

UNIVERSIDAD PERUANA LOS ANDES

FACULTAD DE INGENIERÍA

ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS Y
COMPUTACIÓN



SEMANA 13

ASIGNATURA: Base de Datos II

ESTUDIANTE: Espejo Quispe Luis Enrique

DOCENTE: Mg. Raúl Fernandez Bejarano

CICLO: V

SECCIÓN: A1

HYO-2025

Tema 1: Análisis de rendimiento con SQL Profiler y Extended Events

1. Enunciado del ejercicio

Configurar una herramienta para capturar consultas lentas y eventos críticos en SQL Server utilizando SQL Profiler y Extended Events.

2. Script de la solución en T-SQL

```
-- Crear sesión de Extended Events para consultas lentas
CREATE EVENT SESSION MonitoreoConsultas ON SERVER
ADD EVENT sqlserver.sql_statement_completed
(
    ACTION(sqlserver.sql_text, sqlserver.database_id)
    WHERE duration > 5000 -- consultas mayores a 5 segundos
)
ADD TARGET package0.event_file (SET filename = 'ConsultasLentas.xel')
WITH (STARTUP_STATE = ON);
```

100 %

Messages

Commands completed successfully.

Completion time: 2025-11-27T09:46:44.3113578-05:00

3. Justificación técnica

- Extended Events es más eficiente que SQL Profiler, consume menos recursos.
- Filtrar por duración evita capturar consultas triviales.
- Guardar en archivo permite análisis histórico.

4. Buenas prácticas

- **No usar Profiler en producción** por su impacto.
- **Filtrar eventos relevantes** para reducir carga.
- **Almacenar en archivo** para auditoría y análisis posterior.

Tema 2: Estadísticas e índices (creación, fragmentación, mantenimiento)

1. Enunciado del ejercicio

Crear índices en la tabla Ventas y actualizar estadísticas para optimizar consultas.

2. Script de la solución en T-SQL

```
-- Crear índice por Fecha
CREATE NONCLUSTERED INDEX idx_Ventas_Fecha ON Ventas(Fecha);

-- Ver fragmentación
SELECT * FROM sys.dm_db_index_physical_stats(DB_ID(), OBJECT_ID('Ventas'), NULL, NULL, 'DETAILED');

-- Reorganizar índice si fragmentación > 10%
ALTER INDEX idx_Ventas_Fecha ON Ventas REORGANIZE;

-- Actualizar estadísticas
UPDATE STATISTICS Ventas;
```

3. Justificación técnica

- Índices reducen tiempo de búsqueda.
- Reorganización evita degradación de rendimiento.
- Actualizar estadísticas asegura decisiones óptimas del optimizador.

4. Buenas prácticas

- Evitar exceso de índices para no afectar inserciones.
- Actualizar estadísticas periódicamente.
- Monitorear fragmentación antes de reconstruir.

Tema 3: Administración de transacciones y bloqueos

1. Enunciado del ejercicio

Detectar bloqueos y ajustar niveles de aislamiento para mejorar concurrencia.

2. Script de la solución en T-SQL

```
-- Consultar bloqueos activos
SELECT * FROM sys.dm_tran_locks;

-- Ajustar nivel de aislamiento
SET TRANSACTION ISOLATION LEVEL READ COMMITTED;
BEGIN TRANSACTION;
-- Operaciones aquí
COMMIT;
```

resource_type	resource_subtype	resource_database_id	resource_description	resource_associated_entity_id	resource_lock_partition	request_mode	request_type	request_status	request_reference_count	request_lifetime	request_session_id	request_exec_context_id	request_request_id	request_owner_name
DATABASE		4		0	17	S	LOCK	GRANT	1	0	74	0	0	SHARED_TRA

3. Justificación técnica

- `sys.dm_tran_locks` permite identificar recursos bloqueados.
- Ajustar aislamiento reduce bloqueos sin comprometer integridad.

4. Buenas prácticas

- Evitar transacciones largas.
- Usar niveles adecuados según necesidad.
- Monitorear bloqueos regularmente.

Tema 4: Análisis de planes de ejecución

1. Enunciado del ejercicio

Analizar el plan de ejecución para identificar operaciones costosas.

2. Script de la solución en T-SQL

```
-- Activar plan de ejecución en SSMS
SET SHOWPLAN_XML ON;
GO
SELECT * FROM Ventas WHERE ClienteID = 100;
```

100 %

Results Messages

Microsoft SQL Server 2005 XML Showplan	
1	<ShowPlanXML xmlns="http://schemas.microsoft.com..."

3. Justificación técnica

- El plan muestra cómo el optimizador ejecuta la consulta.
- Permite detectar Table Scan, Nested Loops costosos.

4. Buenas prácticas

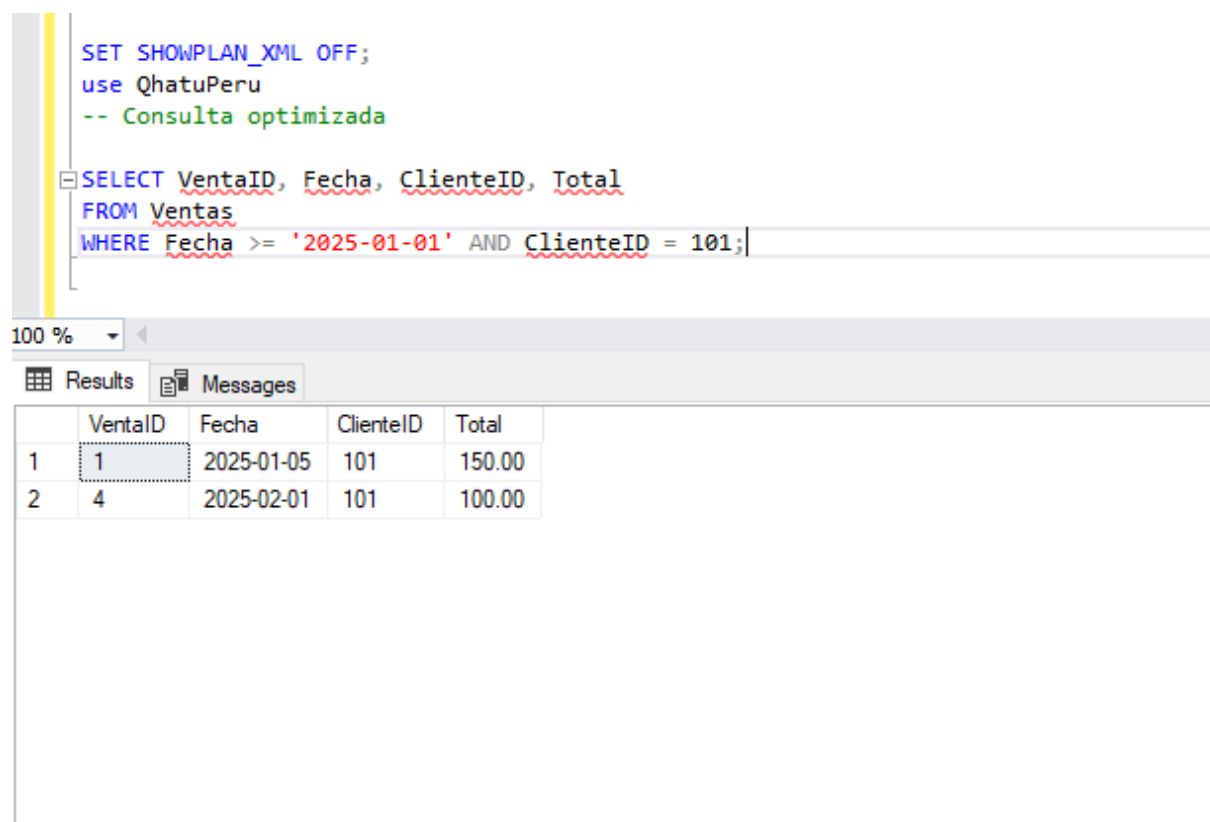
- **Analizar antes de optimizar.**
- ****Evitar SELECT ***** para reducir carga.
- **Agregar índices según plan.**

Tema 5: Optimización de consultas T-SQL

1. Enunciado del ejercicio

Optimizar una consulta que usa SELECT * y múltiples JOINS.

2. Script de la solución en T-SQL



The screenshot displays a SQL Server environment. The top pane shows a T-SQL script with the following content:

```
SET SHOWPLAN_XML OFF;  
use QhatuPeru  
-- Consulta optimizada  
  
SELECT VentaID, Fecha, ClienteID, Total  
FROM Ventas  
WHERE Fecha >= '2025-01-01' AND ClienteID = 101;
```

The bottom pane shows the results of the query in a table format. The table has four columns: VentaID, Fecha, ClienteID, and Total. There are two rows of data.

	VentaID	Fecha	ClienteID	Total
1	1	2025-01-05	101	150.00
2	4	2025-02-01	101	100.00

3. Justificación técnica

- Seleccionar columnas necesarias reduce I/O.
- JOIN optimizado con índice en ClienteID.

4. Buenas prácticas

- ****Evitar SELECT ***.**
- **Usar filtros con índices.**
- **Revisar JOINS y condiciones.**

Tema 6: Control de recursos con Resource Governor

1. Enunciado del ejercicio

Configurar Resource Governor para limitar CPU a un grupo de usuarios.

2. Script de la solución en T-SQL

```
-- Crear el Resource Pool y Workload Group
CREATE RESOURCE POOL PoolVentas WITH (MAX_CPU_PERCENT = 50);
CREATE WORKLOAD GROUP GrupoVentas USING PoolVentas;
GO

-- Crear la función de clasificación
CREATE FUNCTION fnGrupoVentas() RETURNS sysname
WITH SCHEMABINDING
AS
BEGIN
    RETURN CASE
        WHEN SUSER_NAME() = 'usuarioVentas' THEN 'GrupoVentas'
        ELSE 'default'
    END;
END;
GO

-- Asociar la función al Resource Governor
ALTER RESOURCE GOVERNOR WITH (CLASSIFIER_FUNCTION = fnGrupoVentas);
ALTER RESOURCE GOVERNOR RECONFIGURE;
GO
```

100 %

Messages

Commands completed successfully.

Completion time: 2025-11-27T10:06:01.8874822-05:00

3. Justificación técnica

- Resource Governor controla CPU y memoria por grupo.

- Clasificación por usuario asegura segmentación.

4. Buenas prácticas

- **Definir límites realistas** para evitar saturación.
- **Probar en ambiente controlado** antes de producción.
- **Monitorear impacto** tras implementación.