

# INSTITUTO POLITÉCNICO NACIONAL

UNIDAD PROFESIONAL INTERDISCIPLINARIA EN  
INGENIERÍA Y TECNOLOGÍAS AVANZADAS

## BASE DE DATOS DISTRIBUIDAS - 3TM2

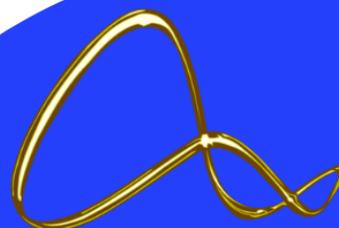
### *Tarea 1. Consulta de optimización con IA*

Alumno:

Gonzalez Luna Marco - **2024640044**

Profesor:

De la Cruz Sosa Carlos



**upiita-ipn**

Consulta a realizar:

```
/*Listar alumnos que hayan cursado y aprobado todas las materias  
que imparte el profesor P0000001*/
```

Solución propuesta en clase:

The screenshot shows a SQL query editor interface. The query itself is as follows:

```
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 calif>=6
and numEmpleado = 'P0000001'

group by c.boleta
having count(distinct c.clave) =
(select count(distinct clave) as cant_mat
from Escuela.Imparte
where numEmpleado = 'P0000001') ;
```

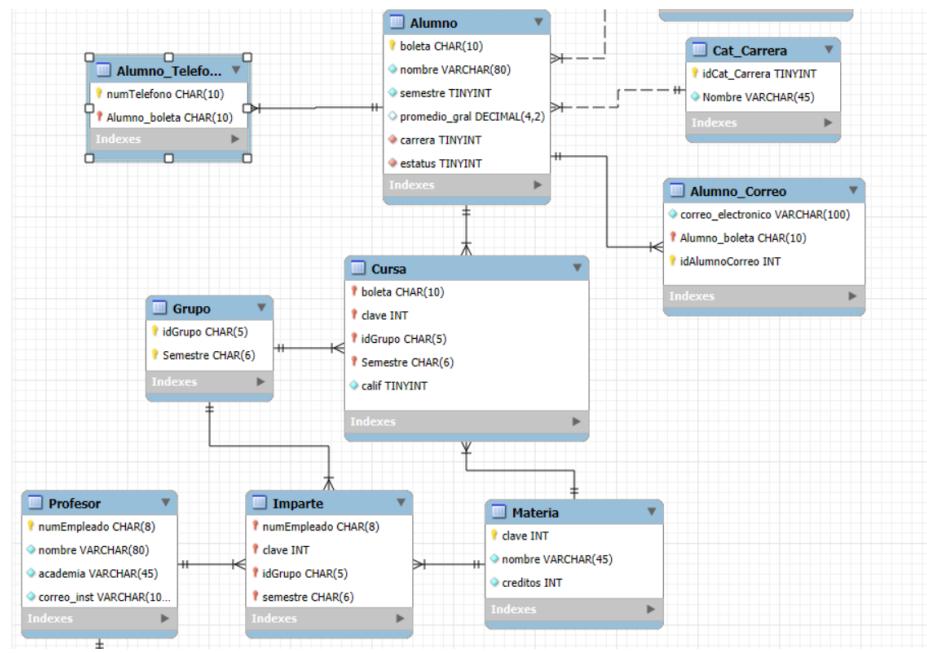
The results pane shows a single row of data:

	boleta	cant_mat
1	2020630002	2

Promt:

<https://gemini.google.com/share/186cc05a6f0b>

Marco González Luna  
 Base de Datos Distribuidas Tarea 1



Responde como un experto en Bases de Datos Relacionales la manera en la que puedes optimizar esta consulta en SQL Server. La base de datos está estructurada de esta manera en el modelo relacional. Explica los pasos que tomas para llegar a la solución, así como las partes de la consulta de manera simple pero concisa por favor.

```

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 calif>=6
and numEmpleado = 'P0000001'

group by c.boleta
having count(distinct c.clave) =
(select count(distinct clave) as cant_mat
from Escuela.Imparte
where numEmpleado = 'P0000001') ;
  
```

Marco González Luna  
Base de Datos Distribuidas Tarea 1

Solución propuesta por Gemini:

```
--/*Consulta gemini*/
-- Definimos cuántas materias imparte el profesor una sola vez
WITH MateriasProfesor AS (
    SELECT COUNT(DISTINCT clave) AS total_materias
    FROM Escuela.Imparte
    WHERE numEmpleado = 'P0000001'
)
SELECT
    c.boleta,
    COUNT(DISTINCT c.clave) AS cant_mat -- Quitamos el DISTINCT si la lógica de negocio garantiza no duplicados en Cursa
    FROM Escuela.Cursa c INNER JOIN Escuela.Imparte i
    ON
        c.clave = i.clave AND
        c.Semestre = i.semestre AND
        c.idGrupo = i.idGrupo
    CROSS JOIN MateriasProfesor mp -- Traemos el número mágico (total de materias)
WHERE
    c.calif >= 6
    AND i.numEmpleado = 'P0000001'
    GROUP BY c.boleta, mp.total_materias
    HAVING COUNT(c.clave) = mp.total_materias;
```

Results

boleta	cant_mat
2020630002	2

Análisis:

```
-- Definimos cuántas materias imparte el profesor una sola vez
WITH MateriasProfesor AS (
    SELECT COUNT(DISTINCT clave) AS total_materias
    FROM Escuela.Imparte
    WHERE numEmpleado = 'P0000001'
)
```

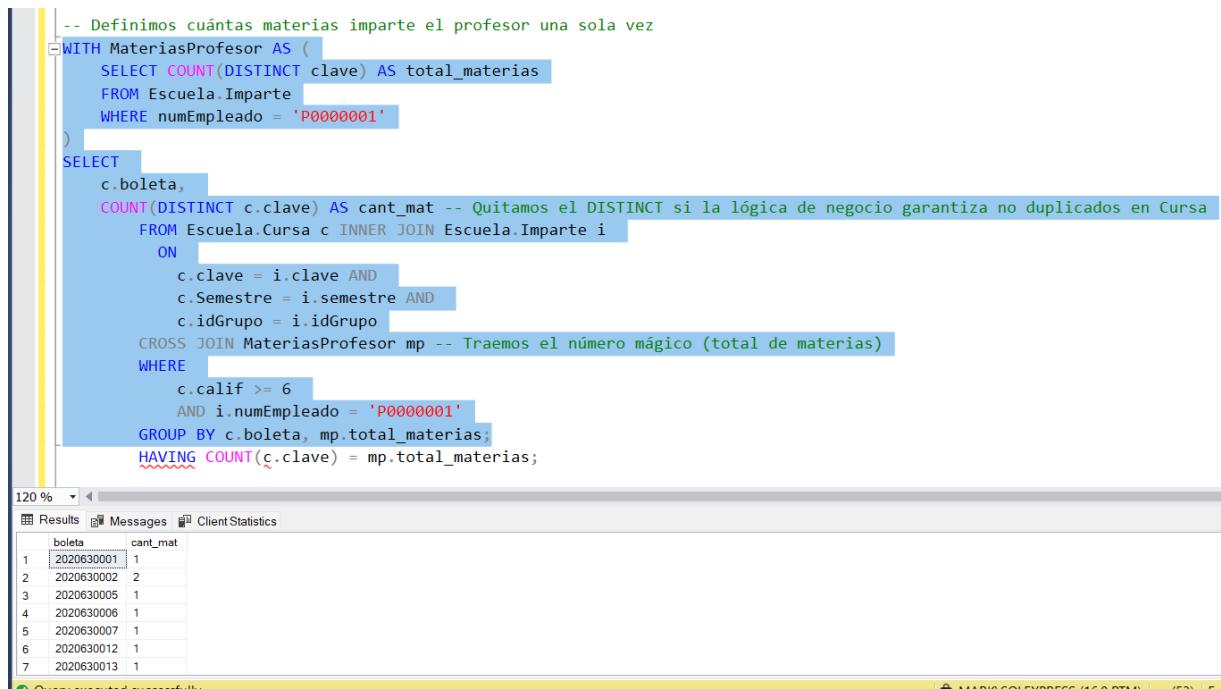
Aquí se realiza el cálculo del número total de materias que imparte el profesor, lo cual se hace una sola vez, reduciendo el uso de memoria, así como las veces que se ejecuta. De la manera anterior, la subconsulta se podría haber ejecutado muchas veces cada que el gestor encuentre más datos en una tabla larga de alumnos.

```
SELECT
    c.boleta,
    COUNT(DISTINCT c.clave) AS cant_mat -- Quitamos el DISTINCT si la lógica de negocio garantiza no duplicados en Cursa
    FROM Escuela.Cursa c INNER JOIN Escuela.Imparte i
    ON
        c.clave = i.clave AND
        c.Semestre = i.semestre AND
        c.idGrupo = i.idGrupo
    CROSS JOIN MateriasProfesor mp -- Traemos el número mágico (total de materias)
WHERE
    c.calif >= 6
    AND i.numEmpleado = 'P0000001'
    GROUP BY c.boleta, mp.total_materias
    HAVING COUNT(c.clave) = mp.total_materias;
```

Ahora bien, aquí se selecciona la boleta y la cantidad de materias del alumno que aprobó un curso con el profesor P0000001, que era la solución que ya teníamos y posteriormente debíamos comprobar con una subconsulta en el having.

La calificación aprobatoria y el profesor se comparan en la cláusula condicional de la consulta, pues eso permite que SQL Server use los índices de manera más eficiente antes de agrupar.

Se agrupa por boleta y por mp.total\_materias porque la consulta externa así lo requiere, además de que la cantidad de materias ahora es un valor constante, pero se tiene que agrupar la boleta con la columna para que se pueda usar en el having y mostrar en el resultado. Hasta este punto la consulta muestra la cantidad de materias que han cursado y aprobado los alumnos con el profesor P0000001.



```
-- Definimos cuántas materias imparte el profesor una sola vez
WITH MATERIASPROFESOR AS (
    SELECT COUNT(DISTINCT clave) AS total_materias
    FROM Escuela.Imparte
    WHERE numEmpleado = 'P0000001'
)
SELECT
    c.boleta,
    COUNT(DISTINCT c.clave) AS cant_mat -- Quitamos el DISTINCT si la lógica de negocio garantiza no duplicados en Cursa
    FROM Escuela.Cursa c INNER JOIN Escuela.Imparte i
    ON
        c.clave = i.clave AND
        c.Semestre = i.semestre AND
        c.idGrupo = i.idGrupo
    CROSS JOIN MATERIASPROFESOR mp -- Traemos el número mágico (total de materias)
    WHERE
        c.calif >= 6
        AND i.numEmpleado = 'P0000001'
    GROUP BY c.boleta, mp.total_materias;
    HAVING COUNT(c.clave) = mp.total_materias;
```

Finalmente, lo que garantiza que sólo se muestren los alumnos que han cursado todas las materias que imparte el profesor P0000001 es el having comparado con el valor obtenido de las materias totales impartidas por el profesor, que vienen del WITH y se pasan al CROSS JOIN y posteriormente a GROUP BY para que la consulta HAVING sea realizable.

Marco González Luna  
Base de Datos Distribuidas Tarea 1

The screenshot shows a SQL query editor window. At the top, there's a dark header bar with the text 'consultasEscuela.sq...SS.Escuela (sa (53))'. Below it is a toolbar with icons for file operations. The main area contains a SQL script titled '/\*Consulta gemini\*/'. The script uses common table expressions (WITH) to calculate the total number of subjects taught by a professor (P0000001) and then joins this with a table of student records (c) to find students who have taken exactly two subjects. The results are displayed in a table at the bottom.

```
--/*Consulta gemini*/
-- Definimos cuántas materias imparte el profesor una sola vez
WITH MateriasProfesor AS (
    SELECT COUNT(DISTINCT clave) AS total_materias
    FROM Escuela.Imparte
    WHERE numEmpleado = 'P0000001'
)
SELECT
    c.boleta,
    COUNT(DISTINCT c.clave) AS cant_mat -- Quitamos el DISTINCT si la lógica de negocio garantiza no duplicados en Cursa
    FROM Escuela.Cursa c INNER JOIN Escuela.Imparte i
    ON
        c.clave = i.clave AND
        c.Semestre = i.semestre AND
        c.idGrupo = i.idGrupo
CROSS JOIN MateriasProfesor mp -- Traemos el número mágico (total de materias)
WHERE
    c.calif >= 6
    AND i.numEmpleado = 'P0000001'
GROUP BY c.boleta, mp.total_materias
HAVING COUNT(c.clave) = mp.total_materias;
```

Results Messages Client Statistics

boleta	cant_mat
1	2020630002