*En SQL, una consulta es una solicitud que se hace a una base de datos para recuperar información específica. Las consultas son una parte fundamental de SQL, ya que permiten al usuario interactuar con los datos almacenados en una base de datos.*

*En general, una consulta se compone de una serie de cláusulas que se combinan para especificar exactamente qué datos se quieren recuperar. Las cláusulas más comunes incluyen:*

* *SELECT: especifica qué columnas de la tabla o tablas se quieren recuperar.*
* *FROM: especifica la tabla o tablas de donde se deben recuperar los datos.*
* *WHERE: especifica las condiciones que deben cumplir las filas para ser incluidas en los resultados.*
* *HAVING: especifica las condiciones que deben cumplir los grupos para ser incluidos en los resultados.*
* *ORDER BY: especifica cómo se deben ordenar los resultados.*

*Las consultas pueden ser muy simples o muy complejas, dependiendo de las necesidades del usuario y la estructura de la base de datos. Es necesario conocer entonces la estructura de la base de datos para realizar consultas sobre ella.*

*Por ejemplo, una consulta simple podría ser:*

| *SELECT nombre, apellido, email FROM estudiantes WHERE fecha\_nacimiento >= '2020-01-01';* |
| --- |

*La consulta anterior devuelve el nombre, apellido y fecha de nacimiento de los alumnos que nacieron de 2020 a la fecha.*

**

*A continuación, revisaremos cada una de las cláusulas más comunes que existen en SQL.*

*2.1 SELECT*

*La cláusula SELECT en SQL se utiliza para recuperar datos de una o varias tablas. Esta cláusula es una de las más utilizadas en SQL, ya que es la forma en que se recuperan los datos que se necesitan de una base de datos.*

*La sintaxis básica de la cláusula SELECT es la siguiente:*

| *SELECT column1, column2, ..., columnN FROM table\_name;* |
| --- |

*Donde “column1”, “column2”, ..., “columnN” son los nombres de las columnas que se quieren recuperar y “table\_name” es el nombre de la tabla de la que se quieren recuperar los datos.*

*Por ejemplo, si queremos hacer una consulta con el id y nombre de los cursos en la base de datos escuela, lo realizaremos con el siguiente comando:*

| *SELECT id, nombre FROM cursos;* |
| --- |

*Y nos devuelve una tabla con ambas columnas:*



*La cláusula SELECT también permite utilizar funciones agregadas, como SUM(), COUNT(), AVG(), MIN() y MAX(), para realizar cálculos en los datos que se están recuperando. Por ejemplo, para obtener el número de registros en la tabla de calificaciones, se puede utilizar la siguiente sintaxis:*

| *SELECT COUNT(\*) AS registros FROM calificaciones;* |
| --- |

*Y nos devuelve la siguiente tabla, que nos indica que hay 100 registros dentro de la tabla calificaciones. Debemos notar el uso de la palabra AS cuya función es renombrar la columna COUNT(\*) como registros.*

**

*Además, la cláusula SELECT también permite utilizar la cláusula WHERE para filtrar los datos que se recuperan según ciertos criterios. Por ejemplo, para recuperar sólo los registros donde la calificación sea 10, se puede utilizar la siguiente sintaxis:*

| *SELECT \* FROM calificaciones WHERE calificacion = 10;* |
| --- |

*La consulta nos devuelve entonces la siguiente tabla, que contiene los alumnos y el curso donde obtuvieron 10.*

**

*En resumen, la cláusula SELECT en SQL es esencial para recuperar los datos necesarios de una base de datos. Es una cláusula muy versátil que permite recuperar datos de una o varias tablas, utilizar funciones agregadas y filtrar los datos según criterios específicos.*

*2.2 WHERE*

*La cláusula WHERE en SQL se utiliza para filtrar registros en una consulta. Permite especificar una o varias condiciones que deben cumplirse para que un registro sea incluido en los resultados de la consulta.*

*La sintaxis básica de la cláusula WHERE es la siguiente:*

| *SELECT columnas FROM tabla WHERE condición;* |
| --- |

*La condición puede incluir comparaciones de valores, operadores lógicos y funciones. Por ejemplo, si se desea obtener las calificaciones del alumno Juan Pérez, cuyo ID es 1, se puede utilizar la siguiente consulta:*

| *SELECT \* FROM estudiantes WHERE estudiante\_id = 1;* |
| --- |

*En esta consulta, la condición especificada en la cláusula WHERE es "estudiante\_id = 1", lo que significa que se seleccionarán todos los registros donde el valor de la columna "estudiante\_id" sea igual a 1.*

*También es posible combinar varias condiciones utilizando operadores lógicos como AND y OR. Por ejemplo, si se desea obtener los registros de la tabla de alumnos donde el curso sea "Matemáticas" y la calificación obtenida sea menor que 6, se puede utilizar la siguiente consulta:*

| *SELECT \* FROM calificaciones WHERE curso\_id = 1 AND calificacion < 6;* |
| --- |

*En esta consulta, se utilizan dos condiciones separadas por el operador lógico AND. La primera condición especifica que el curso debe ser "Matemáticas" (cuyo id es 1), y la segunda condición específica que la calificación obtenida debe ser menor que 6. La consulta da como resultado la siguiente tabla, cuyo registro corresponde al alumno que obtuvo una calificación menor a 6 en matemáticas.*

**

*La cláusula WHERE es una de las cláusulas más importantes en SQL, ya que permite seleccionar los datos específicos que se desean consultar en una tabla. Es importante comprender cómo funciona esta cláusula y cómo se pueden combinar varias condiciones para obtener los resultados deseados.*

*2.3 ORDER BY*

*La cláusula ORDER BY se utiliza en SQL para ordenar los resultados de una consulta por uno o más campos en orden ascendente o descendente. Por defecto, los resultados se ordenan en orden ascendente (de menor a mayor) según el campo especificado.*

*La sintaxis básica de la cláusula ORDER BY es la siguiente:*

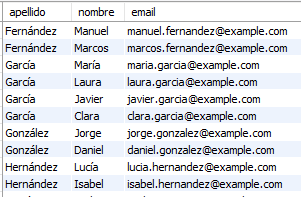
| *SELECT columna1, columna2, ..., columnaN FROM tabla ORDER BY columna1 [ASC|DESC], columna2 [ASC|DESC]* |
| --- |

*En esta sintaxis, se especifican las columnas que se desean seleccionar en la consulta. A continuación, se utiliza la cláusula ORDER BY para ordenar los resultados de la consulta. Se especifica una o varias columnas por las que se desea ordenar los resultados y se utiliza ASC o DESC para especificar el orden ascendente o descendente, respectivamente.*

*Por ejemplo, para obtener una lista de alumnos ordenados por su apellido en orden alfabético, la consulta SQL podría ser:*

| *SELECT apellido, nombre, email FROM estudiantes ORDER BY apellido ASC;* |
| --- |

*En este ejemplo, la cláusula ORDER BY se utiliza para ordenar los resultados de la consulta por el campo "apellido" de la tabla "estudiantes" en orden ascendente. Obtenemos como resultado la siguiente tabla:*

**

*2.4 LIMIT*

*La cláusula LIMIT es una cláusula opcional que se utiliza en SQL para limitar el número de filas devueltas por una consulta. Se utiliza en combinación con la cláusula SELECT para especificar el número máximo de filas que se deben devolver.*

*La sintaxis de la cláusula LIMIT es la siguiente:*

| *SELECT column1, column2, ... FROM table\_name LIMIT number\_of\_rows;* |
| --- |

*Donde “number\_of\_rows” es el número máximo de filas que se deben devolver. Por ejemplo, si se establece en 5, la consulta devolverá las primeras 5 filas que cumplan con la condición de la cláusula WHERE, si existe.*

*Es importante tener en cuenta que el uso de la cláusula LIMIT puede variar según el tipo de base de datos utilizado. Por ejemplo, en MySQL, la cláusula LIMIT se puede combinar con OFFSET para especificar un desplazamiento en el número de filas devueltas, mientras que en Oracle, se utiliza la cláusula ROWNUM en lugar de LIMIT.*

*La cláusula LIMIT es útil cuando se trabaja con grandes conjuntos de datos y se desea limitar el número de filas devueltas para mejorar el rendimiento de la consulta. También puede ser útil para paginar los resultados de la consulta, es decir, para mostrar una cantidad limitada de resultados por página en una aplicación web, por ejemplo.*

*2.5 Funciones de agregación*

*Las funciones de agregación son un conjunto de funciones en SQL que se utilizan para calcular valores resumidos a partir de los datos en una tabla. Las funciones de agregación más comunes son:*

1. *SUM: Calcula la suma de los valores en una columna numérica.*
2. *AVG: Calcula el promedio de los valores en una columna numérica.*
3. *COUNT: Cuenta el número de filas en una tabla o el número de valores distintos en una columna.*
4. *MAX: Encuentra el valor máximo en una columna.*
5. *MIN: Encuentra el valor mínimo en una columna.*

*Estas funciones de agregación son muy útiles para obtener información resumida de grandes cantidades de datos en una tabla y tomar decisiones basadas en ella.*

*Por ejemplo, supongamos que deseamos obtener la calificación promedio de los alumnos, la calificación máxima y la calificación mínima. La sintaxis para obtener estos valores es la siguiente:*

| *SELECT AVG(calificacion) AS cal\_avg,   MAX(calificacion) AS cal\_max,   MIN(calificacion) AS cal\_min FROM calificaciones;* |
| --- |

*El resultado obtenido es la siguiente tabla, con la calificación promedio, la calificación máxima y la calificación mínima:*

**

*Para finalizar el tema, usaremos las cláusulas vistas para construir una consulta más elaborada. Supongamos que deseamos conocer a los dos alumnos con mayor promedio en la escuela, pero sin considerar el curso de historia cuyas calificaciones no son definitivas. La sintaxis para realizar esta consulta es la siguiente:*

| *SELECT estudiante\_id, AVG(calificacion) AS promedio FROM calificaciones WHERE curso\_id != 2 GROUP BY estudiante\_id ORDER BY promedio DESC LIMIT 2;* |
| --- |

*La consulta generada es la siguiente:*

**

*En la tabla se muestra el id de los dos alumnos con mejor promedio considerando cuatro de los cinco cursos:*

*Los comandos utilizados en esta consulta son:*

* *SELECT: especifica las columnas que queremos que aparezcan en los resultados*
* *WHERE: filtra los registros de la tabla según una condición (en este caso, que el id sea diferente de 2, que corresponde al curso de historia)*
* *GROUP BY: agrupa los registros según el valor de una columna (en este caso, por el id del estudiante)*
* *ORDER BY: ordena los resultados según el valor de una columna (en este caso, por la calificación promedio del alumno)*
* *LIMIT: limita el número de resultados devueltos (en este caso, a los dos primeros estudiantes con mayor calificación promedio)*
* *AVG: función de agregación que obtiene el promedio de los valores de una columna en un grupo.*

*Para saber el nombre de los estudiantes nos es suficiente con conocer el id para poder hallar los nombres en la tabla “estudiantes”. En la siguiente sección veremos cómo hacer estos cruces entre tablas utilizando las cláusulas JOIN.*