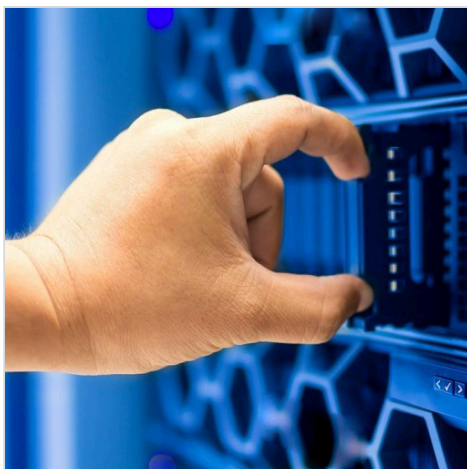


L U M B R E



# Subconsultas y CTEs

**Grado:** 3

**Materia:** SQL y Consultas de Datos · Bases de Datos

03/10/2026

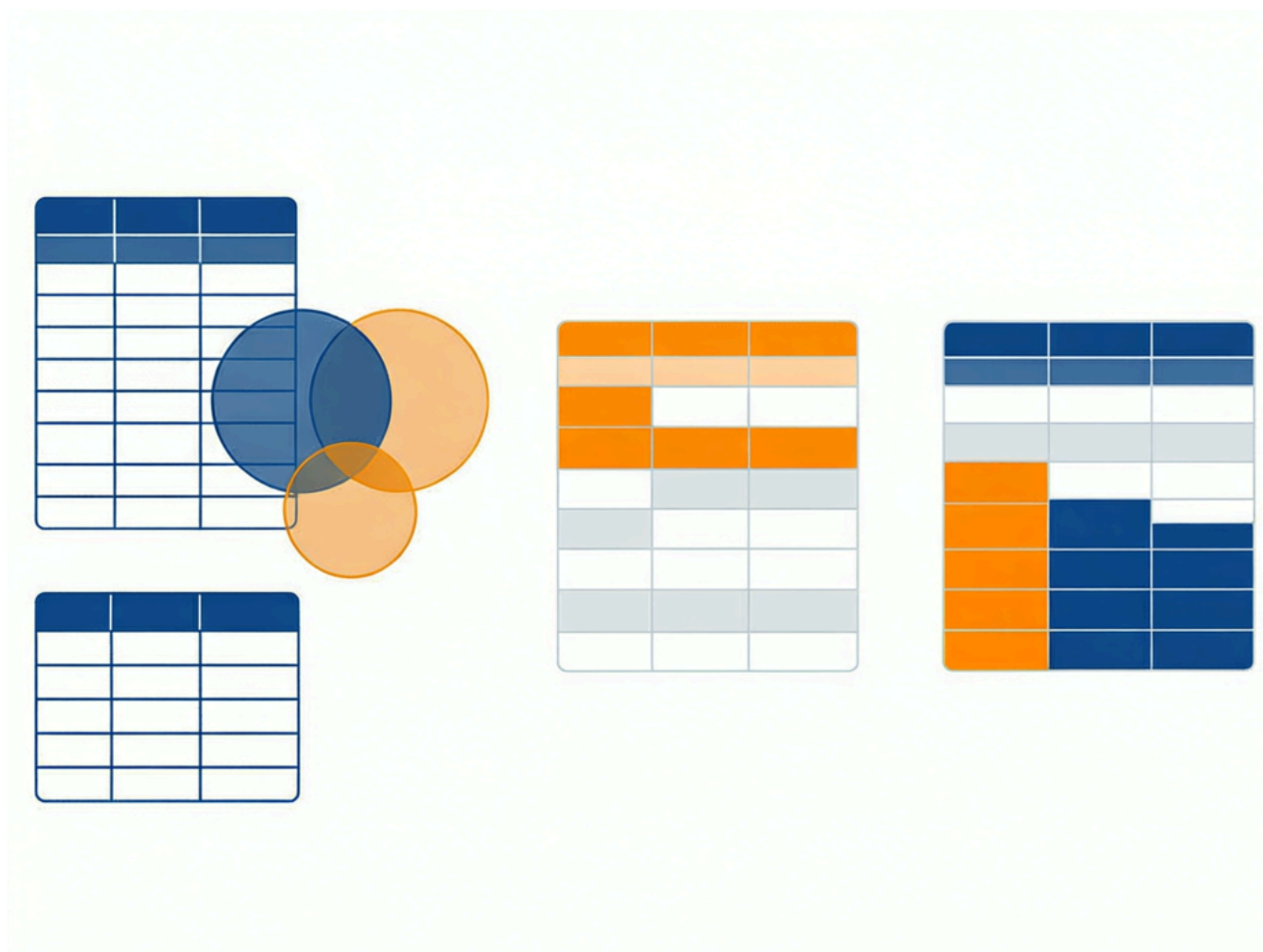
## Antes de empezar

Cuando una sola pasada no basta, se descompone el problema en pasos. Las subconsultas meten una consulta dentro de otra; las CTE con WITH dan nombre a esos pasos intermedios y hacen el SQL complejo legible y mantenible.

### QUÉ APRENDERÁS

- Componer consultas anidando subconsultas y refactorizando con CTE para separar pasos intermedios, leyendo mejor y reutilizando resultados parciales.

## Subconsultas y CTEs



*SQL: consultas, joins, agregacion y optimizacion de consultas.*

## Mapa conceptual

- **SQL**
  - Básico

- SELECT
- WHERE
- ORDER BY
- JOINS
  - INNER
  - LEFT
  - FULL
- Agregación
  - GROUP BY
  - HAVING
- Avanzado
  - Subconsultas
  - CTEs
  - Ventana
- Rendimiento
  - EXPLAIN
  - Índices

El mapa mental interactivo es interactivo en la versión online.



### Pensar por capas

Un análisis real rara vez es una sola consulta plana. Necesitas resultados intermedios: un umbral calculado, un top previo, un resumen para comparar. Aquí aprendes a componer pasos sin perder el hilo.

## Subconsulta escalar

Una subconsulta escalar devuelve un único valor que se usa donde iría un número, por ejemplo en WHERE precio mayor que el promedio de la tabla. El promedio se calcula en una consulta interna y alimenta la externa. Es la forma más directa de comparar filas contra una estadística global.

## Subconsulta de lista

Devuelve una columna de valores para alimentar un IN: productos que pertenecen a cierta categoría, o clientes con al menos una devolución. El IN filtra por pertenencia a

ese conjunto calculado.

## Subconsulta correlacionada

Es una subconsulta interna que referencia valores de la consulta externa y se re-evalúa por cada fila candidata. Potente para comprobaciones tipo «existe un pedido más reciente», pero costosa porque su trabajo se multiplica por el número de filas externas.

## Existir frente a contar

Para preguntar si hay al menos una coincidencia se usa EXISTS, que corta en cuanto encuentra una, en lugar de contar todas. Suele ser más eficiente que un IN sobre una subconsulta grande y maneja bien los NULL, porque no compara valores sino presencia de filas.

## La cláusula WITH (CTE)

WITH introduce una expresión de tabla común (CTE): con nombre un paso intermedio y luego lo usas como si fuera una tabla temporal. Las CTE convierten una maraña de subconsultas anidadas en una secuencia legible de etapas con nombre.

## Por qué preferir CTE

- Legibilidad: cada paso tiene nombre y propósito explícito, se lee de arriba abajo.
- Reutilización: la misma CTE se puede referenciar varias veces dentro de la consulta.
- Depuración: puedes aislar una CTE y probarla por separado.
- Composición: varias CTE encadenadas modelan un flujo paso a paso.

## CTE recursiva

Una CTE puede referirse a sí misma para recorrer jerarquías o generar secuencias: la estructura de árbol de una organización o una cadena de dependencias. Consta de un caso base y una parte recursiva unida por UNION ALL. Es la herramienta natural para datos anidados.

## Anatomía de una CTE

1. WITH seguido de un nombre y, entre paréntesis, la consulta que la produce.
2. Una o más CTE separadas por comas, cada una pudiendo usar las anteriores.
3. La consulta principal que consume las CTE como tablas.

## Del subselect a la CTE

Toda subconsulta se puede reescribir como CTE y viceversa, en esencia. La diferencia es de claridad: anidar cuatro subconsultas crea un barril ilegible; cuatro CTE nombradas cuentan una historia. En producción, la versión legible gana casi siempre.

## Rendimiento comparado

Algunos motores materializan la CTE una sola vez y la reutilizan; otros la expanden como una subconsulta en cada referencia. El comportamiento depende del sistema, así que conviene revisar el plan de ejecución cuando la consulta es grande y crítica.

*Cada mecanismo de composición resuelve una forma distinta de anidar lógica.*

| Estructura                | Devolución           | Buen uso                                   |
|---------------------------|----------------------|--------------------------------------------|
| Subconsulta escalar       | un valor             | comparar filas contra un resumen global    |
| Subconsulta de lista      | una columna          | alimentar un IN de pertenencia             |
| Correlacionada con EXISTS | verdadero o falso    | comprobar presencia fila a fila            |
| CTE con WITH              | una tabla intermedia | descomponer un análisis en pasos nombrados |

### 1. ¿Qué ventaja principal aporta una CTE frente a subconsultas anidadas?

- ☐ Ejecuta siempre más rápido
- ☐ Da nombre a pasos intermedios y hace la consulta legible y reutilizable
- ☐ Permite evitar el GROUP BY
- ☐ Sustituye a los JOIN

## Nombres que documentan

Un buen nombre de CTE describe su contenido: `ventas_mensuales`, `clientes_vip`, `ultimo_pedido`. La consulta final se lee entonces casi como lenguaje natural y se vuelve autoexplicativa, un activo para quien la herede.

## Cuándo NO conviene

Para un filtro puntual que solo se usa una vez, una subconsulta en línea puede ser más clara que abrir una CTE. El criterio es la reaparición: si el paso intermedio se usa una sola vez y es trivial, mantenlo dentro; si se repite o complica, nómbralo.

## Errores frecuentes

- Re-evaluar una subconsulta correlacionada sobre millones de filas sin notar el coste.

- Olvidar el UNION ALL adecuado en una CTE recursiva y provocar un bucle infinito.
- Referenciar una CTE antes de que esté definida en la lista de WITH.

## 2. Una CTE recursiva es apropiada para recorrer una jerarquía de padre e hijo.

☐ Verdadero

☐ Falso

## Plan mental de composición

Ante un problema complejo, identifica primero los resultados intermedios y ponles nombre. Cada uno se vuelve una CTE. Al final queda un guión: preparar, calcular, combinar, presentar. Ese enfoque por capas es transferible a cualquier motor.

ACTIVIDAD INTERACTIVA DISPONIBLE EN  
LUMBRE

### 3. Composición de consultas



## 4. Ordena los pasos para atacar un análisis complejo con CTE:

### Elementos:

- detecta qué resultados intermedios necesitas
- exprésalos en WITH con nombre claro
- usa CTE encadenadas que consuman las anteriores
- escribe la consulta final leyendo las CTE

|                   |       |
|-------------------|-------|
| <b>Intermedio</b> | . . . |
| <b>Nombre</b>     | . . . |
| <b>Encadenar</b>  | . . . |
| <b>Final</b>      | . . . |

Use The Index, Luke <https://use-the-index-luke.com/> Rendimiento de SQL, subconsultas y planes de ejecución.

## VIDEO

**Video complementario**

Recurso audiovisual para reforzar los conceptos.



Escanea para verlo en Lumbre

<https://www.youtube.com/watch?v=HX6q1bsBz1w>

**5. Reescribe como CTE una consulta que lista clientes cuyo gasto supera el gasto medio, y explica qué paso intermedio aíslas y por qué mejora la legibilidad frente a la subconsulta anidada.**

SQLZoo <https://sqlzoo.net/Ejercicios interactivos de SQL en línea>.

## Referencia curricular

|                 |                                           |
|-----------------|-------------------------------------------|
| <b>Grado:</b>   | 3                                         |
| <b>Materia:</b> | SQL y Consultas de Datos · Bases de Datos |

Versión interactiva completa (con actividades, videos y autoevaluación):



<https://lumbre.lat/object/cdia-sql-004-subconsultas-y-ctes>

Material educativo generado desde Lumbre · [lumbre.lat](https://lumbre.lat)