	Centro de Teleinformática y Producción Industrial - Regional Cauca			Pág.1 de40
	Número de Documento:	ENT-WS-DATABASE1	Fecha de Creación: 3/03/2020	Preparado por: Ing. Henry Bastidas
	Nombre del Documento:	Entrenamiento Bases de Datos MySQL		

Capacitación Bases de Datos MySQL Server

1. Descargar e Instalar XAMPP

¿Qué es XAMPP?

XAMPP es un paquete de software libre, que consiste principalmente en el sistema de gestión de bases de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script PHP y Perl. El nombre es en realidad un acrónimo: X (para cualquiera de los diferentes sistemas operativos), Apache, MariaDB/MySQL, PHP, Perl. A partir de la versión 5.6.15, XAMPP cambió la base de datos MySQL por MariaDB, un fork de MySQL con licencia GPL.

2. Descargar la versión para Windows 10, Para poder instalar el programa XAMPP en Windows 10, primero tenemos que descárgalo de su página oficial para ello debemos seguir los siguientes pasos:

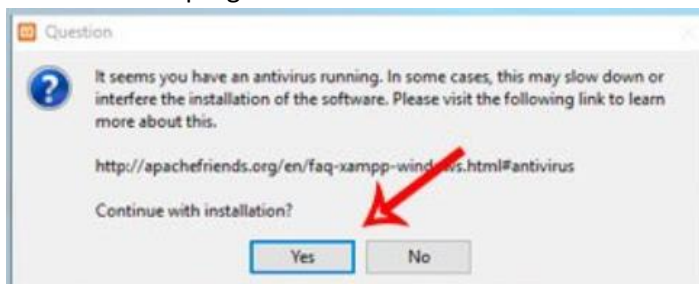
- ✓ Ir a su página oficial de XAMPP, Para descargar el programa XAMPP a tu computadora, primero tienes que ir a su página oficial y desde allí podrás descargar el software para su instalación en Windows 10.
- ✓ Ir a su página oficial del programa XAMPP haciendo clic en este enlace <https://www.apachefriends.org/es/index.html> y automáticamente en tu navegador se abrirá una nueva pestaña con la página oficial de XAMPP en español, y desde allí podrás descargarlo el programa.



- ✓ En su página web oficial podemos ver varias opciones de descarga, por ejemplo: este software XAMPP está disponible para distintas plataformas de Sistemas Operativos como para Windows, Linux y Mac.

	Centro de Teleinformática y Producción Industrial - Regional Cauca			Pág.2 de40
	Número de Documento:	ENT-WS-DATABASE1	Fecha de Creación: 3/03/2020	Preparado por: Ing. Henry Bastidas
	Nombre del Documento:	Entrenamiento Bases de Datos MySQL		

- ✓ Lo que tienes que hacer aquí es, escoger para la plataforma Windows, porque en esta guía te enseñare a cómo instalar el programa XAMPP en Windows 10.
- ✓ Para ejecutar el asistente de instalación del programa XAMPP en Windows 10, tienes que hacer clic derecho en el archivo ejecutable y luego de esta acción se deslizará una barra de menú.
- ✓ Esta acción se realiza para poder ejecutar el archivo como administrador, y con este paso el programa XAMPP en Windows 10 se instalará correctamente y sin errores.
- ✓ Luego que hagas clic en el menú “Ejecutar como administrador” se abrirá otra ventana con esta pregunta ¿Desea permitir que este Programa realice cambios en el Equipo?, en esta ventana de advertencia de Windows 10 tienes que hacer clic en el botón “Si” para continuar con la instalación del programa XAMPP.
- ✓ Si tienes en tu computador instalado un Antivirus, entonces, este mensaje te aparecerá como una advertencia, indicando que esto puede interferir en la instalación del programa XAMPP.



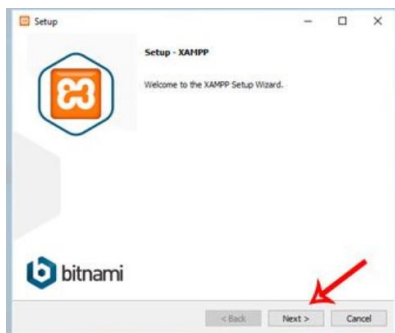
- ✓ Sería recomendable que hasta lograr la instalación del programa XAMPP en Windows 10 desactives por un periodo de 30 minutos tu antivirus.
- ✓ Después de terminar la instalación, entonces vuelves activar tu antivirus para seguir protegiendo tu Windows 10.
- ✓ Después aparecerá otra advertencia, es tedioso ver estos mensajes, pero tenemos que aceptar estos mensajes con paciencia para poder seguir con la instalación de programa XAMPP en Windows 10.
- ✓ Entonces, lo que aremos aquí es presionar el botón “Ok” y continuar con la instalación del programa XAMPP en Windows 10.
- ✓ Desde aquí hacia adelante comenzaremos a configurar el asistente de instalación del programa XAMPP para Windows 10.
- ✓ En esta ventana de bienvenida haremos clic en el botón “Next”, que se encuentra en la parte inferior para continuar con la configuración.



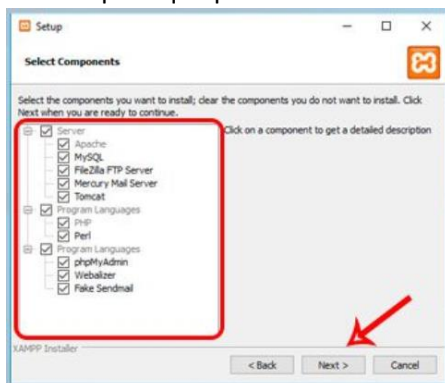
Número de Documento:

ENT-WS-DATABASE1Fecha de Creación:
3/03/2020Preparado por:
Ing. Henry Bastidas

Nombre del Documento:

Entrenamiento Bases de Datos MySQL

- ✓ Seleccionar Componentes, el siguiente paso que nos muestra el asistente de instalación del programa XAMPP es: Elegir los componentes, para que puedan instalarse en tu computador.
- ✓ En esta parte del asistente marcamos todos los componentes que trae el programa XAMPP para que pueda instalarse en tu computadora.



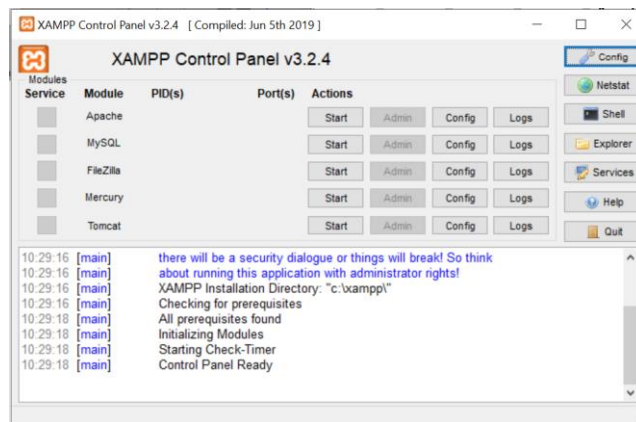
- ✓ Seleccione la Carpeta, ahora lo que nos pide en esta parte es: Elegir la ubicación en donde se podrá instalar el programa XAMPP. No es recomendable modificar esta parte, a menos que seas un usuario avanzado y sabes lo que haces.
- ✓ Después de hacer estas configuraciones en el asistente de instalación, llega el momento de instalar el programa XAMPP en Windows 10, para continuar, tienes que hacer clic en el botón “Next” y el asistente comenzara a instalar el programa XAMPP en Windows 10.
- ✓ Finalizando la Instalación, una vez que termine de instalar el programa XAMPP en Windows 10, aparecerá en la pantalla una última ventana para poder configurar.
- ✓ En esta ventana antes de hacer clic en el botón “Finish”, tienes que deshabilitar, es decir deseleccionar el checkbox, que por defecto el asistente lo tiene seleccionado el checkbox, porque si lo dejamos tal como está y haces clic en el botón “Finish”, se ejecutara el panel de control de XAMPP.
- ✓ El siguiente paso será elegir el idioma, el instalador nos da dos alternativas en inglés o alemán, seleccionamos y damos clic en Save para salvar la elección.
- ✓ Posteriormente debemos cargar el xampp-control desde donde podemos controlar la ejecución de los servicios instalados.



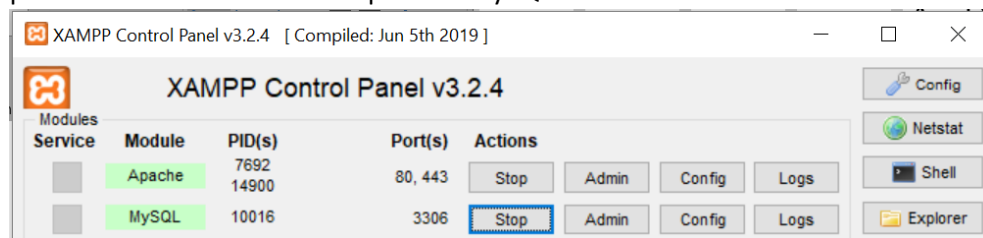
Número de Documento:

ENT-WS-DATABASE1Fecha de Creación:
3/03/2020Preparado por:
Ing. Henry Bastidas

Nombre del Documento:

Entrenamiento Bases de Datos MySQL

- ✓ Para iniciar los servicios solo basta con presionar el botón Start sobre los servicios para nuestro caso vamos a requerir el MySQL nuestro servidor de base de datos.



- ✓ Y listo de esta manera ya tenemos listo todo para iniciar a trabajar con mysql

3. Que es SQL?

¿Qué es SQL? SQL Structured Query Language en español Lenguaje de consulta Estructurado, es un lenguaje específico utilizado en programación, diseñado para administrar y recuperar información de sistemas de gestión de bases de datos relacionales.

Una de sus principales características es el manejo de álgebra y el cálculo relacional para realizar consultas y obtener información de forma sencilla, y además para realizar cambios en ella.

El SQL permite definir y trabajar con diferentes tipos de datos como los siguientes:

- Tipos de datos.
 - Definición de variables.
 - Estructuras de control de flujo.
 - Gestión de excepciones.
 - Funciones predefinidas.

Sin embargo, no nos permite:

	Centro de Teleinformática y Producción Industrial - Regional Cauca			Pág.5 de40
	Número de Documento:	ENT-WS-DATABASE1	Fecha de Creación: 3/03/2020	Preparado por: Ing. Henry Bastidas
	Nombre del Documento:	Entrenamiento Bases de Datos MySQL		

- Crear interfaces de usuario.
- Crear aplicaciones ejecutables.

MySQL es un sistema de gestión de base de datos relacional (RDBMS) de código abierto, basado en lenguaje de consulta estructurado (SQL).

MySQL se ejecuta en prácticamente todas las plataformas, incluyendo Linux, UNIX y Windows. A pesar de que se puede utilizar en una amplia gama de aplicaciones, MySQL se asocia más con las aplicaciones basadas en la web y la publicación en línea y es un componente importante de una pila empresarial de código abierto llamado LAMP. LAMP es una plataforma de desarrollo web que utiliza Linux como sistema operativo, Apache como servidor web, MySQL como sistema de gestión de base de datos relacional y PHP como lenguaje de programación orientado a objetos (a veces, Perl o Python se utiliza en lugar de PHP).

Que son las Tablas, Campos y registros

Las tablas son objetos de base de datos que contienen todos sus datos. En las tablas, los datos se organizan con arreglo a un formato de filas y columnas, similar al de una hoja de cálculo. Cada fila representa un registro único y cada columna un campo dentro del registro. Por ejemplo, en una tabla que contiene los datos de los empleados de una compañía puede haber una fila para cada empleado y distintas columnas en las que figuren detalles de los mismos, como el número de empleado, el nombre, la dirección, el puesto que ocupa y su número de teléfono particular.


Origen y estructura

Las tablas se componen de dos elementos principales que vamos a utilizar muy seguido cuando hablemos de bases de datos y de lenguaje SQL, ellos son los campos y los registros:

Campo: es la denominación que se le da a la columna, el mismo tiene un nombre único, un tipo de datos específico y un formato con una longitud o ancho de campo.

Registro: así se llama a cada fila que compone una tabla, es decir los renglones de la tabla, y están compuestos por cada uno de los campos de la tabla.

Dentro de una base de datos las "Tablas", son objetos, de vital importancia, y se podría decir que constituyen el arreglo principal que pueden tener los datos dentro de la misma, ya que viene siendo la forma más común de guardar y organizar los

	Centro de Teleinformática y Producción Industrial - Regional Cauca			Pág.6 de40
	Número de Documento:	ENT-WS-DATABASE1	Fecha de Creación: 3/03/2020	Preparado por: Ing. Henry Bastidas
	Nombre del Documento:	Entrenamiento Bases de Datos MySQL		

mismos. Su estructura general es comparable a una hoja de cálculo de Excel, es decir que la podríamos imaginar como un conjunto de celdas donde tenemos un campo al lado del otro, los cuales todos juntos conforman un registro y así luego tenemos un registro debajo del otro, completando la estructura final de la tabla.

A continuación, mostraremos un ejemplo.

Campos de la Tabla (Columnas - amarillo)

ID	Nombre	Descripción	Precio
1	Chaqueta	Chaqueta para dama tipo jean	100000
2	Camiseta	Tipo polo marca ACME	30000
3	Pantalón	Para caballero tipo Jean	80000
4	Camisa	Para Caballero	

Registros de la tabla (Filas - azul)

Propiedades especiales


A los campos se les puede asignar, además, propiedades especiales que afectan a los registros insertados, como ser que no se admitan valores nulos, para lo cual dicho campo deberá tener al menos un valor, o bien que sean campos índice, los cuales se usarán como criterio principal a la hora de ordenar los datos de la tabla y demás acciones que veremos más adelante, y por último pueden ser campos autonuméricos o autoincrementables los cuales básicamente sirven como para dar un único identificador o Id a los registros de la tabla.

Todas estas propiedades no son las únicas y las veremos más en detalle cuando veamos cómo se crea y construye una tabla, pero como vimos se resumen en:

- Admitir valores Nulos
- Indexación
- Campo autoincrementable

El espacio de una tabla

Una tabla es una forma o modelo para guardar y organizar los datos dentro de una base de datos, son uno de los objetos principales de dichas bases, y existen como tales desde el momento de crearse hasta que sean eliminadas, permanecen guardadas invariables hasta

	Centro de Teleinformática y Producción Industrial - Regional Cauca			Pág.7 de40
	Número de Documento:	ENT-WS-DATABASE1	Fecha de Creación: 3/03/2020	Preparado por: Ing. Henry Bastidas
	Nombre del Documento:	Entrenamiento Bases de Datos MySQL		

que se les realice alguna modificación y no varían sus datos a no ser que sean actualizados, se inserten registros nuevos, se eliminen registros o bien se elimine la tabla. También se les puede agregar campos nuevos o eliminar alguno de los mismos.

Con todo esto también queremos explicar que la tabla ocupa un espacio dentro de la base de datos, este espacio está determinado por el total de la cantidad de registros y estos a su vez por el total de campos que los conforman.

4. Que es un Tipo de Dato

En ciencias de la computación, un tipo de dato informático o simplemente tipo, es un atributo de los datos que indica al ordenador (y/o al programador/programadora) sobre la clase de datos que se va a manejar. Esto incluye imponer restricciones en los datos, como qué valores pueden tomar y qué operaciones se pueden realizar.


Los tipos de datos más comunes son: números enteros, números con signo (negativos), números de coma flotante (decimales), cadenas alfanuméricas (y unicodes), estados, etc.

Un tipo de dato es, un espacio en memoria con restricciones. Por ejemplo, el tipo "int" representa, generalmente, un conjunto de enteros de 32 bits cuyo rango va desde el -2.147.483.648 al 2.147.483.647, así como las operaciones que se pueden realizar con los enteros, como son la suma, la resta, y la multiplicación. Los colores, por su parte, se representan como tres bytes denotando la cantidad de rojo, verde y azul, y una cadena de caracteres representando el nombre del color (en este caso, las operaciones permitidas incluyen la adición y la sustracción, pero no la multiplicación).

Este es un concepto propio de la informática, y más específicamente de los lenguajes de programación, aunque también se encuentra relacionado con nociones similares de la matemática y la lógica.

En un sentido amplio, un tipo de datos define un conjunto de valores y las operaciones sobre esos valores. Casi todos los lenguajes de programación explícitamente incluyen la notación del tipo de datos, aunque lenguajes diferentes pueden usar terminologías diferentes. La mayor parte de los lenguajes de programación permiten al programador definir tipos de datos adicionales, normalmente combinando múltiples elementos de otros tipos y definiendo las operaciones del nuevo tipo de dato. Por ejemplo, un programador puede crear un nuevo tipo de dato llamado "Persona", contemplando que el dato interpretado como Persona incluya un nombre y una fecha de nacimiento.

Un tipo de dato puede ser también visto como una limitación impuesta en la interpretación de los datos en un sistema de tipificación, describiendo la representación, la interpretación y la estructura de los valores u objetos almacenados en la memoria del ordenador. El

	Centro de Teleinformática y Producción Industrial - Regional Cauca			Pág.8 de40
	Número de Documento:	ENT-WS-DATABASE1	Fecha de Creación: 3/03/2020	Preparado por: Ing. Henry Bastidas
	Nombre del Documento:	Entrenamiento Bases de Datos MySQL		

sistema de tipificación usa información de los tipos de datos para comprobar la verificación de los programas que acceden o manipulan los datos.

Los tipos de datos que puede haber en un campo, se pueden agrupar en tres grandes grupos:

- ✓ Tipos numéricos
- ✓ Tipos de Fecha
- ✓ Tipos de Cadena

Tipos numéricos:

Existen tipos de datos numéricos, que se pueden dividir en dos grandes grupos, los que están en coma flotante (con decimales) y los que no.

- TinyInt: Es un número entero con o sin signo. Con signo el rango de valores válidos va desde -128 a 127. Sin signo, el rango de valores es de 0 a 255
- Bit ó Bool: Un número entero que puede ser 0 ó 1
- SmallInt: Número entero con o sin signo. Con signo el rango de valores va desde -32768 a 32767. Sin signo, el rango de valores es de 0 a 65535.
- MediumInt: Número entero con o sin signo. Con signo el rango de valores va desde -8.388.608 a 8.388.607. Sin signo el rango va desde 0 a 16777215.
- Integer, Int: Número entero con o sin signo. Con signo el rango de valores va desde -2147483648 a 2147483647. Sin signo el rango va desde 0 a 429.4967.295
- BigInt: Número entero con o sin signo. Con signo el rango de valores va desde -9.223.372.036.854.775.808 a 9.223.372.036.854.775.807. Sin signo el rango va desde 0 a 18.446.744.073.709.551.615.
- Float: Número pequeño en coma flotante de precisión simple. Los valores válidos van desde -3.402823466E+38 a -1.175494351E-38, 0 y desde 1.175494351E-38 a 3.402823466E+38.
- xReal, Double: Número en coma flotante de precisión doble. Los valores permitidos van desde -1.7976931348623157E+308 a -2.2250738585072014E-308, 0 y desde 2.2250738585072014E-308 a 1.7976931348623157E+308
- Decimal, Dec, Numeric: Número en coma flotante desempaquetado. El número se almacena como una cadena



Número de Documento:

ENT-WS-DATABASE1Fecha de Creación:
3/03/2020Preparado por:
Ing. Henry Bastidas

Nombre del Documento:

Entrenamiento Bases de Datos MySQL

Tipo de Campo	Tamaño de Almacenamiento
TINYINT	1 byte
SMALLINT	2 bytes
MEDIUMINT	3 bytes
INT	4 bytes
INTEGER	4 bytes
BIGINT	8 bytes
FLOAT(X)	4 ú 8 bytes
FLOAT	4 bytes
DOUBLE	8 bytes
DOUBLE PRECISION	8 bytes
REAL	8 bytes
DECIMAL(M,D)	M+2 bytes sí D > 0, M+1 bytes sí D = 0
NUMERIC(M,D)	M+2 bytes if D > 0, M+1 bytes if D = 0

Tipos fecha:

A la hora de almacenar fechas, hay que tener en cuenta que Mysql no comprueba de una manera estricta si una fecha es válida o no. Simplemente comprueba que el mes está comprendido entre 0 y 12 y que el día está comprendido entre 0 y 31.

- **Date:** Tipo fecha, almacena una fecha. El rango de valores va desde el 1 de enero del 1001 al 31 de diciembre de 9999. El formato de almacenamiento es de año-mes-día
- **DateTime:** Combinación de fecha y hora. El rango de valores va desde el 1 de enero del 1001 a las 0 horas, 0 minutos y 0 segundos al 31 de diciembre del 9999 a las 23 horas, 59 minutos y 59 segundos. El formato de almacenamiento es de año-mes-dia horas:minutos:segundos
- **TimeStamp:** Combinación de fecha y hora. El rango va desde el 1 de enero de 1970 al año 2037. El formato de almacenamiento depende del tamaño del campo:



Número de Documento:

ENT-WS-DATABASE1Fecha de Creación:
3/03/2020

Preparado por:

Ing. Henry Bastidas

Nombre del Documento:

Entrenamiento Bases de Datos MySQL

Tamaño	Formato
14	AñoMesDiaHoraMinutoSegundo aaaammddhhmmss
12	AñoMesDiaHoraMinutoSegundo aammddhhmmss
8	AñoMesDia aaaammdd
6	AñoMesDia aammdd
4	AñoMes aamm
2	Año aa

- Time: Almacena una hora. El rango de horas va desde -838 horas, 59 minutos y 59 segundos a 838, 59 minutos y 59 segundos. El formato de almacenamiento es de 'HH:MM:SS'
- Year: Almacena un año. El rango de valores permitidos va desde el año 1901 al año 2155. El campo puede tener tamaño dos o tamaño 4 dependiendo de si queremos almacenar el año con dos o cuatro dígitos.

Tipo de Campo	Tamaño de Almacenamiento
DATE	3 bytes
DATETIME	8 bytes
TIMESTAMP	4 bytes
TIME	3 bytes
YEAR	1 byte

Tipos de cadena:

- Char(n): Almacena una cadena de longitud fija. La cadena podrá contener desde 0 a 255 caracteres.
- VarChar(n): Almacena una cadena de longitud variable. La cadena podrá contener desde 0 a 255 caracteres.

	Centro de Teleinformática y Producción Industrial - Regional Cauca			Pág.11
	Número de Documento:	ENT-WS-DATABASE1	Fecha de Creación:	3/03/2020
	Preparado por:			Ing. Henry Bastidas
Nombre del Documento:		Entrenamiento Bases de Datos MySQL		

- Dentro de los tipos de cadena se pueden distinguir otros dos subtipos, los tipos Test y los tipos BLOB (Binary large Object).

La diferencia entre un tipo y otro es el tratamiento que reciben a la hora de realizar ordenamientos y comparaciones. Mientras que el tipo test se ordena sin tener en cuenta las Mayúsculas y las minúsculas, el tipo BLOB se ordena teniéndolas en cuenta.

Los tipos BLOB se utilizan para almacenar datos binarios como pueden ser ficheros.

- TinyText y TinyBlob: Columna con una longitud máxima de 255 caracteres.
- Blob y Text: Un texto con un máximo de 65535 caracteres.
- MediumBlob y MediumText: Un texto con un máximo de 16.777.215 caracteres.
- LongBlob y LongText: Un texto con un máximo de caracteres 4.294.967.295. Hay que tener en cuenta que debido a los protocolos de comunicación los paquetes pueden tener un máximo de 16 Mb.
- Enum: Campo que puede tener un único valor de una lista que se especifica. El tipo Enum acepta hasta 65535 valores distintos
- Set: Un campo que puede contener ninguno, uno ó varios valores de una lista. La lista puede tener un máximo de 64 valores.

Tipo de campo	Tamaño de Almacenamiento
CHAR(n)	n bytes
VARCHAR(n)	n +1 bytes
TINYBLOB, TINYTEXT	Longitud+1 bytes
BLOB, TEXT	Longitud +2 bytes
MEDIUMBLOB, MEDIUMTEXT	Longitud +3 bytes
LOBLOB, LONGTEXT	Longitud +4 bytes
ENUM('value1','value2',...)	1 ó dos bytes dependiendo del número de valores
SET('value1','value2',...)	1, 2, 3, 4 ó 8 bytes, dependiendo del número de valores

Diferencia de almacenamiento entre los tipos Char y VarChar



Número de Documento:

ENT-WS-DATABASE1Fecha de Creación:
3/03/2020

Preparado por:

Ing. Henry Bastidas

Nombre del Documento:

Entrenamiento Bases de Datos MySQL

Valor	CHAR(4)	Almacenamiento	VARCHAR(4)	Almacenamiento
"	"	4 bytes	"	1 byte
'ab'	'ab '	4 bytes	'ab'	3 bytes
'abcd'	'abcd'	4 bytes	'abcd'	
'abcdefgh'	'abcd'	4 bytes	'abcd'	5 bytes

5. Restricciones y Propiedades de los campos

PRIMARY KEY Definiciones y reglas generales

- ✓ La clave primaria o primary key, identifica de manera unívoca (única) a cada registro de una tabla.
- ✓ El valor que contiene la columna definida como primary key, debe ser único.
- ✓ El valor debe ser NOT NULL (no permitirá valores nulos)
- ✓ Una tabla puede tener más de un campo PK, a la que llamaremos CLAVE COMPUESTA
- ✓ Sea SIMPLE o COMPUESTA, cada tabla solo podrá tener una clave primaria (PRIMARY KEY)
- ✓ Es decir que sólo habrá una única clave primaria de ese tipo. No se podrá crear otra combinación.

IdPaciente ▾	nombre	apellido	email	idpais
1	Jorge	Rodriguez	a@a.com	MEX
2	Marcelo	Lopez Llano	a@a.com	MEX
3	Kari	Lopreta	a@a.com	COL
4	Juan Manuel	Loperfano	a@a.com	ARG
5	Juan Manuel	Perez Lozano	a@a.com	ESP
6	Karim	Berragas	a@a.com	PER
7	Saul	Lopez Gomez	a@a.com	CHI

6. FOREIGN KEY Definiciones y reglas generales

- ✓ La clave foránea o foreign key, debe ser del mismo tipo de dato que su campo relacionado.
- ✓ El valor del campo definido como FK puede ser NULL
- ✓ Una tabla puede tener más de un campo FK



Número de Documento:

ENT-WS-DATABASE1Fecha de Creación:
3/03/2020

Preparado por:

Ing. Henry Bastidas

Nombre del Documento:

Entrenamiento Bases de Datos MySQL

IdPaciente ▾	nombre	apellido	email	Idpais (FK) ▾
1	Jorge	Rodriguez	a@a.com	MEX
2	Marcelo	Lopez Llano	a@a.com	MEX
3	Kari	Lopreta	a@a.com	COL
4	Juan Manuel	Loperfano	a@a.com	ARG
5	Juan Manuel	Perez Lozano	a@a.com	ESP
6	Karim	Berragas	a@a.com	PER
7	Saul	Lopez Gomez	a@a.com	CHI

Idpais ▾	pais
MEX	México
COL	Colombia
ARG	Argentina

7. Fundamentos y Diseño de Bases de Datos

Modelo entidad-relación ER

El modelo entidad-relación ER es un modelo de datos que permite representar cualquier abstracción, percepción y conocimiento en un sistema de información formado por un conjunto de objetos denominados entidades y relaciones, incorporando una representación visual conocida como diagrama entidad-relación.

Conceptos del modelo ER

Registros: guardan una serie de características similares o que pueden ser agrupados o clasificados dadas sus características comunes en grupos bien delimitados, en términos de abstracción como la extensión de la base de datos. Por ejemplo, es la lista de usuarios de una biblioteca, la lista de productos con sus características, la lista de tipos de documentos y su definición.

Entidad(Tabla). La entidad es cualquier clase de objeto o conjunto de elementos presentes o no, en un contexto determinado dado por el sistema de información o las funciones y procesos que se definen en un plan de automatización. Dicho de otra forma, las entidades las constituyen las tablas de la base de datos que permiten el almacenamiento de los ejemplares o registros del sistema, quedando recogidos bajo la denominación o título de la tabla o entidad. Por ejemplo, la entidad usuarios guarda los datos personales de los usuarios de la biblioteca, la entidad catalogo registra todos los libros catalogados, la entidad circulación todos los libros prestados y devueltos y así sucesivamente con todos los casos.

	Centro de Teleinformática y Producción Industrial - Regional Cauca			Pág.14 de40
	Número de Documento:	ENT-WS-DATABASE1	Fecha de Creación: 3/03/2020	Preparado por: Ing. Henry Bastidas
	Nombre del Documento:	Entrenamiento Bases de Datos MySQL		

Atributos - Intención. Son las características, rasgos y propiedades de una entidad, que toman como valor una instancia particular. Es decir, los atributos de una tabla son en realidad sus campos descriptivos, el predicado que permite definir lo que decimos de un determinado sujeto. Por ejemplo, de una entidad o tabla catálogo, se pueden determinar los atributos título, subtítulo, título paralelo, otras formas del título, autor principal, otras menciones de responsabilidad, edición, mención de edición, editorial, lugar de publicación, fecha de publicación, ...

Relación. Vínculo que permite definir una dependencia entre los conjuntos de dos o más entidades. Esto es la relación entre la información contenida en los registros de varias tablas. Por ejemplo, los usuarios suelen clasificarse según una lista de tipos de usuarios, ya sean profesores, alumnos o investigadores. De esta forma es posible emitir la relación entre el usuario Jorge Martínez como alumno y Enrique Valtierra como profesor. Las relaciones son definidas de forma natural en un diagrama relacional para expresar un modelo cognitivo que dará lugar posteriormente a las interrelaciones de las entidades.

Interrelación. Las interrelaciones las constituyen los vínculos entre entidades, de forma tal que representan las relaciones definidas en el esquema relacional de forma efectiva. Esto no sólo la relación de los registros sino de sus tablas y de las características de la interrelación entre las entidades, a través de un campo clave que actúa como código de identificación y referencia para relacionar (es decir, como nexo de unión y articulación de la relación). Los tipos de interrelaciones entre entidades o tablas se realizan aplicando las reglas de cardinalidad y modalidad.

Entidades fuertes. Lo constituyen las tablas principales de la base de datos que contienen los registros principales del sistema de información y que requieren de entidades o tablas auxiliares para completar su descripción o información. Por ejemplo, la tabla usuario es una entidad fuerte en relación a la tabla tipos de usuarios, que es una entidad débil dada su condición auxiliar para clasificar a los usuarios registrados en la biblioteca.

Entidades débiles. Son entidades débiles a las tablas auxiliares de una tabla principal a la que completan o complementan con la información de sus registros relacionados. Por ejemplo también son consideradas entidades débiles las tablas intermedias que sirven para compartir información de varias tablas principales.

Modelo Entidad – Relación	Objeto de la Base de Datos	Ejemplo
Registros - Conjuntos-Extensiones	Registros de una Tabla	Conjunto-Usuarios{Jorge Martínez(1 alumno), Enrique Valtierra(2 profesor), Miguel dos Santos(3 investigador)}
Entidad	Tabla de la base de datos	Tabla usuarios
Atributos – Intención	Campos de una tabla	id, nombre, apellidos, tipo de usuario, dni, dirección, teléfono.

	Centro de Teleinformática y Producción Industrial - Regional Cauca			Pág.15 de40
	Número de Documento:	ENT-WS-DATABASE1	Fecha de Creación: 3/03/2020	Preparado por: Ing. Henry Bastidas
	Nombre del Documento:	Entrenamiento Bases de Datos MySQL		

Relación	Vínculo entre conjuntos	Jorge Martínez es investigador
Interrelación	Relación entre tablas	Tabla Usuarios relacionada con Tabla Tipo de usuarios
Entidades fuertes	Tabla principal	Tabla Usuarios
Entidades débiles	Tabla auxiliar	Tabla Tipo de usuarios

Restricciones del MER

- ✓ Un único constructor (la relación).
- ✓ La relación es un conjunto (en el sentido matemático) y por lo
- ✓ tanto:
 - No pueden existir tuplas duplicadas
 - Toda relación debe tener una llave primaria 1
 - No hay noción de orden entre las tuplas o entre los atributos.
 - Los atributos de la llave primaria NO pueden ser nulos

Llaves Primarias

- ✓ **Llave:** Atributo o conjunto de atributos cuyo valor es único y diferente para cada tupla, una tabla puede poseer más de una llave.
- ✓ **Llave Candidata:** Es una llave tal que:
 - ✓ Es única (i.e., es una llave).
 - ✓ Es irreducible: No se pueden eliminar componentes de la llave sin destruir la unicidad.

Las llaves no se identifican dependiendo del estado o la instancia de la base de datos.

- ✓ **Llave Compuesta:** Es una llave conformada por más de un atributo.
- ✓ **Llave Primaria:** Llave candidata que se escoge en el modelo para identificar cada tupla.
- ✓ **Llaves alternas:** Las llaves candidatas que no fueron escogidas como llave primaria
- ✓ **Llave Foránea:** atributo(s) de una relación r1 que hacen referencia a otra relación r2 Las FK permiten representar relaciones entre las entidades. r1 y r2 pueden ser la misma relación. Por ejemplo: personas (cc, nombre, apellido, teléfono, dirección, ciu_id) ciudades(id, nombre).
El atributo ciu_id en personas es una llave foránea que referencia al atributo id en la tabla ciudades.

Integridad referencial. Se denomina integridad referencial al tipo de interrelación que se produce entre tablas mediante un campo clave que deberá contener la cadena alfanumérica exacta al identificador de la tabla auxiliar para poder realizar la relación entre los registros. En caso contrario no se produce la relación. Además, se trata de un mecanismo que evita duplicidades e

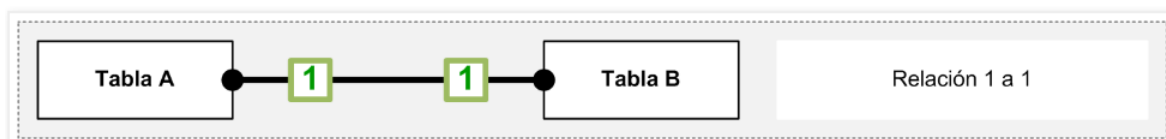
	Centro de Teleinformática y Producción Industrial - Regional Cauca			Pág.16 de40
	Número de Documento:	ENT-WS-DATABASE1	Fecha de Creación: 3/03/2020	Preparado por: Ing. Henry Bastidas
	Nombre del Documento:	Entrenamiento Bases de Datos MySQL		

incorrecciones ya que la propiedad de integridad referencial conmina a que los datos de un usuario además de su identificador ID sean distintos al de los demás. Dicho de otra forma, no pueden existir dos registros iguales con los mismos datos.

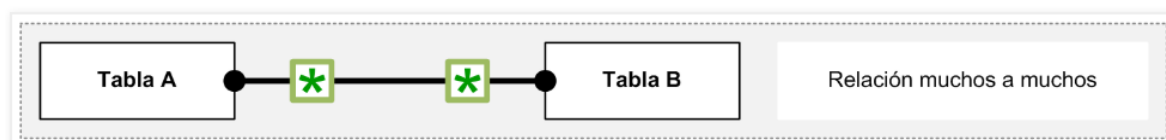
Tipos de relaciones

Según cardinalidad. La cardinalidad se representan en un diagrama ER como una etiqueta que se ubica en ambos extremos de la línea de relación de las entidades y que puede contener diversos valores entre los que destacan comúnmente el 1 y el *, obteniendo los siguientes tipos:

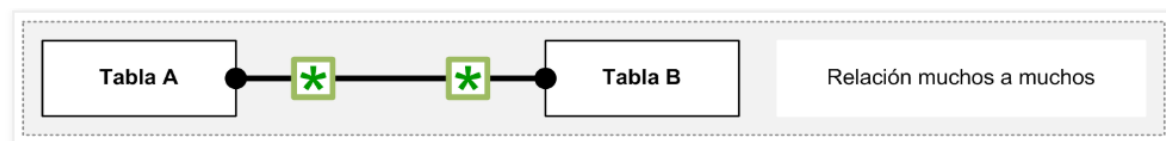
Relación 1 a 1. La relación uno a uno, define que un único registro de la tabla puede estar relacionado con un único registro de la tabla relacionada.



Relación 1 a *. La relación de uno a varios, define que un registro dado de una tabla auxiliar o secundaria sólo puede estar vinculado con un único registro de la tabla principal con la que está relacionada.



Relación * a *. La relación de varios a varios, define que un registro de una tabla puede estar relacionado con varios registros de la tabla relacionada y viceversa.



Según modalidad

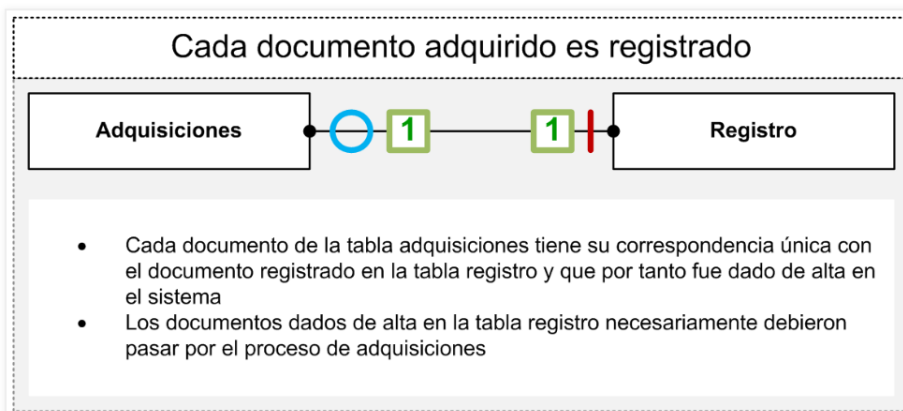
- ✓ **Optativa.** La relación entre un registro de una tabla y varios de la tabla relacionada, puede existir o no.
- ✓ **Obligatoria.** La relación entre un registro de una tabla y otro de la tabla relacionada es obligada, debe existir siempre.

	Centro de Teleinformática y Producción Industrial - Regional Cauca			Pág.17 de40
	Número de Documento:	ENT-WS-DATABASE1	Fecha de Creación: 3/03/2020	Preparado por: Ing. Henry Bastidas
	Nombre del Documento:	Entrenamiento Bases de Datos MySQL		



Ejemplos

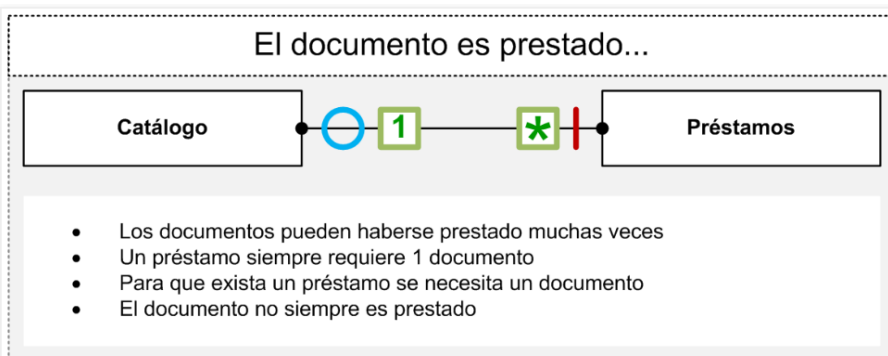
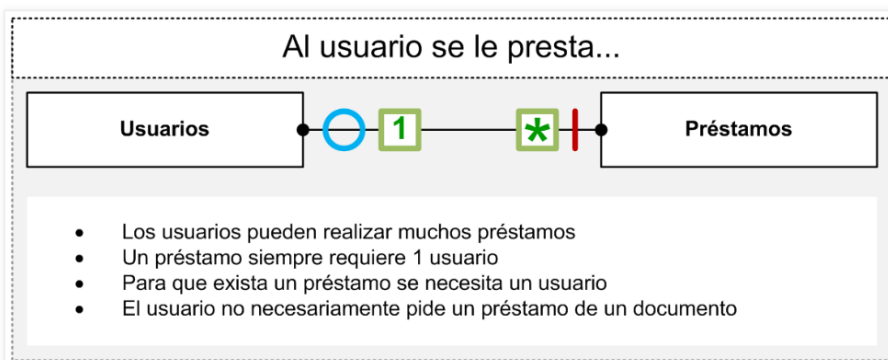
Relación 1 a 1. Cada documento adquirido es registrado podría equipararse a la cardinalidad 1 a 1. Esto significa que cada documento que se introduce en el módulo de adquisiciones (y por ende en su tabla) tiene su correspondencia con los documentos que finalmente se reciben en la biblioteca para ser dados de alta en la tabla registro. De esta forma, puede haber o no documentos en proceso de adquisición (relación optativa). En cambio la tabla registro se encarga de registrar los documentos que se reciben por lo que su relación es de obligatoriedad (todo documento registrado está presente en la tabla de adquisiciones). Todo ello no implica que todos los documentos en fase de adquisición tengan que estar registrados. Pueden existir documentos en fase de adquisición que no hubieran sido registrados.



Relación 1 a *. Los usuarios de una biblioteca suelen solicitar préstamos, por lo tanto, la relación que se produce es de uno a muchos. Un usuario puede pedir o no el préstamo de uno o varios libros o documentos. Por lo tanto, pueden existir uno o ningún usuario solicitando el préstamo, pero para que exista la relación con la tabla préstamos, ésta debe registrar al usuario, su fecha de préstamo y devolución. Véase figura6. Por otra parte, el préstamo se compone no sólo del usuario que lo solicita, sino del documento u objeto que le es prestado, que es el caso que se expone en la figura7. Al igual que en la relación del usuario con la tabla préstamo, un documento puede o no ser prestado. Un documento puede haber sido prestado y devuelto muchas veces y para que exista la relación entre la tabla préstamos y catálogo, debe existir un registro que integre su información (de aquí su modalidad de relación obligatoria). Todo ello conduce a una relación más compleja que puede observarse en la figura8. Lo que en un principio se consideraba una relación de 1 a muchos, termina

	Centro de Teleinformática y Producción Industrial - Regional Cauca			Pág.18 de40
	Número de Documento:	ENT-WS-DATABASE1	Fecha de Creación: 3/03/2020	Preparado por: Ing. Henry Bastidas
	Nombre del Documento:	Entrenamiento Bases de Datos MySQL		

convirtiéndose en una relación de muchos a muchos gracias a una tabla débil o intermedia que almacena la información necesaria del usuario y del documento para poder efectuar el préstamo correspondiente. Por lo tanto, la tabla préstamos, relaciona muchos usuarios con muchos libros en múltiples conjuntos o registros que pueden estar activos o finalizados. Recuérdese que los libros una vez devueltos vuelven a estar disponibles para dar servicio a terceros usuarios. Por ello se concluye que para que un préstamo tenga lugar, deberá estar presente el identificador del usuario y del documento siendo obligatorios en todo proceso de circulación.





Número de Documento:

ENT-WS-DATABASE1

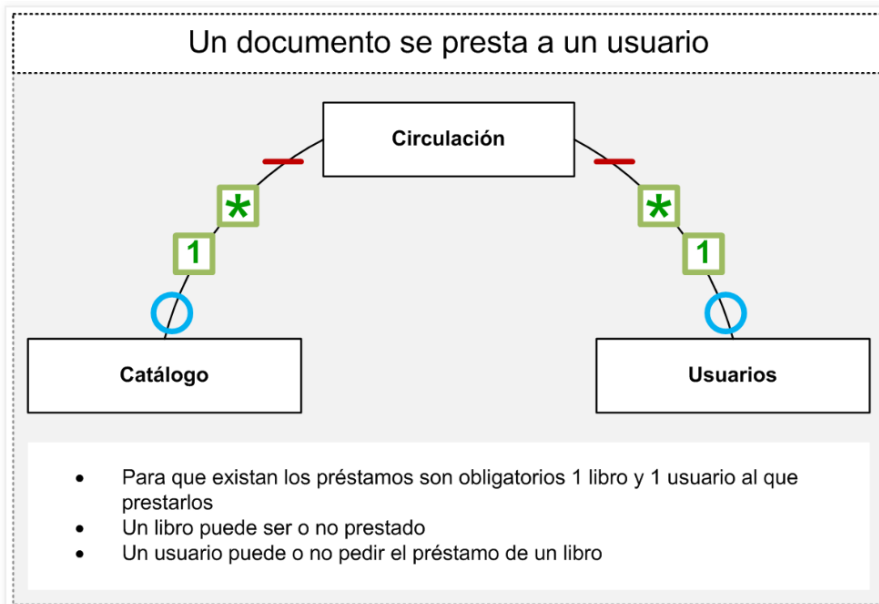
Fecha de Creación:
3/03/2020

Preparado por:

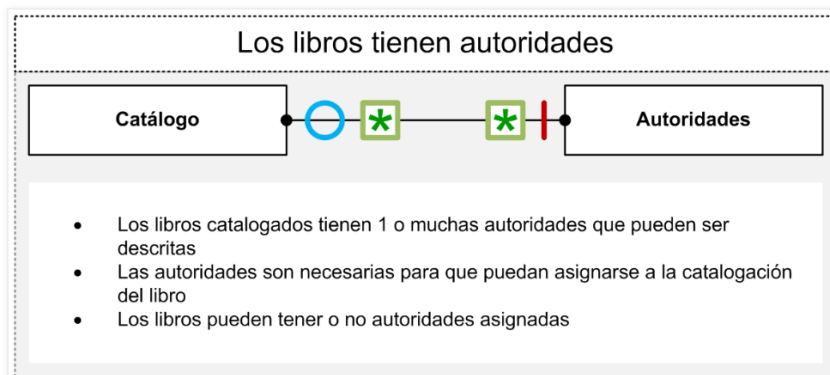
Ing. Henry Bastidas

Nombre del Documento:

Entrenamiento Bases de Datos MySQL



Relación * a *. Cuando se catalogan los documentos en una biblioteca, al seguir las indicaciones de las normas de catalogación, se advierte un apartado de suma importancia; las autoridades. Éstas definen la responsabilidad intelectual, artística, cognitiva, editorial, administrativa, introductoria, del documento. Por ello no es de extrañar que en la catalogación los campos de autoridades sean repetibles, dado que pueden existir 1 o más autoridades. Esta relación es la que se advierte en la figura9. Cada libro puede tener o no 1 o muchas autoridades. Por lo tanto, una autoridad puede estar presente en varios libros o formar parte de varias responsabilidades en el mismo.



8. Acceso a un servidor MySQL y administración de la base de datos por línea de Comandos.

	Centro de Teleinformática y Producción Industrial - Regional Cauca			Pág.20
	Número de Documento:	ENT-WS-DATABASE1	Fecha de Creación:	3/03/2020
	Preparado por:			Ing. Henry Bastidas
	Nombre del Documento:	Entrenamiento Bases de Datos MySQL		

Es muy normal que utilicemos MySQL a través de páginas PHP y para administrar la base de datos utilicemos un programa como PhpMyAdmin, pero a veces no nos queda otro remedio que acceder a la base de datos a través de la línea de comandos.

MySQL tiene un programa, que se llama con el mismo nombre de la base de datos (mysql) que sirve para gestionar la base de datos por línea de comandos. Ese programa, en una instalación de Windows se encuentra en un directorio como.

C:\xampp\mysql\bin\mysql.exe

Conectar con el servidor MySQL

Lo primero que tendremos que hacer es conectar con el sistema gestor de MySQL. Para ello, desde la línea de comandos invocamos a MySQL. Para ello, simplemente tenemos que escribir el comando "mysql" e indicarle unas opciones de conexión.

Con esa sentencia se conecta uno con la base de datos con los parámetros por defecto. Es decir, al servidor local, con usuario y password igual a cadenas vacías.

Lo más normal es que tengamos que indicar algún otro dato para conectar con la base de datos, como el usuario, la clave o la dirección del servidor con el que queremos conectar. La sintaxis sería la siguiente:

```
mysql -h nombre_servidor -u nombre_usuario -p
```

Si deseamos conectarnos a la base de datos en local y con nombre de usuario root tendríamos que escribir:

```
mysql -h localhost -u root -p
```

Lo primero que nos preguntará será el password para el usuario root. Una vez introducida la clave, ya estaremos dentro de la línea de comandos de MySQL. Tenga en cuenta que al inicio de esta capacitación instalamos el XAMP como plataforma pero no incluimos seguridad, el mysql está instalado por defecto por lo que no tiene contraseña, así que bastará solo con presionar la tecla ENTER cuando la solicite, con ello el prompt cambiará a algo como esto:



Número de
Documento:

ENT-WS-DATABASE1

Fecha de Creación:
3/03/2020

Preparado por:
Ing. Henry Bastidas

Nombre del Documento:

Entrenamiento Bases de Datos MySQL

```
C:\xampp\mysql\bin>mysql -h localhost -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 9
Server version: 10.4.11-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>
```

De esta manera ya tenemos acceso a nuestro servidor de base de datos sobre el cual podremos ya iniciar el trabajo con las sentencias y el lenguaje SQL.

Dentro de la consola de MySQL

Una vez dentro, tendremos a nuestra disposición todas las sentencias de MySQL para el trabajo con la base de datos y el lenguaje SQL.

Lo más normal es que primero te tengas que conectar con una base de datos en concreto, de entre todas las que puedes tener creadas en tu servidor MySQL. Eso se hace con el comando use, seguido del nombre de la base de datos que desees conectar.

```
mysql> use mibasedatos;
```

Esto nos conectaría con la base de datos llamada "mibasedatos".

Nota: Hay que fijarse que todas las sentencias dentro de la línea de comandos de MySQL acaban en ";". Si no colocamos el punto y coma, lo más seguro es que no se ejecute el comando y nos vuelva a salir el prompt para que sigamos introduciendo el comando. Si lo que queríamos era ejecutar la sentencia que habíamos escrito antes, con simplemente entrar el ";" será suficiente. Es decir, no debemos escribir de nuevo la sentencia entera, sólo el ";" y volver a apretar "enter".

Si queremos ver una lista de las bases de datos alojadas en nuestro servidor podemos escribir el comando show databases. Así:



```
MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql            |
| performance_schema |
| phpmyadmin       |
| test              |
+-----+
5 rows in set (0.014 sec)

MariaDB [(none)]>
```

9. Comandos SQL: DDL, DML y DCL

DDL: Data Definition Language es la denominación en inglés que determina su sigla, en español se traduce como Lenguaje de Definición de Datos. Muchas veces los pueden llegar a encontrar con el título de OPERACIONES BÁSICAS. Estos comandos permiten crear bases de datos, tablas, vistas, eliminar cualquier objeto, modificar el diseño de los mismos. A manera general, serán vistos como los ingenieros, los cuales permitirán la construcción de la base de datos y el mantenimiento de la misma.

Dentro de los comandos DDL podemos encontrar:

Comando	Descripción
CREATE	Utilizado para crear nuevas tablas, campos e índices.
DROP	Empleado para eliminar tablas e índices
ALTER	Utilizado para modificar las tablas agregando campos o cambiando la definición de los campos

CREATE: Este comando crea un objeto dentro de la base de datos. Puede ser una tabla, vista, índice, trigger, función, procedimiento o cualquier otro objeto que el motor de la base de datos soporte.

Una sentencia CREATE en SQL crea un objeto dentro de un sistema de gestión de bases de datos relacionales (RDBMS).

Los tipos de objetos que se pueden crear depende del RDBMS que esté siendo utilizado, pero la mayoría soportan la creación de tablas, índices, usuarios, sinónimos y bases de datos. Veamos a continuación

	Centro de Teleinformática y Producción Industrial - Regional Cauca			Pág.23 de40
	Número de Documento:	ENT-WS-DATABASE1	Fecha de Creación: 3/03/2020	Preparado por: Ing. Henry Bastidas
	Nombre del Documento:	Entrenamiento Bases de Datos MySQL		

algunos ejemplos del comando CREATE:

Sintaxis y ejemplos del comando CREATE

Objeto a Crear	Sintaxis del Comando
Base de Datos	CREATE DATABASE [IF NOT EXISTS] nombre_BD; Ejemplo: CREATE DATABASE IF NOT EXISTS alumnos;
Tabla	CREATE [TEMPORARY] TABLE [IF NOT EXISTS] nombre_tabla; Ejemplo: CREATE TABLE IF NOT EXISTS nom_alumno;
Índice	CREATE [UNIQUE FULLTEXT SPATIAL] INDEX nombre_index [USING tipo_index] ON nombre_tabla (nom_col_index,...); Ejemplo: CREATE INDEX parte_nombre ON nom_alumno (nombre(10));

Ejemplos:

CREATE DATABASE coleccion; (Crea la base de datos coleccion)

Al crear una tabla de base de datos debemos considerar(Estructura)

```
CREATE TABLE nombre_tabla (
    nombre_columna tipo_columna [ cláusula_defecto ] [ vínculos_de_columna ]
    [ , nombre_columna tipo_columna [ cláusula_defecto ] [ vínculos_de_columna ] ... ]
    [ , [ vínculo_de_tabla ] ... ] )
```

- ✓ Nombre del campo deben empezar por un carácter alfabético y ser lo más comprensible posible para entender que datos se almacenan en ese campo.
- ✓ Tipo del campo (ver tipos de datos en MySQL)
- ✓ Cláusula defecto se le asigna valor por defecto al campo si no se le indica en el momento que se inserta una fila. Puede iniciarse con un valor o nulo.

DEFAULT { valor | NULL }

Vínculos de integridad que se aplica a cada campo

- ✓ NOT NULL no permite valor NULL
- ✓ PRIMARY KEY para indicar que es la clave primaria de la tabla. Puede formarse por más de un campo, directamente debe almacenar valor único y que no sea NULL
- ✓ FOREIGN KEY indica la clave foránea haciendo referencia a otra tabla, estableciendo la relación. Tiene las cláusulas ON DELETE y ON UPDATE indican que acción debe ejecutarse en el caso que la clave foránea (a quién hace referencia) es eliminada o borrada. Las acciones pueden ser:
 - CASCADE: elimina o modifica la tupa que tiene el campo referenciado

	Centro de Teleinformática y Producción Industrial - Regional Cauca			Pág.24 de40
	Número de Documento:	ENT-WS-DATABASE1	Fecha de Creación: 3/03/2020	Preparado por: Ing. Henry Bastidas
	Nombre del Documento:	Entrenamiento Bases de Datos MySQL		

- SET DEFAULT: asigna valor por defecto a la columna referenciada
 - SET NULL: asigna valor NULL a la columna referenciada
- ✓ Control de valor permite asignar o no un valor a la columna dependiendo del resultado de la condición: CHECK { expresión_condicional }

Vínculos de integridad que se pueden aplicar a más campos de la tabla

- Clave primaria PRIMARY KEY (columna1 [, columna2 ...])
- Clave foránea FOREIGN KEY (columna1 [, columna2 ...])

Actividad practica

- ✓ Crear la Base de datos: para el desarrollo de esta práctica vamos a crear una base de datos llamada CentroMedico, cuyo objetivo será realizar una base de datos que maneje la información de un centro médico incluyendo los pacientes, médicos, sus especialidades, citas y demás información que puede ser generada y gestionada en un centro médico.
- Para ello utilizaremos el primer comando de tipo DDL en este caso el CREATE.

```
MariaDB [mysql]> create database CentroMedico;
ERROR 2006 (HY000): MySQL server has gone away
No connection. Trying to reconnect...
Connection id: 11
Current database: mysql

Query OK, 1 row affected (0.007 sec)
```

Podemos verificar si nuestra base de datos fue creada utilizando para ello el comando show databases;



Número de
Documento:

ENT-WS-DATABASE1

Fecha de Creación:
3/03/2020

Preparado por:
Ing. Henry Bastidas

Nombre del Documento:

Entrenamiento Bases de Datos MySQL

```
MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| centromedico |
| information_schema |
| mysql |
| performance_schema |
| phpmyadmin |
| test |
+-----+
6 rows in set (0.001 sec)
```

Efectivamente tenemos creada ya nuestra base de datos, ahora procederemos a crear las tablas que harán parte de la base de datos.

- ✓ El siguiente paso será crear una tabla en este caso la que contendrá la información de pacientes, para ello recuerda que antes debes de elegir la base de datos en la cual vas a trabajar para ello se utiliza el comando USE + NOMBRE DE LA BASE DE DATOS

```
MariaDB [(none)]> use centromedico
Database changed
```

```
CREATE TABLE paciente (
  idPaciente int NOT NULL PRIMARY KEY,
  nombre varchar(50) NULL,
  apellido varchar(50) NULL,
  fNacimiento date NULL,
  domicilio varchar(50) NULL,
  idPais int NOT NULL,
  telefono varchar(20) NULL,
  email varchar(30) NULL,
  observacion varchar(1000) NULL,
  fechaAlta datetime NULL
);
```

El siguiente es el resultado de ejecutar el script:



Número de
Documento:

ENT-WS-DATABASE1

Fecha de Creación:
3/03/2020

Preparado por:
Ing. Henry Bastidas

Nombre del Documento:

Entrenamiento Bases de Datos MySQL

```
MariaDB [(none)]> use centromedico
Database changed
MariaDB [centromedico]> CREATE TABLE paciente(
  -> idPaciente int NOT NULL PRIMARY KEY,
  -> nombre varchar(50) NULL,
  -> apellido varchar(50) NULL,
  -> fNacimiento date NULL,
  -> domicilio varchar(50) NULL,
  -> idPais int NOT NULL,
  -> telefono varchar(20) NULL,
  -> email varchar(30) NULL,
  -> observacion varchar(1000) NULL,
  -> fechaAlta datetime NULL
  -> );
Query OK, 0 rows affected (0.015 sec)

MariaDB [centromedico]>
```

Ahora podemos seguir probando la utilización de algunos otros comandos como por ejemplo DESCRIBE que nos permite ver como es la estructura de la tabla creada.

```
MariaDB [centromedico]> describe paciente;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| idPaciente | int(11) | NO | PRI | NULL | |
| nombre | varchar(50) | YES | | NULL | |
| apellido | varchar(50) | YES | | NULL | |
| fNacimiento | date | YES | | NULL | |
| domicilio | varchar(50) | YES | | NULL | |
| idPais | int(11) | NO | | NULL | |
| telefono | varchar(20) | YES | | NULL | |
| email | varchar(30) | YES | | NULL | |
| observacion | varchar(1000) | YES | | NULL | |
| fechaAlta | datetime | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
10 rows in set (0.008 sec)

MariaDB [centromedico]>
```

Para continuar con el ejercicio crearemos otra tabla llamada país ya que como se observa en la tabla paciente hay un campo o atributo llamado idPais con el cual se debe establecer una relación que permita identificar de que país es el paciente.

Para ello haremos exactamente los mismos pasos del ejercicio anterior, pero para los atributos de la entidad o tabla que llamaremos país.

```
CREATE TABLE pais (
  idPais INT NOT NULL PRIMARY KEY,
  nombre VARCHAR(50)
);
```

Comando DROP: Utilizado para eliminar objetos en la base de datos, Podemos eliminar un índice, una tabla, o la base de datos completa.

	Centro de Teleinformática y Producción Industrial - Regional Cauca			Pág.27 de40
	Número de Documento:	ENT-WS-DATABASE1	Fecha de Creación: 3/03/2020	Preparado por: Ing. Henry Bastidas
	Nombre del Documento:	Entrenamiento Bases de Datos MySQL		

La sintaxis es muy sencilla, solo es utilizar el comando DROP + Objeto + Nombre del Objeto, por ejemplo, si quisiéramos borrar la base de datos llamada prueba el comando a utilizar sería el siguiente:

```
MariaDB [centromedico]> drop database prueba;
Query OK, 0 rows affected (0.002 sec)

MariaDB [centromedico]>
```

Comando ALTER: Utilizado para modificar las tablas agregando o cambiando la definición de los campos, por ejemplo siguiendo el ejemplo de la base de datos CentroMedico decíamos que nos hacía falta indicarle a la tabla paciente que el campo llamado idPais debía ser el que permitiera hacer una relación entre paciente y país de origen por ello modificaremos la estructura de la tabla paciente para agregarle la llave foránea requerida.

```
ALTER TABLE paciente ADD CONSTRAINT idPais FOREIGN KEY (idpais)
REFERENCES pais(idpais);
```

El comando anterior me indica que voy a establecer una llave foránea con el campo país y que la referencia con quien se establecerá la relación es la tabla país con el campo idpais.

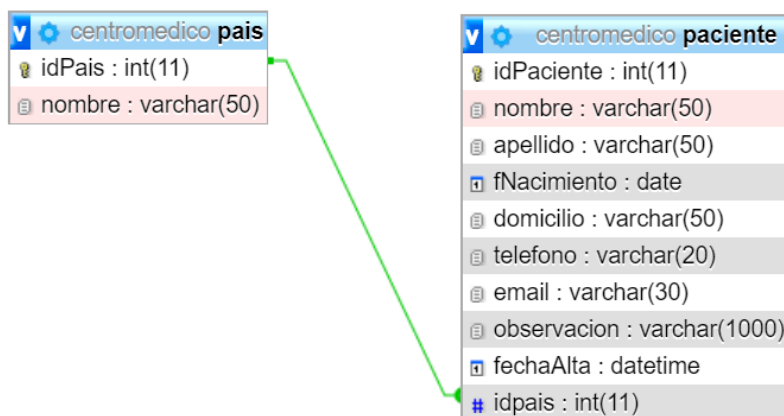
Ahora revisamos de nuevo la estructura de la tabla paciente para validar si efectivamente la llave fue creada.

```
MariaDB [centromedico]> describe paciente;
```

Field	Type	Null	Key	Default	Extra
idPaciente	int(11)	NO	PRI	NULL	
nombre	varchar(50)	YES		NULL	
apellido	varchar(50)	YES		NULL	
fNacimiento	date	YES		NULL	
domicilio	varchar(50)	YES		NULL	
telefono	varchar(20)	YES		NULL	
email	varchar(30)	YES		NULL	
observacion	varchar(1000)	YES		NULL	
fechaAlta	datetime	YES		NULL	
idpais	int(11)	NO	MUL	NULL	

En la vista de diseño podemos observar como la relación entre las dos tablas fue creada de manera exitosa.

	Centro de Teleinformática y Producción Industrial - Regional Cauca			Pág.28 de40
	Número de Documento:	ENT-WS-DATABASE1	Fecha de Creación: 3/03/2020	Preparado por: Ing. Henry Bastidas
	Nombre del Documento: Entrenamiento Bases de Datos MySQL			



Comando TRUNCATE

Este comando permite borrar el contenido de una tabla, es decir elimina los registros que hayan sido previamente ingresado a la tabla.

Para validar el funcionamiento de este comando utilizare como ejemplo la tabla País que contiene los siguientes registros:

```

MariaDB [centromedico]> select * from pais;
+-----+-----+
| idPais | nombre |
+-----+-----+
|      1 | Colombia |
|      2 | Venezuela |
+-----+-----+

```

Como se puede evidenciar la tabla País tiene dos registros con que contienen la información de los países de Colombia y Venezuela.

Ahora haremos uso del comando TRUNCATE y verificaremos de nuevo la tabla para evidenciar el resultado de la ejecución del comando.

```

MariaDB [centromedico]> truncate table pais;
ERROR 1701 (42000): Cannot truncate a table referenced in a foreign key constraint ('centromedico`.`paciente`, CONSTRAINT `idPais` FOREIGN KEY (`idpais`) REFERENCES `centromedico`.`pais` (`idPais`))
MariaDB [centromedico]>

```

Este error que se genera más que un error es una advertencia que nos indica que no podemos borrar los registros de esta tabla por la integridad referencial ya que la información de la tabla País está relacionada con la tabla Paciente.

Para solucionar esto lo que debemos hacer es desactivar de manera temporal la comprobación de tablas externas y una vez ejecutado el TRUNCATE volver a activarlas de la siguiente manera:

```

SET FOREIGN_KEY_CHECKS=0;
TRUNCATE TABLE `YourTableName`;
SET FOREIGN_KEY_CHECKS=1;

```

	Centro de Teleinformática y Producción Industrial - Regional Cauca			Pág.29 de40
	Número de Documento:	ENT-WS-DATABASE1	Fecha de Creación: 3/03/2020	Preparado por: Ing. Henry Bastidas
	Nombre del Documento:	Entrenamiento Bases de Datos MySQL		

De tal manera que se pueda “vaciar” o eliminar el contenido de la tabla País.

```
MariaDB [centromedico]> SET FOREIGN_KEY_CHECKS=0;
Query OK, 0 rows affected (0.000 sec)

MariaDB [centromedico]> truncate table pais;
Query OK, 0 rows affected (0.021 sec)

MariaDB [centromedico]> select * from pais;
Empty set (0.001 sec)

MariaDB [centromedico]>
```

Comando RENAME

Este comando se utiliza cuando se requiera cambiar el nombre de una tabla, se utiliza en conjunto con el comando antes trabajado ALTER TABLE de la siguiente manera:

```
ALTER TABLE pais RENAME TO paises;
```

De esta manera renombramos el nombre de la tabla que anteriormente se llamaba país y cambio a paises.

```
MariaDB [centromedico]> show tables;
+-----+
| Tables_in_centromedico |
+-----+
| paciente               |
| paises                  |
+-----+
```

Sin embargo, al utilizar este comando debemos de tener presente que, si modificamos el nombre, los scripts podrían dejar de funcionar. Hay que tener en cuenta también las relaciones que hay en la tabla, ejemplo si una tabla tiene una clave foránea, generará error.

	Centro de Teleinformática y Producción Industrial - Regional Cauca			Pág.30 de40
	Número de Documento:	ENT-WS-DATABASE1	Fecha de Creación: 3/03/2020	Preparado por: Ing. Henry Bastidas
	Nombre del Documento:	Entrenamiento Bases de Datos MySQL		

10. DML: Data Manipulation Language

Lenguaje Manipulación de Datos) para recuperar y trabajar con datos.

Comando **SELECT**

Utilizado para consultar registros de la base de datos que satisfagan un criterio determinado

Sintaxis:

```
SELECT [ ALL | DISTINCT ] lista_elementos_selección
FROM lista_referencias_tabla
[ WHERE expresión_condicional ]
[ GROUP BY lista_columnas ]
[ HAVING expresión_condicional ]
[ ORDER BY lista_columnas ]
```

Se puede buscar todos los valores de un campo o varios campos de una tabla, o con DISTINCT eliminar aquellas filas cuyo campo se repite. Si no se indica ALL | DISTINCT por defecto se realizará ALL. Nunca se debe usar DISTINCT en la clave primaria, ni en las candidatas ya que de por sí son únicas.

WHERE se utiliza para generar una condición que debe cumplirse, aquellos campos que no la cumplan no se seleccionarán.

GROUP BY para agrupar en una sola línea el campo o campos especificados.

HAVING del resultado de GROUP BY elimina las que no cumplan su expresión condicional.

ORDER BY para ordenar el orden de las columnas de formas ascendente (ASC) o descendente (DESC). Por defecto el orden se hace de forma ascendente.

Además de la sintaxis, podemos encontrarnos con consultas que utilicen las siguientes funciones de agregado: COUNT, SUM, AVG, MAX, MIN.

Operadores lógicos: AND, OR, NOT.

Operadores de comparación: >, <, =, !=, !, BETWEEN para intervalo de valores, LIKE para comparación entre cadenas de caracteres con los pattern '%' y '_'.

IN para especificar registros de una base de datos.

Para ver un ejemplo, vamos a crear una base de datos llamada viedojuegos y a ver las tablas de nuestra base de datos agregando campo STOCK a la tabla CONSOLA_JUEGO, para saber cuántas unidades hay de cada juego y por consola.



Número de
Documento:

ENT-WS-DATABASE1

Fecha de Creación:
3/03/2020

Preparado por:

Ing. Henry Bastidas

Nombre del Documento:

Entrenamiento Bases de Datos MySQL

CONSOLA

ID_CONSOLA	NOMBRE	TIPO	MARCA
00001	PS4	Sobremesa	Sony
00002	XBOX ONE	Sobremesa	Microsoft
00003	3DS	Portátil	Nintendo

JUEGO

ID_JUEGO	TITULO	GENERO
0001	Darks Souls III	ROL
0002	Project Cars	Conducción
0003	Metal Gear Solid V	Acción
0004	Killer Instinct	Lucha
0005	Metroid Prime	Acción
0006	Dragon Quest VII	ROL

CONSOLA_JUEGO

ID_CONSOLA	ID_JUEGO	STOCK
0001	0001	45
0001	0002	30
0001	0003	15
0002	0003	15
0002	0004	19
0003	0005	34
0003	0006	23

Consulta: Mostrar los tipos de consolas que hay registrados

```
SELECT DISTINCT tipo
```

```
FROM consola
```

Consulta: Nombre de los juegos que sólo están en 3DS

```
SELECT titulo
```

	Centro de Teleinformática y Producción Industrial - Regional Cauca			Pág.32 de40
	Número de Documento:	ENT-WS-DATABASE1	Fecha de Creación: 3/03/2020	Preparado por: Ing. Henry Bastidas
	Nombre del Documento:	Entrenamiento Bases de Datos MySQL		

```
FROM juego
WHERE juego.id_juego IN(
SELECT id_juego
FROM consola_juego
WHERE consola_juego.id_consola IN(
SELECT id_consola
FROM consola
WHERE consola.nombre='3DS')
)
```

Consulta: Stock total de aquellos juegos cuyo género sea de Acción

```
SELECT SUM(consola_juego.stock)
FROM consola_juego
WHERE consola_juego.id_juego IN(
SELECT id_juego
FROM juego
WHERE juego.genero='accion')
```

Consulta: Titulo de aquellos juegos con stock total que supere las 20 unidades

```
SELECT juego.titulo
FROM juego
WHERE juego.id_juego IN(
SELECT id_juego
FROM consola_juego
GROUP BY id_juego
HAVING SUM(consola_juego.stock) > 20)
```

Comando INSERT

Utilizado para cargar lotes de datos en la base de datos en una única operación

Sintaxis:

```
INSERT INTO table_name VALUES (value1,value2,value3,...);
```

Ejemplos:

Vamos a adicionar los valores mostrados en las tablas de la base de datos videojuegos como se muestran en la gráfica anterior.

	Centro de Teleinformática y Producción Industrial - Regional Cauca			Pág.33 de40
	Número de Documento:	ENT-WS-DATABASE1	Fecha de Creación: 3/03/2020	Preparado por: Ing. Henry Bastidas
	Nombre del Documento:	Entrenamiento Bases de Datos MySQL		

Insertando nombres y marcas de consolas de videojuegos

```
INSERT INTO consola (nombre, marca) VALUES ('PS4','Sony');
INSERT INTO consola (nombre, marca) VALUES ('XBOX ONE','Microsoft');
INSERT INTO consola (nombre, tipo, marca) VALUES ('3DS','Portatil','Nintendo');
```

Agregando título y género en los tipos de juegos

```
INSERT INTO juego (titulo, genero) VALUES ('Dark Souls III', 'ROL');
INSERT INTO juego (titulo, genero) VALUES ('Project Cars', 'Conduccion');
INSERT INTO juego (titulo, genero) VALUES ('Metal Gear Solid V', 'Accion');
INSERT INTO juego (titulo, genero) VALUES ('Killer Instinct', 'Lucha');
INSERT INTO juego (titulo, genero) VALUES ('Metroid Prime', 'Accion');
INSERT INTO juego (titulo, genero) VALUES ('Dragon Quest VII', 'ROL');
```

Agregando stock en las tablas de juegos y por consola

```
INSERT INTO consola_juego (id_consola, id_juego, stock) VALUES (1,1, 45);
INSERT INTO consola_juego (id_consola, id_juego, stock) VALUES (1,2, 30);
INSERT INTO consola_juego (id_consola, id_juego, stock) VALUES (1,3, 15);
INSERT INTO consola_juego (id_consola, id_juego, stock) VALUES (2,3, 15);
INSERT INTO consola_juego (id_consola, id_juego, stock) VALUES (2,4, 19);
INSERT INTO consola_juego (id_consola, id_juego, stock) VALUES (3,5, 34);
INSERT INTO consola_juego (id_consola, id_juego, stock) VALUES (3,6, 23);
```

De esta manera quedan llenas nuestras tablas con las cuales podemos hacer las pruebas de los comandos SELECT y probar el resultado de las consultas antes diseñadas.

Comando **UPDATE**

Utilizado para modificar los valores de los campos y registros especificados

Sintaxis:

UPDATE table_name

SET column1=value1, column2=value2,...

WHERE some_column=some_value;

Ejemplo:

	Centro de Teleinformática y Producción Industrial - Regional Cauca			Pág.34 de40
	Número de Documento:	ENT-WS-DATABASE1	Fecha de Creación: 3/03/2020	Preparado por: Ing. Henry Bastidas
	Nombre del Documento:	Entrenamiento Bases de Datos MySQL		

UPDATE juego SET titulo='The Legend of Zelda' WHERE titulo='Metroid Prime';

Comando **DELETE**

Utilizado para eliminar registros de una tabla de una base de datos

Sintaxis:

DELETE FROM nombretabla WHERE condicion

Ejemplo:

DELETE FROM juego WHERE id_juego=5;

Comando **MERGE**

Cuando se quiere actualizar/insertar datos de una tabla, actualiza cuando se cumple la condición e inserta cuando no se cumple.

Sintaxis:

MERGE INTO [tabla_destino]

USING ([tabla o vista o consulta])

ON ([condición de existencia de registro])

WHEN MATCHED THEN [sentencia de actualización]

WHEN NOT MATCHED THEN [sentencia de inserción];

Ejemplo:

Para este ejemplo creamos la tabla consola_temporal con los siguientes valores.

```
INSERT INTO consola_temporal(nombre, marca) VALUES('PS4','Sony');
INSERT INTO consola_temporal(nombre, marca) VALUES('XBOX ONE','Microsoft');
INSERT INTO consola_temporal(nombre, tipo, marca) VALUES('NEW 3DS','Portatil','Nintendo');
INSERT INTO consola_temporal(nombre, tipo, marca) VALUES('PSVITA','Portatil','Sony');
```

	Centro de Teleinformática y Producción Industrial - Regional Cauca			Pág.35 de40
	Número de Documento:	ENT-WS-DATABASE1	Fecha de Creación: 3/03/2020	Preparado por: Ing. Henry Bastidas
	Nombre del Documento:	Entrenamiento Bases de Datos MySQL		

CONSOLA_TEMPORAL

ID_CONSOLA	NOMBRE	TIPO	MARCA
00001	PS4	Sobremesa	Sony
00002	XBOX ONE	Sobremesa	Microsoft
00003	NEW3DS	Portátil	Nintendo
00004	PSVITA	Portátil	Sony

```

MERGE INTO consola C
USING (SELECT * FROM consola_temporal) CT
ON(C.id_consola=CT.id_consola)
WHEN MATCHED THEN UPDATE SET C.nombre=CT.Nombre
WHEN NOT MATCHED THEN insert (nombre, tipo, marca)
VALUES (CT.nombre, CT.tipo,CT.marca);

```

Este sería el resultado de la ejecución del comando anterior.

id_consola	nombre	tipo	marca
1	PS4		Sony
2	PS4	(NULL)	Sony
3	XBOXONE	(NULL)	Microsoft
4	3DS	Portatil	Nintendo
5	PS4	(NULL)	Sony
6	XBOXONE	(NULL)	Microsoft
7	3DS	Portatil	Nintendo

11. PROCEDIMIENTOS ALMACENADOS

Un procedimiento almacenado MySQL no es más que una porción de código que puedes guardar y reutilizar. Es útil cuando repites la misma tarea repetidas veces, siendo un buen método para encapsular el código. Al igual que ocurre con las funciones, también puede aceptar datos como parámetros, de modo que actúa en base a éstos.

	Centro de Teleinformática y Producción Industrial - Regional Cauca			Pág.36 de40
	Número de Documento:	ENT-WS-DATABASE1	Fecha de Creación: 3/03/2020	Preparado por: Ing. Henry Bastidas
	Nombre del Documento:	Entrenamiento Bases de Datos MySQL		

Para poder crear un procedimiento almacenado es necesario que tengas permisos INSERT y DELETE sobre la base de datos.

Sintaxis:

```
CREATE PROCEDURE nombre_procedimiento
AS
sentencias_sql
GO;
```

Para ejecutar un procedimiento almacenado lo invocamos así:

```
EXEC nombre_procedimiento (param1, param2, ....);

CALL nombre_procedimeinto(param1, param2, ....);
```

Parámetros

Como has visto, los parámetros se definen separados por una coma. Los parámetros de los procedimientos almacenados de MySQL pueden ser de tres tipos:

IN: Es el tipo de parámetro que se usa por defecto. La aplicación o código que invoque al procedimiento tendrá que pasar un argumento para este parámetro. El procedimiento trabajará con una copia de su valor, teniendo el parámetro su valor original al terminar la ejecución del procedimiento.

OUT: El valor de este parámetro puede ser cambiado en el procedimiento, y además su valor modificado será enviado de vuelta al código o programa que invoca el procedimiento.

INOUT: Es una mezcla de los dos conceptos anteriores. La aplicación o código que invoca al procedimiento puede pasarle un valor a éste, devolviendo el valor modificado al terminar la ejecución.

La ventaja de un procedimiento almacenado es que podemos crearlo en cualquier editor de texto e incluso en el servidor, es ejecutado por el motor de bases de datos y no es accesible a los usuarios sino solo al administrador.

Un procedimiento almacenado (Store Procedure), es un pequeño algoritmo en lenguaje SQL que se almacena junto a la base de datos y permite realizar tareas sobre estos datos.

	Centro de Teleinformática y Producción Industrial - Regional Cauca			Pág.37
	Número de Documento:	ENT-WS-DATABASE1	Fecha de Creación:	3/03/2020
	Preparado por:			Ing. Henry Bastidas
	Nombre del Documento:	Entrenamiento Bases de Datos MySQL		

Las Ventajas de los procedimientos almacenados son:

- ✓ Se pueden acceder desde distintos programas externos, si la necesidad de hacer pública la estructura de la base de datos.
- ✓ Se pueden reutilizar por lo tanto ganaremos tiempo al estar ya programado y testeado.

Utilizar procedimientos mysql almacenados, con phpmyadmin

Iniciaremos el trabajo de crear y consultar procedimientos almacenados con la herramienta phpmyadmin, pero puede utilizarse cualquiera que soporte consultas SQL a partir de MYSQL 5.0

En este caso tomaremos la base de datos de prueba llamada videojuegos con la que veníamos trabajando.

Existen 2 tipo motores que manejan datos en Mysql

MyISAM: motor por defecto, muy rápido para consultas, no provee integridad de datos, ni protección referencial es Ideal para sistemas con muchas consultas

InnoDB: provee protección referencial e integridad de datos además de bloqueo de registros, ideal si se va a insertar, editar o eliminar mucha información constantemente. Generalmente para procedimientos almacenados es mejor utilizar InnoDB.

Ejemplos:

Vamos a desarrollar a continuación un procedimiento almacenado para cada transacción consultar, insertar, modificar y eliminar un registro.

Podemos utilizar Phpmyadmin o un gestor como HeidiSQL que es gratuito y funciona en Windows o Linux.

Creamos un procedimiento almacenado para consultar la tabla consola:

```
DELIMITER //
CREATE PROCEDURE pa_listainmuebles()
BEGIN
SELECT * FROM inmuebles;
END//
DELIMITER ;
```

De esta manera se crea el procedimiento almacenado, si accedemos a nuestro servidor de MySQL desde phpMyAdmin seleccionamos la pestaña Rutinas y en ese espacio encontraran los procedimientos que vayan siendo creados.



Número de Documento:

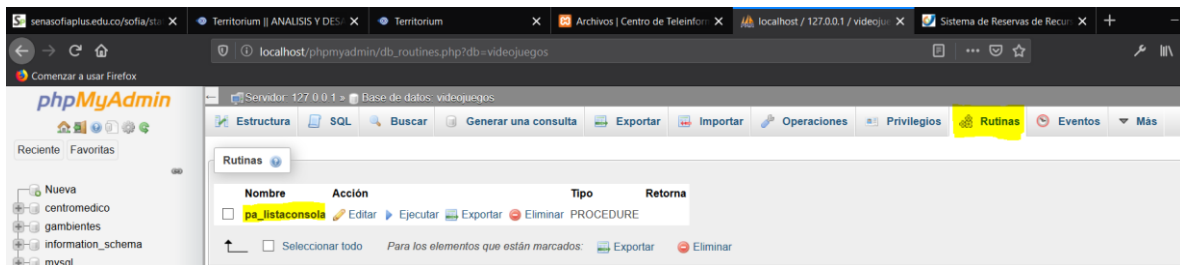
ENT-WS-DATABASE1

Fecha de Creación:
3/03/2020

Preparado por:
Ing. Henry Bastidas

Nombre del Documento:

Entrenamiento Bases de Datos MySQL



MYSQL entiende que una sentencia finaliza con un punto y coma. La sentencia DELIMITER cambia el carácter de finalización por cualquier otro carácter, por convención se utiliza // para indicar el fin del procedimiento almacenado para que MySQL no finalice el procedimiento almacenado al encontrar el primer punto y coma.

Podemos ir a la **pestaña Rutinas** para ver cada transacción que hayamos creado y desde allí podremos modificar, ejecutar, exportar o eliminar el código.

Para ejecutar un procedimiento almacenado usamos el comando CALL desde la pestaña SQL o también desde un lenguaje de programación como .NET o Java. A continuación, invocamos el procedimiento almacenado creado con el comando.

```
CALL pa_listaconsola();
```

Resultado de la ejecución del procedimiento

consola (7r x 4c)			
id_consola	nombre	tipo	marca
1	PS4		Sony
2	PS4	(NULL)	Sony
3	XBOXONE	(NULL)	Microsoft
4	3DS	Portatil	Nintendo
5	PS4	(NULL)	Sony
6	XBOXONE	(NULL)	Microsoft
7	3DS	Portatil	Nintendo

A continuación, crearemos un procedimiento almacenado para insertar una nueva consola, para esto necesitaremos parámetros tipo IN es decir que al procedimiento almacenado le asignaremos datos y variables de entrada para hacer alguna transacción, en este caso guardarlos en la base de datos.

	Centro de Teleinformática y Producción Industrial - Regional Cauca			Pág.39 de40
	Número de Documento:	ENT-WS-DATABASE1	Fecha de Creación: 3/03/2020	Preparado por: Ing. Henry Bastidas
	Nombre del Documento:	Entrenamiento Bases de Datos MySQL		

DELIMITER //

```
CREATE PROCEDURE pa_nuevaconsola (
IN nombre VARCHAR(30) ,
IN tipo VARCHAR(30),
IN marca VARCHAR(50)
)
BEGIN

INSERT INTO consola(nombre, tipo, marca ) VALUES (nombre,tipo,marca);

END//

DELIMITER ;
```

Luego podemos ejecutar el procedimiento almacenado invocando y asignado los parámetros.

```
CALL pa_nuevaconsola('XBOXTN','portatil','POLLITO');
```

Al hacer el llamado del procedimiento anterior se puede evidenciar que el nuevo registro efectivamente se insertó.

```
1 CALL pa_nuevaconsola('XBOXTN','portatil','POLLITO');
2
3 CALL pa_listaconsola();
4
```

id_consola	nombre	tipo	marca
1	PS4		Sony
2	PS4	(NULL)	Sony
3	XBOXONE	(NULL)	Microsoft
4	3DS	Portatil	Nintendo
5	PS4	(NULL)	Sony
6	XBOXONE	(NULL)	Microsoft
7	3DS	Portatil	Nintendo
8	XBOXTN	portatil	POLLITO
9	XBOXTN	portatil	POLLITO

A continuación, crearemos el procedimiento almacenado para editar una consola desde el editor de Phpmyadmin, en este caso solo modificaremos el nombre.

	Centro de Teleinformática y Producción Industrial - Regional Cauca			Pág.40 de40
	Número de Documento:	ENT-WS-DATABASE1	Fecha de Creación: 3/03/2020	Preparado por: Ing. Henry Bastidas
	Nombre del Documento:	Entrenamiento Bases de Datos MySQL		

DELIMITER //

```
CREATE DEFINER=root@localhost
PROCEDURE pa_editarconsola(
IN nombrenuevo VARCHAR(30),
IN id_consolan INT(11))
BEGIN
UPDATE consola SET nombre=nombrenuevo WHERE idconsola=id_consolan;
END//
```

DELIMITER ;

Ahora llamaremos el procedimiento y probaremos el resultado de su ejecución, para este ejemplo le cambiaremos el nombre al registro con id 2 en la tabla consola.

```
call pa_editarconsola('XBOSTREE',2);
```

nuevamente probamos que el procedimiento almacenado halla hecho lo solicitado.

```
CALL pa_listaconsola();
```

consola (9r x 4c)			
id_consola	nombre	tipo	marca
1	PS4		Sony
2	XBOSTREE	(NULL)	Sony
3	XBOXONE	(NULL)	Microsoft

Podemos realizar igual procedimientos con parámetros OUT, como ejemplo crearemos un procedimiento que me de el número de consolas según la marca.

```
DELIMITER //
CREATE PROCEDURE contarconsolaPornombre(
IN nombrec VARCHAR(30),
OUT numero INT)
BEGIN
SELECT count(id_consola)
INTO numero
FROM consola
WHERE nombre = nombrec;
END//
DELIMITER ;
```

Ahora lo ejecutaremos como los anteriores

```
CALL contarconsolaPornombre('sony', @numero);
```