**Clase 11:** Introducción a las bases de datos – parte 2

Objetivos de la clase

* Aprender los fundamentos del lenguaje SQL (Structured Query Language).
* Familiarizarse con las operaciones básicas de consulta, inserción, actualización y eliminación de datos en una base de datos relacional.
* Comprender cómo utilizar SQL para interactuar con bases de datos relacionales.

**Preguntas orientadoras para la lectura**

* *¿Qué es SQL y para qué se utiliza en la gestión de bases de datos relacionales?*
* *¿Cómo podemos armar una consulta a una tabla en SQL ? es posible combinar distintas tablas?*
* *¿Cómo puedo filtrar, incorporar diferentes criterios y ordenar la información?*
* *¿Puedo realizar consultas avanzadas? ¿Qué ventajas tiene?*

**Introducción a SQL**

SQL (Structured Query Language) es un lenguaje de programación diseñado para gestionar y manipular bases de datos relacionales.

Permite a los usuarios consultar, actualizar y gestionar los datos almacenados en un sistema de gestión de bases de datos (DBMS).

*Características*

* **Acceso a datos:** SQL proporciona un medio estandarizado para acceder a datos almacenados en bases de datos relacionales.
* **Manipulación de datos:** Permite modificar, eliminar y añadir datos en una base de datos.
* **Integridad de datos:** Facilita la implementación de restricciones para mantener la precisión y validez de los datos.

*Breve reseña sobre SQL*

* **Orígenes:** SQL fue desarrollado en la década de 1970 por IBM como parte de su sistema de gestión de bases de datos relacionales, System R.
* **Estándares:** La primera versión estándar de SQL fue publicada por ANSI en 1986. Desde entonces, ha habido varias actualizaciones (SQL-89, SQL-92, SQL:2003, SQL:2011, etc.) que han introducido nuevas características y mejoras.

**Operaciones básicas de SQL**

*Consultas SELECT*

Estructura: **SELECT** columna1, columna2 **FROM** tabla;

Ejemplo: **SELECT** nombre, edad **FROM** empleados;

Explicación:

**SELECT**: Esta palabra clave se utiliza para especificar las columnas que queremos recuperar de la base de datos.

**nombre**: Especifica que queremos recuperar los datos de la columna nombre.

**edad**: Especifica que también queremos recuperar los datos de la columna edad.

**FROM**: Esta palabra clave indica la tabla de la que queremos recuperar los datos.

**empleados**: Es el nombre de la tabla de la que estamos extrayendo los datos

La consulta:.

* **Recupera datos específicos**: Obtiene únicamente las columnas nombre y edad de cada fila en la tabla empleados.
* **Muestra todos los registros**: Por defecto, si no se especifica una condición, se devuelven todas las filas presentes en la tabla empleados.

Veamos la tabla **empleados:**

|  |  |  |  |
| --- | --- | --- | --- |
| id | nombre | edad | Puesto |
| 1 | Ana | 28 | Desarrolladora |
| 2 | Luis | 34 | Diseñador |
| 3 | Marta | 45 | Gerente |

El resulta de la consulta es:

|  |  |
| --- | --- |
| **nombre** | **edad** |
| Ana | 28 |
| Luis | 34 |
| Marta | 45 |

*Filtrado de datos con la cláusula WHERE*

Estructura: SELECT columna1, columna2 FROM tabla WHERE condicion;

Ejemplo: SELECT nombre, edad FROM empleados WHERE edad > 30;

Explicación:

La consulta realiza las siguientes acciones:

1. **Filtra los Datos**: Solo selecciona las filas donde el valor de edad es mayor que 30.
2. **Recupera Columnas Específicas**: Para las filas que cumplen con la condición, devuelve los valores de las columnas nombre y edad.

El resultado de la consulta es:

|  |  |
| --- | --- |
| **nombre** | **edad** |
| Luis | 34 |
| Marta | 45 |

*Ordenamiento de resultados con ORDER BY*

Sintaxis: SELECT columna1, columna2 FROM tabla ORDER BY columna1 [ASC|DESC];

Ejemplo: SELECT nombre, edad FROM empleados ORDER BY edad DESC;

El resultado de la consulta es:

|  |  |
| --- | --- |
| **nombre** | **edad** |
| Marta | 45 |
| Luis | 34 |
| Ana | 28 |

**Operaciones de inserción, actualización y eliminación**

Sintaxis: INSERT INTO tabla (columna1, columna2) VALUES (valor1, valor2);

Ejemplo: INSERT INTO empleados (nombre, edad) VALUES ('Ana', 28);

*Actualización de datos*

Sintaxis: UPDATE tabla SET columna1 = valor1 WHERE condicion;

Ejemplo: UPDATE empleados SET edad = 29 WHERE nombre = 'Ana';

*Eliminación de datos*

Sintaxis: DELETE FROM tabla WHERE condicion;

Ejemplo: DELETE FROM empleados WHERE nombre = 'Ana';

**Joins y Subconsultas**

*Tipos de Joins*

**INNER JOIN:** Devuelve solo las filas que tienen coincidencias en ambas tablas.

Sintaxis: SELECT tabla1.columna1, tabla2.columna2 FROM tabla1 INNER JOIN tabla2

ON tabla1.columna\_clave = tabla2.columna\_clave;

**LEFT JOIN (o LEFT OUTER JOIN):**Devuelve todas las filas de la tabla de la izquierda y las filas coincidentes de la tabla de la derecha.

Sintaxis: SELECT tabla1.columna1, tabla2.columna2 FROM tabla1 LEFT JOIN tabla2

ON tabla1.columna\_clave = tabla2.columna\_clave;

**RIGHT JOIN (o RIGHT OUTER JOIN):** Devuelve todas las filas de la tabla de la derecha y las filas coincidentes de la tabla de la izquierda.

Sintaxis: SELECT tabla1.columna1, tabla2.columna2 FROM tabla1 RIGHT JOIN tabla2

ON tabla1.columna\_clave = tabla2.columna\_clave;

**FULL JOIN (o FULL OUTER JOIN):** Devuelve todas las filas cuando hay una coincidencia en una de las tablas.

Sintaxis: SELECT tabla1.columna1, tabla2.columna2 FROM tabla1 FULL JOIN tabla2

ON tabla1.columna\_clave = tabla2.columna\_clave;

*Subconsultas*

Ejemplo 1: SELECT nombre FROM empleados WHERE edad > (SELECT AVG(edad) FROM empleados);

Ejemplo 2: SELECT sub.nombre, sub.edad FROM (SELECT nombre, edad FROM empleados WHERE edad > 30) AS sub;