Trabajo 4 de Base De Datos 1

Gustavo Ángel Ortega Pérez

Hennys Sebastian Tous De Avila

Universidad de Cartagena

**Texto, Carta

Descripción generada automáticamente**

Programa: Ingeniería de Software

Asignatura: Estructura de Datos

Tutor: Jhon Arrieta Arrieta

Cartagena de Indias – agosto 4 del año 2022

**Tabla de Contenido**

1. Introducción.
2. Objetivo.
3. Justificación.
4. Resultados
5. Síntesis y argumentos individuales.
6. Bibliografía

**Introducción**

En el presente documento profundizaremos en las sentencias DML, específicamente las de tipo Select, las cuales nos permiten realizar la consulta de los registros almacenados en nuestras tablas. A su vez exploraremos los extensos usos de la sentencia JOIN y su importante papel dentro de las relaciones entre tablas.

**Objetivos**

* Manejar de manera adecuada las sentencias DML de tipo Select al momento de realizar consultas.
* Conocer los conceptos básicos de IN y Distinct, y como estos afectan nuestras consultas a registros.
* Reconocer los usos de la sentencia JOIN y como esta puede consultar las relaciones entre tablas, y sus respectivos registros.

**Justificación**

Principalmente hemos decidido tomar riendas de este documento para ahondar nuestros conocimientos dentro del uso de Gestores para bases de Datos, teniendo en cuenta sentencias DML.

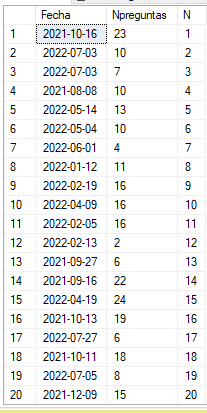
Esto con la finalidad de aprender a cómo manejar registros mediante sentencias DML de tipo Insert, Update y Delete,

**Resultados**

Tomar alguna de las tablas (TablaX, TablaY, TablaZ) de la base de datos bd\_grupo\_xyz y realizar las siguientes operaciones.

1. Mostrar **todos los registros** de la TablaX:

SELECT \* FROM Examen

Imagen que contiene Calendario

Descripción generada automáticamente

Usamos SELECT \* para mostrar todos los registros de la tabla Examen.

1. **Mostrar el valor solo de 2 columnas** de todos los registros de la TablaX

SELECT Fecha, Npreguntas FROM Examen

Imagen que contiene Tabla

Descripción generada automáticamente

Usamos SELECT y especificamos las columnas que queremos mostrar.

1. Mostrar el valor de 2 columnas de todos los registros de la TablaX, pero en el resultado de la consulta el **nombre de esas dos columnas debe ser** Columna1, Column2 respectivamente:

SELECT Fecha as Columna1, Npreguntas as Columna2 FROM Examen

Imagen que contiene Tabla

Descripción generada automáticamente

Usamos SELECT, especificamos la tabla que queremos seleccionar y la nombramos como queramos usando AS.

1. Mostrar el **número de registros** almacenados en la TablaX:

SELECT COUNT(\*) FROM Examen



Usamos SELECT seguido de COUNT (conteo) para que nos arroje el numero de registros de la tabla.

1. Mostrar todos los registros de la TablaX **ordenados descendentemente**:

SELECT \* FROM Examen ORDER BY Npreguntas ASC

Imagen que contiene Gráfico

Descripción generada automáticamente

Usamos SELECT \* para que nos de todos los registros de la tabla Examen y usamos ORDER BY seguido de la tabla que queremos usar como factor para ordenar los registros (en este caso Npreguntas), y por ultimo usamos ASC para especificar que queremos ordenarlas ascendentemente.

1. Mostrar el **número de registros** almacenados en la TablaX, pero el valor del resultado debe estar almacenado en una columna llamada EL\_CONTEO.

SELECT COUNT(\*) as EL\_CONTEO FROM Examen

Imagen que contiene Forma

Descripción generada automáticamente

Usamos SELECT seguido de COUNT (conteo) para que nos arroje el numero de registros de la tabla y usamos AS para nombrarla como queramos.

1. Realizar una consulta en la TabalX o la Tabla Y o la TablaZ en la que se pueda obtener una **sumatoria** de valores de alguna columna numérica, pero el valor del resultado debe estar almacenado en una columna llamada LA\_SUMATORIA.

SELECT SUM(Npreguntas) as LA\_SUMA FROM Examen

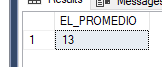
Imagen que contiene Calendario

Descripción generada automáticamente

Usamos SELECT seguido de SUM(nombre\_columna) para que nos arroje la sumatoria de todos los datos de dicha columna.

1. Realizar una consulta en la TablaX o la TablaY o la TablaZ en la que se pueda obtener el **promedio** de valores de alguna columna numérica, pero el valor del resultado debe estar almacenado en una columna llamada El\_PROMEDIO.

SELECT AVG(Npreguntas) as EL\_PROMEDIO FROM Examen



Usamos SELECT seguido de AVG (nombre\_columna) para que nos arroje el promedio de todos los datos de dicha columna.

1. Realizar una consulta en la TablaX o la TablaY o la TablaZ en la que se pueda obtener el **el valor máximo** de alguna columna numérica, pero el valor del resultado debe estar almacenado en una columna llamada El\_MAXIMO.

SELECT MAX(Npreguntas) as EL\_MAXIMO FROM Examen

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

Usamos SELECT seguido de MAX (nombre\_columna) para que nos arroje el máximo dato numérico de la columna.

1. Realizar una consulta en la TablaX o la TablaY o la TablaZ en la que se pueda obtener el **el valor mínimo** de alguna columna numérica, pero el valor del resultado debe estar almacenado en una columna llamada El\_MINIMO.

SELECT MIN(Npreguntas) as EL\_MINIMO FROM Examen

Imagen que contiene Forma

Descripción generada automáticamente

Usamos SELECT seguido de MIN (nombre\_columna) para que nos arroje el mínimo dato numérico de la columna.

1. Realizar una consulta en la TablaX o la TablaY o la TablaZ que permita mostrar los valores de dos columnas **concatenados** en una sola columna, pero el valor del resultado debe estar almacenado en una columna llamada VALORES\_UNIDOS.

SELECT CONCAT(Fecha,' || ', Npreguntas) as VALORES\_UNIDOS FROM Examen

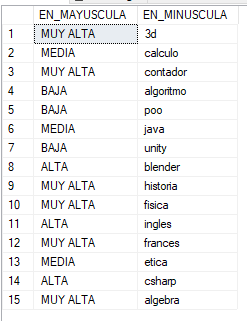
Tabla

Descripción generada automáticamente con confianza media

Usamos SELECT seguido de CONCAT(nombre\_columna1,nombre\_columna2) para que nos arroje la concatenación de ambas columnas y usamos AS para nombrar la consulta.

1. Realizar una consulta en la TablaX o la TablaY o la TablaZ que permita mostrar el **MAYÚSCULAS** los valores de una columna, y en **Minúsculas** los valores de otra columna, pero en el resultado estas columnas deben llamarse EN\_MAYUSCULA, EN\_MINUSCULA respectivamente.

SELECT UPPER(Dificultad) as EN\_MAYUSCULA, LOWER(Titulo) as EN\_MINUSCULA FROM Practica



Usamos SELECT seguido de UPPER(nombre\_columna) para cambiar los datos de dicha columna a mayúsculas, LOWER(nombre\_columna) para cambiar los datos de dicha columna a minúsculas y usamos AS para nombrar las consultas.

1. Realizar una consulta en la TablaX o la TablaY o la TablaZ que permita mostrar sólo **el AÑO** de alguna columna Tipo Fecha, pero en el resultado esta columna debe llamarse EL\_AÑO.

SELECT YEAR(Fecha) as EL\_AÑO From Examen

Tabla

Descripción generada automáticamente

Usamos SELECT seguido de YEAR(nombre\_columna) para obtener solo ñps años de la columna que contenga fechas y usamos AS para nombrar la consulta.

1. Realizar una consulta en la TablaX o la TablaY o la TablaZ que permita mostrar sólo **el MES** de alguna columna Tipo Fecha, pero en el resultado esta columna debe llamarse EL\_MES:

SELECT MONTH(Fecha) as EL\_MES From Examen

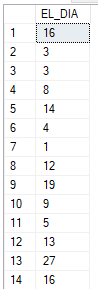
Imagen que contiene Tabla

Descripción generada automáticamente

Usamos SELECT seguido de MONTH(nombre\_columna) para obtener solo los meses de la columna que contenga fechas y usamos AS para nombrar la consulta.

1. Realizar una consulta en la TablaX o la TablaY o la TablaZ que permita mostrar sólo **el DIA** de alguna columna Tipo Fecha, pero en el resultado esta columna debe llamarse EL\_DIA..

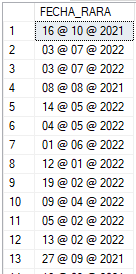
SELECT DAY(Fecha) as EL\_DIA From Examen



Usamos SELECT seguido de DAY(nombre\_columna) para obtener solo los dias de la columna que contenga fechas y usamos AS para nombrar la consulta.

1. Realizar una consulta en la TablaX o la TablaY o la TablaZ que permita mostrar sólo **la fecha en formato personalizado,** así **(dia@mes@año)** de alguna columna Tipo Fecha, pero en el resultado esta columna debe llamarse FECHA\_RARA.:

SELECT FORMAT(Fecha,'dd @ MM @ yyyy') AS FECHA\_RARA From Examen



Usamos SELECT seguido de FORMAT(nombre\_columna,’formato\_fecha’) para mostrar las fechas almacenadas en una columna usando un formato personalizado y usamos AS para nombrar la consulta.

1. Realizar una consulta en la TablaX o la TablaY o la TablaZ que permita mostrar sólo **la FECHA ACTUAL** en una nueva columna**,** pero en el resultado esta columna debe llamarse FECHA\_ACTUAL.

SELECT GETDATE() AS FECHA\_ACTUAL;



Usamos SELECT seguido de la función GETDATE() para obtener la fecha actual del sistema y usamos AS para nombrar la consulta.

1. Realizar una consulta en la TablaX o la TablaY o la TablaZ que permita mostrar en 3 nuevas columnas el resultado de **SUMAR**, **RESTAR** Y **MULTIPLICAR** los valores de 2 columnas Tipo numérico, pero en el resultado esta columna debe llamarse LA\_SUMA, LA\_RESTA, LA\_MULTIPLICACION respectivamente.

SELECT (Npreguntas + N) as LA\_SUMA, (Npreguntas - N) as LA\_RESTA, (Npreguntas \* N) as LA\_MULTIPLICACION from Examen

Tabla

Descripción generada automáticamente

Usamos SELECT seguido de (columna1 + columna2) para sumar los valores de ambas columnas, (columna1 – columna2) para restar los valores de ambas columnas, (columna1 \* columna2) para multiplicar los valores de ambas columnas y usamos AS para nombrar las consultas.

1. Investigar cual es el objetivo de la Palabra DISTINCT de SQL

SELECT DISTINCT nos permite consultar solo datos que sean diferentes.

Dentro de una tabla, normalmente una tabla puede llegar a tener valores repetidos y esta función permite apartar dichos valores repetidos.

Ejemplo:

* SELECT DISTINCT columna1, columna2, ...

FROM nombre\_tabla;

1. Seleccionar alguna de las tablas TablaX o la TablaY o la TablaZ y realizar una consulta donde se pueda apreciar el uso de la palabra **DISTINCT**

SELECT DISTINCT Npreguntas From Examen

Imagen que contiene Tabla

Descripción generada automáticamente

Usamos SELECT DISTINCT nombre\_columna para que nos arroje datos únicos de una columna.

1. Realizar una consulta en la TablaX que permita mostrar los datos de un registro cuya columna llave primaria sea **igual** al valorY

SELECT \* FROM Examen Where N = 15



Usamos SELECT \* y Where para seleccionar todos los registros de la tabla donde N sea 15.

1. Realizar una consulta en la TablaX que permita mostrar los registros con que cumplan con la conducción columnaY **igual** datoZ

SELECT \* FROM Examen Where Npreguntas = 10

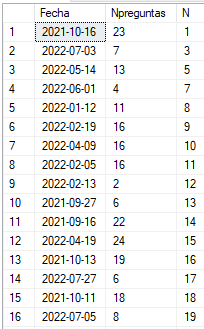
Tabla

Descripción generada automáticamente

Usamos SELECT \* y Where para seleccionar todos los registros de la tabla donde Npreguntas sea 10.

1. Realizar una consulta en la TablaX que permita mostrar los registros con que cumplan con la conducción columnaY **diferente**  datoZ

SELECT \* FROM Examen Where Npreguntas != 10

 Imagen que contiene Tabla

Descripción generada automáticamente

Usamos SELECT \* y Where para seleccionar todos los registros de la tabla donde Npreguntas sea diferente de 10.

1. Realizar una consulta en la TablaX que permita mostrar los registros con que cumplan con la conducción columnaY **menor**  datoZ

SELECT \* FROM Examen Where Npreguntas < 10

Imagen que contiene Tabla

Descripción generada automáticamente

Usamos SELECT \* y Where para seleccionar todos los registros de la tabla donde Npreguntas sea menor a 10.

1. Realizar una consulta en la TablaX que permita mostrar los registros con que cumplan con la conducción columnaY **mayor** datoZ

SELECT \* FROM Examen Where Npreguntas > 10

Tabla

Descripción generada automáticamente con confianza baja

Usamos SELECT \* y Where para seleccionar todos los registros de la tabla donde Npreguntas sea mayor a 10.

1. Realizar una consulta en la TablaX que permita mostrar los registros con que cumplan con la conducción columnaY **entre valor** datoX y valor

SELECT \* FROM Examen Where Npreguntas BETWEEN 10 and 20

Imagen que contiene Tabla

Descripción generada automáticamente

Usamos SELECT \*, Where y Between para seleccionar todos los registros de la tabla donde el valor de Npreguntas este entre 10 y 20.

1. Realizar una consulta en la TablaX que permita mostrar los registros con que cumplan con la conducción columnaY **Sea NULO**

SELECT \* FROM Alumno WHERE Grupo IS NULL

Tabla

Descripción generada automáticamente

Usamos SELECT \*, Where e IS NULL para seleccionar todos los registros de la tabla donde Grupo sea NULL.

1. Realizar una consulta en la TablaX que permita mostrar los registros con que cumplan con la conducción columnaY **No Sea NULO**

SELECT \* FROM Alumno WHERE Grupo IS NOT NULL

Calendario

Descripción generada automáticamente con confianza baja

Usamos SELECT \*,Where e IS NOT NULL para seleccionar todos los registros de la tabla donde Grupo no sea sea NULL.

1. Realizar una consulta en la TablaX que permita mostrar los registros con que cumplan con la conducción columnaY **Contenga el valor** DatoX en su interior

SELECT \* FROM Examen WHERE Fecha like '%3%'

Interfaz de usuario gráfica, Tabla

Descripción generada automáticamente

Usamos SELECT \*, Where y Like ‘%dato%’ para seleccionar todos los registros de la tabla en las cuales este el valor 3 (sin importar su posición) en cualquier Fecha.

1. Realizar una consulta en la TablaX que permita mostrar los registros con que cumplan con la conducción columnaY **Inicie con el valor** DatoX

SELECT \* FROM Alumno WHERE Nombre like 'a%'

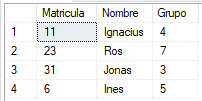
Tabla

Descripción generada automáticamente

Usamos SELECT \*, Where y Like ‘dato%’ para seleccionar todos los registros de la tabla en las cuales este el valor a al inicio de cualquier Nombre.

1. Realizar una consulta en la TablaX que permita mostrar los registros con que cumplan con la conducción columnaY **Termine con el valor** DatoX

SELECT \* FROM Alumno WHERE Nombre like '%s'



Usamos SELECT \*, Where y Like ‘%dato’ para seleccionar todos los registros de la tabla en las cuales este el valor s al final de cualquier Nombre.

1. Investigar cual es el objetivo del comando IN.

El operador IN nos permite especificar múltiples valores para una clausula WHERE. Podría decirse que IN es la forma acortada de una cadena de OR’s

1. Realizar una consulta en la TablaX que permita mostrar el uso y aplicación de la palabra **IN**

SELECT \* FROM Alumno WHERE Grupo IN(2,5)

Imagen que contiene Gráfico

Descripción generada automáticamente

Usamos SELECT \*, Where e IN para arrojar todos los registros donde su grupo sea 2 o 5.

Para realizar las siguientes consultas se asume que TablaA, TablaB, TablaC, …, TablaZ, representan relacionadas entre sí, mediante llaves foráneas, que se describen en El Modelo Relacional del ejercicio que han seleccionado.

Asumimos que TablaA y TablaB están relacionados así:

[**TablaA**] ->----<**R**>----|- [**TablaB**] ->----<**R**>----|- [**TablaC**]

Es decir, [**TablaA**] tiene una llave foránea llamada por ejemplo **tablaB\_id**, la cual se conecta con la llave primaria **id** de la [**TablaB**], mientras que [**TablaB**] tiene una llave foránea llamada por ejemplo **tablaC\_id** la cual se conecta con la llave primaria **id** de la [**TablaC**],

1. Realizar una consulta que permitan mostrar los registros de la **TablaA que están relacionados** con la **TablaB**

SELECT \* From Alumno

JOIN AlumnoExamen

ON Alumno.Matricula = AlumnoExamen.Matricula

WHERE Alumno.Nombre IS NOT NULL;



Usamos SELECT \* From Alumno para seleccionar todos los registros de la tabla, luego usamos JOIN AlumnoExamen ON Alumno.Matricula = AlumnoExamen.Matricula para relacionar los registros de la primera tabla con la segunda tabla teniendo en cuenta la columna Matricula presente en ambas tablas.

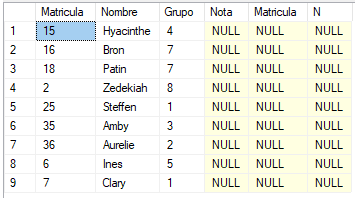
1. Realizar una consulta que permitan mostrar los registros de la **TablaA** que **NO están relacionados** con la **TablaB**

SELECT \* FROM Alumno

FULL JOIN AlumnoExamen

ON Alumno.Matricula = AlumnoExamen.Matricula

WHERE AlumnoExamen.Matricula IS NULL and Alumno.Nombre IS NOT NULL;



Usamos SELECT \* From Alumno para seleccionar todos los registros de la tabla, luego usamos JOIN AlumnoExamen ON Alumno.Matricula = AlumnoExamen.Matricula para relacionar los registros de la primera tabla con la segunda tabla teniendo en cuenta la columna Matricula presente en ambas tablas, seguido de un Where AlumnoExamen.Matricula IS NULL para especificar cuales son las Matriculas de la tabla Alumno que no se encuentran en la tabla AlumnoExamen.

1. Realizar una consulta que permitan mostrar los registros de la **TablaB** que **están relacionados** con la **TablaA**

**INSTRUCCIÓN SQL**:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Recorte de pantalla:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Explicación:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Opcionalmente GUI:**

Recorte de pantalla:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Explicación:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Realizar una consulta que permitan mostrar los registros de la **TablaB** que **NO están relacionados** con la **TablaA**

**INSTRUCCIÓN SQL**:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Recorte de pantalla:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Explicación:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Opcionalmente GUI:**

Recorte de pantalla:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Explicación:

1. Realizar una consulta que permitan mostrar los registros de la [**TablaA] que están relacionados** con la **TablaC** mediante la [**TablaB**]

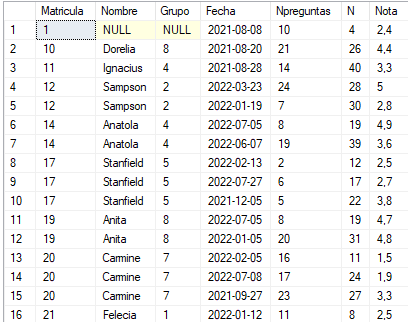
SELECT Alumno.\*, Examen.\*, AlumnoExamen.Nota From Alumno

JOIN AlumnoExamen

ON Alumno.Matricula = AlumnoExamen.Matricula

JOIN Examen

ON Examen.N = AlumnoExamen.N



Usamos SELECT Alumno.\*, Examen.\* FROM Alumno para seleccionar todos los registros de las tablas Alumno y Examen, a su vez, colocamos AlumnoExamen.Nota para obtener solo la columna de notas de la tabla AlumnoExamen.

Luego usamos JOIN AlumnoExamen ON Alumno.Matricula = AlumnoExamen.Matricula para juntar los registros de la tabla Alumno con la tabla AlumnoExamen teniendo en cuenta la columna Matricula presente en ambas tablas.

Luego usamos otro JOIN Examen ON Examen.N = AlumnoExamen.N para juntar los registros de la tabla Examen con la tabla AlumnoExamen teniendo en cuenta la columna N presente en ambas tablas.

1. Supongamos que necesitamos buscar un registro en [**TablaA]** cuya columna **llave primaria** tiene el **valor X,** pero también necesitamos obtener los registros de [**TablaC**] que están relacionados con ese registro de [**TablaA**]. Escriba la consulta SQL necesaria para lograr obtener la información necesaria.

SELECT Alumno.\*, Examen.\* From Alumno

JOIN AlumnoExamen ON Alumno.Matricula = AlumnoExamen.Matricula

JOIN Examen ON Examen.N = AlumnoExamen.N

Where Alumno.Matricula = 17

Tabla

Descripción generada automáticamente

Usamos SELECT Alumno.\*, Examen.\* FROM Alumno para seleccionar todos los registros de las tablas Alumno y Examen, a su vez, colocamos AlumnoExamen.Nota para obtener solo la columna de notas de la tabla AlumnoExamen.

Luego usamos JOIN AlumnoExamen ON Alumno.Matricula = AlumnoExamen.Matricula para juntar los registros de la tabla Alumno con la tabla AlumnoExamen teniendo en cuenta la columna Matricula presente en ambas tablas.

Luego usamos otro JOIN Examen ON Examen.N = AlumnoExamen.N para juntar los registros de la tabla Examen con la tabla AlumnoExamen teniendo en cuenta la columna N presente en ambas tablas.

Por ultimo usamos un WHERE para especificar que solo queremos que nos muestre los registros donde la columna Matricula de la tabla Alumno sea igual a 17.

**Síntesis**

Gestor de Base de Datos

Un Sistema Gestor de Base de Datos (SGBD es un sistema que permite la creación, gestión y administración de bases de datos, así como la selección y gestión de las estructuras necesarias para almacenar y recuperar información de la forma más eficiente posible.

Actualmente podemos organizarlos en dos tipo, a su vez daremos ejemplos:

* Gestores Relacionales: Estos gestores se basan principalmente en los Modelos relacionales, de tal manera que estos establecen relaciones entre sus datos. Entre ellos podemos encontrar:
  + MySQL
  + SQL Server
  + PostgreSQL
  + Oracle
* Gestores no relacionales: Estos gestores no requieren estructurar datos fijamente como en tablas. Son usados principalmente para gestionar grandes cantidades de datos. Entre ellos podemos encontrar:
  + MongoDB
  + Cassandra
  + Redis

Sentencias DML

Las sentencias DML (Data Manipulation Language) permiten a los usuarios la introducción, modificación, consulta o eliminación de registros que se encuentran en una base de datos. Las principales sentencias DML son:

* SELECT: Esta sentencia se utiliza para realizar consultas sobre los registros.
* INSERT: Esta sentencia se utiliza para insertar registros en una base de datos.
* UPDATE: Esta sentencia se utiliza para modificar los valores de uno o varios registros.
* DELETE: Esta sentencia se utiliza para eliminar los registros de una tabla.

Sentencia JOIN

Esta sentencia es usada para combinar dos filas de dos o mas tablas, las cuales posean alguna columna relacionada y común entre ellas. La sentencia Join posee muchas variantes, las cuales son:

* Inner Join: La cual selecciona los registros que tengan datos iguales en ambas tablas.
* Left Join: La cual selecciona todos los registros de la tabla a la izquierda y los registros relacionados de la tabla a la derecha.
* Right Join: La cual selecciona todos los registros de la tabla a la derecha y los registros relacionados de la tabla a la izquierda.
* Full Join: La cual selecciona todos los registros relacionados de ambas tablas.

**Conclusión**

En el presente documento, hemos podido observar las consultas de los registros frente a las sentencias Select ejecutadas para la realización de los ítems. El uso de estas es de vital importancia al momento de mostrar organizadamente los datos de nuestro esquema.

Principalmente podemos destacar el uso de la sentencia Join para realizar consultas a tablas que contengan datos relacionados entre sí.

**Referencias**

W3 Schools. (s. f.). *SQL tutorial*. <https://www.w3schools.com:443/sql/>

I. (2019, 28 junio). *Custom Date/Time Format Strings Supported by FORMAT() in SQL Server*. SQL Server Custom Date Format. https://database.guide/list-of-the-custom-date-time-format-strings-supported-by-the-format-function-in-sql-server/

*PostgreSQL TO\_CHAR Function By Practical Examples*. (2021, 14 marzo). PostgreSQL Tutorial. https://www.postgresqltutorial.com/postgresql-string-functions/postgresql-to\_char/