

Modelo entidad relación

**Breve descripción:**

Una vez conocidos los principios del diseño de bases de datos relacionales, se hace importante fortalecer las habilidades de análisis, debido a que diseñar un modelo de datos no solamente es una tarea que intenta cumplir un conjunto de requisitos funcionales de un sistema informático, sino también el requerimiento de rendimiento, escalabilidad y flexibilidad.

**Abril 2024**

Tabla de contenido

[Introducción 4](#_Toc165021504)

[1. Establecer un entorno de trabajo 5](#_Toc165021505)

[1.1. Sistema gestor de bases de datos MySQL 5](#_Toc165021506)

[1.2. Sistema de diseño y gestión de bases de datos 6](#_Toc165021507)

[1.3. Sistemas basados en MySQL 8](#_Toc165021508)

[2. Ejercicios de diseño e implementación 9](#_Toc165021509)

[2.1. Análisis y diseño sistema de facturación 9](#_Toc165021510)

[Análisis y diseño 10](#_Toc165021511)

[2.2. Identificación de sentencias DDL 19](#_Toc165021512)

[2.3. Generar la base de datos 24](#_Toc165021513)

[2.4. Operaciones de mantenimiento y actualización 25](#_Toc165021514)

[2.5. Actualización por interfaz de línea de comandos 28](#_Toc165021515)

[Diseño conceptual 31](#_Toc165021516)

[Diseño lógico 34](#_Toc165021517)

[Diseño físico 35](#_Toc165021518)

[3. Administración de bases de datos 37](#_Toc165021519)

[3.1. Copia de seguridad de base de datos con MySQL “Workbench” 38](#_Toc165021520)

[3.2. Restaurar una copia de seguridad con MySQL “Workbench” 39](#_Toc165021521)

[Síntesis 41](#_Toc165021522)

[Material complementario 42](#_Toc165021523)

[Glosario 43](#_Toc165021524)

[Referencias bibliográficas 44](#_Toc165021525)

[Créditos 45](#_Toc165021526)

Introducción

Bienvenido al presente componente formativo

Una vez definidos los conceptos básicos de las bases de datos, se está listo para afrontar su análisis y diseño, el cual consiste en tomar los datos más importantes que están implicados en un problema o qué datos son necesarios para llevar el control por un sistema informático a implementar.

Para extraer los datos, se debe hacer un análisis detallado del problema, para poder descartar los que no son susceptibles a la administración por el sistema a desarrollar, luego de tener extraídos los datos esenciales, se puede empezar a construir el modelo adecuado, por lo que se necesitará una herramienta de diseño de base datos y a través de ella expresar todos los aspectos técnicos de la solución a implementar.

Este componente brinda una serie de ejercicios prácticos de análisis y diseño, a la vez que profundiza en los temas de modelado.

Le deseamos muchos éxitos en este proceso de aprendizaje.

# Establecer un entorno de trabajo

Es importante saber cómo obtener, instalar, configurar y probar las herramientas necesarias para realizar las operaciones de diseño. Existen muchas tecnologías para diseñar e implementar bases de datos, en este caso, utilizaremos MySQL como sistema gestor de bases de datos y MySQL “Workbench” como herramienta de diseño, es importante que en la medida que se van presentando los contenidos, realice de forma local y en su computador el trabajo con los ejercicios y ejemplos propuestos.

## Sistema gestor de bases de datos MySQL

MySQL es un sistema de gestión de bases de datos relacional, desarrollado bajo licencia dual, lo que significa que es una licencia pública general y/o licencia comercial por Oracle Corporation y, que según medios de comunicación, es considerada como la base de datos de código abierto más popular del mundo. Tener licencia dual significa que se puede usar la licencia de uso público sin incurrir en gastos; sin embargo, se puede adquirir una licencia de pago que incluye paquetes que robustecen el producto y servicios de soporte técnico especializado.

* **Proceso de descarga de MySQL Server “Community” y MySQL “Workbench”**

Lo invitamos a consultar el PDF **Proceso de descarga de MySQL Server “Community” y MySQL “Workbench**”, el cual se encuentra en la carpeta Anexos.

* **Proceso de instalación de MySQL**

Lo invitamos a consultar el PDF “**Proceso de instalación de MySQL Server “Community”**”, el cual se encuentra en la carpeta Anexos.

## Sistema de diseño y gestión de bases de datos

Es una herramienta visual unificada para arquitectos, desarrolladores y administradores de bases de datos. MySQL “Workbench” proporciona modelado de datos, desarrollo de SQL y herramientas de administración integrales para la configuración del servidor, administración de usuarios, respaldo y muchas otras características y, lo más importante es que es multiplataforma, lo que indica que puede descargarse para sistemas operativos Windows, Linux y Mac.

Algunas de las características más importantes de “Workbench” son las siguientes:

* **Diseño**

MySQL “Workbench” permite a un administrador de bases de datos, desarrollador o arquitecto de datos, diseñar, modelar, generar y administrar bases de datos visualmente. Incluye todo lo que un modelador de datos necesita para crear modelos ER complejos, ingeniería directa e inversa, y también ofrece características clave para realizar tareas difíciles de gestión de cambios y documentación, que normalmente, requieren mucho tiempo y esfuerzo.

En “Workbench” se pueden crear nuevos modelos, lo que equivale a crear una nueva base de datos e ir agregando los objetos requeridos como tablas, atributos, tipos de datos, relaciones, entre otras.

* **Desarrollar**

MySQL “Workbench” ofrece herramientas visuales para crear, ejecutar y optimizar consultas SQL. El editor de SQL proporciona resaltado de sintaxis de color, autocompletado, reutilización de fragmentos de código SQL e historial de ejecución de SQL. El panel de conexiones de base de datos permite a los desarrolladores administrar fácilmente las conexiones de base de datos estándar, incluido MySQL Fabric. El explorador de objetos proporciona acceso instantáneo al esquema y los objetos de la base de datos.

* **Administrar**

MySQL “Workbench” proporciona una consola visual para administrar fácilmente los entornos MySQL y obtener una mejor visibilidad de las bases de datos. Los desarrolladores y administradores de bases de datos pueden utilizar las herramientas visuales para configurar servidores, administrar usuarios, realizar copias de seguridad y recuperación, inspeccionar datos de auditoría y ver el estado de la base de datos.

* **Panel de rendimiento visual**

MySQL “Workbench” proporciona un conjunto de herramientas para mejorar el rendimiento de las aplicaciones MySQL. Los administradores de bases de datos pueden ver rápidamente los indicadores clave de rendimiento mediante el panel de rendimiento. Los informes de rendimiento proporcionan una fácil identificación y acceso a puntos de acceso de E / S (entrada y salida), declaraciones SQL de alto costo y más. Además, con un clic, los desarrolladores pueden ver dónde optimizar su consulta con el plan visual explain mejorado y fácil de usar.

* **Migración de bases de datos**

MySQL “Workbench” proporciona una solución completa y sencilla de usar para migrar Microsoft SQL Server, Microsoft Access, Sybase ASE, PostreSQL y otras tablas, objetos y datos RDBMS a MySQL. Los desarrolladores y administradores de bases de datos pueden convertir rápida y fácilmente las aplicaciones existentes para que se ejecuten en MySQL, tanto en Windows como en otras plataformas. La migración también admite la migración de versiones anteriores de MySQL a las últimas versiones.

## Sistemas basados en MySQL

El diseño de un cuestionario está directamente relacionado con el tipo de información que se desea obtener, ya sea información exploratoria o para probar una hipótesis.

Son instrumentos de investigación que consisten en conjunto de preguntas u otro tipo de indicadores que son diseñados específicamente para recolectar información del usuario encuestado. Esta técnica incluye preguntas cerradas y abiertas, que al final se traducen como herramientas de índole cualitativas o cuantitativas.

A continuación, encontrará el PDF Manual de instalación de MariaDB en Windows. En él solo se instala el SGBD, por lo que debe instalar por aparte MySQL “Workbench”, para que administre las bases de datos de MariaDB.

* **Manual de instalación de MariaDB en Windows**

Lo invitamos a consultar el PDF “**Manual de Instalación de MariaDB en Windows**”, el cual se encuentra en la carpeta Anexos.

# Ejercicios de diseño e implementación

El aprendizaje basado en problemas es una buena estrategia a seguir en este tipo de actividades, debido a que es el análisis de un problema el que da origen a un modelo de base de datos. Por lo tanto, se propondrán dos problemas a considerar, y en torno a ellos se van reforzando los conocimientos adquiridos a la vez que se va profundizando en algunos conceptos. Para lograrlo, se toma como problema los casos comunes como lo son el análisis y diseño de un sistema de facturación en inventarios y un sistema para gestionar las matrículas de estudiantes en un instituto de formación técnica.

## Análisis y diseño sistema de facturación

Analicemos el primer caso, llamado Sistema de facturación.

Suponga que lo contratan para diseñar una base de datos de una droguería, el objetivo principal es llevar las existencias de cada producto, a la vez que se va facturando lo que se vende, también interesa la información de los clientes, porque a futuro se quiere hacer campañas publicitarias de email marketing, según los productos que cada cliente consume.

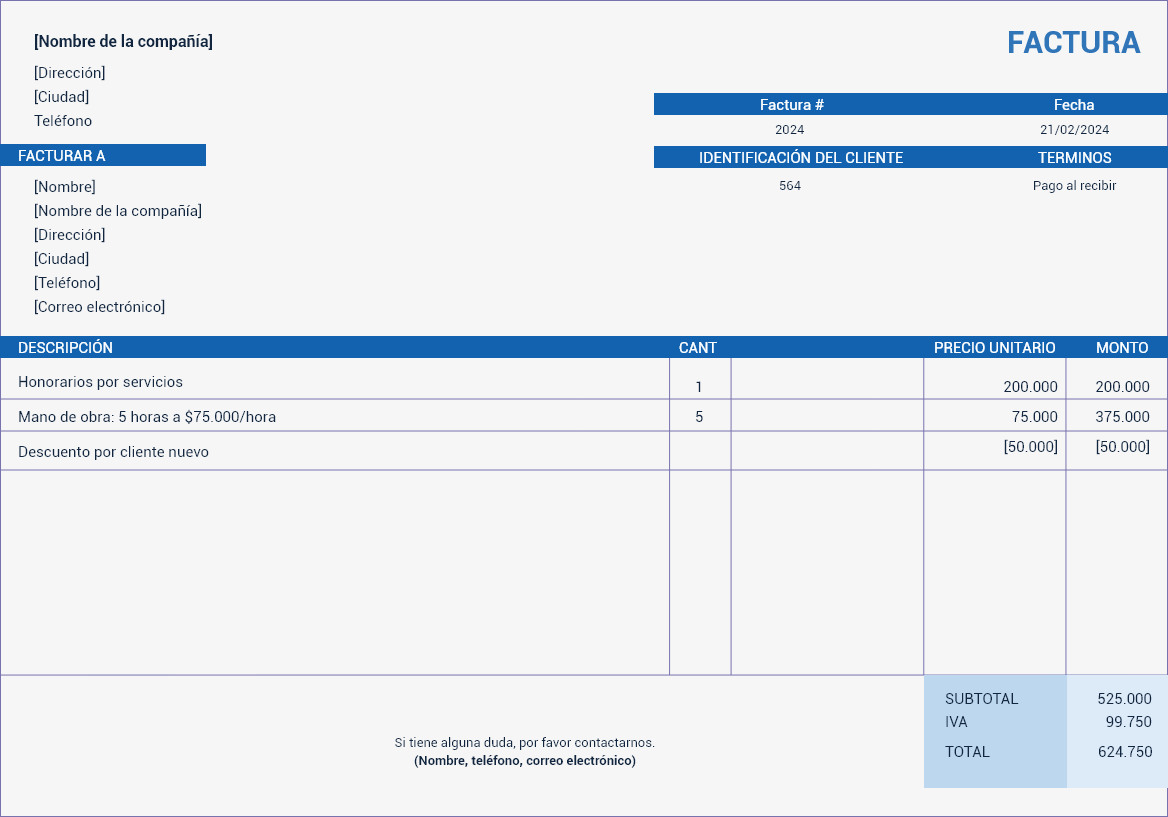
En una entrevista, el cliente dijo que necesitaba algunos reportes como:

1. Artículo más vendido.
2. Artículos que más utilidades dejan por unidad.
3. Empresa o persona que más factura (cantidad de factura).
4. Empresa o persona que más factura (volumen de ventas en pesos).
5. Productos sin existencia.

También fijó los productos que se organizan por categoría, porque le interesa saber el precio de compra y de venta, para conocer qué descuento hacer a una factura.

El cliente ofrece un modelo sobre cómo debe ser la factura, como se muestra en la siguiente figura.

1. Modelo de factura



### Análisis y diseño

Una buena forma de empezar el análisis es con el formato de la factura; se inicia revisando la cabecera de la factura.

Un primer impulso puede hacer pensar que se debe crear una entidad para la compañía o empresa y pensar en definir una entidad o empresa con atributos “nombre\_compañia”, “dirección”, “ciudad”, etc.; sin embargo, considere lo siguiente:

El cliente le ha pedido que modele un sistema de facturación para su droguería, es decir, para una droguería, no para un conjunto de droguerías. Es diferente, si solicitaran “se necesita un sistema de facturación que administre el inventario y facturación de múltiples droguerías”, a estos problemas se les llama “multitenant” (multiinquilino), pero este no es el caso. Por lo tanto, no es la entidad droguería parte del modelo, porque no se está modelando la facturación de múltiples droguerías.

1. Factura con información resaltada



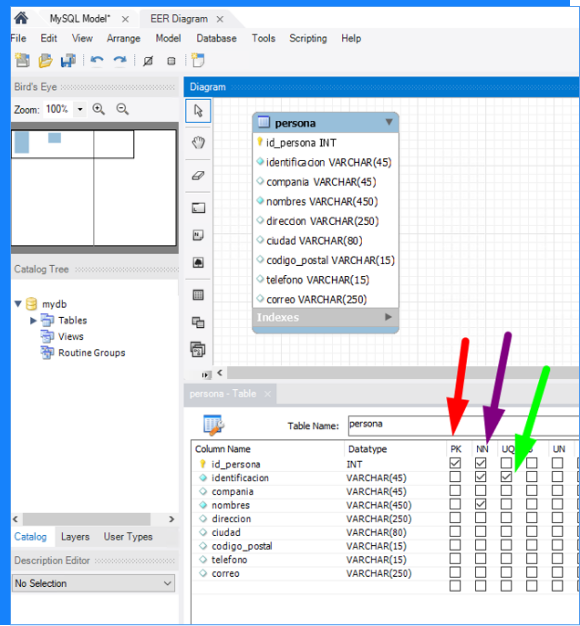
Tal parece que los datos más relevantes de los clientes están presentados en la parte resaltada de la imagen, lo que hace pensar que se requiere una entidad, cliente, usuario o persona y, que la persona puede tener relacionada una empresa o tal vez una persona hace la compra para una empresa.

Lo anterior se puede modelar con una entidad, la entidad persona que relaciona los datos (nombre, compañía, dirección, ciudad, teléfono, correo electrónico).

Una forma simplificada sería creando una entidad persona y para poder identificar cada fila, usar una llave primaria (id\_persona), pero también es importante usar un identificador único de persona como lo es el documento de identificación (porque las personas nunca tienen el mismo número), de igual forma, debe tener en cuenta que no puede existir una persona si al menos no se define el número de identificación y el nombre.

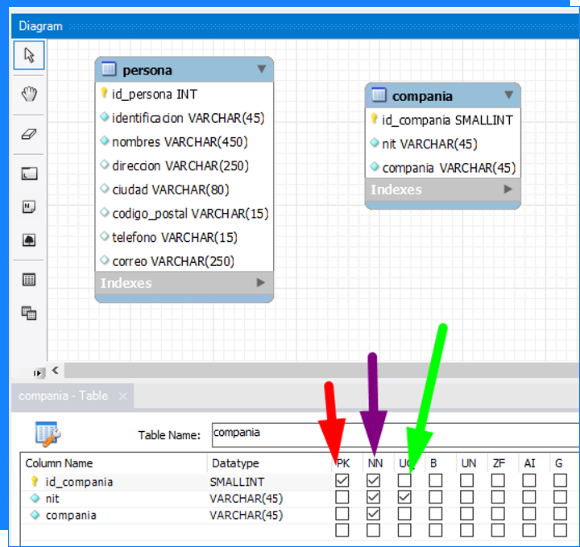
Lo anterior se puede ver en la siguiente imagen (flecha roja para definir la llave primaria, morada para definir los datos que deben ser registrados obligatoriamente, y la verde una columna cuyo valor no puede repetirse en toda la tabla).

1. Identificación



Tal parece que los datos más relevantes de los clientes están presentados en la parte resaltada de la imagen, lo que hace pensar que se requiere una entidad, cliente, usuario o persona y, que la persona puede tener relacionada una empresa o tal vez una persona hace la compra para una empresa.

1. Creación de tabla con las compañías



Como hay menos compañías que personas, se ha tomado id\_compania como SMALLINT. Esta decisión permite cumplir con dos requerimientos no funcionales:

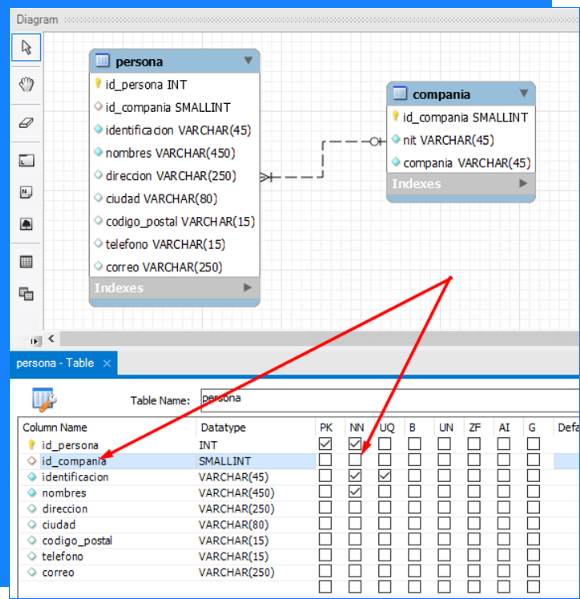
* Ahorro de espacio de almacenamiento.
* Ahorro en el tiempo computacional de las consultas.

**Nota:**

* Es importante comprender cómo la decisión de tipo de dato apunta a cumplir con los requerimientos no funcionales.

Es posible que piense que debe haber una relación entre la persona y la compañía de la siguiente forma:

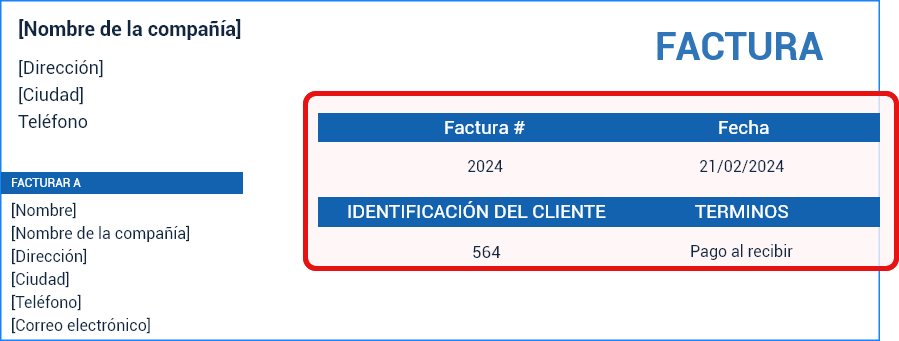
1. Relación entre la persona y la compañía



Al hacerlo de esta forma, una persona puede pertenecer a una empresa (solo a una), pero esa misma persona podría comprar en la farmacia unas veces a nombre propio y otras a nombre de la empresa, y este modelo no permitiría saber cada caso. Otro problema es que esa persona un día compra a nombre de una empresa y otro día puede comprar a nombre de otra empresa (si cambia de empresa) y al actualizar la empresa de la persona, se estarían actualizando las facturas pasadas, lo cual sería un error.

De esta forma, la manera correcta sería relacionar la persona y la empresa con la factura.

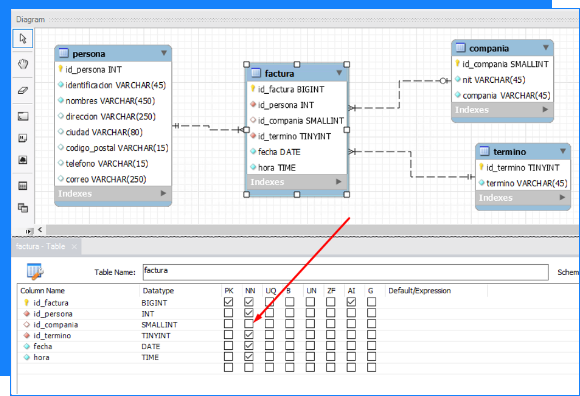
1. Información de la factura



Ahora, se presentan los datos de la factura y sus datos, como número de factura (que debe ser único y secuencial), fecha, la identificación del cliente (código interno id\_persona), y términos de pago.

Como se puede ver en la siguiente imagen, se creó la tabla término de pago, que almacenará los medios de pago (crédito, contado, contra entrega, etc.), note también que se usó el tipo de datos TINYTINT que solo almacena números de 0-126 y se hace una relación de 1 a muchos (un término puede tener relacionadas muchas facturas, o lo que es igual muchas facturas tienen un mismo término de pago).

1. Creación de la tabla término de pagos



Note que una factura necesariamente debe tener relacionada la referencia de una persona, el término, la fecha y la hora, pero la compañía (id\_compania) no es obligatoria. Esto es porque hay facturas que las compran las personas naturales y otras que las compran las empresas o compañías, pero en un caso o en el otro, siempre habrá una persona que hace la compra, es por esto que el id\_persona es obligatoria.

La factura tiene un tipo de dato BIGINT, porque se espera que pueda facturar muchos al ser minorista, note también la columna AI (Auto Incrementable), esto hace que el gestor de base datos le asigne automáticamente y secuencialmente un número a cada fila para la columna id\_factura.

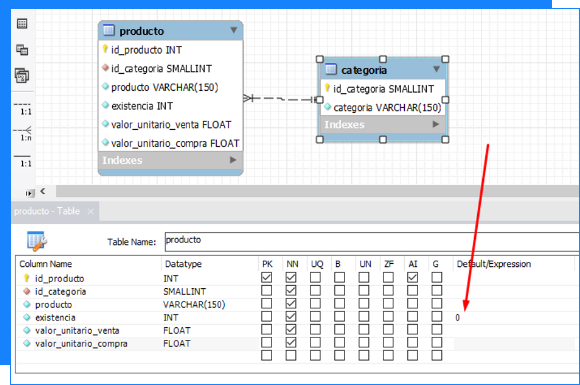
Otro segmento de la factura son los ítems o productos que la componen:

1. Ítems que componen la factura



La imagen sugiere que los productos tienen una descripción, un precio unitario y como mencionó el cliente, un precio de compra:

1. Apartados que tienen los productos



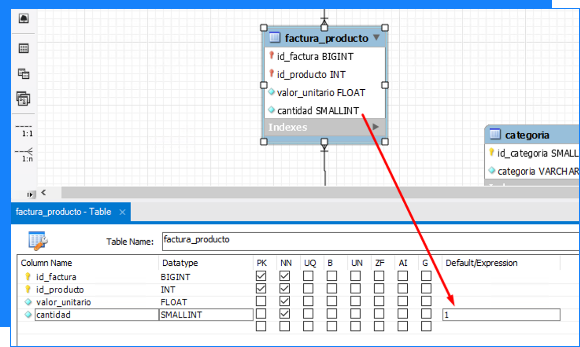
Una categoría puede tener varios productos o varios productos pertenecen a una categoría, esto es una relación de 1 a muchos, que queda definida en la imagen anterior. Note los tipos de datos definidos y también que la columna existencia no puede ser null, además de eso, si el usuario no pone un dato o trata de poner dato null en la existencia, entonces el sistema gestor de base de datos le pondrá “0” a esa fila en esa columna. La imagen sugiere que los productos tienen una descripción, un precio unitario y como mencionó el cliente, un precio de compra:

Ahora, resta relacionar los productos con la factura, y una factura puede relacionar varios productos, y un mismo producto puede estar en muchas facturas, por lo tanto, es una relación de muchos a muchos (n:m), esto hace que se genere una entidad intermedia que se llama producto\_factura.

La usabilidad es una medida de calidad que busca medir la eficacia, eficiencia y satisfacción en la experiencia del usuario con respecto a un producto, lo ideal es que este tipo de interacción entre usuario/producto sea lo más fácil y simple posible.

La entidad factura\_producto es una relación de muchos a muchos y agrega los datos necesarios para saber en cuánto y cuántos productos se venden por factura. También en la siguiente imagen se muestra cómo son obligatorias todas las columnas de este modelo y que la cantidad por defecto tiene valor de 1, debido a que, si se intenta guardar un valor nulo, no debe ser posible y en lugar de insertar nulo insertará 1, porque no tiene sentido agregar un producto a una factura y no vender al menos un producto.

1. Columnas del modelo



## Identificación de sentencias DDL

Suponga que lo contratan para diseñar una base de datos de una droguería, el objetivo principal es llevar las existencias de cada producto, a la vez que se va facturando lo que se vende, también interesa la información de los clientes, porque a futuro se quiere hacer campañas publicitarias de email marketing, según los productos que cada cliente consume.

MySQL “Workbench” ofrece la interfaz gráfica con la que se pueden generar estos códigos. Se mostrará cómo se hace con algunos ejemplos, siga los siguientes pasos:

1. Haga clic derecho a la entidad persona y luego a “Copy SQL to Clipboard” (copiar SQL en portapapeles).
2. Después se debe pegar (Ctrl+V) el contenido en un editor de texto y el resultado será:

CREATE TABLE IF NOT EXISTS `mydb`.`persona` (

`id\_persona` INT NOT NULL,

`identificacion` VARCHAR(45) NOT NULL,

`nombres` VARCHAR(450) NOT NULL,

`direccion` VARCHAR(250) NULL,

`ciudad` VARCHAR(80) NULL,

`codigo\_postal` VARCHAR(15) NULL,

`telefono` VARCHAR(15) NULL,

`correo` VARCHAR(250) NULL,

PRIMARY KEY (`id\_persona`),

UNIQUE INDEX `identificacion\_UNIQUE` (`identificacion` ASC) VISIBLE)

ENGINE = InnoDB;

Como puede apreciar, es posible obtener la definición de la base de datos en lenguaje SQL (DDL), si no se tuviera MySQL “Workbench”, se tendría que elaborar cada una de las sentencias.

Pruebe sus conocimientos a través del siguiente ejercicio, en el que identifique en la sintaxis:

* Cómo en SQL define una llave primaria.
* Cómo en SQL se define un índice único.

Es importante que se familiarice con estas sentencias, porque las operaciones de mantenimiento de bases de datos como adición o sustracción de columnas o tablas, son más fáciles cuando se hacen a través de la ejecución de scripts en lenguaje SQL.

**Nota:**

* Como no se ha especificado el nombre de la base de datos MySQL “Workbench” pone por defecto `mydb` a la base de datos.

A continuación, se presenta una entidad más compleja, como lo es la entidad factura:

CREATE TABLE IF NOT EXISTS `mydb`.`factura` (

`id\_factura` BIGINT NOT NULL AUTO\_INCREMENT,

`id\_persona` INT NOT NULL,

`id\_compania` SMALLINT NULL,

`id\_termino` TINYINT NOT NULL,

`fecha` DATE NOT NULL,

`hora` TIME NOT NULL,

PRIMARY KEY (`id\_factura`),

INDEX `fk\_factura\_persona\_idx` (`id\_persona` ASC) VISIBLE,

INDEX `fk\_factura\_compania1\_idx` (`id\_compania` ASC) VISIBLE,

INDEX `fk\_factura\_termino1\_idx` (`id\_termino` ASC) VISIBLE,

CONSTRAINT `fk\_factura\_persona`

FOREIGN KEY (`id\_persona`)

REFERENCES `mydb`.`persona` (`id\_persona`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_factura\_compania1`

FOREIGN KEY (`id\_compania`)

REFERENCES `mydb`.`compania` (`id\_compania`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_factura\_termino1`

FOREIGN KEY (`id\_termino`)

REFERENCES `mydb`.`termino` (`id\_termino`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB

Observe cómo las llaves foráneas les define un índice:

INDEX `fk\_factura\_persona\_idx` (`id\_persona` ASC) VISIBLE,

Con esta sintaxis, lo que hace MySQL “Workbench” es decirle al gestor de base de datos MySQL que esta columna id\_persona debe guardarse en estructura de árbol binario (que se llame fk\_factura\_persona\_idx) y no de manera secuencial. Este tipo de estructura de datos en árbol mejora la velocidad de las operaciones de consulta y búsqueda.

También se crean índices para las columnas id\_compania e id\_termino.

**Nota:**

* Usar MySQL “Workbench” ayuda a definir la estructura de almacenamiento de los datos para que cumpla con el requerimiento no funcional de rendimiento en las consultas.

También MySQL “Workbench” crea unas restricciones que permiten mantener la integridad de los datos.

CONSTRAINT `fk\_factura\_persona`

FOREIGN KEY (`id\_persona`)

REFERENCES `mydb`.`persona` (`id\_persona`)

ON DELETE NO ACTION ACTION,]

A continuación, se presenta línea a línea, recordando que la columna id\_persona es una llave foránea de la tabla persona:

1. Restricciones

| Nombre restricción | Descripción |
| --- | --- |
| CONSTRAINT `fk\_factura\_persona` | Indica al SGDB que debe crear una restricción (“CONSTRAINT”) que se llama fk\_factura\_persona |
| FOREIGN KEY (`id\_persona`) | Indica la columna que es llave foránea, es decir, que id\_persona es una referencia de una fila de otra tabla. |
| REFERENCES `mydb`.`persona` (`id\_persona`) | La fila de la otra tabla de la que es referencia de la base de datos mydb, tabla persona columna id\_persona |
| ON DELETE NO ACTION | Qué acción debe hacer el SGDB si borran la fila de la tabla persona con los datos de la tabla factura que estén relacionados, acá dice que no le permite borrar personas que ya tienen facturas. |
| ON UPDATE NO ACTION, | Qué acción debe hacer el SGDB si actualiza la columna id\_persona una fila de la tabla persona con los datos de la tabla factura que estén relacionados, acá dice que no le permita actualizar la columna id\_persona de la tabla persona que ya tiene facturas. |

No es de preocuparse si no es muy claro hasta la decisión del SQL de llaves foráneas, dado que a lo largo de la práctica y la revisión del material complementario, podría complementar los diferentes elementos y conceptos.

## Generar la base de datos

A continuación, se presentará cómo obtener el modelo físico de una base de datos en formato natural de las bases de datos (esto es SQL). Con el siguiente procedimiento se obtendrá un modelo de datos en sentencias DDL, en un archivo con extensión .sql, estos documentos son los códigos fuente de las bases de datos, generalmente en el proceso de desarrollo de “software” son versionados como cualquier archivo de código fuente.

De acuerdo con lo anterior, una forma de generar la base de datos es la siguiente:

1. Ponga un nombre a la base de datos dando clic derecho en editar Schema.
2. Seleccione Ingeniería hacia adelante (“Forward Ingingger”).
3. Entre los datos de “root” base de datos.
4. Seleccione las opciones de la base de datos y se deja por defecto.
5. Luego, pide la contraseña de usuario “root” que se puso en el proceso de instalación de MySQL.
6. Después muestra un resumen de lo que creará.
7. Finalmente, le da en “Next” y el sistema generará el DDL de la base datos.

Se guarda un archivo “Save to file” y se copia al portapapeles “Copy to clipboard”. Se sugiere antes de dar “Next” y continuar el proceso, primero guardar como un archivo y ponerle este nombre, luego de ponerlo le da “Next” para que ejecute ese “script” en el SGDB instalado.

El archivo generado guardado en formato de texto con extensión .sql es lo que se llama el “script” DDL de la base de datos, es el principal producto de un modelo de datos, es el entregable de un proyecto y por lo que materialmente se puede pasar una factura al cliente.

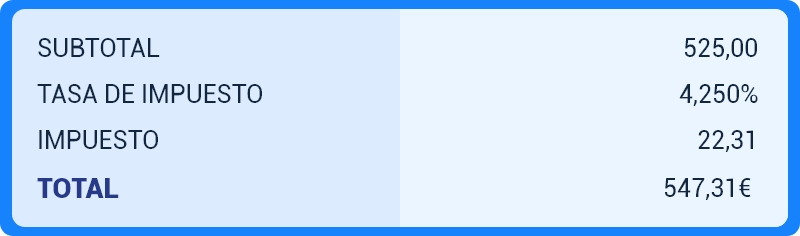
## Operaciones de mantenimiento y actualización

Por lo general, la base datos se diseña y se despliega en servidores en la nube, hosting o en instancias de almacenamiento, sistemas operativos o de plataforma que no tienen interfaz gráfica, es por eso que es importante el archivo de texto DDL con extensión .sql, la base de datos hasta ahora generada presenta una inexactitud que se ha puesto para ejemplificar una operación de actualización o mantenimiento.

Suponga que el cliente ha contratado un servidor en la nube para la base de datos y al darle subir el archivo de extensión .sql ha generado la base de datos, pero, se ha dado cuenta de que su modelo está mal o incompleto y lleva varios meses trabajando la base de datos, hay muchos datos, por lo que no se puede volver a empezar con una base de datos vacía, sino que se debe ajustar la base de datos actual (base en producción) sin borrar o alterar los datos que la base ya tiene.

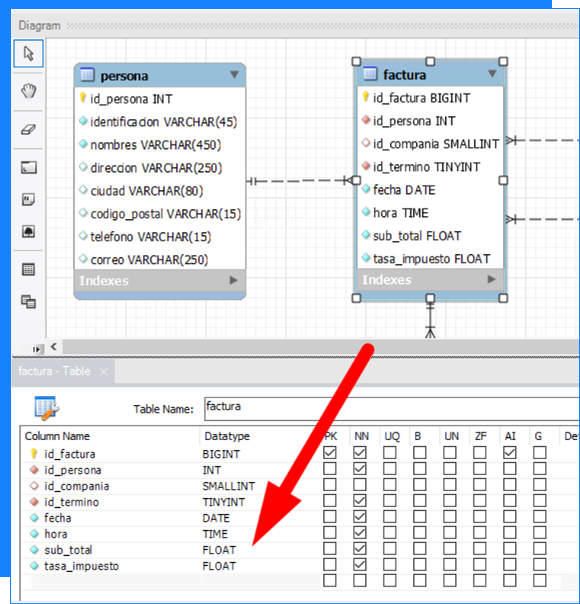
Además, junto a su equipo de trabajo ha detectado que no modelaron el pie de la factura:

1. Pie de la factura



Este ajuste consiste en agregar en la tabla factura, el valor del subtotal y la tasa de impuesto a aplicar; con estos datos se puede calcular el total:

1. Información a agregar en la tabla factura



Ahora, se necesitan sincronizar estos cambios (los del diagrama) con la base de datos generada. De acuerdo con lo anterior, puede revisar los siguientes pasos:

* Esto se logra dando clic en Database > Synchronize Model.
* Luego, seleccione la conexión local.
* Luego, seleccione los datos en la parte de generación de “script” adicionales.
* Después de pedir la contraseña se conecta; analice las bases de datos presentes y características del servidor.
* Seleccione la base de datos con la que se va a sincronizar.
* Analice las diferencias.
* Al ubicarse en cada una de las entidades o tablas, se puede ignorar la actualización.
* En lugar de actualizar la base de datos, si lo que se quiere es actualizar el modelo (diagrama) para que quede conforme a la base de datos, se cliquea en el botón “Update Model”. Pero este no es el caso de interés porque se necesita actualizar la base de datos, entonces se da clic en “Next”.

Al final del proceso se debe tener el DDL donde se genera la base datos y el DDL donde se actualiza la tabla factura.

## Actualización por interfaz de línea de comandos

Muchas veces no es el computador local el que interesa actualizar, sino más bien, un servidor en la nube. Algunas plataformas tienen un entorno gráfico que permite aplicar el “script” con unos cuantos clics desde una interfaz web, pero la mayoría ofrece una interfaz de consola o terminal, de manera que los recursos computacionales no se desperdician en interfaces gráficas.

Verifique los siguientes archivos:

Lo invitamos a consultar el PDF “**DDL drogueria**”, el cual se encuentra en la carpeta Anexos.

Lo invitamos a consultar el PDF “**DDL update**”, el cual se encuentra en la carpeta Anexos.

Lo invitamos a consultar el PDF “**Modelo Facturación Drogueria**”, el cual se encuentra en la carpeta Anexos.

A continuación, se muestra cómo aplicar el “script” desde una terminal. Esta base de datos se encuentra diseñada para cumplir el requerimiento funcional de administración de facturas de una droguería:

* **Paso 1.** Dar clic en el menú inicio, buscar MySQL, una terminal o interface de línea de comandados (Command Line Interface, CLI).
* **Paso 2.** A continuación, pedirá la contraseña de “root”.
* **Paso 3.** Luego, se debe seleccionar la base de datos con la sentencia SQL: use farmacia.

A continuación, se describe un problema del cual se pretende obtener un modelo de datos que permita gestionar la información, a través de un sistema gestor de base de datos, como se realizó en el caso anterior, donde se fue empleando una estrategia empírica, luego de conocer los conceptos de bases de datos.

Se presentará un problema más sencillo con el ánimo de concentrar esfuerzos en el análisis desde un enfoque metodológico, que se puede basar en tres fases:

**Diseño conceptual**

Encargado de entender el problema y la naturaleza de los datos (qué son y qué representan en el mundo real). Las actividades a desarrollar, son:

* Identificar entidades.
* Identificar atributos.
* Identificar las relaciones.
* Identificar las cardinalidades de las relaciones.
* Diagrama entidad relación.

**Diseño lógico**

Se encarga de analizar la estructura de los datos y que se cumpla con los requisitos del negocio, modelo de negocio o problema. Las actividades a desarrollar, son:

* Mapear diagrama entidad relación a un modelo relacional.
* Análisis de restricciones de integridad.
* Validación de requisitos funcionales.

**Diseño físico**

Se preocupa por cómo se almacenarán los datos. Las actividades a desarrollar, son:

* Definición de tipos de datos.
* Analizar las restricciones de llaves foráneas.
* Definición de índices.

Estudiemos ahora el caso 2, gestión de matrículas.

Se necesita un sistema que permita controlar las matrículas de un instituto de formación técnica profesional; donde interesa la información pertinente de los estudiantes como son los datos personales y la edad; y de los profesores parte de los datos personales como la titulación profesional o técnica y el año de graduación.

Cada programa técnico se desarrolla por semestres y un semestre tiene varias materias, cada semestre un estudiante puede matricular máximo 7 materias, un profesor puede orientar varias materias y la misma materia puede ser dictada por varios profesores.

Un estudiante puede estudiar varios programas, aunque no a la vez, y el estudiante se inscribe a un programa y cada año puede cursar 2 semestres (de enero a junio período 1, y de julio a diciembre período 2), si registra por semestre una matrícula con las materias a cursar. Algunos programas duran 5, 6 o 7 semestres.

### Diseño conceptual

Para la elaboración del modelo conceptual es fundamental identificar las entidades, luego los atributos de cada una de ellas, luego las relaciones entre entidades y, si se desea, se elabora un diagrama de entidad relación como resultado del diseño conceptual:

**Lista de entidades**

* Persona (estudiantes y profesores): existen personas estudiantes y personas profesores.
* Programa: representa la carrera técnica.
* Inscripción: de un estudiante a un programa.
* Materia.
* Matrícula: dura un semestre y puede contener máximo 7 materias.

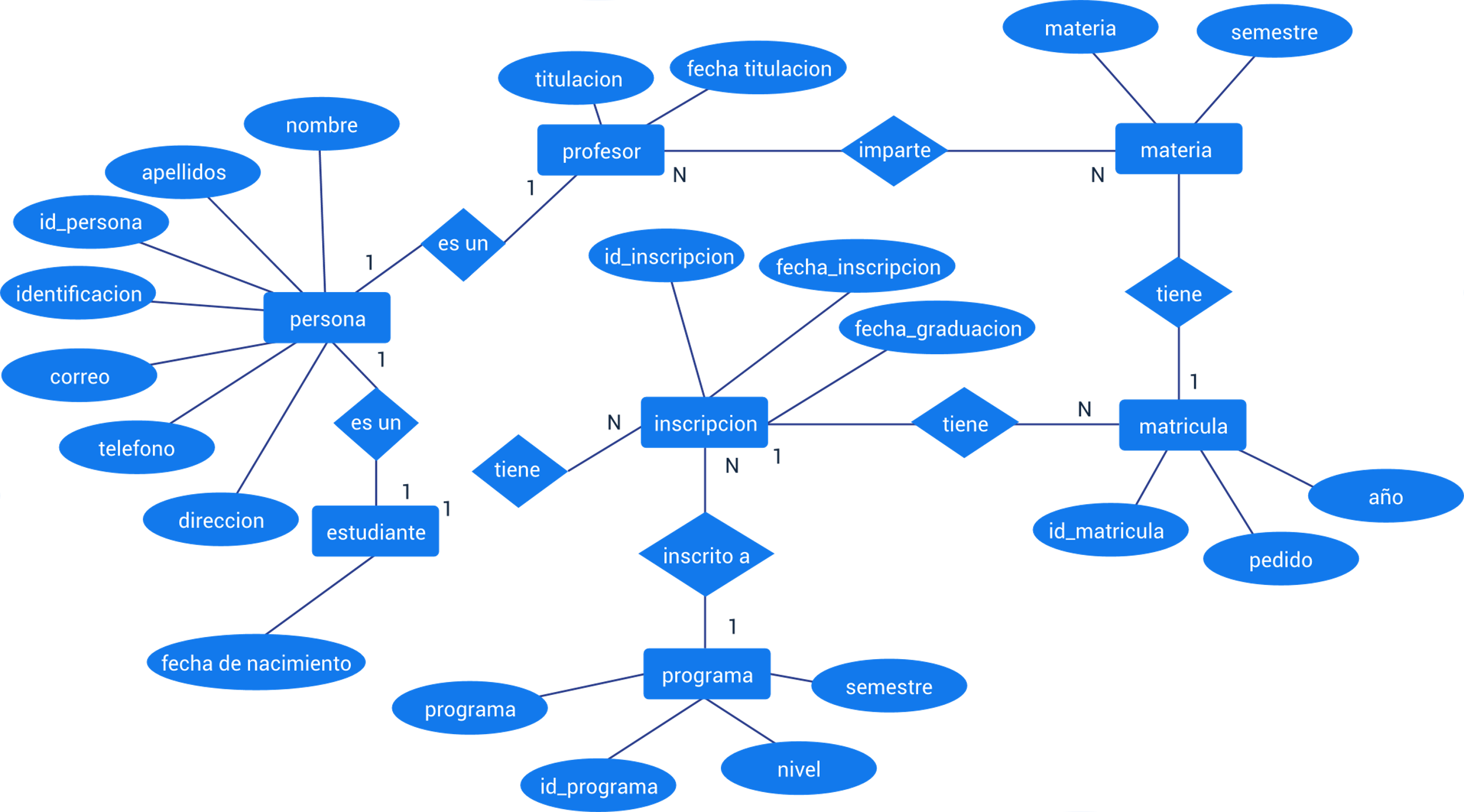
**Lista de atributos**

* Estudiante (nombre, apellidos, identificación, fecha\_nacimiento, correo, teléfono, dirección).
* Profesor (nombre, apellidos, identificación, correo, teléfono, dirección, titulación, fecha de titulación).
* Programa (nombre del programa, nivel, número de semestres).
* Inscripción (fecha, fecha de graduación).
* Materia (nombre de la materia y semestre).
* Matrícula (año, período).

**Relaciones y cardinalidad**

* El estudiante puede tener varias inscripciones, pero una inscripción por un solo estudiante.
* Un profesor puede dictar varias materias y una materia ser dictada por varios profesores.
* Una inscripción puede ser la relación entre un único estudiante y un único programa.
* Una inscripción puede tener varias matrículas, pero una matrícula corresponde a una única inscripción.
* Una matrícula se relaciona con varios estudiantes y un estudiante con varias matrículas.
* Una matrícula puede tener más de una materia, y la misma materia puede estar relacionada con varias matrículas.

1. Diagrama entidad relación



Actualmente, algunos diseñadores de bases de datos no diseñan el modelo entidad relación, por lo general directamente diseñan el modelo lógico o diagrama relacional, en algunos casos se emplea el modelo que se denomina modelo conceptual, que es equivalente al modelo entidad relación, pero difiere en la notación del diagrama al hacerse más parecido al modelo relacional. Por este mismo motivo es poco usado, porque supone hacer un nuevo paso o transformación del modelo conceptual a un modelo lógico, siendo este último el que en realidad puede convertirse en una base de datos.

### Diseño lógico

Una vez se tiene el modelo conceptual, se mapea el diagrama entidad relación o también llamado modelo conceptual a un modelo lógico, esto mediante las reglas de multiplicidad entre entidades (de uno a uno, de uno a muchos y de muchos a muchos).

Puede verificar algunos modelos, pero es recomendable que lo cree de manera que ejercite, lo hasta ahora visto.

Lo invitamos a consultar el archivo “**Modelos**”, el cual se encuentra en la carpeta Anexos.

Ahora, analicemos los siguientes pasos:

**Paso 1. Análisis de restricciones de integridad**

La entidad matrícula tiene relación con una inscripción, y esta a su vez con un estudiante y con el programa al que pertenece, también podría haberse realizado la matrícula con el programa y el estudiante, pero convenientemente, al relacionar matrícula con inscripción con menos datos, se está relacionando el programa y el estudiante de forma indirecta.

Las demás relaciones responden al proceso de análisis de mapeo, por lo que no se entrará en detalles.

**Paso 2. Validación de requisitos funcionales**

Para conocer si se cumplen los requisitos funcionales de lógica es importante prestar atención a los siguientes puntos:

* Revise por cada atributo si es conveniente que sea No Null (No null o NN), o si no nulo representa algo conceptualmente en el proceso.
* Si un atributo o columna es Not Null, revise si es importante definir un valor por defecto según la lógica del problema.

### Diseño físico

Finalmente, se debe revisar cuidadosamente la naturaleza de los datos antes de generar el archivo, conocer los tipos de datos que soporta el SGDB y estar seguro de que están bien definidos (si pueden no ser nulos, la longitud máxima y mínima, etc.):

**Definición de tipos de datos**

Revise:

* Que el valor (entero y flotante) corresponda a los valores que en un momento dado puede tener, por ejemplo, mire que las materias id\_materia es tipo SMALLINT (lo que permitiría 32767 materias) mientras que en personas id\_persona es tipo INT (lo que permitiría 2147483647 personas, es decir, el más del 30% de la población mundial).
* Preste especial atención en las cadenas de texto, tamaño máximo y mínimo en longitud.

**Analizar las restricciones**

Considere:

* Este es el punto estratégico del modelo, revisar la no nulidad o nulidad de las llaves foráneas es de vital importancia, porque ello determina en gran porcentaje si su modelo cumple los requerimientos funcionales del sistema, por ejemplo, una inscripción debe tener relacionado un id\_estudiante y un id\_programa, porque la lógica de una inscripción es que un estudiante se matricula a un programa de formación, por lo tanto, ni id\_estudiante, ni id\_programa deben ser Null, por lo tanto, son atributo con restricción Not Null (NN).

**Definición de índices**

Valide:

Cómo se usa MySQL “Workbench”, esta hace por el diseñador muchas cosas, una de ellas es identificar los índices; sin embargo, si en algún momento decide hacer la base de datos en esta herramienta, no cree índices en los siguientes casos:

* Por cada llave foránea se crea un índice.
* Si va a hacer consultas por rango de fechas, a las columnas por las que va a filtrar por rango de fechas, créeles un índice.
* No es necesario crear índices para las llaves primarias, por defecto MySQL lo hace.

Si algún día usa otro motor de base de datos como Oracle, Postgres o SQL Server, verifique manualmente la creación de índices, porque según la herramienta que esté usando, puede ayudarle a crearlos para las llaves foráneas o no.

# Administración de bases de datos

Las funciones de un administrador de bases de datos (Data Base Administrator, DBA), es la gestión general de la base de datos.

Debe conocer las reglas de la tecnología particular, que en este caso sería MySQL, pero también está entre sus responsabilidades:

* **Validar o revisar el modelado y diseño de las bases de datos**

Debe concentrar su interés en los requerimientos de rendimiento y administración de índices.

* **Procesos de auditoría**

Identificar qué usuarios tienen acceso a qué tablas o recursos a insertar, actualizar o eliminar datos, y cuándo.

* **Resguardo y recuperación de datos**

Esto es hacer copias de seguridad periódicas de los datos, mantenerlos a salvo de la destrucción accidental o intencional. Además, diseñar, implementar y probar un plan de recuperación, considerando que pueden presentarse problemas, y validar los datos que se pueden restaurar rápidamente (que las copias generadas sean útiles).

* **Planificación de capacidad**

Gestionar el volumen de datos y diseñar los planes apropiados para administrarlos, esto ya que los datos crecen rápidamente y se debe llevar un registro de la taza de este crecimiento.

* **Administración de cambios**

A menudo es la responsabilidad del DBA realizar el análisis de impacto antes de generar los cambios dentro de un SGDB.

De las anteriores actividades, se puede decir que al validar o revisar el modelado y diseño, ya se cuenta con la competencia para hacerlo, al igual que la administración de cambios, cuando se vio operaciones de mantenimiento y actualización por el MySQL “Workbench” o por la interfaz línea de comandos.

A continuación, se verá una aproximación sobre cómo hacer y restaurar una copia de la base de datos.

## Copia de seguridad de base de datos con MySQL “Workbench”

Durante todo el ciclo de vida del “software” (desde que se construye hasta que está en operación), se necesitan operaciones de copias y restablecimiento de las bases de datos. Es por esto que se explicará una de las formas para obtener una copia de la base de datos, tanto en estructura como de los datos:

* Antes de cualquier cosa, se debe hacer una conexión a la base de datos si no existe dando clic en la casa y luego en el icono más “+”.
* Posteriormente, en test connection y luego en él se escribe la contraseña del “root”.
* Se le da doble clic a la conexión creada.
* Luego, se da clic en la pestaña, administración y Data Export.
* Se debe seleccionar:
* La base de datos a respaldar.
* Si se va a restaurar la estructura (metadatos) o solo los datos, o solo la estructura (se selecciona estructura y datos).
* La copia en un solo archivo y la ruta.
* Se crea la base de datos si no existe.
* Finalmente, en la carpeta se generará el “backup”.
* El proceso termina diciendo que la copia ha terminado.

A continuación, se verá cómo se restablece la copia que se ha generado, usando la herramienta MySQL “Workbench”.

## Restaurar una copia de seguridad con MySQL “Workbench”

El proceso complementario de restablecer una copia de base de datos es fundamental para la instalación de un sistema o el restablecimiento de la operación, cuando el sistema falla o se quiere volver a una versión anterior de la base de datos. Es por esto que se presentará un método sencillo para restablecer la base de datos a partir de un “backup” en formato SQL.

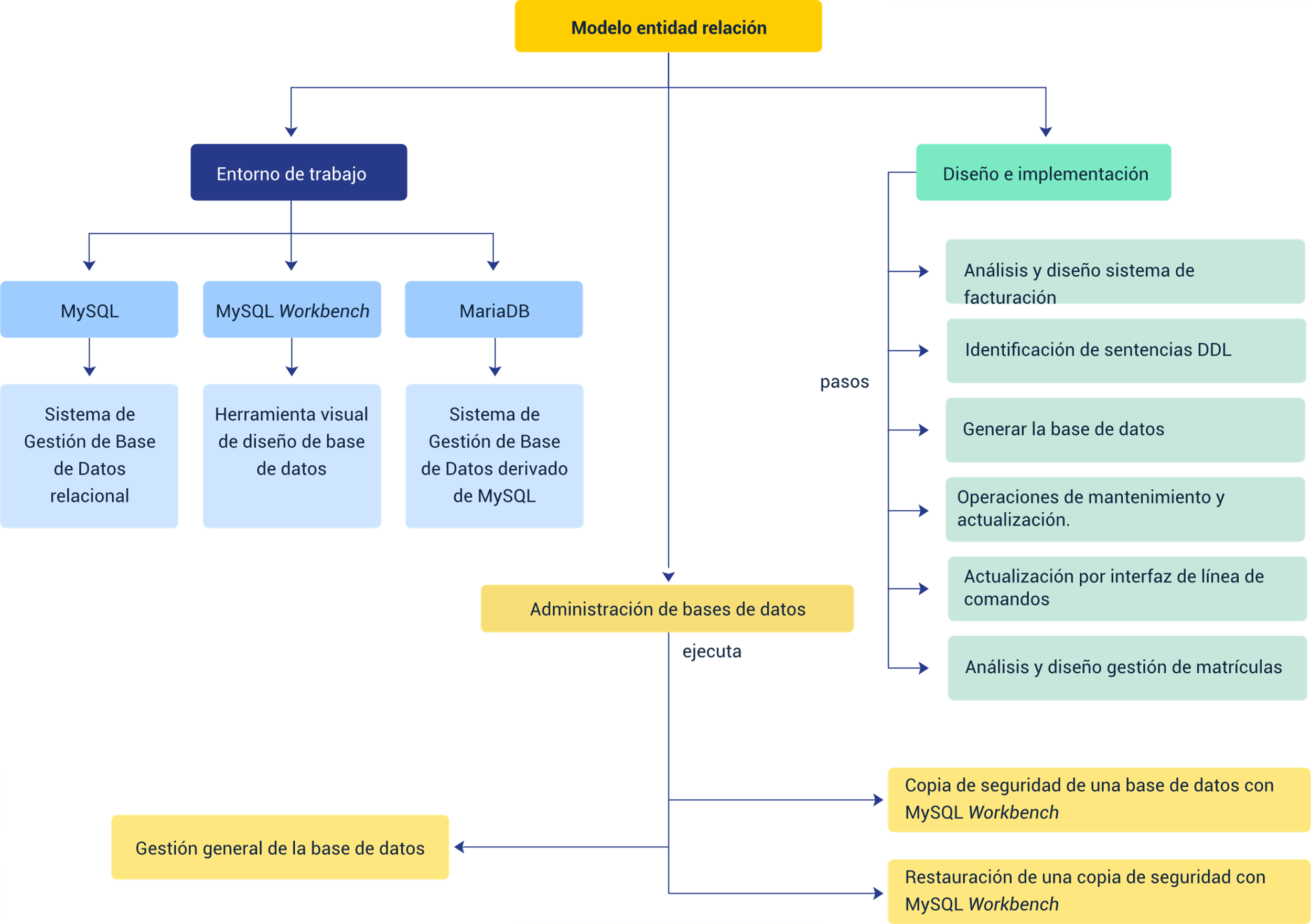
Siga los siguientes pasos luego de conectar a la base de datos con MySQL “Workbench” como se muestra a continuación.

1. Ingrese a la pestaña de administración.
2. Seleccione Data “Import/Restore”.
3. Marque la opción “Import from Self-Contained File”.
4. Busque el archivo generado antes.
5. Seleccione la base de datos a restaurar.
6. Haga clic en el botón “Star Import”.

Hasta este punto, puede tener la seguridad que está en capacidad de respaldar y restablecer la base de datos, por lo menos en un entorno de desarrollo y producción para bases pequeñas. Este proceso le puede servir para bases de datos grandes en las que debe usar la interfaz de comando “línea” con el comando “source”. Esto debido a que si la copia de seguridad es del tamaño de gigas, MySQL “Workbench” no podrá abrirlo y puede bloquearse, mientras que por la interfaz de línea de comando, el tamaño de la copia puede ser tan grande como se desee.

Síntesis

Lo invitamos a revisar el siguiente mapa conceptual, en el cual podrá hacer un breve recorrido sobre las temáticas abordadas.



Material complementario

| Tema | Referencia | Tipo de material | Enlace del recurso |
| --- | --- | --- | --- |
| Establecer un entorno de trabajo | Bustos, H. (2019). Ejemplo introductorio DDL+DML (MySQL y “Workbench”). YouTube. | Video | <https://www.youtube.com/watch?v=xwfzw9paFwo> |

Glosario

**Base datos**: conjunto ordenado y finito de operaciones que permite hallar la solución de un problema.

**DBA**: data base administrador es el rol que diseña un ingeniero responsable de las bases de datos en una organización.

**Diccionario de datos**: conjunto de programas, instrucciones y reglas informáticas para ejecutar ciertas tareas en una computadora.

**Metadatos**: conjunto unitario de instrucciones que permite a una computadora realizar funciones diversas como el tratamiento de textos, el diseño de gráficos, la resolución de problemas matemáticos y el manejo de bancos de datos.

**Modelo conceptual**: enfocado en prestar las entidades, los atributos y las relaciones entre dualidades con su multiplicidad.

**Modelo de datos**: representación de la base de datos a través de un diagrama.

“**Multitenant**”: tenencia múltiple o multiinquilino, hace referencia a aplicaciones que se diseñan para soportar el mismo sistema para distintos usuarios o empresas, a pesar de estar en la misma base de datos y usar el mismo código fuente el sistema sabe distinguir entre los datos de un usuario o empresa de los otros, sin confundirlos.

Referencias bibliográficas

Vertex42. (s.f.). Modelo de facturas gratis. <https://www.vertex42.com/es/excel-factura.html>

Wikipedia. (2021). MariaDB. Wikipedia, la enciclopedia libre. <https://es.wikipedia.org/w/index.php?title=MariaDB&oldid=135391766>

Wikipedia. (2021). MySQL. Wikipedia, la enciclopedia libre. <https://es.wikipedia.org/wiki/MySQL>

Créditos

| Nombre | Cargo | Centro de Formación y Regional |
| --- | --- | --- |
| Milady Tatiana Villamil Castellanos | Responsable del Ecosistema | Dirección General |
| Olga Constanza Bermúdez Jaimes | Responsable de Línea de Producción | Centro de Servicios de Salud - Regional Antioquia |
| Henry Eduardo Bastidas Paruma | Experto Temático | Centro de Teleinformática y Producción Industrial - Regional Cauca |
| Peter Emerson Pinchao Solís | Experto Temático | Centro de Teleinformática y Producción Industrial - Regional Cauca |
| Ana Catalina Córdoba Sus | Evaluadora Instruccional | Centro de Servicios de Salud - Regional Antioquia |
| Yerson Fabián Zárate Saavedra | Diseñador de Contenidos Digitales | Centro de Servicios de Salud - Regional Antioquia |
| Edward Leonardo Pico Cabra | Desarrollador Fullstack | Centro de Servicios de Salud - Regional Antioquia |
| Edgar Mauricio Cortés García | Actividad Didáctica | Centro de Servicios de Salud - Regional Antioquia |
| Luis Gabriel Urueta Álvarez | Validador de Recursos Educativos Digitales | Centro de Servicios de Salud - Regional Antioquia |
| Jaime Hernán Tejada Llano | Validador de Recursos Educativos Digitales | Centro de Servicios de Salud - Regional Antioquia |
| Margarita Marcela Medrano Gómez | Evaluador para Contenidos Inclusivos y Accesibles | Centro de Servicios de Salud - Regional Antioquia |
| Daniel Ricardo Mutis Gómez | Evaluador para contenidos inclusivos y accesibles | Centro de Servicios de Salud - Regional Antioquia |