**GESTIÓN DE FORMACIÓN PROFESIONAL INTEGRAL PROCEDIMIENTO DESARROLLO CURRICULAR**

**Daniel Esteban González Marroquín**

**Servicio Nacional de Aprendizaje**

**Análisis y Desarrollo de Sistemas de Información**

**Bogotá, D.C., mayo 17 de 2022**

Índice general

[**1. ACTIVIDADES DE TRANSFERENCIA DEL CONOCIMIENTO (CONCEPTUALIZACIÓN Y TEORIZACIÓN)** 4](#_Toc96081792)

[**1.1 EXPLIQUE CUÁLES SON LOS TIPOS DE SENTENCIAS SQL (DDL, DML, DCL Y TCL). 4**](#_Toc96081793)

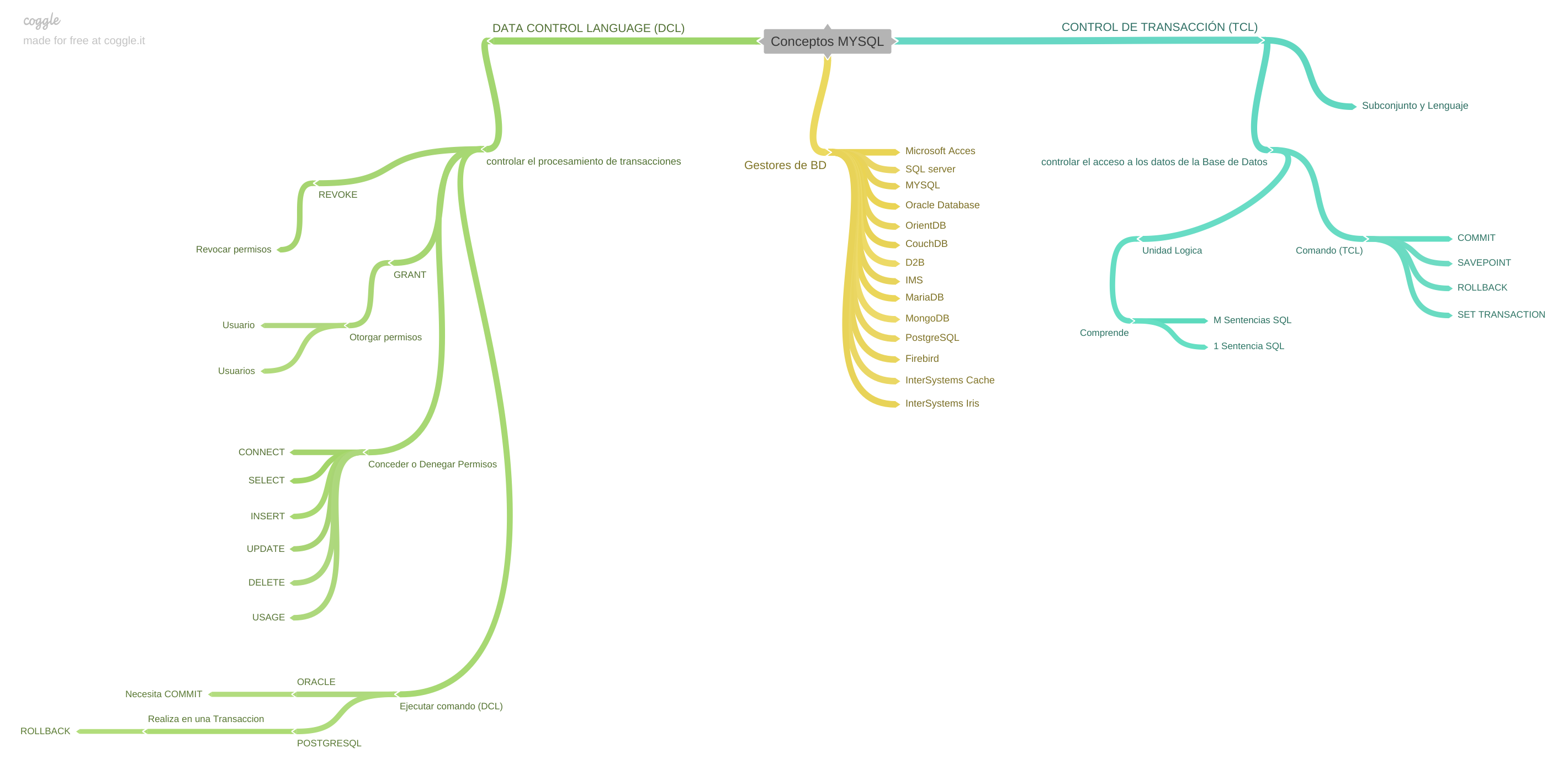
**1**[**.2 LISTE LOS TIPOS DE SMBD CLASIFICADO POR MODELO LÓGICO**. 5](#_Toc96081794)

[**1.3 DILIGENCIE LA SIGUIENTE TABLA Y REALICE UN EJEMPLO EN SQL MYSQL APLICADO A LA BASE DE DATOS “CITAS MÉDICAS”:** 5](#_Toc96081795)

[**Referencias** 6](#_Toc96081796)

# **ACTIVIDADES DE TRANSFERENCIA DEL CONOCIMIENTO (CONCEPTUALIZACIÓN Y TEORIZACIÓN)**

Elaborar un mapa mental que reúna y sintetice los siguientes conceptos:



[**EXPLIQUE CUÁLES SON LOS TIPOS DE SENTENCIAS SQL (DDL, DML, DCL Y TCL)**](#_Toc96081793)

**DLL:**

Se utilizan para definir la estructura de base de datos o esquema.

Es un lenguaje proporcionado por el sistema de gestión de base de datos que permite a los usuarios de la misma llevar a cabo las tareas de definición de las estructuras que almacenarán los datos así como de los procedimientos o funciones que permitan consultarlos.

Un Data Definition Language o Lenguaje de descripción de datos ( DDL ) es un lenguaje de programación para definir estructuras de datos .

El término DDL fue introducido por primera vez en relación con el modelo de base de datos CODASYL, donde el esquema de la base de datos ha sido escrito en un lenguaje de descripción de datos que describe los registros, los campos, y "conjuntos" que conforman el usuario modelo de datos.

Más tarde fue usado para referirse a un subconjunto de SQL, pero ahora se utiliza en un sentido genérico para referirse a cualquier lenguaje formal para describir datos o estructuras de información, como los esquemas XML.

* CREATE - PARA CREAR OBJETOS EN LA BASE DE DATOS
* ALTER - ALTERA LA ESTRUCTURA DE LA BASE DE DATOS
* DROP - ELIMINA LOS OBJETOS DE LA BASE DE DATOS
* TRUNCATE - ELIMINAR TODOS LOS REGISTROS DE UNA TABLA , INCLUYENDO TODOS LOS ESPACIOS ASIGNADOS A LOS REGISTROS SE ELIMINAN
* COMMENT - AGREGAR COMENTARIOS AL DICCIONARIO DE DATOS • RENAME - CAMBIAR EL NOMBRE DE UN OBJETO

LISTE LOS TIPOS DE SMBD CLASIFICADO POR MODELO LÓGICO

**DML:**

Se utilizan para la gestión de datos dentro de los objetos de esquema.

Es un idioma proporcionado por los sistemas gestores de bases de datos que permite a los usuarios de la misma llevar a cabo las tareas de consulta o modificación de los datos contenidos en las Bases de Datos del Sistema Gestor de Bases de Datos. El lenguaje de manipulación de datos más popular hoy día es SQL, usado para recuperar y manipular datos en una base de datos relacional. Otros ejemplos de DML son los usados por bases de datos IMS/DL1, CODASYL u otras.

* SELECT - RECUPERAR DATOS DE LA BASE DE DATOS.
* INSERT - INSERTAR DATOS EN UNA TABLA .
* UPDATE - ACTUALIZACIONES DE DATOS EXISTENTES EN UNA TABLA.
* DELETE - ELIMINA TODOS LOS REGISTROS DE UNA TABLA.
* MERGE - OPERACIÓN UPSERT (INSERCIÓN O ACTUALIZACIÓN).
* CALL - LLAMA A UN PL / SQL O SUBPROGRAMA JAVA.
* EXPLAIN PLAN - EXPLICAR LA RUTA DE ACCESO A LOS DATOS.
* LOCK TABLE - CONCURRENCIAS DE CONTROL.

DCL:

Un Lenguaje de Control de Datos (DCL por sus siglas en inglés: Data Control Language) es un lenguaje proporcionado por el Sistema de Gestión de Base de Datos que incluye una serie de comandos SQL que permiten al administrador controlar el acceso a los datos contenidos en la Base de Datos.

Algunos ejemplos de comandos incluidos en el DCL son los siguientes:

* GRANT: Permite dar permisos a uno o varios usuarios o roles para realizar tareas determinadas.
* REVOKE: Permite eliminar permisos que previamente se han concedido con GRANT.
* Las tareas sobre las que se pueden conceder o denegar permisos son las siguientes:
* CONNECT
* SELECT
* INSERT
* UPDATE
* DELETE
* USAGE

En Oracle, la ejecución de un comando DCL implica un COMMIT de forma implícita. Sin embargo, en PostgreSQL, la ejecución de un comando DCL forma parte de una transacción, por lo que puede ser deshecha mediante el comando ROLLBACK.

TCL:

Es un lenguaje de programación y un subconjunto de SQL , que se utiliza para controlar el procesamiento de transacciones en una base de datos.

Una transacción es una unidad lógica de trabajo que comprende una o más sentencias SQL, por lo general un grupo de Data Manipulation Language (DML).

Ejemplos de comandos de TCL incluyen:

* COMMIT - GUARDA EL TRABAJO REALIZADO.
* SAVEPOINT - IDENTIFICA UN PUNTO EN UNA TRANSACCIÓN A LA QUE MÁS TARDE SEVPUEDE VOLVER.
* ROLLBACK - RESTAURAR LA BASE DE DATOS A LA ORIGINAL, HASTA EL ÚLTIMO COMMIT.
* SET TRANSACTION - CAMBIA LAS OPCIONES DE TRANSACCIÓN COMO NIVEL DE AISLAMIENTO Y QUÉ SEGMENTO DE CANCELACIÓN UTILIZA.

Liste los tipos de SMBD clasificado por modelo lógico.

**MYSQL**:

es un sistema manejador de base de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones

**Características:**

* Soporte a multiplataforma. Procedimientos almacenados. Disparadores.
* Cursores. Vistas actualizables.
* Soporte a VARCHAR. INFORMATION\_SCHEMA Modo Strict Soporte X/Open XA de transacciones distribuidas

**PostgreSQL:**

sistema manejador de base de datos relacional orientada a objetos de software libre, publicado bajo la licencia BSD

**Características:**

* Alta concurrencia. Mediante un sistema denominado MVCC PostgreSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos.
* provee nativamente soporte para: Números de precisión arbitraria. Texto de largo ilimitado. Figuras geométricas. Direcciones IP. Bloques de direcciones estilo CIDR. Direcciones MAC. Arrays.

**ORACLE:**

Sistema manejador de base de datos relacional, desarrollado por Oracle Corporation.

**Características:**

* Soporta transacciones. Estabilidad. Escalabilidad. Soporta multiplataforma.

**FIREBIRD:**

Sistema manejador de base de datos relacional de código abierto, basado en la versión 6 de Interbase, cuyo código fue liberado por Borland en 2000. Su código fue reescrito de C a C++.

**Características:**

* Es multiplataforma, y actualmente puede ejecutarse en los sistemas operativos: Linux, HP-UX, FreeBSD, Mac OS, Solaris y Microsoft Windows.
* Existencia de controladores ODBC, OLEDB, JDBC, PHP, Perl, .net

**Microsoft SQL SERVER:**

Es un sistema manejador de bases de datos relacionales (SGBD) basado en el lenguaje Transact-SQL, y específicamente en Sybase IQ, capaz de poner a disposición de muchos usuarios grandes cantidades de datos de manera simultánea.

**Características:**

* Soporta transacciones. Escalabilidad, estabilidad y seguridad. Soporta procedimientos almacenados.
* Permite el uso de comandos DDL y DML.
* Server incluye interfaces de acceso para varias plataformas de desarrollo.

**PervasiveSQL:**

Motor de base de datos embebible que sustenta la integridad de los datos, el alto rendimiento, flexibilidad, escalabilidad y un bajo coste total de propiedad.

**Características:**

* Proporciona un mayor rendimiento SQL. ODBC, JDBC, ADO / OLE DB y PDAC. Con Turbo Write Accelerator y maximiza la eficacia de las E/S del disco.
* Con la Compatibilidad retroactiva completa, aplicaciones y bases de datos heredadas. Capacidad de incrustación: Biblioteca de desarrollo de clase.

**SQlite:**

sistema manejador de bases de datos relacional compatible con ACID, y que está contenida en una relativamente pequeña biblioteca

**Características:**

La biblioteca implementa la mayor parte del estándar SQL-92, incluyendo transacciones de base de datos atómicas, consistencia de base de datos, aislamiento, y durabilidad (ACID), triggers y la mayor parte de las consultas complejas.

**APACHE DERBY:**

Sistema manejador de base de datos relacional escrito en Java que puede ser embebido en aplicaciones Java y utilizado para procesos de transacciones online.

**Características:**

* Soporta cifrado completo, roles y permisos.
* Soporta internamente procedures, cifrado y compresión.
* Trae soporte multilenguaje y localizaciones específicas.

Diligencie la siguiente tabla y realice un ejemplo en SQL MySQL aplicado a la base de datos “Citas   
médicas”:

|  |  |  |
| --- | --- | --- |
| Sentencia | Función | Ejemplo |
| GRANT | permite dar permisos a uno o varios usuarios o roles para realizar tareas determinadas |  |
| REVOKE | * permite eliminar permisos que previamente se han concedido con GRANT. las tareas sobre las que se pueden conceder o denegar permisos son las siguientes: * CONNECT * SELECT * INSERT * UPDATE * DELETE * USAGE |  |
| COMMIT | guarda el trabajo realizado |  |
| ROLLBACK | restaurar la base de datos a la original, hasta el último COMMIT |  |
| SAVEPOINT | identifica un punto en una transacción a la que más tarde se puede volver |  |

Defina qué es un procedimiento almacenado en MySQL, cuáles son sus ventajas y usos.

**Procedimiento almacenado:** Es un objeto que se crea con la sentencia CREATE PROCEDURE y se invoca con la sentencia CALL. Un procedimiento puede tener cero o muchos parámetros de entrada y cero o muchos parámetros de salida.

Un procedimiento almacenado es un conjunto de instrucciones SQL que se almacena asociado a una base de datos. Es un objeto que se crea con la sentencia CREATE PROCEDURE y se invoca con la sentencia CALL. Un procedimiento puede tener cero o muchos parámetros de entrada y cero o muchos parámetros de salida.

* Pueden recibir y devolver parámetros.
* Pueden manejar tablas, ejecutando operaciones e iteraciones de lectura/escritura.
* Pueden devolver una tabla como resultado.
* Se almacenan en la base de datos en la cual se crean.
* No dependen de ninguna tabla en particular.
* Pueden aceptar recursividad.

Los procedimientos almacenados MySQL se han diseñado para aligerar a las aplicaciones clientes, pudiendo ejecutar directamente en el servidos aquellas tareas pesadas y que necesitan muchos recursos.

**DELIMITER**

Para definir un procedimiento almacenado es necesario modificar temporalmente el carácter separador que se utiliza para delimitar las sentencias SQL.

El carácter separador que se utiliza por defecto en SQL es el punto y coma (;). En los ejemplos que vamos a realizar en esta unidad vamos a utilizar los caracteres $$ para delimitar las instrucciones SQL, pero es posible utilizar cualquier otro carácter.

**Ventajas:**

* **Compatibilidad:** los procedimientos almacenados MySQL permiten ejecutar la misma operación sobre la base de datos también en el caso de que las aplicaciones cliente estén implementadas en distintas plataformas y programadas en distintos lenguajes.
* **Integridad:** gracias a los procedimientos almacenados MySQL es posible centralizar el acceso a la información.
* **Seguridad:** cuando necesitamos evitar el acceso directo a la base de datos, los procedimientos almacenados MySQL permiten establecer un entorno seguro, otorgando permisos y privilegios para la ejecución.
* **Rendimiento:** con los procedimientos almacenados MySQL todo el trabajo se ejecuta en el servidor, por lo cual se minimiza de forma importante la cantidad de información intercambiada con las aplicaciones cliente y se reduce el tráfico de acceso a la base de datos además del número de accesos.
* **Centralización:** los procedimientos almacenados MySQL permiten centralizar la lógica comercial ofreciendo a todas las aplicaciones clientes la misma versión actualizada, lo que hace el mantenimiento más sencillo.
* **Reutilización del código:** al escribir un mismo código que se ejecuta por todas las aplicaciones, es posible reducir las inconsistencias.
* **Sencillez:** los procedimientos almacenados permiten la creación de bibliotecas o funciones, lo que facilita mucho el trabajo del programador ya que se trata de características compartidas por los principales lenguajes de programación modernos.

**Defina qué es una función en MySQL, cuáles son sus ventajas y usos.**

**Función almacenada:** Es un objeto que se crea con la sentencia CREATE FUNCTION y se invoca con la sentencia SELECT o dentro de una expresión. Una función puede tener cero o muchos parámetros de entrada y siempre devuelve un valor, asociado al nombre de la función.

Las funciones son piezas de código que reciben datos de entrada, realizan operaciones con ellos y luego devuelven un resultado. Podemos utilizarlas en nuestras consultas para que el motor de base de datos resuelva eficazmente los cálculos de algunos datos, que, de otra manera, resolviéndolos en nuestra capa de negocio nos podría traer problemas de performance en la aplicación.

Existen dos tipos de funciones, las integradas y las almacenadas:

**Funciones integradas:**

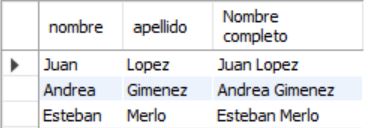
Son las funciones que ya vienen implementadas en el motor de MySQL y nos permiten manipular los datos al momento de hacer una consulta

* **Funciones de cadenas:**

Sirven para operar sobre tipos de datos cadena.

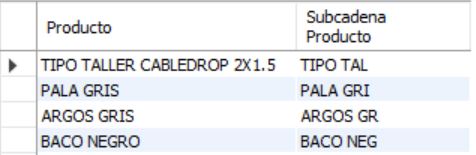
**CONCAT:** podemos concatenar dos o más cadenas en una sola columna. Por ejemplo, en una tabla empleados, en base a la columna nombre y apellido podemos obtener una columna con el nombre completo.

SELECT nombre, apellido, CONCAT(nombre, ' ' ,apellido) AS 'Nombre completo' FROM empleados;



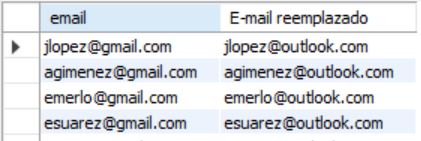
**SUBSTRING:** podemos obtener una subcadena, desde una cierta posición extraer los siguientes caracteres que definimos en el ultimo parametro.

SELECT descripcion AS 'Producto', SUBSTRING(descripcion, 1, 8) AS 'Subcadena Producto' FROM productos;



**REPLACE:** podemos reemplazar una subcadena de una ocurrencia, con otra subcadena. Por ejemplo, en todos los emails, vamos a reemplazar gmail por outlook.

SELECT email, REPLACE(email, 'gmail', 'outlook') AS 'E-mail reemplazado' FROM empleados;

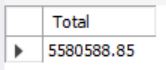


**Funciones numéricas:**

Sirven para operar sobre tipos de datos numéricos.

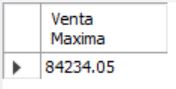
**SUM:** podemos sumar todos los valores de una columna. Por ejemplo, vamos a obtener el valor total de ventas sumando el total de cada una de las ventas.

SELECT SUM(total) AS 'Total' FROM ventas;

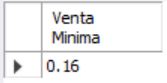


**MAX / MIN:** podemos obtener el máximo o mínimo valor de un registro. Por ejemplo, obtenemos la venta con valor MAXIMO y otra con el valor MINIMO.

SELECT MAX(total) AS 'Venta Maxima' FROM ventas;



SELECT MIN(total) AS 'Venta Minima' FROM ventas;



**AVG:** podemos obtener el promedio de todos los registros en una columna. Por ejemplo, obtenemos el valor promedio de las ventas.

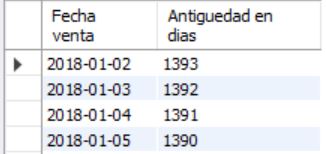
SELECT AVG(total) AS 'Venta Promedio' FROM ventas;



**Funciones fechas:**

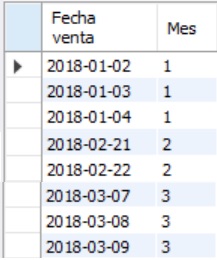
**DATEDIFF:** podemos obtener la diferencia en días entre dos fechas. Por ejemplo, vamos a obtener los días que pasaron entre la fecha de una venta y el día actual con la función NOW().

SELECT fecha AS 'Fecha venta', DATEDIFF(NOW(), fecha) AS 'Antiguedad en dias' FROM ventas;



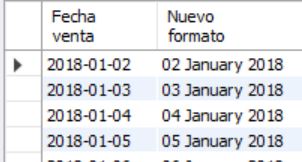
**DAY / MONTH / YEAR:** Podemos obtener el dia, el mes o el año de una fecha respectivamente. Veamos un ejemplo con el MES.

SELECT fecha AS 'Fecha venta', MONTH(fecha) AS 'Mes' FROM ventas;



**DATE\_FORMAT:** nos permite formatear una fecha en base al criterio que creamos necesario. Por ejemplo, en este caso vamos a formatear la fecha con el día en número, el mes con su nombre, y el año en números.

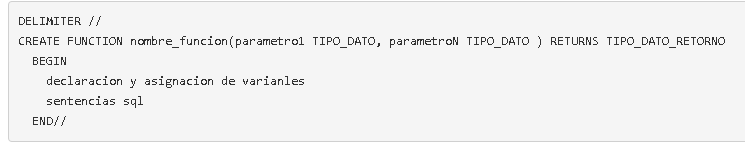
SELECT fecha AS 'Fecha venta', DATE\_FORMAT(fecha, '%d %M %Y') AS 'Nuevo formato' FROM ventas;



**Funciones almacenadas:**

Las funciones almacenadas son aquellas que podemos definir y reutilizar en sentencias SQL como cualquier otra función definida por el motor de MySQL.

La sintaxis básica de una funcione almacenada es la siguiente:



Donde:

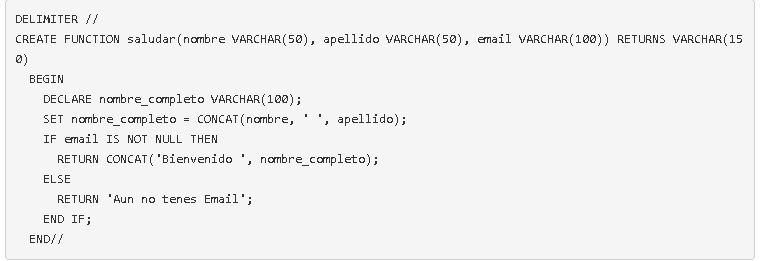
CREATE FUNCTION nombre\_funcion(parametro1 TIPO\_DATO, parametroN TIPO\_DATO ): define el nombre de la función y los parámetros que recibirá con sus tipos de datos correspondientes.

RETURNS: indica el tipo de dato que tendrá el valor de salida de la función

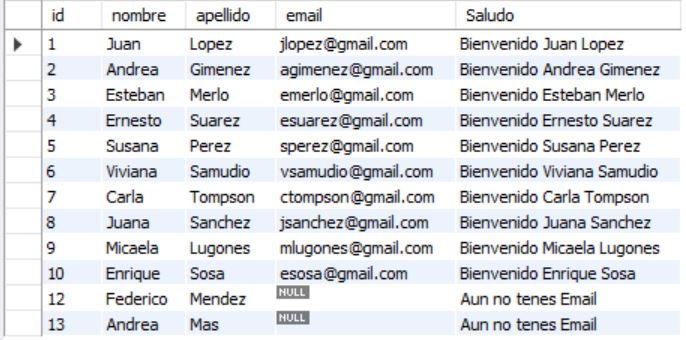
BEGIN: determina el inicio de las sentencias que podemos realizar en la función

END: determina el fin de las sentencias

DELIMITER //: es un delimitador que nos permite indicarle al motor de MySQL sobre donde comienza y finaliza un bloque de código, ya que el delimitador predeterminado es el 'punto y coma', y si no definimos uno (en este caso las dos barras), la ejecución de nuestra función va a finalizar al encontrarse el primer 'punto y coma'.



SELECT id, nombre, apellido, email, saludar(nombre, apellido, email) AS 'Saludo' FROM empleados;



**Defina qué es un disparador o Trigger en MySQL, cuáles son sus ventajas y usos.**

Trigger: Es un objeto que se crea con la sentencia CREATE TRIGGER y tiene que estar asociado a una tabla. Un trigger se activa cuando ocurre un evento de inserción, actualización o borrado, sobre la tabla a la que está asociado.

Un trigger o disparador es un script que se usa en lenguaje de programación SQL, en especial en bases de datos como MySQL o PostgreSQL.

Consiste en una serie de reglas predefinidas que se asocian a una tabla. Estas reglas se aplican a la [base de datos](https://ayudaleyprotecciondatos.es/bases-de-datos/) cuando se realizan determinadas operaciones en la tabla, por ejemplo, al añadir, actualizar o eliminar registros.

Dicho de otra manera, el trigger desencadena determinadas acciones de forma automática en las tablas de la base de datos cuando se insertan, modifican y se añaden nuevos datos.

Estos disparadores se llevan usando en MySQL desde la versión 5.0.2., mientras que PostgreSQL ya los incluyó en el año 1997.

**Funcion:**

La principal función de los TRIGGER es contribuir a mejorar la gestión de la base de datos. Gracias a ellos muchas operaciones se pueden realizar de forma automática, sin necesidad de intervención humana, lo que permite ahorrar mucho tiempo.

Otra de sus funciones es aumentar la seguridad e integridad de la información. Esto lo consiguen gracias a la programación de restricciones o requerimientos de verificación que permiten minimizar los errores y sincronizar la información.

* Se produce una llamada de activación al código que se ha de ejecutar.
* Aplica las restricciones necesarias para poder realizar la acción, por ejemplo, una determinada condición o una nulidad.
* Una vez verificadas las restricciones, se ejecuta la acción, en base a las instrucciones recibidas en el primer punto.

**¿Cuándo se debe usar?**

Los TRIGGER se puede ejecutar cuando el usuario realizar alguna acción relacionada con añadir, actualizar o eliminar información de una tabla. Es decir, al usar los comandos INSERT, UPDATE o DELETE.

Por tanto, para poder usar un TRIGGER es necesario que el usuario posea permisos INSERT y DELETE e dicha base de datos.

**Tipos de TRIGGER**

Existen diferentes tipos de disparadores, en función de las ejecuciones que realizan.

* Disparadores de fila: también llamados ROW TRIGGERS, son aquellos cuya ejecución se realiza a través de llamadas desde una tabla asociada al trigger.
* Disparadores de secuencia. también llamados STATEMENT TRIGGERS, son aquellos que se ejecutan solo una vez, independientemente de la cantidad de veces que se cumplan las condiciones para su ejecución.

**Ejemplos:**

* Crear una copia automatizada que registre cada cliente que entra en un eCommerce.
* Contabilizar estadísticas sobre accesos a una base de datos y las acciones que se han llevado a cabo en ellas.
* Duplicar tablas con información sincronizada.
* Programar la base de datos para que se realice un pedido de forma automática cuando el stock ha legado a un mínimo determinado de unidades.
* Bloquear la inserción de datos incorrectos o de transacciones inválidas.
* Crear alertas que avisen de cuando se ha modificado algún dato en una tabla.

**En MySQL, realice un ejemplo de procedimiento almacenado sin incluir parámetros para la base de datos “Citas médicas”, muestre la sintaxis y el resultado.**

**En MySQL, realice un ejemplo de procedimiento almacenado incluyendo parámetros para la base de datos “Citas médicas”, muestre la sintaxis y el resultado.**

# **Referencias**

http://www.tierradelazaro.com/wp-content/uploads/2016/12/DDL-DML-DCL-TCL.pdf

<https://josejuansanchez.org/bd/unidad-12-teoria/index.html#triggers>

<https://ayudaleyprotecciondatos.es/bases-de-datos/trigger/#:~:text=Un%20trigger%20o%20disparador%20es,se%20asocian%20a%20una%20tabla>.

<https://blog.mdcloud.es/procedimientos-almacenados-mysql-que-son-como-crearlos-y-ventajas/#:~:text=Los%20procedimientos%20almacenados%20MySQL%20se,Pueden%20recibir%20y%20devolver%20par%C3%A1metros>.

<https://coggle.it/diagram/Xpdd4epGh1FdomSW/t/tipos-de-smbd-sistemas-manejadores-de-base-datos>

<http://vyanez-combas-sca.blogspot.com/2011/09/diferentes-tipos-de-smbd.html>

<https://somospnt.com/blog/230-funciones-en-mysql#:~:text=Las%20funciones%20son%20piezas%20de,y%20luego%20devuelven%20un%20resultado>.