



**Instituto Politécnico Nacional
Unidad Profesional Interdisciplinaria en
Ingeniería y Tecnologías Avanzadas
(UPIITA).**



MATERIA: Bases de Datos Distribuidas

Telemática

**Consultas de la sesión del 100226 y
120226**

Nombre:

Nieves Hernández Efraín

1. Reescritura y Optimización del Código

El objetivo es que la consulta sea más legible y eficiente. Una forma moderna de optimizar esto en SQL Server es usando un CTE (Common Table Expression) para separar la lógica del conteo de materias impartidas.

/* Consulta 2 Optimizada

Listar alumnos que aprobaron TODAS las materias del profesor P0000001

*/

WITH MateriasDelProfesor AS (

-- Calculamos una sola vez cuántas materias distintas imparte el profesor

SELECT COUNT(DISTINCT clave) AS total_requerido

FROM Escuela.Imparte

WHERE numEmpleado = 'P0000001'

)

SELECT

c.boleta,

COUNT(DISTINCT c.clave) AS materias_aprobadas

FROM Escuela.Cursa c

JOIN Escuela.Imparte i

ON c.clave = i.clave

AND c.Semestre = i.semestre

AND c.idGrupo = i.idGrupo

WHERE

c.calif >= 6

AND i.numEmpleado = 'P0000001'

GROUP BY c.boleta

HAVING COUNT(DISTINCT c.clave) = (SELECT total_requerido FROM MateriasDelProfesor);

2. Análisis

El enfoque fue pasar de una subconsulta "anidada" que se ejecuta muchas veces a una estructura más limpia.

La solución original ejecutaba la subconsulta (SELECT COUNT(DISTINCT clave)...) dentro del HAVING por cada grupo de boletas. En tablas con miles de registros, esto puede ser lento. La versión optimizada utiliza un CTE (WITH) que separa el cálculo del "objetivo" (cuántas materias son en total) del proceso de filtrado de alumnos.

¿Cómo lo arregla/mejora?

1. **Legibilidad:** Al usar el CTE MateriasDelProfesor, el código se lee como una receta: "primero dime cuántas materias son, y luego compáralas".
2. **Rendimiento:** SQL Server puede optimizar mejor el plan de ejecución cuando el valor de comparación es constante o está pre-calculado en un CTE.

Lo que entendí del ejercicio:

"Para este ejercicio, se analizó la Solución 2 de la Consulta 2, la cual busca identificar a los alumnos que han aprobado la totalidad de las asignaturas impartidas por un docente específico. La versión original empleaba una subconsulta directa en la cláusula HAVING. A través de la interacción con la IA, se propuso una reestructuración utilizando una Expresión de Tabla Común (CTE).

La mejora principal radica en la separación de responsabilidades: se calcula de forma aislada el universo de materias del profesor para luego realizar una comparación directa contra el conteo de materias aprobadas por el alumno. Esto no solo mejora el tiempo de respuesta del motor de base de datos al evitar cálculos redundantes, sino que también facilita la depuración del código en entornos de producción académica y profesional."