



Unidad Profesional  
Interdisciplinaria en Ingeniería y  
Tecnologías Avanzadas



# Base de datos distribuidas

Alumna: Muñoz Reyes Citlali

Profesor: De la Cruz Sosa  
Carlos

Tarea 1 - entrega 13/02/2026  
Consulta de IA del código de la consulta

# Consulta de IA del código de la consulta

## • Código de consulta

```
✓ select c.boleta, count(distinct c.clave) cant_mat
  from Escuela.Cursa c
  join Escuela.Imparte i
    on c.clave = i.clave
   and c.Semestre = i.semestre
   and c.idGrupo = i.idGrupo
   and c.calif >= 6
   and i.numEmpleado = 'P0000001'
 group by c.boleta
 having count(distinct c.clave) =
  (select count(distinct clave) as cant_mat
   from Escuela.Imparte
   where numEmpleado = 'P0000001')
```

Para que la IA me diera la respuesta tuve que darle el contexto de toda la información que estaba manejando para que fuera más certera y mejor su respuesta ante mi duda y mejora del código

ChatGPT ▾

Obtener Plus ×

aquí está la consulta: Consulta 1  
Listar alumnos que hayan cursado y aprobado todas las materias que imparte el profesor P0000001 y su solución fue:

```
select c.boleta, count(distinct c.clave) cant_mat
  from Escuela.Cursa c
  join Escuela.Imparte i
    on c.clave = i.clave
   and c.Semestre = i.semestre
   and c.idGrupo = i.idGrupo
   and c.calif >= 6
   and i.numEmpleado = 'P0000001'
 group by c.boleta
 having count(distinct c.clave) =
  (select count(distinct clave) as cant_mat
   from Escuela.Imparte
   where numEmpleado = 'P0000001')
```

Como era previsto la parte de la escalabilidad presentaba un problema a futuro... es una solución pero siempre y cuando los datos no sea demasiados

### Respuesta que recibí de la IA:

Excelente, ahora sí ya se puede optimizar bien 🌟

Tu consulta es correcta lógicamente, pero se puede hacer más eficiente y más clara.

#### 🔍 Qué hace tu consulta actual (rápido)

- Cruza `Cursa` con `Imparte`
- Filtra materias del profesor `P0000001`
- Solo considera materias **aprobadas** (`calif ≥ 6`)
- Cuenta cuántas materias aprobó cada alumno
- Compara contra el total de materias que imparte el profesor

Funciona 🌟, pero:

- Usa **subconsulta correlacionada innecesaria**
- Hace conteos repetidos
- Puede escalar mal con muchos datos

ChatGPT

Esa es una solución clásica y efectiva utilizando **agregación y comparación de conteos**.

Básicamente, estás diciendo: "Dame a los alumnos que aprobaron un número de materias de este profesor igual al total de materias que ese profesor imparte".

Aquí te doy un breve análisis de por qué tu lógica es correcta y una alternativa común por si necesitas otro enfoque.

#### Análisis de tu solución

- **El JOIN:** Es fundamental que unas por `clave`, `Semestre` e `idGrupo`. Esto asegura que el alumno realmente tomó la clase específicamente con ese profesor y no con otro que diera la misma materia.
- **Filtro de aprobación ( `c.calif >= 6` ):** Correcto, esto descarta a quienes cursaron pero reprobaron.
- **La cláusula HAVING :** Es el corazón de la consulta. Compara el "conteo real" del alumno contra el "conteo objetivo" de la subconsulta.

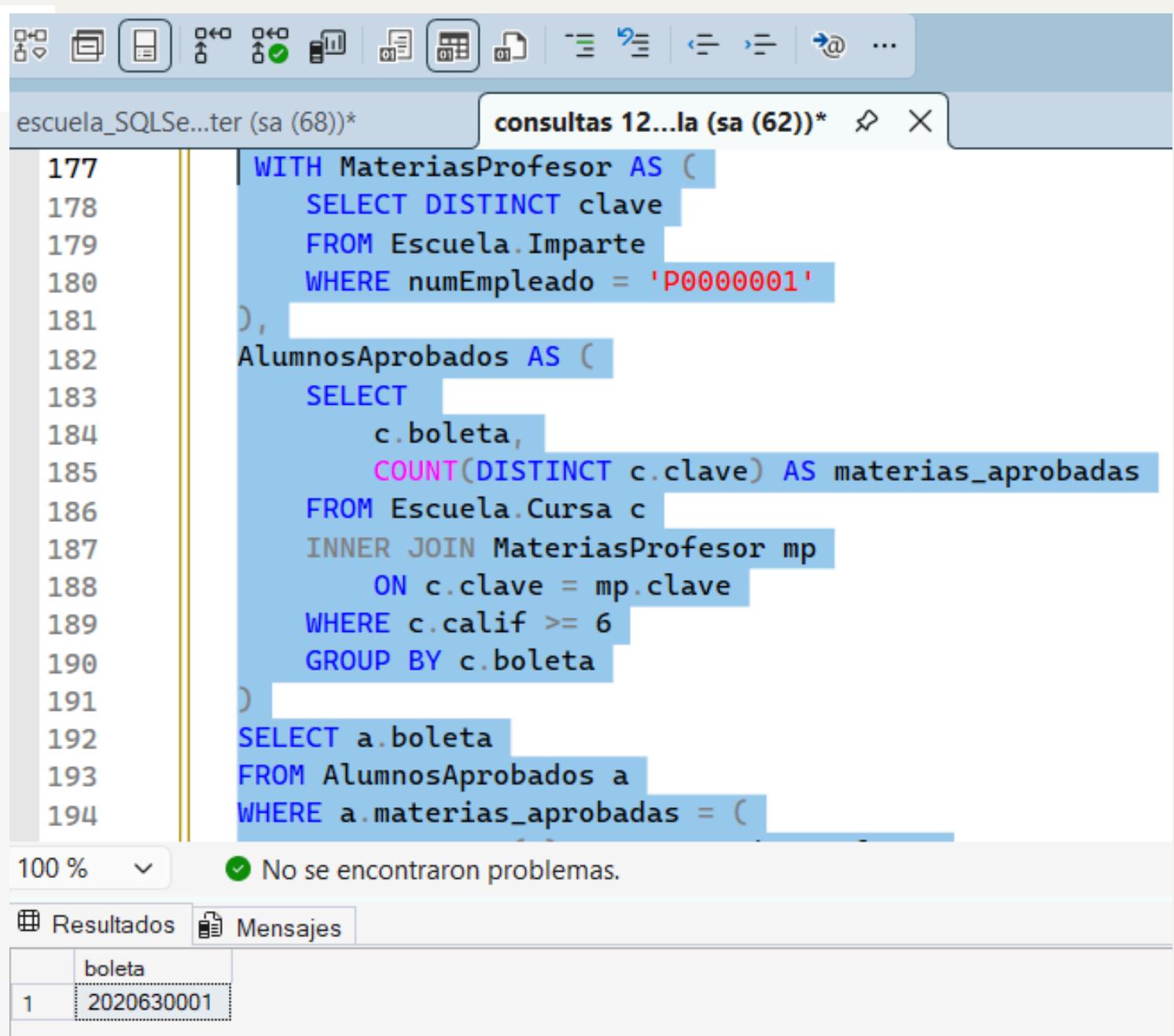
Gemini

## • Código respuesta

 Opción 1: usando CTE (más eficiente y legible)

```
sql

WITH MateriasProfesor AS (
    SELECT DISTINCT clave
    FROM Escuela.Imparte
    WHERE numEmpleado = 'P0000001'
),
AlumnosAprobados AS (
    SELECT
        c.boleta,
        COUNT(DISTINCT c.clave) AS materias_aprobadas
    FROM Escuela.Cursa c
    INNER JOIN MateriasProfesor mp
        ON c.clave = mp.clave
    WHERE c.calif >= 6
    GROUP BY c.boleta
)
SELECT a.boleta
FROM AlumnosAprobados a
WHERE a.materias_aprobadas = (
    SELECT COUNT(*) FROM MateriasProfesor
);
```



**Los beneficios que me dio fueron:**

- El total de materias del profesor se calcula una sola vez
  - Menos joins pesados
  - Mejor para planes de ejecución
  - Más fácil de mantener

A mi consideración creo que el tamaño del código generado es importante, pero si nos damos cuenta nos evita algunos problemas en cuanto a que nuestra base tome un aspecto mas escalable.

 Opción 2: sin subconsulta ( NOT EXISTS )

Esta es la que más le gusta al optimizador de SQL Server

```
sql

SELECT DISTINCT c.boleta
FROM Escuela.Cursa c
WHERE c.calif >= 6
AND NOT EXISTS (
    SELECT 1
    FROM Escuela.Imparte i
    WHERE i.numEmpleado = 'P00000001'
    AND NOT EXISTS (
        SELECT 1
        FROM Escuela.Cursa c2
        WHERE c2.boleta = c.boleta
        AND c2.clave = i.clave
        AND c2.idGrupo = i.idGrupo
        AND c2.Semestre = i.semestre
        AND c2.calif >= 6
    )
);
```

Me mostro esta solución como la mas eficiente, y es donde notamos la sentencia de Not exists, por ventaja tenemos que se puede ocupar para la parte de cualquier materia y evita el uso de algunas sentencias

SQL

```
SELECT DISTINCT c1.boleta
FROM Escuela.Cursa c1
WHERE NOT EXISTS (
    -- Materias que imparte el profesor
    SELECT i.clave
    FROM Escuela.Imparte i
    WHERE i.numEmpleado = 'P0000001'
    AND NOT EXISTS (
        -- Materias que el alumno aprobó con ese profesor
        SELECT c2.clave
        FROM Escuela.Cursa c2
        WHERE c2.boleta = c1.boleta
        AND c2.clave = i.clave
        AND c2.idGrupo = i.idGrupo
        AND c2.Semestre = i.semestre
        AND c2.calif >= 6
    )
);
```

Al igual la otra IA prefiere evitar las otras sentencias y recurrir al uso de NOT EXISTS debido a que suele ser más eficiente en bases de datos con volúmenes de datos masivos, ya que los motores de búsqueda pueden optimizar las búsquedas existenciales más rápido que los conteos globales

## • Respuesta

Aqui podemos ver que solo se mostraran datos siempre y cuando se cumplan las condiciones, es decir que los alumnos pasaran todas las materias que el profesor imparte



```
escuela_SQLServer (sa (68))* consultas 12...la (sa (62))*
159  SELECT DISTINCT c1.boleta
160  FROM Escuela.Cursa c1
161  WHERE NOT EXISTS (
162      -- Materias que imparte el profesor
163      SELECT i.clave
164      FROM Escuela.Imparte i
165      WHERE i.numEmpleado = 'P0000001'
166      AND NOT EXISTS (
167          -- Materias que el alumno aprobó con ese profesor
168          SELECT c2.clave
169          FROM Escuela.Cursa c2
170          WHERE c2.boleta = c1.boleta
171          AND c2.clave = i.clave
172          AND c2.idGrupo = i.idGrupo
173          AND c2.Semestre = i.semestre
174          AND c2.calif >= 6
175      )
176  );
100 % ✓ No se encontraron problemas.
```

Resultados Mensajes

	boleta
1	2020630001