

Subconsultas Sintaxis, ejecución y características esenciales

Conceptos clave para consultas eficaces en bases de datos

Temario

- Concepto y propósito de las subconsultas en SQL
- Sintaxis y proceso de ejecución de una subconsulta
- Características fundamentales de las subconsultas
- Uso de operadores específicos con subconsultas
- Guía para el Uso Efectivo de Subconsultas

Concepto y propósito de las subconsultas en SQL

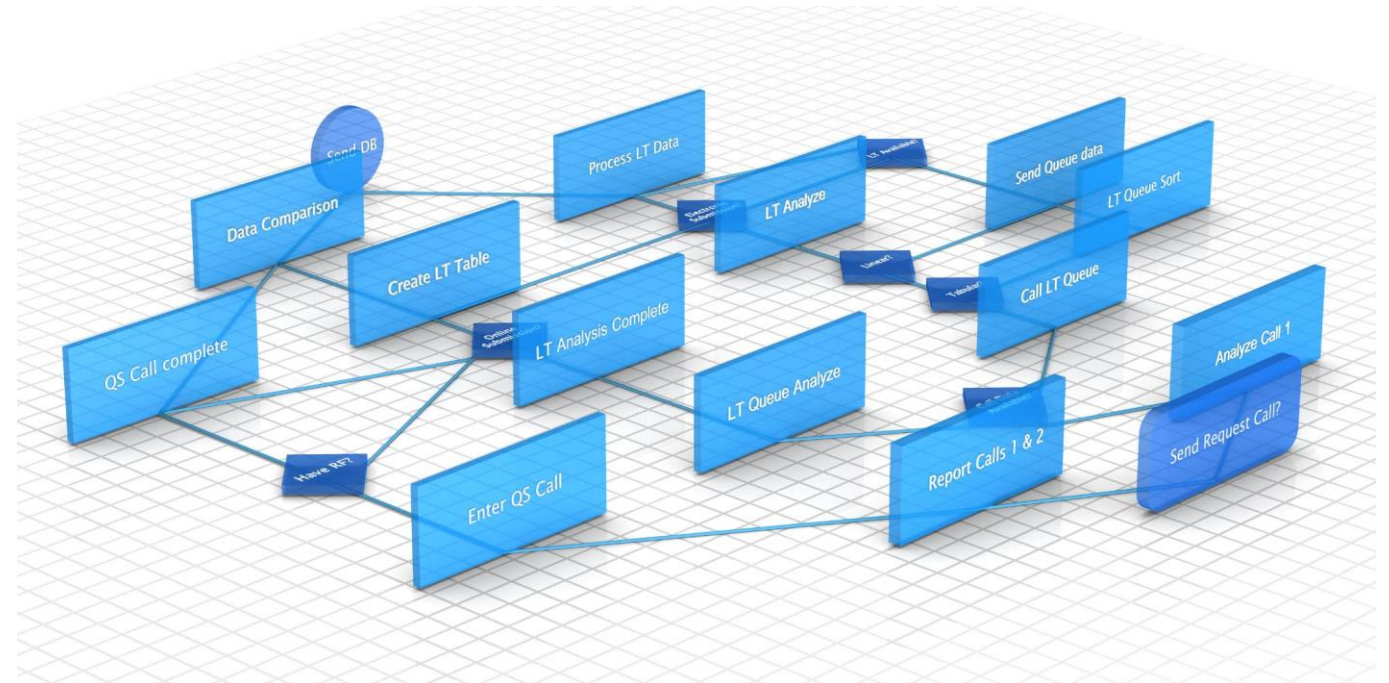
Definición de subconsulta y consulta principal

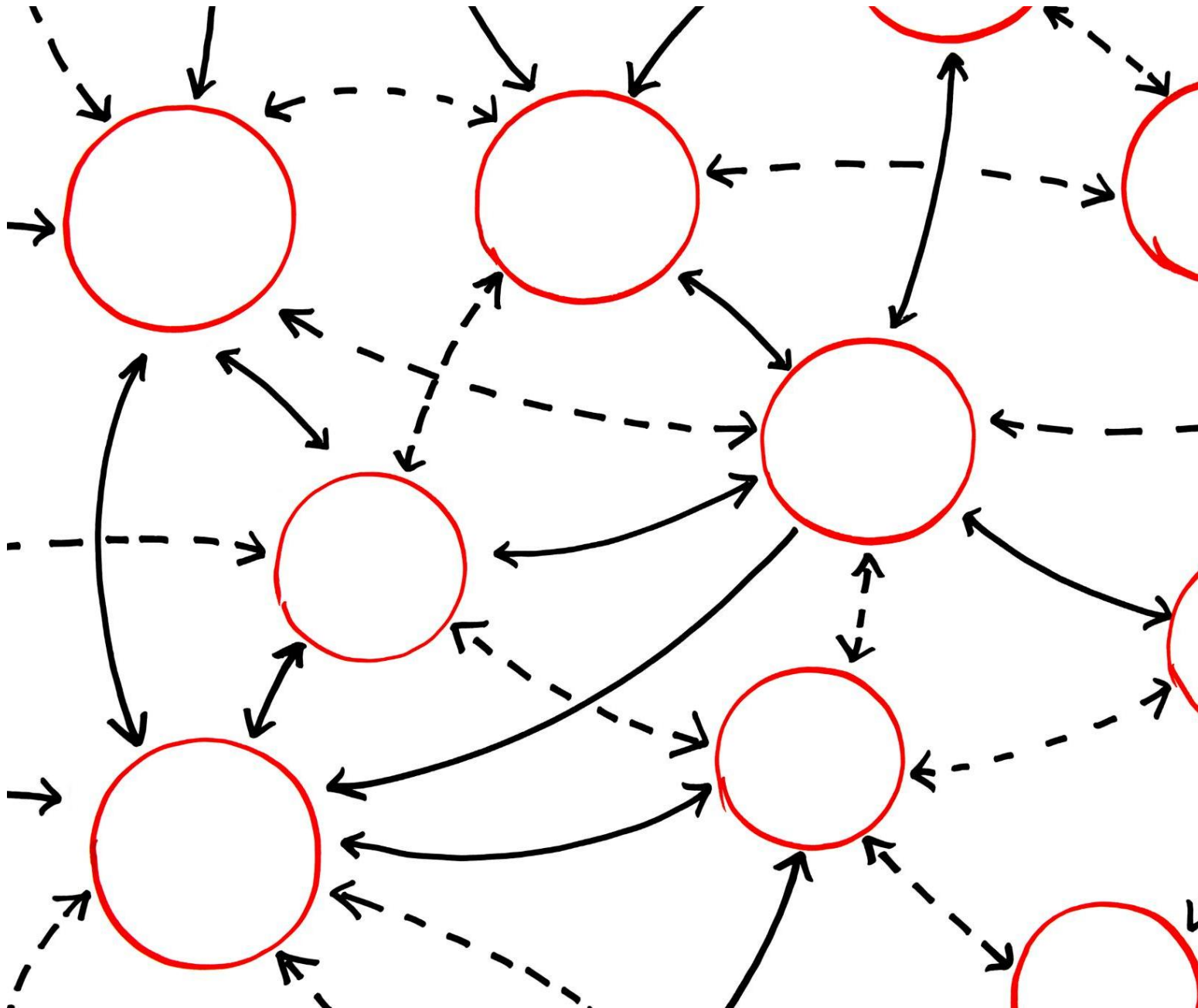
Concepto de subconsulta

Una subconsulta es una consulta dentro de otra que se evalúa primero para proporcionar resultados.

Función de la consulta principal

La consulta principal utiliza los resultados de la subconsulta para filtrar o calcular datos finales.





Ventajas de utilizar subconsultas

Simplificación de Consultas

Las subconsultas reducen la complejidad al integrar varias consultas en una sola estructura coherente.

Mejora de Legibilidad

Facilitan la comprensión del código al organizar mejor las consultas y sus resultados parciales.

Reutilización de Resultados

Permiten usar resultados intermedios sin ejecutar múltiples consultas independientes.

Diferencia entre subconsulta y consulta tradicional

Consulta Tradicional

Una consulta tradicional se ejecuta de forma independiente y devuelve resultados sin depender de otra consulta.

Subconsulta

La subconsulta se ejecuta dentro de otra consulta y puede afectar sus resultados, siendo correlacionada o no correlacionada.



Sintaxis y
proceso de
ejecución de una
subconsulta

La subconsulta se ejecuta una vez y antes de la consulta principal.
El resultado de ella es usado por la consulta principal externa.

```
SELECT lista_de_campos  
FROM tabla  
WHERE condición | HAVING condición  
                (SELECT lista_de_campos  
                 FROM tabla);
```


Estructura y formato correcto de una subconsulta

Delimitación con paréntesis

Las subconsultas deben estar siempre entre paréntesis para ser reconocidas correctamente en la consulta principal.

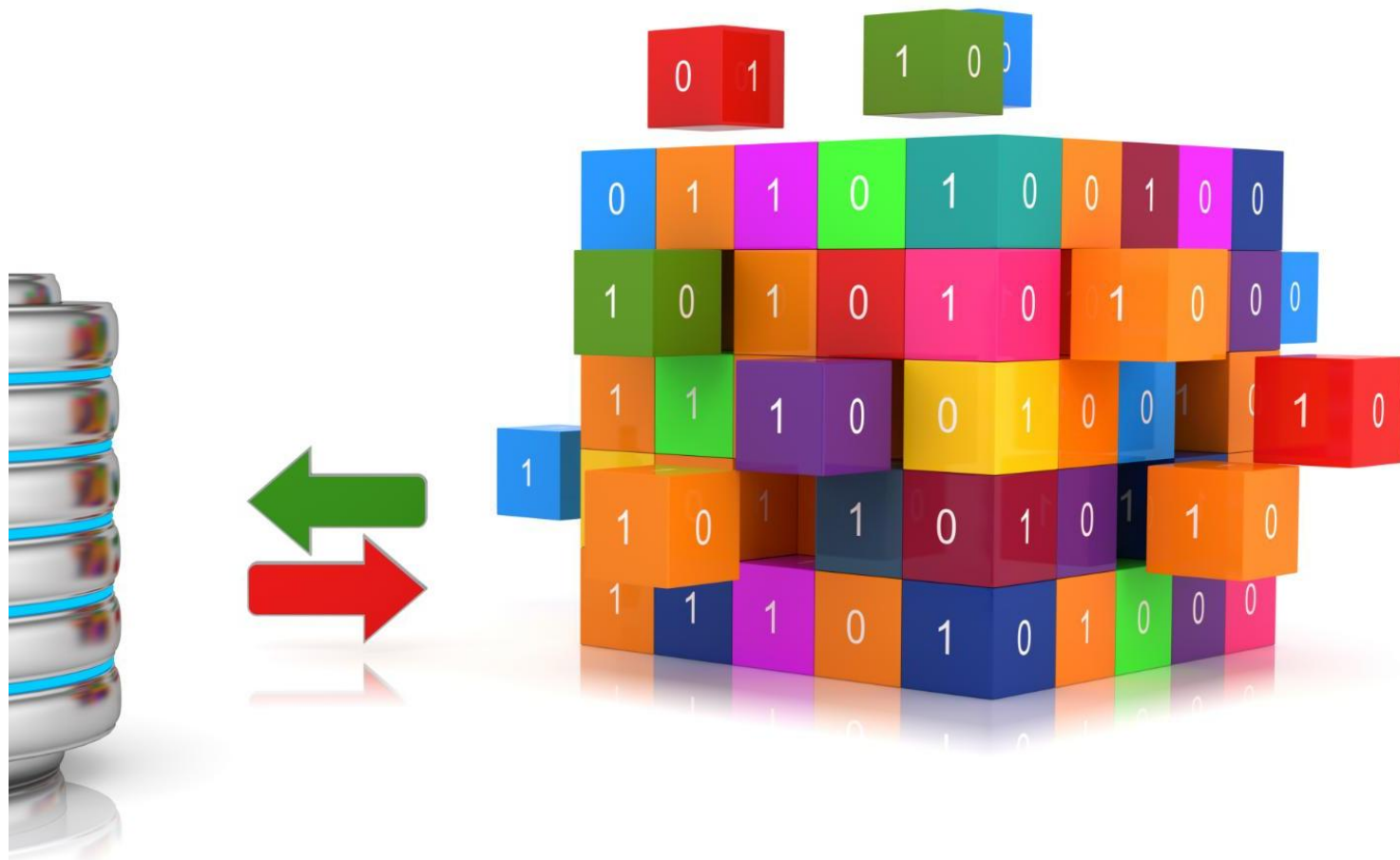
Ubicación en cláusulas permitidas

Las subconsultas pueden ubicarse en las cláusulas WHERE, FROM o SELECT dentro de una consulta principal.

Respeto a la sintaxis SQL

Es fundamental respetar la sintaxis SQL para que la subconsulta funcione correctamente dentro de la consulta principal.





Orden de ejecución entre subconsulta y consulta externa

Ejecución de la Subconsulta

La subconsulta se ejecuta primero para obtener los datos o resultados necesarios para la consulta externa.

Uso de Resultados en Consulta Externa

La consulta externa utiliza los resultados de la subconsulta para completar su procesamiento y obtener el resultado final.

```

135 m_ftw = float.Positive
136 m_fts = float.Positive
137 m_fNW = float.Positive
138 m_fNS = float.Positive
139 if (ro>1)
140     float ro = lambda\mu;
141     float mu = l\Etp;
142     float lambda = l\Eta;
143 }
144 void CalcG1(float Eta, float
145 {
146     if (ro>1)
147     {
148         m_ftw = W;
149         m_fts = float.Positive;
150         m_fNW = float.Positive;
151         m_fNS = float.Positive;
152     }
153     return;
154 }
155 CalcP(v, ro, m_APN;
156 float v = 0.5f*(1+(float
157 double vp = (s*a)\(Etp*Etf
158 double s = (double)Etp\Ms
159 m_ftw = ((kfoat+1) \ (2*
160 m_fts \ lambda;
161 m_fNW = (lambda*lambda\k
162 m_fNS = (ro \ (1-ro)) *
163 {
164     m_ftw = W;
165     m_fts = float.Positive;
166     m_fNW = float.Positive;
167     m_fNS = float.Positive;
168     if (ro>1)
169     {
170         m_ftw = W;
171         m_fts = float.Positive;
172         m_fNW = float.Positive;
173         m_fNS = float.Positive;
174     }
175     return;
176 }
177 CalcP(0.5f, ro, m_APN;

```

Los paréntesis delimitan subconsultas, asegurando que el motor SQL interprete correctamente la jerarquía.

El uso correcto de paréntesis garantiza el orden adecuado en la ejecución de consultas anidadas.

Características fundamentales de las subconsultas

Anidación y construcción de lógicas complejas

Uso de Subconsultas

Las subconsultas permiten anidar consultas dentro de otras para construir análisis detallados.

Construcción de Lógicas Complejas

Se pueden aplicar filtros avanzados combinando múltiples condiciones para un análisis específico.

Ubicación flexible en cláusulas SQL

Subconsultas en cláusula WHERE

Permite filtrar resultados basados en condiciones específicas evaluadas por la subconsulta en la cláusula WHERE.

Subconsultas en cláusula FROM

Facilita la creación de tablas derivadas para análisis complejos usando subconsultas dentro de FROM.

Subconsultas en cláusula SELECT

Permite calcular valores dinámicos para cada fila mediante subconsultas en la cláusula SELECT.

Subconsultas en cláusula HAVING

Usa subconsultas para aplicar condiciones a grupos de resultados en la cláusula HAVING.



Tipos de resultados de retorno: mono-registro y multi-registro

Resultado Mono-registro

Las subconsultas mono-registro devuelven un único valor o fila según la lógica de la consulta.

Resultado Multi-registro

Las subconsultas multi-registro pueden devolver múltiples valores o filas basadas en los criterios de búsqueda.



Uso de
operadores
específicos con
subconsultas

Operadores IN, NOT IN, ANY y ALL

Operadores para comparaciones múltiples

IN y NOT IN comparan valores con conjuntos para incluir o excluir registros fácilmente.

Uso de ANY y ALL

ANY verifica si algún valor cumple la condición, ALL exige que todos los valores la cumplan.



El operador EXISTS y su utilidad

Verificación de existencia

EXISTS comprueba si existe al menos un registro que cumple con la condición en una subconsulta.

Optimización de consultas

Utilizar EXISTS mejora el rendimiento al evitar búsqueda innecesarias de datos completos.

Condiciones booleanas eficientes

EXISTS permite construir condiciones booleanas basadas en la presencia o ausencia de datos específico



Aplicación del operador NOT en condiciones

Función del operador NOT

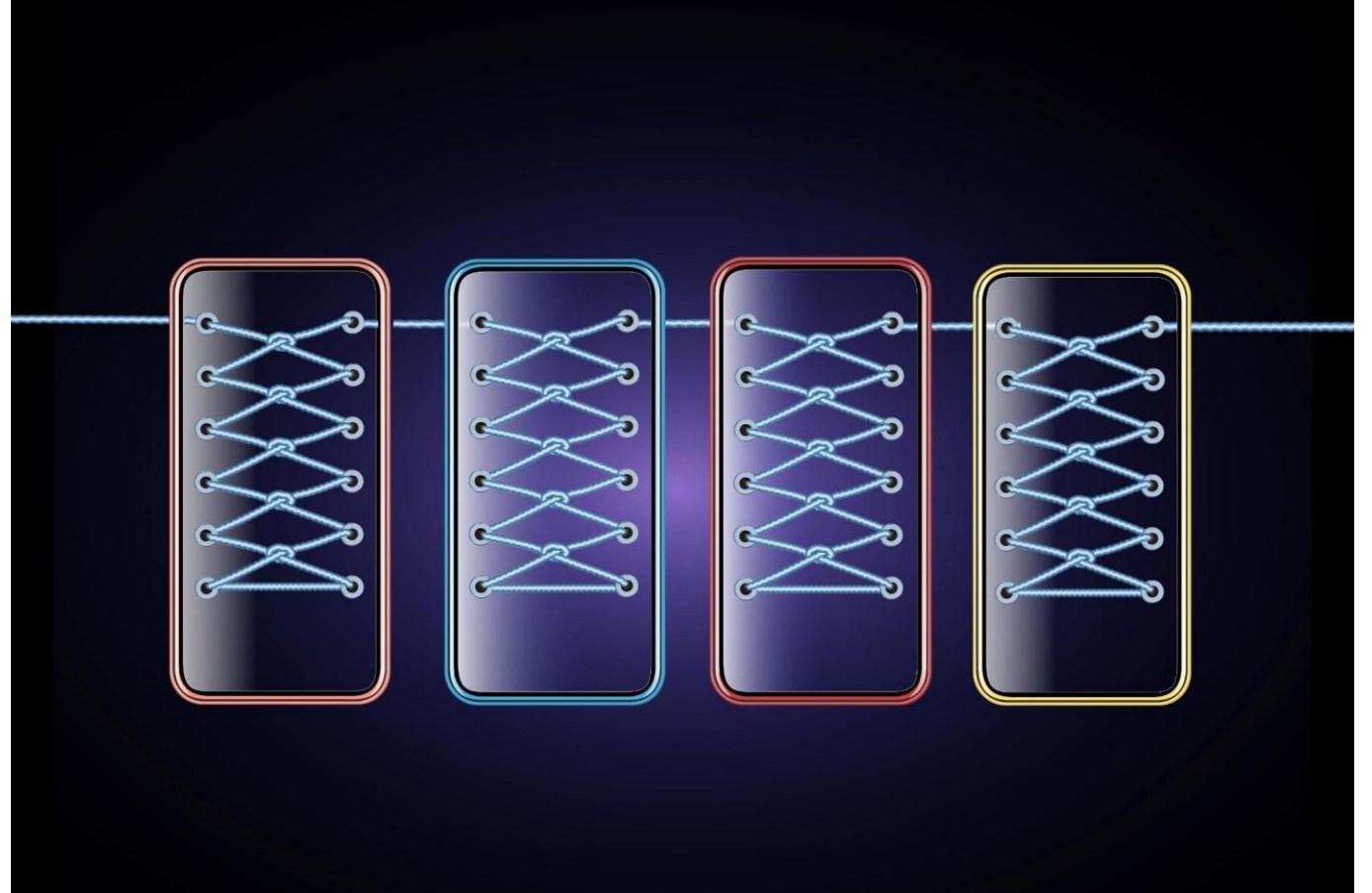
El operador NOT se utiliza para invertir el resultado de una condición lógica en consultas.

Uso con EXISTS

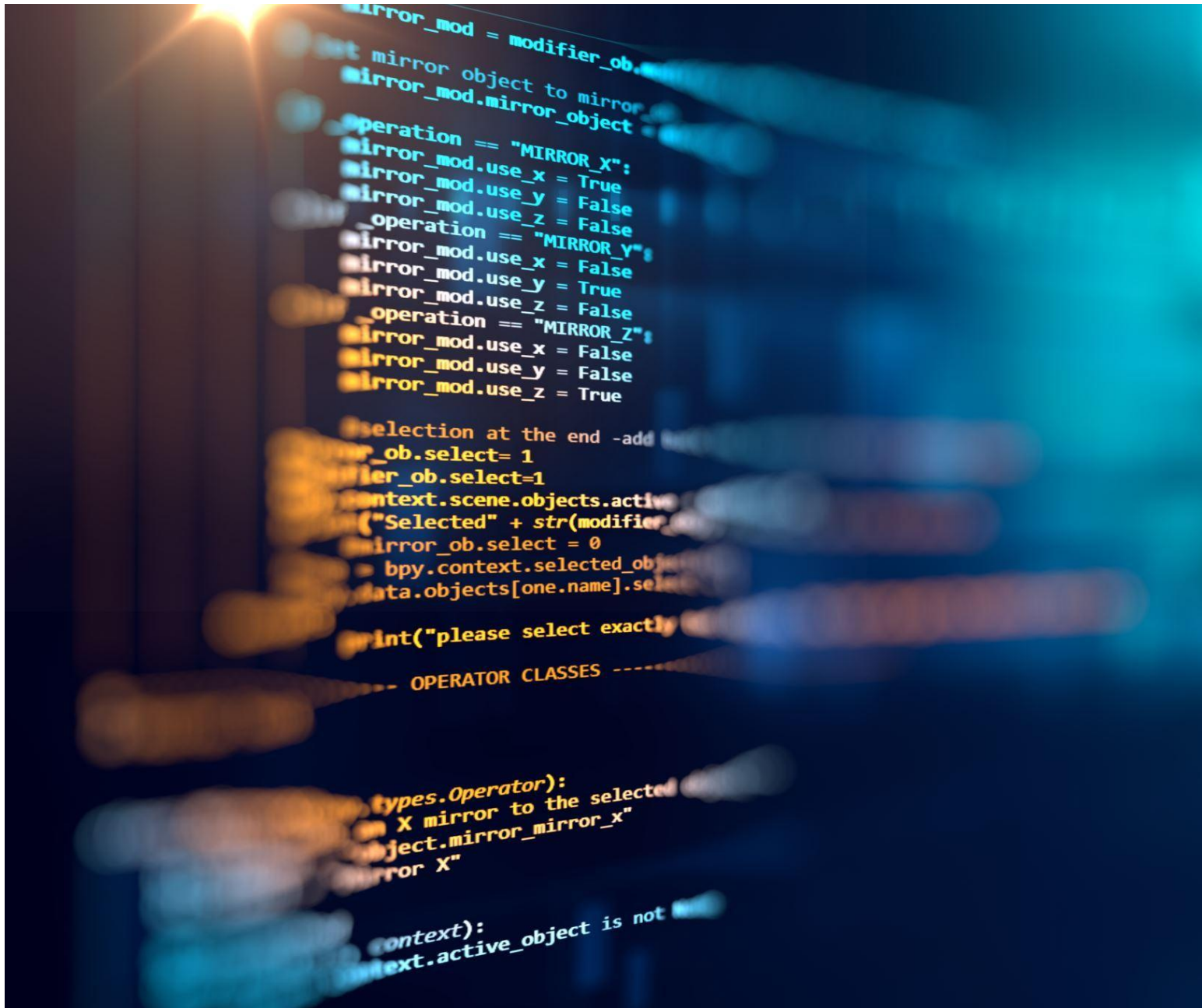
NOT combinado con EXISTS permite excluir registros que existen según una subconsulta.

Uso con IN

NOT junto con IN filtra datos que no están en una lista determinada por la subconsulta.



Guía para el Uso Efectivo de Subconsultas



Directrices para sintaxis y operadores en subconsultas

Sintaxis clara

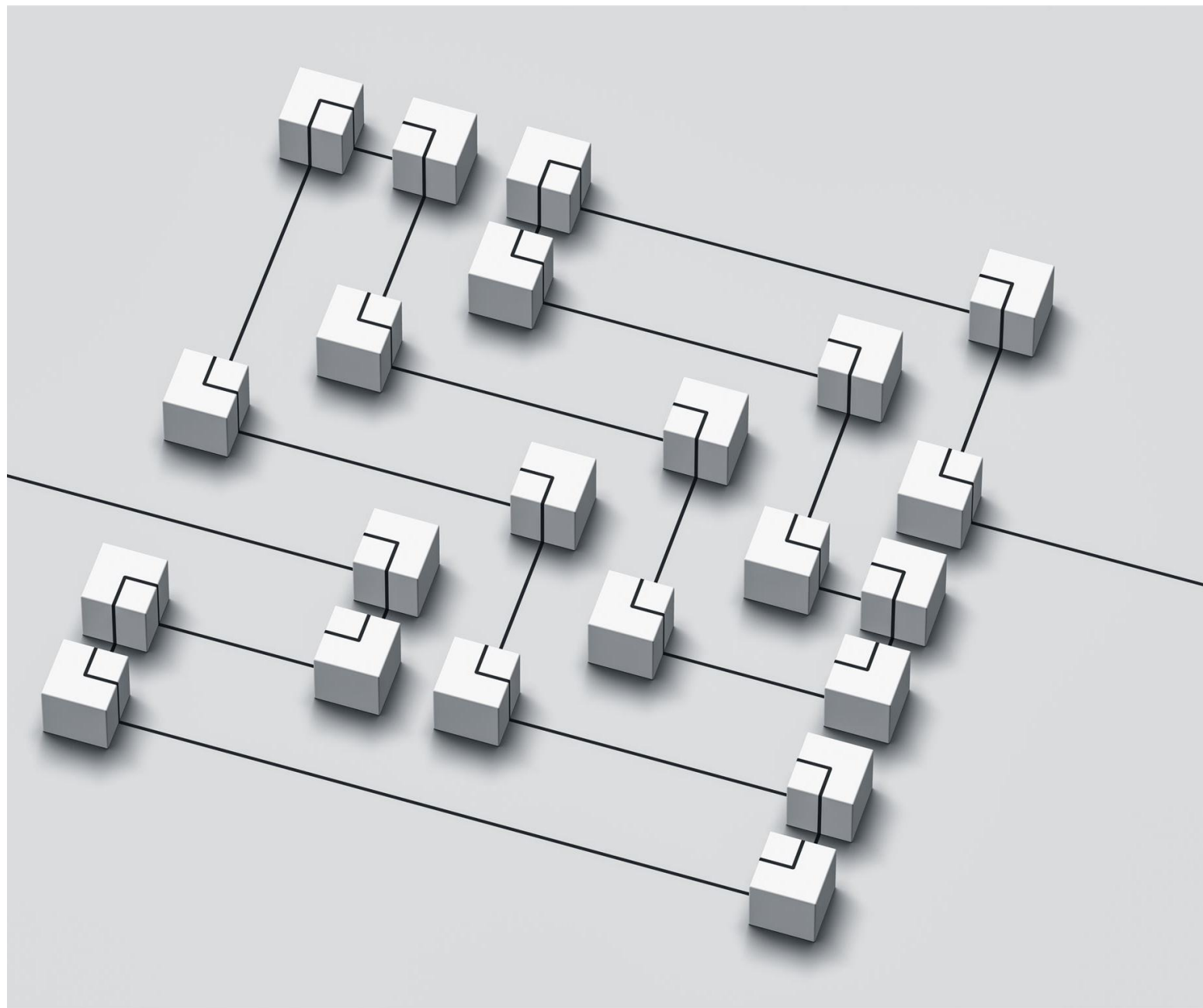
Una sintaxis clara facilita la comprensión y el mantenimiento de las subconsultas en bases de datos.

Uso correcto de paréntesis

El uso preciso de paréntesis asegura la correcta ejecución y agrupación de las subconsultas.

Selección adecuada de operadores

Elegir operadores apropiados garantiza que las subconsultas realicen las operaciones deseadas eficazmente.



Subconsultas mono-registro y multi-registro: características y ejemplos

Subconsultas Mono-registro

Retornan un único valor, perfectas para comparaciones directas en consultas SQL.

Subconsultas Multi-registro

Retornan conjuntos de valores, usadas para comparaciones múltiples en bases de datos.

Empleo de subconsultas en la cláusula FROM

Tablas Derivadas Temporales

Las subconsultas en FROM crean tablas derivadas temporales para organizar datos de manera eficiente.

Facilita Consultas Complejas

Permite realizar consultas complejas al dividir el procesamiento en pasos manejables.

Reorganización de Datos

Subconsultas reorganizan y preparan los datos antes del análisis final, mejorando la precisión.

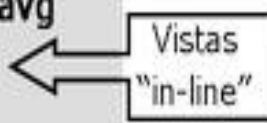


Guía Uso de Subconsultas

- Encierre las subconsultas entre paréntesis.
- No añada una cláusula ORDER BY a una subconsulta.
- Utilice operadores a nivel de fila para subconsultas que devuelvan solo una fila-->MONOREGISTRO.
- Utilice operadores que actúan sobre varios registros para subconsultas que devuelven más de una fila → MULTI REGISTRO.

```
SQL> SELECT ename, job
      FROM emp
      WHERE job = (SELECT job FROM EMP
                  WHERE empno = 7369) → CLERK
      AND    sal > (SELECT sal FROM EMP
                  WHERE empno = 7876); → 1100
```

```
SQL> SELECT a.ename, a.sal, a.deptno, b.salavg
      FROM emp a, (SELECT deptno, avg(sal) salavg
                  FROM emp
                  GROUP BY deptno) b
      WHERE a.deptno = b.deptno
      AND a.sal > b.salavg;
```



Vistas
"in-line"

```
SQL> SELECT ename, sal, deptno
      FROM emp
      WHERE sal IN ( SELECT MIN(sal) FROM emp
                    GROUP BY deptno );
```

```
SQL> SELECT ename, sal, job
      FROM emp
      WHERE sal < ANY ( SELECT sal FROM emp
                      WHERE job = 'CLERK')
      AND job <> 'CLERK';
```

```
SQL> SELECT empno, ename, job
      FROM emp
      WHERE sal > ALL ( SELECT avg(sal) FROM emp
                      GROUP BY deptno);
```


Subconsultas Mono registro

- Devuelven un único registro.
- Se utilizan operadores de comparación (=, >, >=, <, <= y <>).

Ejemplo:

```
SQL> SELECT ename, job
      FROM emp
      WHERE job = (SELECT job FROM EMP
                   WHERE empno = 7369) → CLERK
      AND    sal > (SELECT sal FROM EMP
                   WHERE empno = 7876); → 1100
```

Subconsultas Multi registro

Devuelven más de un registro

Se utilizan comparadores multiregistro:

El operador NOT puede ser utilizado con los operadores IN, ANY y ALL.

- IN → TRUE si se encuentra en la lista.
- ANY (y sinónimo SOME) → TRUE si la condición se cumple con algún registro de la lista devuelta por la subconsulta.
- ALL → TRUE si la condición se cumple con todos los registros de la lista devuelta por la subconsulta.

Ejemplo subc. Multi registro

```
SQL> SELECT ename, sal, deptno  
      FROM emp  
      WHERE sal IN ( SELECT MIN(sal) FROM emp  
                    GROUP BY deptno );
```

```
SQL> SELECT ename, sal, job  
      FROM emp  
      WHERE sal < ANY ( SELECT sal FROM emp  
                       WHERE job = 'CLERK')  
      AND job <> 'CLERK';
```

```
SQL> SELECT empno, ename, job  
      FROM emp  
      WHERE sal > ALL ( SELECT avg(sal) FROM emp  
                       GROUP BY deptno);
```

Conclusión

Importancia de las subconsultas

Las subconsultas incrementan la capacidad analítica al permitir consultas complejas y específicas en bases de datos.

Sintaxis y ejecución

Conocer la sintaxis correcta y la ejecución eficiente es clave para diseñar consultas robustas y efectivas.