**1. Introducción a la sentencia SELECT**

La sentencia **SELECT** se utiliza para recuperar información almacenada en una base de datos. Es la consulta más básica pero también una de las más poderosas de SQL, porque nos permite:

1. **Seleccionar** registros específicos de una tabla utilizando criterios (cláusulas) que filtran qué datos queremos ver.
2. **Proyectar** columnas específicas, seleccionando solo las que necesitamos, y no todas las columnas disponibles.
3. **Unir** datos de diferentes tablas relacionadas, utilizando enlaces como claves primarias y foráneas.

**Estructura básica de SELECT:**

sql

Copiar código

SELECT [DISTINCT] columna1, columna2, ...

FROM tabla1 [, tabla2, ...]

[WHERE condición]

**2. Desglose de la sintaxis**

**a) Seleccionar todas las columnas**

Si deseas obtener **todas las columnas** de una tabla, utiliza el carácter asterisco (\*):

sql

Copiar código

SELECT \* FROM empleados;

Esto mostrará todos los datos de la tabla empleados.

**b) Seleccionar columnas específicas**

Si solo necesitas ciertas columnas, indica sus nombres después de **SELECT**:

sql

Copiar código

SELECT nombre, salario FROM empleados;

Esto devolverá únicamente las columnas nombre y salario.

**c) Eliminación de duplicados con DISTINCT**

Cuando existen valores repetidos, la palabra clave **DISTINCT** elimina los duplicados:

sql

Copiar código

SELECT DISTINCT departamento FROM empleados;

Esto mostrará cada departamento una sola vez, sin repetirlos.

**3. Manipulación de datos en SELECT**

**a) Expresiones aritméticas**

Puedes realizar cálculos directamente en la consulta:

sql

Copiar código

SELECT nombre, salario, salario + 500 AS "Nuevo Salario"

FROM empleados;

Esto sumará 500 al salario de cada empleado y mostrará el resultado como una columna llamada "Nuevo Salario".

* Operadores básicos:
  + **+** suma
  + **-** resta
  + **\*** multiplicación
  + **/** división

**Regla de precedencia:** La multiplicación y división tienen prioridad sobre la suma y resta. Usa paréntesis para cambiar el orden de las operaciones:

sql

Copiar código

SELECT (salario + 500) \* 1.1 AS "Ajuste Total"

FROM empleados;

**b) Trabajar con valores nulos (NULL)**

Un valor **NULL** no es igual a 0 ni a un espacio en blanco; simplemente significa que no hay un valor almacenado. Si usas un cálculo donde hay **NULL**, el resultado será **NULL**:

sql

Copiar código

SELECT nombre, salario, salario + NULL AS "Resultado"

FROM empleados;

Esto devolverá **NULL** en la columna "Resultado".

**4. Alias de columnas**

Los alias permiten renombrar columnas en los resultados, facilitando su comprensión. Para esto puedes usar:

* La palabra clave **AS**
* O directamente un nombre nuevo (con o sin comillas dobles si contiene espacios o caracteres especiales).

Ejemplo:

sql

Copiar código

SELECT nombre AS "Empleado", salario\*12 AS "Salario Anual"

FROM empleados;

**5. Concatenación de cadenas**

El operador **||** permite combinar textos y columnas en una sola salida:

sql

Copiar código

SELECT nombre || ' trabaja como ' || puesto AS "Información del Empleado"

FROM empleados;

Esto crea una salida como: Juan trabaja como Analista.

**6. Filtrar datos con la cláusula WHERE**

La cláusula **WHERE** filtra registros según una condición específica. Es esencial para limitar los resultados.

**Ejemplo básico:**

sql

Copiar código

SELECT nombre, puesto

FROM empleados

WHERE puesto = 'CLERK';

* **Cadenas de texto**: Distinguen entre mayúsculas y minúsculas (case sensitive).
* **Números**: No necesitan comillas.
* **Fechas**: Utiliza el formato estándar SQL DATE 'YYYY-MM-DD':

sql

Copiar código

SELECT nombre

FROM empleados

WHERE fecha\_contratacion = DATE '2023-01-01';

**Operadores de comparación en WHERE**

| **Operador** | **Significado** |
| --- | --- |
| = | Igual a |
| > | Mayor que |
| < | Menor que |
| >= | Mayor o igual que |
| <= | Menor o igual que |
| <> o != | Distinto de |

**7. Rangos y patrones en WHERE**

**a) BETWEEN ... AND ...**

Para buscar valores dentro de un rango (incluyendo los extremos):

sql

Copiar código

SELECT nombre, salario

FROM empleados

WHERE salario BETWEEN 1000 AND 2000;

**b) LIKE**

Busca patrones en cadenas utilizando comodines:

* %: Cualquier secuencia de caracteres.
* \_: Un solo carácter.

Ejemplo:

sql

Copiar código

SELECT nombre

FROM empleados

WHERE nombre LIKE 'J%'; -- Nombres que empiezan con J

**c) Valores nulos (IS NULL)**

Para identificar valores que no tienen contenido:

sql

Copiar código

SELECT nombre

FROM empleados

WHERE bono IS NULL;

**8. Ordenar resultados con ORDER BY**

La cláusula **ORDER BY** organiza los resultados en un orden específico:

* **ASC** (ascendente) es el valor predeterminado.
* **DESC** (descendente) debe especificarse explícitamente.

Ejemplo:

sql

Copiar código

SELECT nombre, salario

FROM empleados

ORDER BY salario DESC;

**9. Operadores lógicos**

Permiten combinar múltiples condiciones:

* **AND**: Ambas condiciones deben ser verdaderas.
* **OR**: Una de las condiciones debe ser verdadera.
* **NOT**: Invierte el resultado.

Ejemplo:

sql

Copiar código

SELECT nombre

FROM empleados

WHERE salario > 1000 AND puesto = 'ANALYST';

**10. Consultas avanzadas**

**Unir tablas con relaciones**

Las consultas pueden combinar información de diferentes tablas usando claves compartidas:

sql

Copiar código

SELECT e.nombre, d.nombre AS "Departamento"

FROM empleados e

JOIN departamentos d ON e.departamento\_id = d.id;

El INNER JOIN ya filtra automáticamente las filas donde haya coincidencia

/////////////////////////////////////////////////////////////////////////////////////////////////////

**¿Qué es una subconsulta?**

Una subconsulta es una sentencia SELECT que se encuentra dentro de otra sentencia SQL (como SELECT, INSERT, UPDATE, o DELETE). Se utiliza para devolver un valor o conjunto de valores que luego serán usados en la consulta principal.

**Cómo funcionan las subconsultas**

1. **Ejemplo conceptual**  
   Si queremos encontrar a los empleados que ganan más que Jones:
   * Primero, necesitamos saber cuánto gana Jones.
   * Luego, con ese dato, encontramos a los empleados que superan ese salario.

Esto equivale a ejecutar dos consultas:

* + Localizar el salario de Jones:

sql

Copiar código

SELECT sal FROM emp WHERE ename = 'Jones';

* + Usar ese salario en otra consulta para comparar:

sql

Copiar código

SELECT ename FROM emp WHERE sal > (resultado anterior);

La subconsulta permite combinarlas:

sql

Copiar código

SELECT ename

FROM emp

WHERE sal > (SELECT sal FROM emp WHERE ename = 'Jones');

**Sintaxis básica**

sql

Copiar código

SELECT columna1, ..., columnax

FROM tabla

WHERE expresión operador (SELECT columna1, ..., columnaz

FROM tabla);

* **Explicación**:
  1. La subconsulta (entre paréntesis) se ejecuta primero.
  2. El resultado de la subconsulta se usa en la condición de la consulta principal.
  3. El operador puede ser =, >, <, IN, ANY, o ALL, dependiendo del contexto.

**Reglas importantes para las subconsultas**

1. **Encerrar en paréntesis**  
   Siempre debes usar paréntesis para delimitar la subconsulta.
2. **No usar ORDER BY en subconsultas**  
   La cláusula ORDER BY sólo puede aparecer en la consulta principal.
3. **Tipos de operadores según la subconsulta**:
   * **Operadores mono-registro**: Se usan cuando la subconsulta devuelve un solo valor. Ejemplos: =, >, <, <=, <>.
   * **Operadores multi-registro**: Se usan cuando la subconsulta devuelve múltiples valores. Ejemplos: IN, ANY, ALL.

**Ejemplos prácticos**

**Mono-registro**

1. Buscar el nombre del empleado con el salario mínimo:

sql

Copiar código

SELECT ename

FROM emp

WHERE sal = (SELECT MIN(sal) FROM emp);

1. Empleados que tienen el mismo puesto que el empleado con código 7369:

sql

Copiar código

SELECT ename, job

FROM emp

WHERE job = (SELECT job FROM emp WHERE empno = 7369);

**Multi-registro**

1. Empleados cuyo salario sea uno de los mínimos por departamento:

sql

Copiar código

SELECT ename, sal, deptno

FROM emp

WHERE sal IN (SELECT MIN(sal) FROM emp GROUP BY deptno);

1. Empleados que ganan menos que cualquiera de los empleados con puesto "CLERK":

sql

Copiar código

SELECT ename, sal

FROM emp

WHERE sal < ANY (SELECT sal FROM emp WHERE job = 'CLERK');

1. Empleados cuyo salario supera el salario promedio de todos los departamentos:

sql

Copiar código

SELECT ename, sal

FROM emp

WHERE sal > ALL (SELECT AVG(sal) FROM emp GROUP BY deptno);

**Errores comunes con subconsultas**

1. **Subconsulta devuelve más de una fila con un operador de comparación único (=):**

sql

Copiar código

SELECT empno, ename

FROM emp

WHERE sal = (SELECT MIN(sal) FROM emp GROUP BY deptno);

**Error**: La subconsulta devuelve varios valores (mínimos por departamento).  
**Corrección**: Cambiar = por IN:

sql

Copiar código

SELECT empno, ename

FROM emp

WHERE sal IN (SELECT MIN(sal) FROM emp GROUP BY deptno);

1. **NULL inesperado en subconsulta:** Si la subconsulta devuelve NULL (por ejemplo, buscar algo que no existe), la consulta principal puede no devolver filas.

**Notas importantes**

1. **Operadores multiregistro:**
   * IN: Devuelve filas que coinciden con cualquier valor de la lista de la subconsulta.
   * ANY: Compara con cada valor de la subconsulta y evalúa si cumple la condición con al menos uno.
   * ALL: Compara con cada valor de la subconsulta y evalúa si cumple la condición con todos.
2. **Cláusulas que aceptan subconsultas:**
   * WHERE: Condiciones simples o complejas.
   * HAVING: Condiciones sobre datos agrupados.
   * FROM: Permite usar subconsultas como tablas temporales.