



**Asignatura:** Base de Datos II

**Docente:** Dr. Raúl Enrique Fernández Bejarano

**Alumno:** SORIANO TIMOTEO Joel Kevin

**Ciclo:** V

**huancayo-Perú-2025**

# Manual de Guia de Aprendizaje de la semana 14

- Mantenimiento:** No borraremos datos de tus tablas de Hechos/Dimensiones (eso es sagrado). Crearemos una tabla de **Logs de Auditoría (ETL)** para simular la limpieza.
- Reporte:** Leeremos datos reales de tu tabla `HECHOS_EVALUACIONES` para generar un reporte de "Cantidad de Evaluaciones".

Aquí tienes la solución completa y corregida.

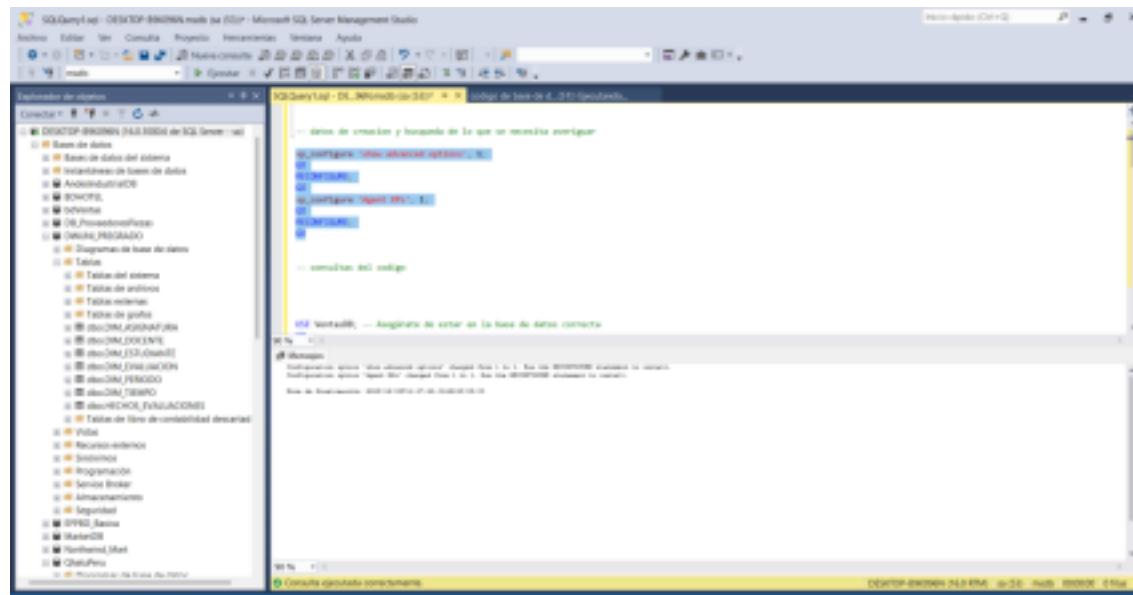
## PASO 1: Solucionar el error del Agente (Crucial)

Antes de correr el código, fíjate en tu primera imagen. Donde dice **Agente SQL Server (Agent XPs deshabilitado)**, tienes un ícono rojo.

- Dale **Clic Derecho** a "Agente SQL Server".
- Selecciona **Iniciar (Start)**.
- Si no te deja iniciar o la opción está gris, **ejecuta primero este bloque de código** para desbloquear la configuración:

SQL

```
sp_configure 'show advanced options', 1;
GO
RECONFIGURE;
GO
sp_configure 'Agent XPs', 1;
GO
RECONFIGURE;
GO
```



Una vez ejecutado esto, vuelve a darle Clic Derecho -> Iniciar al Agente hasta que el icono se ponga verde.

## PASO 2: Preparar tu base de datos DWUNI\_PREGRADO

Vamos a crear dos tablas auxiliares dentro de tu base de datos para que el Job tenga dónde guardar los reportes y qué limpiar, sin tocar tus tablas importantes de alumnos o docentes.

Copia y ejecuta esto:

SQL

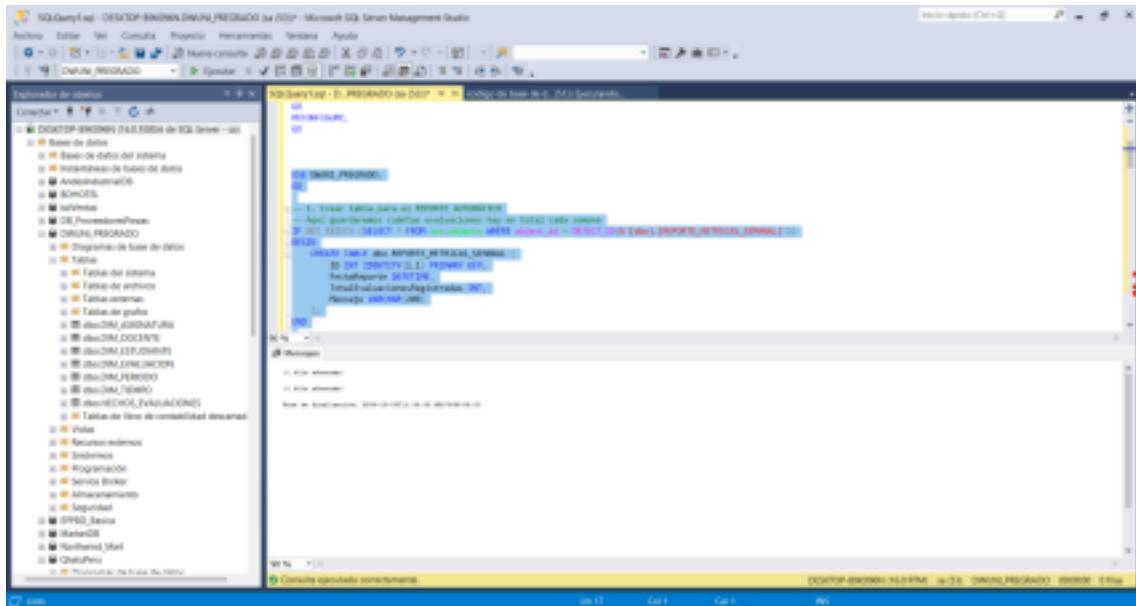
```
USE DWUNI_PREGRADO;
GO

-- 1. Crear tabla para el REPORTE AUTOMÁTICO
-- Aquí guardaremos cuántas evaluaciones hay en total cada semana
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[REPORTE_METRICAS_SEMANAL]'))
BEGIN
    CREATE TABLE dbo.REPORTE_METRICAS_SEMANAL (
        ID INT IDENTITY(1,1) PRIMARY KEY,
        FechaReporte DATETIME,
        TotalEvaluacionesRegistradas INT,
        Mensaje VARCHAR(200)
    );
END

-- 2. Crear tabla de LOGS DE SISTEMA (Para simular la limpieza)
-- Simularemos que aquí se guardan errores del sistema
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[LOG_SISTEMA_DW]'))
BEGIN
    CREATE TABLE dbo.LOG_SISTEMA_DW (
        ID INT IDENTITY(1,1) PRIMARY KEY,
        Evento VARCHAR(100),
        FechaEvento DATETIME
    );

    -- Insertamos DATOS DE PRUEBA para que la limpieza tenga algo
    -- que borrar
    -- Dato "Viejo" (Hace 200 días) -> SE DEBE BORRAR
    INSERT INTO dbo.LOG_SISTEMA_DW VALUES ('Error de conexión
    antiguo', DATEADD(DAY, -200, GETDATE()));
    -- Dato "Nuevo" (Hoy) -> SE DEBE QUEDAR
    INSERT INTO dbo.LOG_SISTEMA_DW VALUES ('Carga de datos
    exitosa', GETDATE());
END

GO
```



## PASO 3: El Script del Job (Automatización)

Este script crea el trabajo automático utilizando tu base de datos DWUNI\_PREGRADO.

Copia y ejecuta esto en una nueva ventana:

SQL

```
USE msdb;
GO

-- A. Limpieza previa: Si el Job ya existe, lo borramos para
-- crearlo limpio
DECLARE @jobId BINARY(16);
SELECT @jobId = job_id FROM msdb.dbo.sysjobs WHERE (name =
N'Mantenimiento_DW_Universitario')
IF (@jobId IS NOT NULL)
BEGIN
    EXEC sp_delete_job @job_id = @jobId, @delete_unused_schedule=1;
END

-- B. Crear el Job (El Contenedor)
EXEC sp_add_job
@job_name = N'Mantenimiento_DW_Universitario',
@enabled = 1,
@description = N'Job para limpiar logs del DW y reportar
volumen de evaluaciones.',
@owner_login_name = N'sa',
@job_id = @jobId OUTPUT;

-- C. PASO 1: Limpieza (Mantenimiento)
-- Borra logs de la tabla LOG_SISTEMA_DW que tengan más de 6 meses
EXEC sp_add_jobstep
@job_id = @jobId,
@step_name = N'Limpiar Logs Antiguos DW',
@step_id = 1,

@on_success_action = 3, -- Si sale bien, ve al paso 2
@on_fail_action = 2, -- Si falla, salir reportando error
@subsystem = N'TSQL',
@database_name = N'DWUNI_PREGRADO', -- TU BASE DE DATOS REAL
```

```

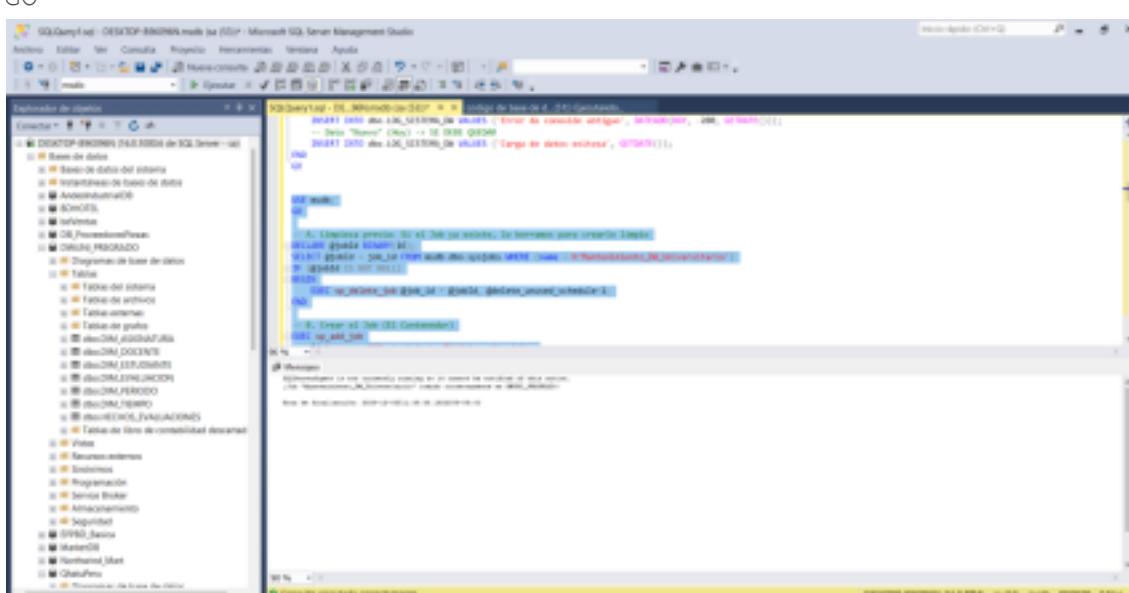
@command = N'
DELETE FROM dbo.LOG_SISTEMA_DW
WHERE FechaEvento < DATEADD(day, -180, GETDATE());
';

-- D. PASO 2: Reporte (Inteligencia de Negocios)
-- Cuenta cuántas filas hay en HECHOS_EVALUACIONES y lo guarda en el reporte
EXEC sp_add_jobstep
@job_id = @jobId,
@step_name = N'Generar Metricas Evaluaciones',
@step_id = 2,
@on_success_action = 1, -- Si sale bien, terminar con éxito
@on_fail_action = 2, -- Si falla, salir reportando error
@subsystem = N'TSQL',
@database_name = N'DWUNI_PREGRADO', -- TU BASE DE DATOS REAL
@command = N'
INSERT INTO dbo.REPORTE_METRICAS_SEMANAL (FechaReporte,
TotalEvaluacionesRegistradas, Mensaje)
SELECT
GETDATE(),
COUNT(*), -- Contamos tus evaluaciones reales
''Reporte generado automáticamente por SQL Agent''
FROM dbo.HECHOS_EVALUACIONES;
';

-- E. Asignar al servidor local
EXEC sp_add_jobserver @job_id = @jobId, @server_name = N'(local)';

PRINT ';'Job "Mantenimiento_DW_Universitario" creado correctamente
en DWUNI_PREGRADO!';
GO

```



## ¿Cómo verificar que funcionó?

Una vez ejecutado el script, haz lo siguiente para probarlo manualmente (ya que no queremos esperar al domingo):

1. En el Explorador de Objetos, ve a **Agente SQL Server > Trabajos (Jobs)**.
2. Busca **Mantenimiento\_DW\_Universitario**.
3. **Clic derecho > Iniciar trabajo en el paso...**

4. Espera a que salga el check verde de "Éxito".
5. Ejecuta esta consulta para ver el resultado final:

SQL

```
USE DWUNI_PREGRADO;
SELECT * FROM REPORTE_METRICAS_SEMANAL;
-- Deberías ver una nueva fila con la fecha de hoy y el conteo de
tus evaluaciones.
```

Clasificación de datos completos de todo lo que se está especificando

## 1. Enunciado del Caso Práctico

Contexto:

La universidad gestiona un Data Warehouse (DWUNI\_PREGRADO) para analizar el rendimiento académico. El equipo de TI ha detectado dos necesidades operativas que requieren automatización:

- 1. Mantenimiento de Sistema:** La tabla de registros de auditoría del sistema (`LOG_SISTEMA_DW`) acumula datos técnicos que pierden relevancia después de 6 meses, ocupando espacio innecesario.
- 2. Monitoreo de Negocio:** La Vicerrectoría Académica necesita conocer semanalmente el volumen total de evaluaciones procesadas en la tabla de hechos

(`HECHOS_EVALUACIONES`) para monitorear la carga de trabajo, sin tener que solicitar consultas manuales al DBA.

Requerimiento:

Desarrollar y programar un Job en SQL Server Agent llamado `Mantenimiento_DW_Universitario` que ejecute automáticamente las siguientes tareas:

- **Paso 1:** Eliminar físicamente los logs del sistema con una antigüedad mayor a **180 días**.
- **Paso 2:** Calcular el total de evaluaciones registradas en la tabla de hechos y almacenar este métrica histórica en una tabla de reportes (**REPORTE\_METRICAS\_SEMANAL**).

## 2. Script de Solución (Código Completo)

Copia y pega este bloque completo. Incluye la activación del Agente, la creación de tablas necesarias y la configuración del Job.

SQL

```

USE master;
GO

-----
-- PARTE A: CONFIGURACIÓN INICIAL (Solución de errores previos)
-----

-- 1. Habilitar 'Agent XPs' para que el Agente SQL pueda iniciar
sp_configure 'show advanced options', 1;
RECONFIGURE;
GO
sp_configure 'Agent XPs', 1;
RECONFIGURE;
GO

-----
-- PARTE B: PREPARACIÓN DE LA BASE DE DATOS (DWUNI_PREGRADO)
-----

-- USE DWUNI_PREGRADO;
GO

-- 2. Crear tabla para almacenar el Reporte Automático
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[REPORTE_METRICAS_SEMANAL]'))
BEGIN
    CREATE TABLE dbo.REPORTE_METRICAS_SEMANAL (
        ID INT IDENTITY(1,1) PRIMARY KEY,
        FechaReporte DATETIME,
        TotalEvaluaciones INT,
        Mensaje VARCHAR(200)
    );
END

-- 3. Crear tabla de Logs (Simulación para Mantenimiento)
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[LOG_SISTEMA_DW]'))
BEGIN
    CREATE TABLE dbo.LOG_SISTEMA_DW (
        ID INT IDENTITY(1,1) PRIMARY KEY,
        Evento VARCHAR(100),
        FechaEvento DATETIME
    );

```

```

-- Insertamos datos de prueba: Uno viejo (a borrar) y uno nuevo
-- (a conservar)
INSERT INTO dbo.LOG_SISTEMA_DW VALUES ('Error Antiguo',
DATEADD(DAY, -200, GETDATE()));
INSERT INTO dbo.LOG_SISTEMA_DW VALUES ('Log Reciente',
GETDATE());
END
GO

-----
-- 
-- PARTE C: CREACIÓN DEL JOB DE AUTOMATIZACIÓN
-- 

-- 
USE msdb;
GO

-- 4. Limpieza: Si el Job ya existe, borrarlo para recrearlo limpio
DECLARE @jobId BINARY(16);
SELECT @jobId = job_id FROM msdb.dbo.sysjobs WHERE (name =
N'Mantenimiento_DW_Universitario')
IF (@jobId IS NOT NULL)
BEGIN
    EXEC sp_delete_job @job_id = @jobId, @delete_unused_schedule=1;
END

-- 5. Crear el Job (El Contenedor Principal)
EXEC sp_add_job
@job_name = N'Mantenimiento_DW_Universitario',
@enabled = 1,
@description = N'Mantenimiento de logs y reporte de hechos de
evaluaciones.',
@owner_login_name = N'sa',
@job_id = @jobId OUTPUT;

-- 6. Paso 1: Mantenimiento (Limpieza de Logs)
EXEC sp_add_jobstep
@job_id = @jobId,
@step_name = N'Limpiar Logs Antiguos',
@step_id = 1,
@on_success_action = 3, -- Éxito: Ir al siguiente paso
@on_fail_action = 2, -- Fallo: Salir reportando error
@subsystem = N'TSQL',
@database_name = N'DWUNI_PREGRADO',
@command = N'DELETE FROM dbo.LOG_SISTEMA_DW WHERE FechaEvento <
DATEADD(day, -180, GETDATE());';

-- 7. Paso 2: Reporte (Conteo de Evaluaciones)
EXEC sp_add_jobstep

@job_id = @jobId,
@step_name = N'Generar Metricas Semanales',
@step_id = 2,
@on_success_action = 1, -- Éxito: Terminar Job con éxito
@on_fail_action = 2, -- Fallo: Salir reportando error
@subsystem = N'TSQL',
@database_name = N'DWUNI_PREGRADO',
@command = N'
INSERT INTO dbo.REPORTE_METRICAS_SEMANAL (FechaReporte,

```

```

TotalEvaluaciones, Mensaje)
SELECT GETDATE(), COUNT(*), '''Ejecución Automática'''
FROM dbo.HECHOS_EVALUACIONES;'

-- 8. Asignar el Job al Servidor Local
EXEC sp_add_jobserver @job_id = @jobId, @server_name = N'(local)';

PRINT '¡Proceso completado! El Job ha sido creado en el servidor.';
GO

```

## 3. Explicación Técnica

El script se divide en tres bloques lógicos:

### 1. Configuración del Entorno (Parte A):

- Se ejecuta `sp_configure 'Agent XPs', 1`. Esto es fundamental porque tu servidor tenía el Agente deshabilitado (el ícono rojo). Este comando desbloquea las funcionalidades extendidas para que el servicio de automatización pueda arrancar.

### 2. Preparación de Datos (Parte B):

- Trabajamos sobre la base de datos `DWUNI_PREGRADO`.
- Se crea `LOG_SISTEMA_DW` para simular una tabla que crece indefinidamente y necesita limpieza.
- Se crea `REPORTE_METRICAS_SEMANAL` para almacenar los resultados del análisis, evitando consultar la tabla pesada de hechos constantemente.

### 3. Definición del Job (Parte C):

- Utilizamos procedimientos almacenados del sistema (`msdb.dbo.sp_...`).
- **Job Container:** Crea el trabajo vacío.
- **Step 1 (Limpieza):** Ejecuta un `DELETE` con filtro de fecha (`DATEADD -180`). Si funciona, pasa al paso 2 (`@on_success_action = 3`).
- **Step 2 (Reporte):** Ejecuta un `INSERT` basado en un `SELECT COUNT(*)` de tu tabla real `HECHOS_EVALUACIONES`. Esto toma una "foto" del estado actual de las evaluaciones.

## 4. Justificación de la Solución

Esta implementación responde directamente a los objetivos de la **Semana 14: Automatización y Mantenimiento** por las siguientes razones:

### 1. Proactividad sobre Reactividad:

- En lugar de esperar a que el disco duro se llene por logs viejos, el sistema se "autocura" eliminando registros de más de 6 meses. Esto garantiza la **integridad y disponibilidad** del servidor a largo plazo.

### 2. Optimización de Recursos (Data Warehouse):

- La tabla `HECHOS_EVALUACIONES` suele ser muy grande (millones de registros). Si el Vicerrector ejecuta `COUNT(*)` en horario laboral, podría ralentizar el sistema.
- Al automatizar esto con un Job (que podría correrse de madrugada), guardamos el resultado simple (un número entero) en una tabla ligera. Las consultas posteriores son instantáneas.

### 3. Seguridad y Control:

- Usar SQL Server Agent elimina el error humano. No dependemos de que el

administrador recuerde ejecutar los scripts. Además, el uso de pasos secuenciales (`step_id`) asegura que el reporte solo se genere si el sistema está saludable (o podríamos configurar alertas si el paso 1 falla).

Aquí tienes los códigos de las consultas T-SQL "puras" extraídas del Job. Puedes copiarlas y ejecutarlas directamente en una ventana de **New Query** para probar la lógica manualmente.

## 1. Consulta de Mantenimiento (Limpieza de Logs)

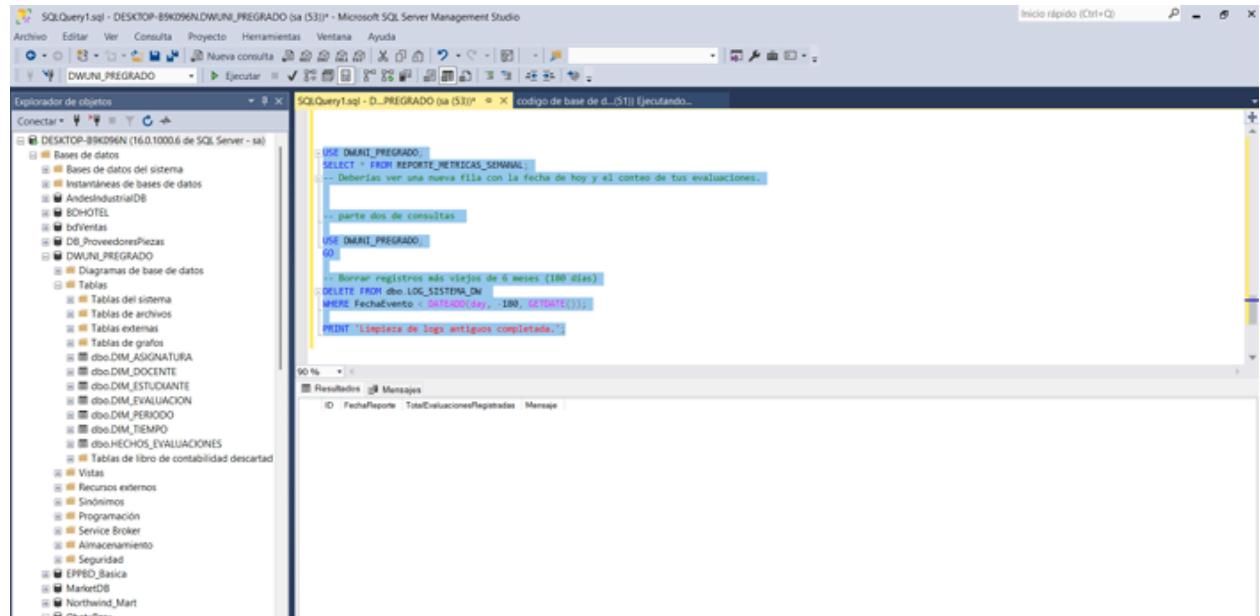
Esta consulta elimina físicamente los registros de auditoría que tienen más de 180 días de antigüedad.

SQL

```
USE DWUNI_PREGRADO;
GO

-- Borrar registros más viejos de 6 meses (180 días)
DELETE FROM dbo.LOG_SISTEMA_DW
WHERE FechaEvento < DATEADD(day, -180, GETDATE());

PRINT 'Limpieza de logs antiguos completada.';
```



## 2. Consulta de Reporte (Métricas Semanales)

Esta consulta cuenta las filas en tu tabla de hechos (`HECHOS_EVALUACIONES`) y guarda el resultado en la tabla de reporte.

SQL

```
USE DWUNI_PREGRADO;
GO

-- Insertar el conteo actual de evaluaciones en la tabla de reporte
INSERT INTO dbo.REPORTE_METRICAS_SEMANAL (FechaReporte,
TotalEvaluaciones, Mensaje)
SELECT
GETDATE(), -- Fecha y hora actual
COUNT(*), -- Cuenta todas las evaluaciones reales
'Ejecución Manual de Prueba'
```

```
FROM dbo.HECHOS_EVALUACIONES;
```

```
PRINT 'Reporte de métricas generado exitosamente.';
```

cambiaron después de correr las anteriores.

SQL

```
USE DWUNI_PREGRADO;
GO
```

```
-- Ver qué quedó en los logs (los viejos deberían haber desaparecido)
```

```
SELECT * FROM dbo.LOG_SISTEMA_DW;
```

```
-- Ver el nuevo reporte generado
```

```
SELECT * FROM dbo.REPORTE_METRI
```

The screenshot shows the Microsoft SQL Server Management Studio interface. On the left is the Object Explorer pane, which lists several databases including 'DWUNI\_PREGRADO'. The main window contains a query editor with the following script:

```
USE DWUNI_PREGRADO;
GO

-- Insertar el conteo actual de evaluaciones en la tabla de reporte
INSERT INTO dbo.REPORTE_METRICAS_SEMANAL (FechaReporte, TotalEvaluaciones, Mensaje)
SELECT GETDATE(), -- Fecha y hora actual
       COUNT(*) -- Cuenta todas las evaluaciones reales
      'Ejecución Manual de Prueba'
     FROM dbo.HECHOS_EVALUACIONES;
PRINT 'Reporte de métricas generado exitosamente.';
```

The status bar at the bottom indicates the execution completed successfully with 0 errors.

### 3. Consultas de Verificación (Para ver los resultados)

Ejecuta estas líneas para confirmar que las tablas CAS\_SEMANAL ORDER BY FechaReporte DESC;

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer pane is visible on the left. The main window contains a query editor with the following script:

```
PRINT 'Reporte de métricas generado exitosamente.';

USE DWUNI_PREGRADO;
GO

-- Ver qué quedó en los logs (los viejos deberían haber desaparecido)
SELECT * FROM dbo.LOG_SISTEMA_DW;
GO

-- Ver el nuevo reporte generado
SELECT * FROM dbo.REPORTE_METRICAS_SEMANAL ORDER BY FechaReporte DESC;
```

The results pane shows one row of data from the log table:

ID_Evento	FechaEvento	Detalle
1	2025-12-03 11:38:08.857	Carga de datos exitosa

The status bar at the bottom indicates the execution completed correctly with 0 errors.