

# Explicación y correcciones de la consulta

Tema: Validación de alumnos (boleta) que aprobaron materias impartidas por el profesor **P0000001**. Este documento resume la corrección de la consulta, por qué puede fallar y el significado de las palabras clave SQL usadas.

## 1. Consulta original

```
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')
```

## 2. ¿Qué se modificó y por qué?

**Modificación A: separar llaves del JOIN (ON) de filtros (WHERE).** En tu consulta, condiciones como **c.calif >= 6** e **i.numEmpleado = 'P0000001'** estaban dentro del ON. Aunque con *INNER JOIN* suele dar el mismo resultado, es mejor práctica mover filtros a **WHERE** para mantener el JOIN como relación entre tablas y evitar sorpresas si luego se cambia a *LEFT JOIN*.

**Modificación B (clave): alinear el 'universo' que se está contando.** Tu JOIN relaciona filas a nivel de impartición (clave + semestre + grupo), pero el conteo se hace solo por **clave**. Si el profesor imparte la misma clave en varios grupos o semestres, contar solo clave puede aceptar alumnos que no cubrieron todas las imparticiones.

### 3. Consulta corregida (misma idea, más limpia)

```
SELECT
    c.boleta,
    COUNT(DISTINCT c.clave) AS cant_mat
FROM Escuela.Cursa AS c
JOIN Escuela.Imparte AS i
    ON i.clave      = c.clave
    AND i.semestre = c.semestre
    AND i.idGrupo   = c.idGrupo
WHERE
    c.calif >= 6
    AND i.numEmpleado = 'P0000001'
GROUP BY
    c.boleta
HAVING
    COUNT(DISTINCT c.clave) =
    (
        SELECT COUNT(DISTINCT i2.clave)
        FROM Escuela.Imparte AS i2
        WHERE i2.numEmpleado = 'P0000001'
    );
```

**Importante:** esta versión valida por materia (**clave**). Si necesitas validar por impartición (clave + semestre + grupo), usa la alternativa del siguiente apartado.

### 4. ¿Por qué puede fallar la consulta original?

- **Fallo lógico (universos distintos):** se comparan conteos por clave, pero el JOIN trabaja a nivel de impartición.
- **Repetición de clave:** si el profesor imparte la misma clave en más de un grupo/semestre, el DISTINCT por clave oculta faltantes.
- **Duplicados por el JOIN:** si Imparte o Cursa tienen filas duplicadas para la misma combinación, el JOIN puede multiplicar filas. DISTINCT ayuda, pero puede no representar el requisito real.
- **Casos borde:** si el profesor no tiene registros en Imparte, el conteo esperado podría ser 0; según el FROM, podrían colarse resultados no deseados.

### 5. Alternativa correcta si tu requisito es 'todas las imparticiones'

```
WITH mat_prof AS (
    SELECT DISTINCT clave, semestre, idGrupo
    FROM Escuela.Imparte
    WHERE numEmpleado = 'P0000001'
)
SELECT
    c.boleta
FROM Escuela.Cursa c
JOIN mat_prof m
    ON m.clave      = c.clave
    AND m.semestre = c.semestre
    AND m.idGrupo   = c.idGrupo
WHERE c.calif >= 6
GROUP BY c.boleta
HAVING COUNT(*) = (SELECT COUNT(*) FROM mat_prof);
```

## **Notas de SQL Server - Consulta de alumnos vs materias del profesor**

---

Aquí el 'universo' del profesor se define como combinaciones únicas (clave, semestre, idGrupo). El alumno cumple si su número de coincidencias aprobadas es igual al total de imparticiones del profesor.

## 6. Glosario: ¿para qué sirve cada palabra clave?

**GROUP BY:** Agrupa filas para obtener una fila por grupo. Permite usar agregados como COUNT/SUM/AVG. En tu caso, agrupa por boleta para resumir resultados por alumno.

**DISTINCT:** Elimina duplicados. Dentro de COUNT(DISTINCT ...), evita contar la misma clave más de una vez.

**JOIN:** Combina filas de dos tablas. Con INNER JOIN solo quedan filas que existen en ambos lados.

**ON:** Define la condición de emparejamiento (las 'llaves' del JOIN). Debe describir cómo se relacionan las tablas.

**AND:** Operador lógico: todas las condiciones unidas por AND deben cumplirse.

**HAVING:** Filtra grupos después del GROUP BY. Se usa para condiciones sobre agregados (p. ej., COUNT).

**WITH (CTE):** Define una expresión de tabla común. Sirve para claridad, reutilización y para construir consultas por pasos.

**NOT EXISTS:** Devuelve verdadero si la subconsulta no devuelve filas. Útil para lógica tipo 'no hay faltantes'.

**EXCEPT:** Diferencia de conjuntos: devuelve filas que están en A y no están en B (A - B).

**COUNT:** Función agregada que cuenta filas o valores. En COUNT(DISTINCT clave) cuenta claves únicas por grupo.

## 7. ¿Qué hace COUNT en tu consulta?

En el SELECT, **COUNT(DISTINCT c.clave)** cuenta cuántas materias únicas (clave) tiene el alumno que cumplen el filtro (calif >= 6 y que además coinciden con imparticiones del profesor). En el HAVING, ese conteo se compara contra el número de materias (o imparticiones) que el profesor impartió para validar si el alumno cubre el conjunto completo.

## 8. Tres opciones alternativas para resolver el problema

**Opción A: NOT EXISTS (división relacional).** Expresa 'no existe una impartición del profe que el alumno no haya aprobado'.

```
SELECT a.boleta
FROM (SELECT DISTINCT boleta FROM Escuela.Cursa) AS a
WHERE NOT EXISTS (
    SELECT 1
    FROM Escuela.Imparte i
    WHERE i.numEmpleado = 'P0000001'
        AND NOT EXISTS (
            SELECT 1
            FROM Escuela.Cursa c2
            WHERE c2.boleta      = a.boleta
                AND c2.clave       = i.clave
                AND c2.semestre   = i.semestre
                AND c2.idGrupo    = i.idGrupo
                AND c2.calif      >= 6
        )
)
```

**Opción B: EXCEPT.** Calcula imparticiones faltantes: (impartidas por el profe) EXCEPT (aprobadas por el alumno). Si queda vacío, cumple.

```
SELECT a.boleta
FROM (SELECT DISTINCT boleta FROM Escuela.Cursa) AS a
WHERE NOT EXISTS (
    SELECT i.clave, i.semestre, i.idGrupo
    FROM Escuela.Imparte i
    WHERE i.numEmpleado = 'P0000001'

EXCEPT

SELECT c.clave, c.semestre, c.idGrupo
FROM Escuela.Cursa c
WHERE c.boleta = a.boleta
    AND c.calif >= 6
)
```

## Opción C: CTE + conteo.

Define el universo del profe y las aprobadas, luego compara conteos al mismo nivel (impartición).

```
WITH mat_prof AS (
    SELECT DISTINCT clave, semestre, idGrupo
    FROM Escuela.Imparte
    WHERE numEmpleado = 'P0000001'
),
aprobadas AS (
    SELECT DISTINCT boleta, clave, semestre, idGrupo
    FROM Escuela.Cursa
    WHERE calif >= 6
)
SELECT a.boleta
FROM aprobadas a
JOIN mat_prof m
    ON m.clave = a.clave
    AND m.semestre = a.semestre
    AND m.idGrupo = a.idGrupo
GROUP BY a.boleta
HAVING COUNT(*) = (SELECT COUNT(*) FROM mat_prof);
```

Nota: elige la opción según el enunciado. Si el requisito es por 'materia' usa COUNT(DISTINCT clave). Si el requisito es por 'impartición' (clave+semestre+grupo), compara a ese mismo nivel.