios alto

Vistas indexadas

- Requieren para su creación y uso de una configuración específica

Configuración	Vista indexada	OLEDB y ODBC	DB-Library
ANSI_NULLS	ON	ON	OFF
ANSI_PADDING	ON	ON	OFF
ANSI_WARNINGS	ON	ON	OFF
ARITHABORT	ON	OFF	OFF
CONCAT_NULL_YIELDS_NULL	ON	ON	OFF
NUMERIC_ROUNDABORT	OFF	OFF	OFF
QUOTED_IDENTIFIER	ON	ON	OFF

- Debemos testear si nuestras aplicaciones se pueden adaptar
- Podemos encapsular su uso dentro de un procedimiento almacenado anidado

Demo

- Utilizando una vista indexada para reducir el consumo de CPU

Resultados

Test sin vista indexada	Media por operación	Operaciones por segundo
OLTP	37.3 ms	27.02
BI	6130 ms	0.16
OLTP (+BI)	158ms	6.33
BI (+OLTP)	11801 ms	0.08

Test con vista indexada	Media por operación	Operaciones por segundo
OLTP	39.5 ms	25.31
BI	1280 ms	0.78
OLTP (+BI)	60.4 ms	16.67
BI (+OLTP)	21363 ms	0.046

Resultados

- La vista indexada sacrifica rendimiento OLTP aumentando el rendimiento en la consulta tipo BI
 - Baja mucho el consumo de CPU → concurrencia

Versión	CPU	Duración	10 consultas paralelo
Sin vista indexada	2183 ms	2434 ms	18753 ms (8.59x)
Con vista indexada	103 ms	1280 ms	4730 ms (3.69x)

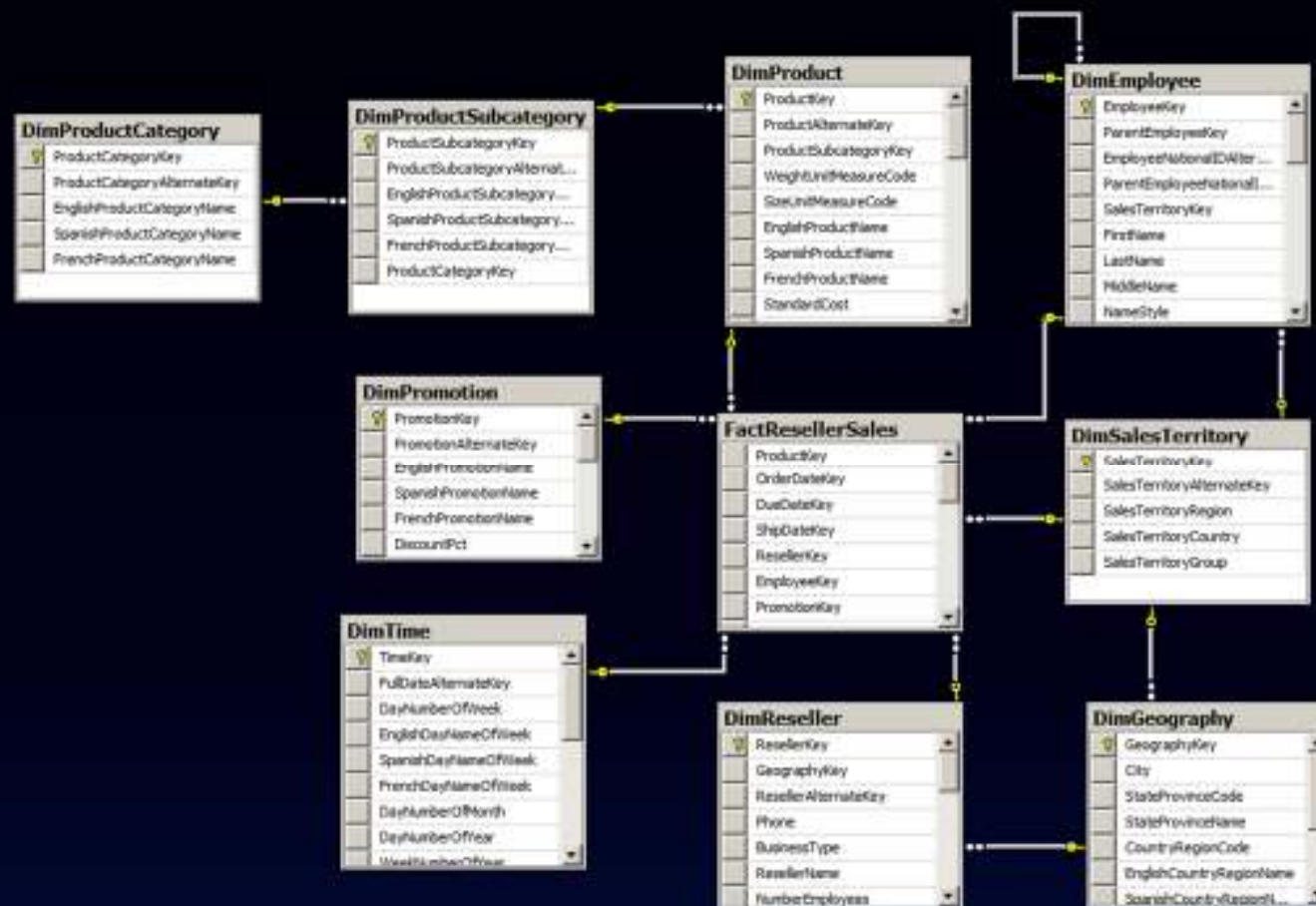
- Cuando hay concurrencia OLTP y BI
 - Se perjudica más el rendimiento de la consulta BI
 - Pueden producirse bloqueos por las actualizaciones de la vista indexada
 - Analizar cada caso y sus probabilidades

Tablas resumen

- Tablas que almacenan información calculada/acumulada para optimizar las consultas
- Normalmente se inicializan y se les da mantenimiento síncronamente
 - Triggers
 - Procedimientos almacenados
 - Replicación → Asincronismo
- Acumulación «inline»
 - Utilizar un identificador especial para insertar la fila con el valor acumulado
 - Para consultar el acumulado simplemente filtramos por dicha fila
 - Realizamos el mantenimiento de dicha fila

Diseño estrella / copo de nieve

- Reducimos la complejidad de los joins
- AdventureworksDW



Demo

- Ejemplo de tabla resumen inline

Escalabilidad horizontal

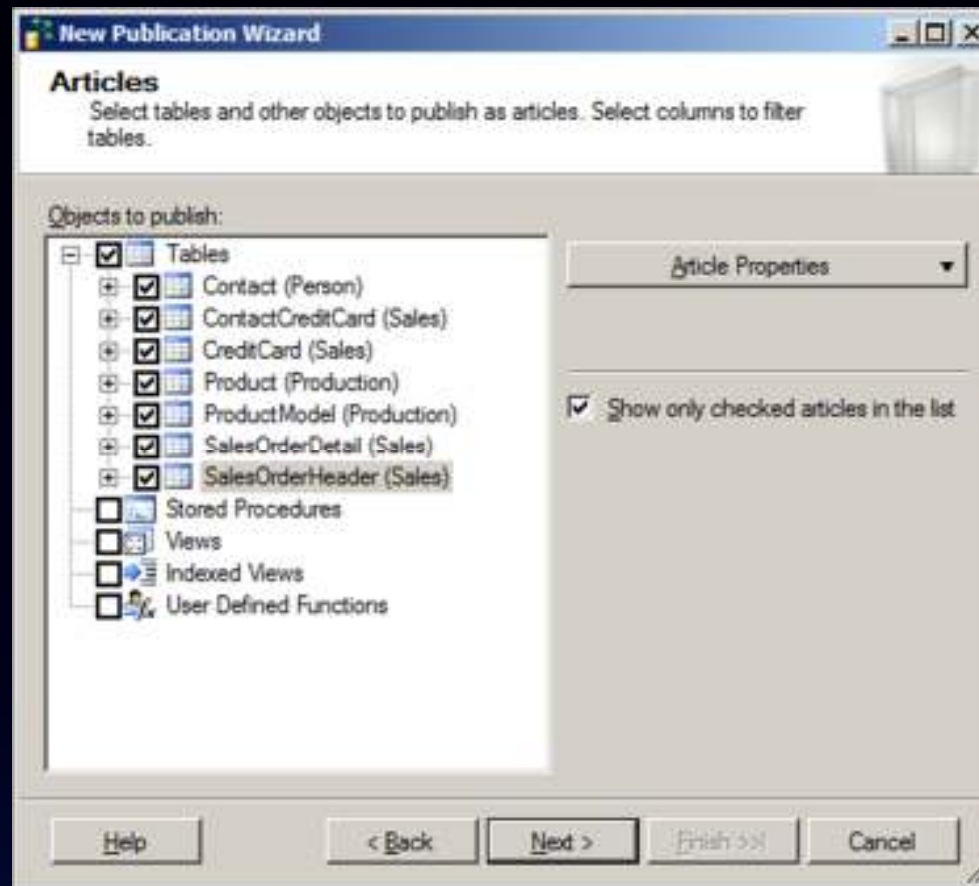
- Log Shipping
 - 1 a N
 - BD en standby
 - Echar a los usuarios cada vez que hacemos un restore
- Database Mirroring
 - 1 a 1
 - Crear snapshots periódicamente
 - La aplicación debe saber a que snapshot conectar
- Replicación
 - 1 a N
 - Base de datos subscriptora R/W → Permisos
 - Posibilidad de filtrados, transformaciones, etc.

Escalabilidad horizontal

- Si prevemos N servidores
 - Distribuir uniformemente consumo de CPU
 - Descartar mirroring
- Ojo si el mecanismo va a ser también parte de la estrategia de HA
- Si solo lo hacemos para reducir los bloqueos
 - Read committed snapshot
 - Snapshot
 - Read uncommitted
- Puede ser una alternativa interesante para los típicos clusters de dos nodos monoinstancia
 - Mejor aprovechamiento de recursos ya disponibles

Demo

- Desacoplando la vista indexada con replicación transaccional



Resultados

Test con vista indexada	Media por operación	Operaciones por segundo
OLTP	39.5 ms	25.31
BI	1280 ms	0.78
OLTP (+BI)	60.4 ms	16.67
BI (+OLTP)	21363 ms	0.046

Test con replicación + vista indexada	Media por operación	Operaciones por segundo
OLTP	39.5 ms	25.31
BI	1280 ms	0.78
OLTP (+BI)	40.6 ms	24.63
BI (+OLTP)	1345 ms	0.74

Resultados

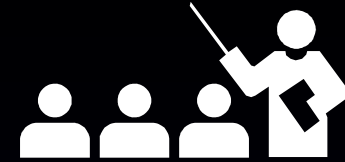
- Se reduce el impacto de mantener la vista indexada
 - Bases de datos diferentes
 - La réplica transaccional empaqueta las pequeñas operaciones OLTP en batches mayores
- Si desacoplamos en dos servidores no competiríamos por CPU, memoria, disco, etc.
- Utilizando la réplica es viable desnormalizar al vuelo en una tabla resumen, diseño en estrella, etc.

Conclusiones

- Utilizar un servidor OLTP para operaciones tipo BI puede ser un comienzo... pero no un final
- Si el volumen de consultas analíticas aumenta escalaremos pobremente
- La desnormalización nos puede ayudar pero debemos de analizar bien las alternativas y su impacto
- El uso de técnicas de desacoplamiento nos ayudarán a aliviar un servidor extenuado.

Sesiones relacionadas

- PDW → A continuación
- Diseño de DW relacionales → Miércoles 11:00
- Carga de tablas grandes → Miércoles 12:30
- Particionamiento → Miércoles 16:45



¿Preguntas?