



Desarrollo de software

Semestre 04 (Licenciatura)

Semestre 05 (TSU)

Programa de la unidad didáctica:
Administración de Base de Datos

Unidad 2. Operaciones básicas

Clave:

Licenciatura
15142422

TSU
16142523

Ciudad de México, enero de 2025

Universidad Abierta y a Distancia de México





Índice

Presentación de la unidad	3
Logros de la unidad	3
Competencia específica.....	3
Temario de la unidad	4
Unidad 2. Operaciones básicas	5
2.1. Data Definition Language (DDL)	5
2.2. Data Manipulation Language (DML).....	12
2.3. Data Control Language (DCL).....	14
Cierre de la unidad	15
Para saber más.....	16
Fuentes de consulta	16



Presentación de la unidad

Bienvenido(a) a la segunda unidad de la unidad didáctica de Administración de Bases de Datos. Actualmente, las bases de datos son utilizadas para administrar la información de las empresas. En muchas ocasiones, no es suficiente con que la información simplemente esté almacenada; es necesario poder manipularla. Esto implica realizar operaciones específicas en los datos, como observar y extraer el contenido de una base de datos, es decir, las tablas y los atributos especificados en cada una de ellas.

El administrador de bases de datos frecuentemente necesita actualizar algún dato en el sistema o, en algunos casos, eliminar una tabla que ya no es necesaria para la empresa. En esta unidad, identificarás las operaciones básicas para administrar bases de datos, y aprenderás sobre la definición del lenguaje de datos, incluyendo cómo mostrar, modificar, reparar y borrar tablas, entre otras funciones. Además, aplicarás los comandos utilizados para la manipulación de datos, como seleccionar, insertar, actualizar y borrar datos específicos, así como ejecutar consultas según las necesidades de la empresa.

Finalmente, en esta unidad se presentarán los comandos del lenguaje de control de datos, con los cuales aprenderás a otorgar o cancelar permisos de acceso, así como a crear usuarios y roles dentro de la base de datos. Como has podido notar, en esta unidad conocerás los comandos básicos que, como administrador de bases de datos, utilizarás. Saber manipular correctamente la información en las bases de datos es una herramienta importante que te servirá en materias posteriores, como programación o diseño web, donde también se aplican bases de datos.

Logros de la unidad

En esta unidad lograrás:

- Identificar las funciones básicas de MySQL para almacenar datos.
- Crear bases de datos con MySQL para la manipulación de la información.
- Ejecutar las funciones de inserción, consulta y búsqueda desde MySQL.

Competencia específica

Utilizar MySQL para manipular información mediante las operaciones básicas de inserción, consulta y búsqueda de datos.



Temario de la unidad

2. Operaciones básicas

2.1. Data Definition Language (DDL)

- 2.1.1. Create, database y create table
- 2.1.2. Show database y show table
- 2.1.3. Alter table, Repair table, Rename table
- 2.1.4. Drop table, truncate table y drop database
- 2.1.5. Check table y Optimize table

2.2. Data Manipulation Language (DML)

- 2.2.1. Select
- 2.2.2. Insert
- 2.2.3. Update
- 2.2.4. Delete

2.3. Data Control Language (DCL)

- 2.3.1. Grant
- 2.3.2. Revoke

Unidad 2. Operaciones básicas

En la unidad anterior identificaste: cuáles son las funciones específicas de un *administrador de bases de datos* y cómo es su relación con otras áreas; hiciste un análisis de los manejadores de bases de datos existentes en el mercado y la forma de administrar los datos; revisaste el proceso de instalación del sistema gestor llamado MySQL, sus requisitos, paquetes, el procedimiento y configuración para Windows y Linux; y el proceso de la administración del disco duro partiendo de un análisis, la segmentación del disco y preparar el espacio del mismo.

Una vez que has aprendido y practicado todo lo anterior, el siguiente paso es, a partir de un contexto de procesamiento de información, obtener un modelo y codificarlo en un gestor. Debes ser capaz de considerar y manipular la información para garantizar que siempre esté disponible, todo ello utilizando un Sistema Gestor de Bases de Datos. Para el desarrollo de esta materia, utilizaremos MySQL.

El primer Tema 2.1. Está enfocado al lenguaje de definición de datos (DDL), éste estará apoyado en: la página oficial de MySQL; en los libros de Coronel, C (2011), Nevado, V. (2010) y Stanek, W (2007).

Posteriormente, el Tema 2.2. referente al *lenguaje de manipulación de datos*, lo podrás revisar directamente la página oficial del MySQL, donde encontrarás a profundidad la descripción y uso de



estos comandos.

Por último, en el Tema 2.3. *Lenguaje de control de datos (DCL)*, nos apoyaremos en la página oficial de MySQL. Todos estos temas están enfocados a cumplir con el logro de la competencia en donde podrás utilizar MySQL para manipular la información mediante las operaciones básicas de inserción consulta y búsqueda.

2.1. Data Definition Language (DDL).

El DDL (Lenguaje de Definición de Datos) es el lenguaje con el que podemos comenzar la creación de bases de datos, tablas y sus relaciones. Aunque puede variar de un lenguaje a otro, en este caso se utilizará el lenguaje estándar ANSI-ISO SQL (Structured Query Language), que es el utilizado en MySQL. Con el uso de DDL, podemos definir las estructuras que seguirán los datos.

Durante el desarrollo de este tema nos apoyaremos con varios textos tales como: Nevado, V. (2010), Coronel, C. (2011), Stanek, W. (2007) y con en el sitio oficial de MySQL; en los cuales, durante el desarrollo de esta unidad, te podrás apoyar.

Los comandos que usaremos para la primera parte de la definición de datos (DDL) son: CREATE, SHOW, DROP, TRUNCATE, CHECK Y OPTIMIZE; en los libros mencionados para este tema, revisarás a detalle dichos comandos y cómo se aplican en cada uno para la definición de los datos.

Para comenzar, te invitamos a **leer** el *tema 2.15 Lenguaje de definición de datos (DDL)* en: Nevado, V.(2010, pp. 36-37). En dicho texto encontrarás: el punto de vista lógico y el del administrador, las estructuras físicas o lógicas y el crecimiento de la información o de la estructura de la misma.

Una vez que se tienen: el análisis completo de la información que va a contener la base de datos, la forma en que va a estar organizada, cómo quedarán los atributos de las tablas y las relaciones entre esas tablas, el siguiente paso para el análisis y diseño de bases de datos, es normalizar la base de datos para su codificación; muchos de los comandos que se usan en la codificación ya los revisaste en la materia de diseño de bases de datos, pero es importante que, como administrador de una base de datos, los domines para que cuando se presenten problemas, identifiques las herramientas para resolverlos.

Como menciona Camuña (2015 pp.44) las instrucciones DDL pueden:

- Crear y definir tablas
- Eliminar tablas
- Modificar una tabla creada previamente
- Crear vistas
- Crear índices
- Controlar el almacenamiento físico de los datos



Create Database y create table

Antes de comenzar con el uso de la base de datos, es indispensable **leer** Coronel, C. (2011, p. 225), el tema 7.2.2 *Creación de la base de datos*. En este texto, se presentan algunos aspectos importantes que se deben considerar antes de crear una base de datos, las siglas RDBMS que ahí son usadas se refieren a (SISTEMAS RELACIONALES DE BASES DE DATOS). Una vez que ya tienes los antecedentes de cómo diseñar la base de datos, el siguiente paso es crearla, para lo cual el comando a usar es el:

Create Database

Para conocer más el uso del comando CREATE DATABASE **lee** en la siguiente dirección de internet, perteneciente a la página oficial del MySQL: <https://downloads.mysql.com/docs/refman-5.0-es.pdf> En esta página, te darás cuenta de que es muy simple crear una base de datos es muy simple y sólo requiere de pocas acciones. Un ejemplo para crear la base de datos llamada DIRECTORIO basta con escribir en la línea de comandos de MySQL CREATE DATABASE DIRECTORIO.

Una vez creada la base de datos, el siguiente paso es crear las tablas, por lo tanto, **complementa** tu información con la lectura de Coronel, C. (2011, p. 226-228), el tema 7.2.4 *Tipos de datos*, en donde se explican los tipos de datos que se permiten en SQL. En la página 227, se muestra una tabla que contiene un diccionario de datos que es la estructura, descripción y forma de los datos, así como su tipo. En la página 228 hay una tabla de los tipos comunes de datos para SQL.

Enriquece tu actividad leyendo a Stanek, W. (2007, pág. 269-271), en el ejemplo 9-4 *Sintaxis y uso de CREATE TABLE*, en donde usando los tipos de datos y los atributos de las tablas, podrás crear una tabla usando el comando *create y table*. A continuación, aplicaremos un ejemplo:

Este ejemplo incluye uso de las instrucciones *CREATE DATABASE Y CREATE TABLE* para que veas el uso de las mismas.



Aplicación de comandos *CREATE DATABASE* Y *CREATE TABLE*

La empresa “EL DESARROLLO FELIZ” se dedica al desarrollo de software con la realización de proyectos. Esta empresa desea realizar una base de datos para el control de los proyectos, además de delegar a quien los analiza y quien los desarrolla o codifica, para lo cual considera almacenar datos de los analistas; los datos que recabará serán su dni, nombre, dirección, email. Se consideran también almacenar información de los programadores, por lo tanto, se recabará su dni, nombre, dirección, email. Esta empresa desarrolladora de software trabaja a través de proyectos de los que nos interesa guardar: código, nombre, costo y duración del proyecto; dichos proyectos son realizados para clientes de los que nos interesa conocer: su dni, nombre, dirección y teléfono. La base de datos debe considerar los siguientes puntos:

- En un proyecto puede participar un sólo analista y un analista puede participar en varios proyectos.
- En un proyecto pueden participar varios programadores y un programador puede participar en varios proyectos.
- Un proyecto pertenece a un sólo cliente, pero un cliente puede solicitar varios proyectos.

```
create database Desarrolladora;
```

```
create table analista
```

```
(  
dni int primary key not null, nombre varchar(60), direccion varchar(30), email varchar(30)  
);
```

```
create table Programador(  
dni int primary key not null, nombre varchar(60), direccion varchar(30), email varchar(30)
```

```
);
```

```
create table Proyecto(  
);
```



```
codigo int primary key not null,  
nombre varchar(60),  
costo float,  
duracion int,  
analista int foreign key references Analista(dni),  
cliente int foreign key references Cliente(dni)  
);  
  
create table Cliente  
(  
dni int primary key not null,  
nombre varchar(60),  
direccion varchar(50),  
telefono varchar(10)  
);  
  
create table Programadores_Proyecto  
(  
codigo int foreign key references Proyecto(codigo),  
programador int foreign key references Programador(dni)  
);
```

En este ejemplo, creamos la base datos, llamada *desarrolladora*, y 5 tablas llamadas: *analista*, *programador*, *proyecto*, *cliente*, *programadores_proyecto*. Observa que las tablas tienen las llaves primarias y las llaves foráneas que les corresponden; también aplican los tipos de datos que van a usar para llegar a este punto, debes recordar la materia de análisis y diseño en la que viste porqué se asignan llaves foráneas y primarias y porqué sale una tabla nueva producto de las relaciones.

Existen más comandos DDL tales como:

Show database y show table

Los comandos show database y show table se usan para mostrar la información, ya sea de la base de datos o de la tabla y su forma de usarlo:

En el portal de MySQL encontrarás el comando Show database <https://downloads.mysql.com/docs/refman-5.0-es.pdf>, en el que te darás cuenta, que es un comando muy sencillo, el cual se ejecuta y muestra la base dedatos que estás usando.

Ahora, **revisa** esta dirección, para ver una explicación al comando show table: <https://downloads.mysql.com/docs/refman-5.0-es.pdf>

Ejecuta estos comandos para las bases de datos que acabas de capturar y para sus tablas y ve los resultados.



Y podrás darte cuenta de que muestra una salida como ésta:

```
mysql> show tables;
+-----+
| Tables_in_hospital |
+-----+
| ingreso            |
| medico            |
| paciente          |
+-----+
3 rows in set (0.00 sec)
```

Alter table, repair table, rename table

Otros comandos importantes que debes considerar como administrador son el *Alter table*, *repair table* y *rename table* los cuales se aplican directamente a las tablas.

El comando *alter table* es un comando que en pocas ocasiones se usa ya que, como su nombre lo dice, sirve para modificar la estructura de una tabla existente y eso implica un previo análisis muy detallado de lo que se quiere modificar. En el sitio oficial de MySQL, podrás encontrar la forma de ejecutar los comandos, por lo tanto, ingresa a: <https://downloads.mysql.com/docs/refman-5.0-es.pdf>. Por ejemplo, con el uso de este comando, podrás añadir o borrar columnas, crear o destruir índices, cambiar el tipo de columnas existentes o renombrar columnas o la misma tabla. Puede cambiar el comentario de la tabla y su tipo.

Un ejemplo aplicado a nuestra base de datos, que ya tenemos capturada, sería:

```
alter table Programadores_Proyecto add fecha_incorporacion datetime.
```

En este ejemplo pudiste observar que a la tabla *Programadores_Proyecto* le estamos agregando el atributo *fecha_incorporacion* y es del tipo *datetime*.

Ejecuta el ejemplo aplicándolo a la base de datos que tenemos.

El comando *repair table* es un comando que no debe usarse más que en casos extremos, sólo se usa si la información está muy dañada, ya que sí es capaz de recuperar la información, pero, regresa también mucha información.

Ahora, **ingresa** al siguiente link para consultar la forma de usarlo:

<https://downloads.mysql.com/docs/refman-5.0-es.pdf>

Revisa el link para que tengas una referencia de lo que puedes hacer por tus bases de datos si en algún momento son dañadas, e identifica cómo lo usarías en esta base de datos, recuerda que antes



de usar este comando es recomendable hacer una copia de respaldo, pero es una herramienta útil que servirá para recuperar las tablas cuando se han dañado, regresándolas a un estado anterior.

El comando *rename table* es de lo más simple, únicamente cambia el nombre de la tabla. En el siguiente link encontrarás la forma de usarlo:

<https://downloads.mysql.com/docs/refman-5.0-es.pdf>

Lee el texto y **observa** el siguiente ejemplo de cómo se aplicaría para cambiar el nombre de una tabla:

```
Rename table Programadores_Proyecto to program_proy.
```

Es importante que conozcas esta lista de comandos como administrador y los apliques cuando sea necesario aplicarlos. Desde luego, depende su aplicación al enfoque de cada uno.

Drop table, truncate table y drop database

Estas tres instrucciones implican borrado, pero un borrado de diferentes formas, dependiendo del comando y las condiciones ya sea de una tabla, base de datos o su estructura:

El comando DROP aparte de eliminar los datos, también sirve para eliminar la estructura de la tabla. DROP TABLE nombre_tabla: es idéntico el comando al anterior, pero en lugar de borrar una tabla, borra una base de datos completa.

```
DROP DATABASE nombre_de_la_base_de_datos;
```

La explicación del comando Drop la encontrarás en el siguiente link:

<https://downloads.mysql.com/docs/refman-5.0-es.pdf>

En el manual se apunta que: hay que tener cuidado al usar este comando ya que elimina de manera definitiva datos y estructura según donde se esté usando. **Revisa** el link y una vez que lo hayas leído prueba la instrucción eliminando una de las tablas para que compruebes lo fácil que es usarlo.

Otro comando de eliminación es el *Truncate*. El comando *delete* y *truncate* son muy parecidos las diferencias las podrás encontrar en <http://dev.mysql.com/doc/refman/5.0/es/truncate.html>

Revisa el link y observa que el comando *delete* se describe en el siguiente tema; pero, a diferencia de DELETE, el comando TRUNCATE elimina todas las filas de la tabla, sin borrar la tabla, por lo que luego se pueden hacer inserciones. La sintaxis es: TRUNCATE TABLE nombre_tabla;

En esta sección se describieron comandos cuyo objetivo es eliminación o pérdida de la información, ello permite que el administrador decida si desea borrar sólo la información o la estructura, ya sea de la base de datos o de las tablas.



Check table y Optimize table

Existen dos comandos más que se aplican sobre las tablas en la bases de datos, el primero de ellos revisa las tablas.

El comando *check table* es una herramienta para saber verificar lo que está sucediendo en un sistema que utiliza una base de datos MySQL, seguramente como administrador te encontrarás en una situación en la que el sistema está fallando, o no guarda o no está actualizando la información en la base de datos; debido a que este tipo de errores no son fácilmente detectables, ya que otras tablas de la misma base de datos no presentan esta anomalía, esto se debe a que el error está en una tabla específica de la base de datos; cuando un sistema funciona sin ser apagado puede sufrir altibajos de voltaje y dañar cierta parte de la información, entonces lo más aconsejable es utilizar el comando *check table*, este comando simplifica el diagnóstico, ya que cuando se ejecuta muestra si hay errores o advertencias indicando si la tabla no fue cerrada correctamente o si alguna consulta quedó sin terminarse dejó la misma tabla abierta.

Por tanto el comando *check table* busca errores en las tablas y si no tiene errores sólo manda el aviso que está correcta. Para conocer más de este comando lee la información del siguiente link: <https://downloads.mysql.com/docs/refman-5.0-es.pdf>

Aplica el comando *check table* a algunas de las tablas de ejemplo que se tienen capturadas y verifica como se encuentran.

Otro comando importante para la correcta administración de las tablas es el *OPTIMIZE TABLE*; éste es aconsejable utilizarlo, si la tabla durante su uso ha borrado una gran parte de la tabla o si ha hecho varios cambios en una tabla con registros de longitud variable, pues los registros que son borrados se encuentran aún presentes en modo de lista enlazada y operaciones posteriores utilizan posiciones de antiguos registros. Este comando realiza una reorganización y acomodado del archivo que contiene los datos.

Revisa este link: <https://downloads.mysql.com/docs/refman-5.0-es.pdf> para conocer más sobre el comando.

Durante este tema pudiste recordar y profundizar en comandos que pertenecen a la definición de datos; desde crear la base de datos, las tablas con sus respectivos atributos, el uso de comandos que muestran qué tablas y base de datos se están usando.

Es indispensable que, como administrador de una base de datos, puedas agregar atributos o modificar las tablas, repararlas si es necesario o cambiarles de nombre tal como pudiste en este tema aprender a usarlos, también cómo puedes modificar la estructura de la tabla, borrar su información, ya sea sólo los datos, la estructura o el uso de los comandos *drop* y *truncate*. Por último, en este tema viste otros comandos indispensables de un administrador tales como *check table* y *optimize table*.



Para concluir este tema, realiza la *Actividad 1. Encontrar errores*. Que se presentan en el documento: *Actividades de la unidad*.

En ella crearás una base de datos y sus tablas. Con un código que se te proporcionará y en el que encontrarás cuál de esas instrucciones tiene error y lo corregirás aplicando todo lo referente a lo visto en este tema.

2.2. Data Manipulation Language (DML)

Para continuar con la temática de esta Unidad, toca revisar el Lenguaje de Manipulación de Datos (DML) (Data Manipulation Language), el cual es un lenguaje que puede ser aplicado en el sistema gestor de bases de datos, en este caso, MySQL; este lenguaje, permite a los usuarios realizartareas de consulta o manipulación de los datos, organizados por el modelo de datos adecuado. Los comandos más utilizados son: *Select*, *Insert*, *Delete* y *Update*.

Para documentar este tema, **lee** Pérez, C. (2009, pp. 151-155) en este texto encontrarás una ampliación y explicación del comando que veremos a continuación que es el *Select* el cual es para hacer una selección de la información.

Select

La instrucción *Select* sirve para obtener ciertas filas o registros de una o más tablas, que pueden cumplir o no con cierta condición que se especifique. Por ejemplo: puedes, con el uso de este comando, mostrar la lista de clientes que deben más de 3,000 pesos. Estos comandos se pueden usaren combinación con otras instrucciones o palabras clave, las cuales son: *from*, para indicar de qué tabla interesa la información, y la cláusula *where*, para especificar la condición, la cual deseamos que se cumpla para que el *select* muestre la información. Recuerda que el * representa toda la información se puede especificar los campos que de esa tabla nos interesan. Para conocer más acerca de cómo se usa el select **consulta**: <https://downloads.mysql.com/docs/refman-5.0-es.pdf> Algunos ejemplos que servirán de apoyo y que puedes aplicar a las bases de datos que tenemos capturadas descritas en el primer ejemplo son los siguientes.

Ejemplos aplicados a la base de datos que creamos sería:



```
select *
```

```
from Proyecto
```

```
where codigo = 25;
```

Este ejemplo de select muestra todas las columnas de la tabla proyecto donde se cumpla con la condición de que el código sea igual a 25.

```
select *
```

```
from Proyecto
```

```
order by Proy.codigo
```

Este ejemplo de select muestra todas las columnas de la tabla proyecto ordenado por el número de proyecto.

```
select *
```

```
from Programadores_Proyecto
```

```
where codigo <> 1240
```

Este ejemplo de select muestra todas las columnas de la tabla Programadores_Proyecto donde se cumpla con la condición de que el código sea diferente al 1240.

Insert

Muchas de las ocasiones, será necesario insertar nuevas filas en las columnas y esto se puede realizar con la instrucción *insert*, la cual inserta nuevas filas en una tabla existente. Los nombres indican las columnas explícitamente con la palabra SET.

Revisa la liga <https://downloads.mysql.com/docs/refman-5.0-es.pdf> y **observa** cómo se puede usar la función para agregar a la base de datos. A continuación se muestran ejemplos, ejecútalos para la base de datos que se tiene creada desde el tema 1:

EJEMPLOS DE INSERCIÓN DE DATOS A TABLAS

```
insert Analista values(2,'Arturo Ramos','Azul 445','artur222@gmail.com')
```

```
insert Analista values(3,'Daniela López','sprinf 789','Ldany@gmail.com')
```

```
insert Programador values(4,'Rodrigo A','tamazula 666','elrodri@gmail.com')
```

```
insert Programador values(5,'Manuel jass','teran 345','manujass@gmail.com')
```

```
insert Cliente values(15,'Asociados ','blvd jacarandas 13','555890543')
```

```
insert Cliente values(20,'Construc ROMA','Rio Pedraza 45','443524549')
```

```
insert Proyecto values(12,'Sistema XXX',22000,80,2,15)
```

```
insert Proyecto values(123,'Control de Inventario',35000,160, 3,20)
```

```
insert Programadores_Proyecto values (12,4)
```

```
insert Programadores_Proyecto values (12,5)
```



Una vez que ya realizaste estas inserciones, vuelve a ejecutar varias veces los comandos de selección de manera que muestres el contenido de todas las tablas con todas sus columnas y **observa** como ya se encuentran con información.

Update y delete

Ahora, como administrador, te puede pasar que ya se tienen las tablas con datos, pero es necesario cambiar algún dato; una manera de hacerlo es utilizando el comando *Update* el cual actualiza columnas en registros de tabla existentes con nuevos valores. Para el uso de este comando, es necesaria la cláusula **SET** que indica qué columna modificar y los valores que puede recibir; para su uso se puede apoyar de la cláusula **WHERE**, la cual es para especificar en qué registros debe actualizarse la información. De otro modo, se actualizan todos los registros.

Revisa la liga: <https://downloads.mysql.com/docs/refman-5.0-es.pdf>

Y **realiza** cambios a algún dato que desees. Recuerda especificar el registro al que quieres afectar con la cláusula *where*. A continuación, veremos el comando *Delete* el cual tiene la estructura similar a la *update*, pero en lugar de modificar datos, borra filas; este comando también se apoya de la cláusula *where* para delimitar cuales filas se desean borrar, ya que serían las que cumplan con la condición especificada.

Lee Pérez, C. (2010, pp. 498-501), en donde encontrarás una explicación del comando *delete*. **Abre** la siguiente liga de la página oficial de MySQL, en donde encontrarás más del uso de este comando, posterior a esto teclea el comando en la base de datos como lo muestra el ejemplo en:

<https://downloads.mysql.com/docs/refman-5.0-es.pdf>

Ejemplo del comando *delete* para borrar de la tabla proyecto de la fila que cumpla con ser el código 123.

```
DELETE FROM proyecto
```

```
WHERE codigo = 123
```

Durante este tema se enfatizó el uso de los comandos, para la manipulación de los datos, la cual puede ser apoyada por los siguientes comandos: SELECT, INSERT, UPDATE, y DELETE que pertenecen al lenguaje DML; como su nombre lo dice y, como lo pudiste ver, sirven para la manipulación de la base de datos, tal como: mostrar información específica de ciertas tablas con el comando *Select*, agregar datos con el comando *Insert*, modificar los mismos con el comando *Update* y desde luego borrar datos con el comando *Delete*; éstos, se pueden apoyar con el uso de cláusulas tales como como: la *Where*, para ser más específico, sobre la información que se desea manipular.

Para concluir este tema, **realiza** la *Actividad 2. Data Manipulation Language*, en la cual tendrás como propósito aplicar los comando de manipulación de datos vistos en este tema, para lo cual será necesario resolver un problema planteado con la obtención del código de los comandos: SELECT, INSERT, UPDATE y DELETE en la base de datos creada en la actividad anterior.



2.3. Data Control Language (DCL)

En este último tema se estudiarán los comandos **GRANT** y **REVOKE**, los cuales permiten a los administradores de sistemas crear cuentas de usuario MySQL, otorgar permisos y eliminar cuentas. Los comandos GRANT y REVOKE se utilizan para conceder y retirar los derechos a los usuarios de MySQL.

Hay cuatro niveles de privilegio:

- Global
- Base de datos
- Tabla
- Columna

Ahora, **revisa** Pérez, C. (2010, pp. 325-328), en este texto encontrarás los temas de la sentencia *Granty Revoke* y, en una tabla en la página 326, **puedes analizar** una explicación que se da, de los niveles mencionados.

Comando GRANT

Se utiliza para crear usuarios y concederle privilegios. La sintaxis general del comando GRANT es la siguiente:

```
GRANT privilegios (columnas)
ON elemento
TO nombre_usuario IDENTIFIED BY 'contraseña'
(whith grant option);
```

La información de MySQL se almacena en las tablas de la base de datos.

Consulta la siguiente liga la cual es de la página oficial de MySQL, para revisar más a profundidad el uso de estos comandos:

<https://downloads.mysql.com/docs/refman-5.0-es.pdf>

A lo largo de este tercer tema de la unidad, pudiste darte cuenta que se puede controlar el acceso a la base de datos otorgando y quitando permisos a los usuarios una herramienta muy útil para una buena administración. Ya que el tener un buen control sobre quien puede acceder a la información y qué es lo que puede o no hacer, sobre los datos, cada usuario es vital para conservar la integridad de la información.

Para concluir este tema, **realiza** la *Actividad 3. Data Control Language (DCL)*, en la cual tendrás como propósito aplicar los comandos DCL de control de lenguaje, para lo cual será necesario resolver un problema planteado obteniendo el código para otorgar permisos y el código para retirarlos.



Cierre de la unidad

Has concluido la segunda unidad del curso. A lo largo de esta recordaste conceptos básicos tales como: los comandos DDL (Data Definition Language) que se usan en la creación y definición de los datos, los comandos DML (Data Manipulation Language) que son usados en la manipulación de los datos y DCL (Data control Language) para el control del acceso a los mismos con esto serás capaz de tener un control sobre la información y conociste varias herramientas para la correcta administración de una base de datos.

Es aconsejable que revises nuevamente la unidad en caso de que los temas que se acaban de mencionar no te sean familiares, o no los recuerdes, de no ser este tu caso, ya estás preparado(a) para seguir con la *Unidad 3. Administrar bases de datos*, en donde continuarás con el tema de conectividad de la base de datos, modos de operación, bitácoras, respaldo y recuperación de la base de datos; esta será la última unidad del curso de *Administración de bases de datos*.

Para saber más...

Si deseas saber más acerca de la Administración de bases de datos puedes consultar un documento elaborado por el Departamento de Lenguajes y Ciencias de la Computación de la Universidad de Málaga:

- *Sistemas de Bases de Datos: Introducción y Conceptos Básicos para la Administración y de Bases de Datos Relacionales.*
<https://rua.ua.es/dspace/bitstream/10045/2990/1/ApuntesBD1.pdf>

Fuentes de consulta

- Camuña Rodríguez, J. F. (2015). Lenguajes de definición y modificación de datos SQL (UF1472):
- (ed.). IC Editorial. <https://elibro.net/es/ereader/unadmexico/44141?page=44>
- Coronel, C. (2011). *Bases de datos: Diseño, implementación y administración*. México: CENGAGE
- MySQL. (2009). *Manuales de Referencia del Manejador de Base de Datos*.
<https://downloads.mysql.com/docs/refman-5.0-es.pdf>
- Nevado, V. (2010) *Introducción a las bases de datos relacionales*. España: Visión Libros.
- Pérez, C (2009). *MySQL para Windows y Linux*. México: Alfaomega.
- Pérez, C (2010) *SQL server 2008*. México: Alfaomega.
- Stanek, W (2007). *SQL SERVER 2005 Manual del administrador*. México: McGraw Hill.