UNIDAD 1: Estructura SQL – Estructura NoSQL

1. [UNIDAD 1: Introducción](https://learn.nextu.com/mod/page/view.php?id=3553&pid=P_WEBDEV_V2)

Bienvenido a Interactuando con Bases de Datos

En esta Unidad aprenderás qué es una base de datos y cómo funciona. En computación llamamos base de datos a un conjunto de datos que tienen una estructura organizacional que se guarda en un servidor. Estos servidores cuentan con un sistema gestor de base de datos que permite a los usuarios acceder a la información que se esté almacenando desde diferentes computadores. Además entenderás, con detalle, qué es una base de datos relacional y no relacional e instalarás un Sistema Gestor de Base de Datos orientado a objetos y relaciones llamado PostgreSQL.

Puntos de aprendizaje

**Unidad 1: Estructura SQL – Estructura NoSQL**  
¡Aquí comienza la Unidad 1: Estructura SQL – Estructura NoSQL!

Los objetivos del Programa que corresponden a esta Unidad son:

* **Usar metodologías sincrónicas y asincrónicas para gestionar datos desde el servidor.**
* **Crear ambientes de programación y a gestionar recursos.**

Los objetivos del Curso 7 que corresponden a esta Unidad son:

* **Aprender qué es y cómo funciona una base de datos.**
* **Asimilar, diseñar y crear bases de datos SQL.**
* **Entender, diseñar y construir bases de datos NoSQL.**

1. Lección 1: ¿Qué es una base de datos
   1. [Definición de base de datos](https://learn.nextu.com/mod/lesson/view.php?id=3554&pid=P_WEBDEV_V2)

Conjunto de datos que tienen una estructura organizacional que se guarda en un servidor 🡪 Los servidores cuentan con un gestor.

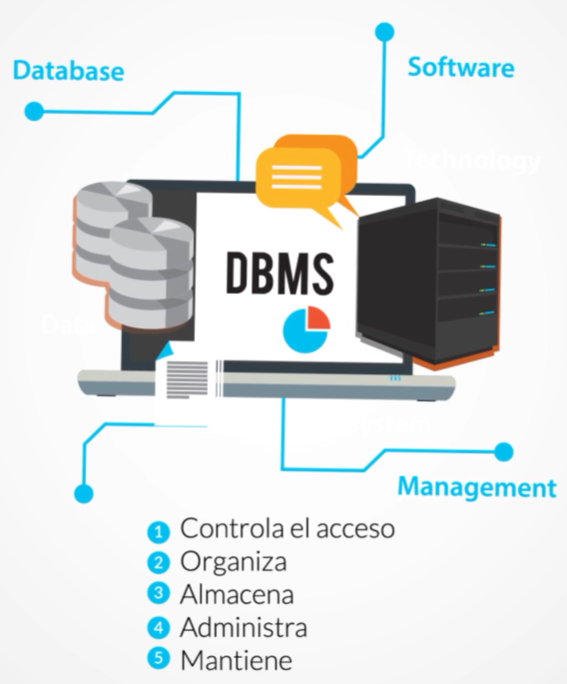
Requerimientos mínimos:



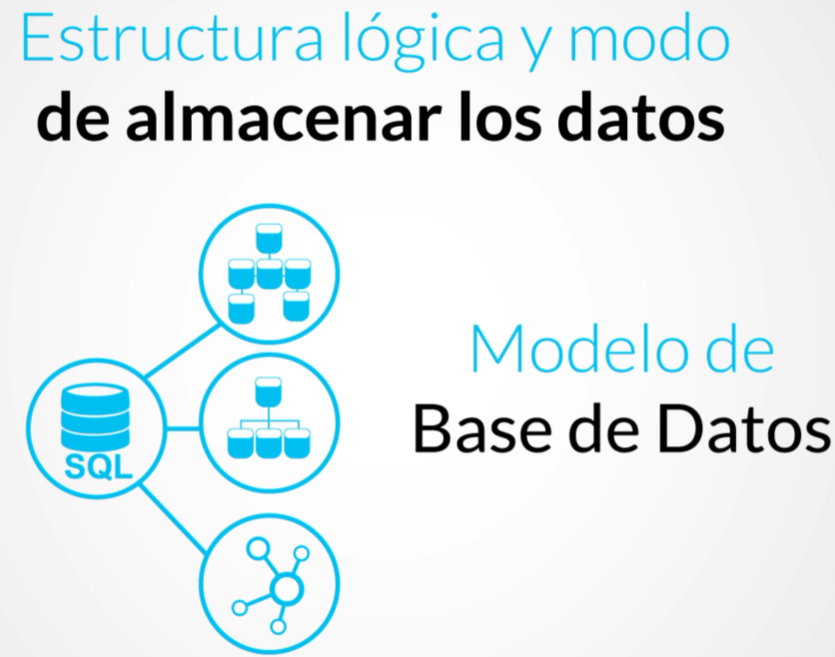
Especialidades:



Gestor de Datos CGBD o DBMS



* Calidad de datos (Redundancia puede provocar pérdida)



Los 3 más utilizados son:

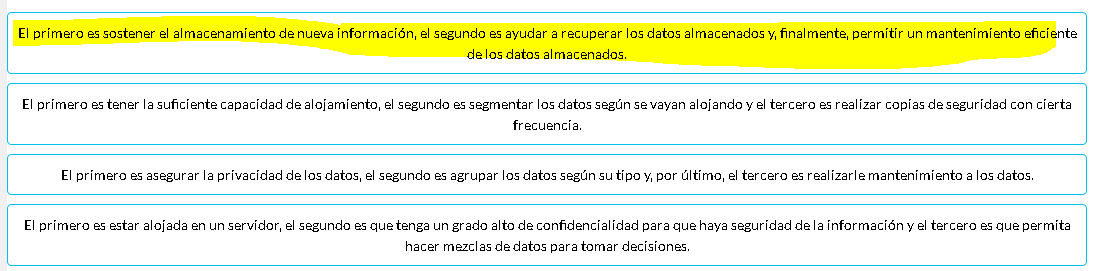
* Modelo Relacional – Es el más común (entidad – relación). Conexiones existentes, un registro es una fila de una tabla de la base de datos.
* Modelo Jerárquico – Organiza los datos en forma de árbol (padre – raíz)
* Modelo en Red - Se Construye en el Modelo Jerárquico (muchos – muchos). Registros padre múltiples



* 1. Actividad 1

Tiempo de pensar

En computación llamamos base de datos a un conjunto de datos que tienen una estructura organizacional que se guarda en un servidor, estos servidores cuentan con un sistema gestor de base de datos que permite a los usuarios acceder a la información que se esté almacenando desde diferentes computadores.  
  
Instrucciones:  
Lee el enunciado y, según las opciones de respuesta que se te plantean a continuación, elige el texto que lo complementa.  
  
Enunciado:  
Las bases de datos deben tener unos requerimientos mínimos:



* 1. Lectura: Por qué utilizar una base de datos

¿Por qué utilizar una Base de Datos?

Traducir:  
-DataBase Management System por Sistema de Gestión de Bases de datos  
-DBMS por SGBD  
-Programming por Programación  
-Data por Datos  
  
Como hemos aprendido en esta lección, las bases de datos son de mucha ayuda para almacenar datos en un servidor. Para que estos datos se puedan administrar de una manera más eficiente, existe un conjunto de aplicaciones que, unidas, forman un Sistema Gestor de Base de Datos (SGBD).  
  
Pero de esto nos surgen varios interrogantes: ¿Por qué utilizar una base de datos? ó ¿Por qué utilizar un SGBD? ¿Por qué no simplemente almacenar los datos en archivos de texto?

Las respuestas a estas preguntas se resuelven analizando las ventajas que tienen las bases de datos y los SGBDs. Entre estas ventajas tenemos:

1. Potencial para reforzar estándares
2. Flexibilidad
3. Aislamiento de los datos y el Software
4. Reducción del tiempo de desarrollo del Software
5. Seguridad
6. Restricción de Acceso No Autorizado
7. Compartir datos y sistema multiusuario
8. Backup y recuperación de datos

1. Potencial para reforzar estándares  
  
Una Base de Datos permite al Administrador (DBA) definir y reforzar estándares entre los usuarios que acceden a los datos en una organización, facilitando la comunicación y la cooperación entre varios departamentos, proyectos y usuarios en la organización.  
Los estándares pueden estar definidos por nombres y formatos de datos, reporte de estructuras, terminología, entre otros.  
En una base de datos, el administrador puede reforzar estándares de una manera centralizada sin tener el problema de asignar las reglas de uso de datos a cada usuario o a cada aplicación.

2. Flexibilidad  
  
En ocasiones, los requerimientos de la aplicación cambian y es necesario cambiar la estructura de la base de datos. Por ejemplo, puede surgir un nuevo grupo de usuarios que necesite información que no está actualmente en la base de datos. Los SGBDs modernos permiten algunos cambios en la estructura sin afectar los datos guardados y los programas de la aplicación existente.

3. Aislamiento de los datos y el Software  
  
Al utilizar una base de datos tienes los datos en un lugar independiente de tu aplicación, lo que permite el acceso desde diferentes lugares o programas. Esto facilita que los cambios en la estructura de datos sean manejados por el SGBD y no estén embebidos en el Software.

4. Reducción del tiempo de desarrollo del Software  
  
Diseñar e implementar una base de datos desde 0 puede tomar más tiempo que escribir un solo archivo especializado. Sin embargo, una vez que la base de datos está configurada y funcionando, generalmente se requiere menos tiempo para crear nuevas aplicaciones utilizando las facilidades de un SGBD.

5. Seguridad  
  
No todos los usuarios tienen los mismos permisos o privilegios de acceso a los datos. Los SGBDs facilitan el manejo y el control de la Seguridad de las Bases de Datos que almacena; permitiendo crear usuarios, grupos de usuarios y roles. A cada uno de estos grupos se les pueden asignar unos permisos y privilegios.

6. Restricción de Acceso No Autorizado  
  
Cuando múltiples usuarios comparten una base de datos es probable que algunos usuarios no estén autorizados para acceder a toda la información. Por ejemplo, para datos confidenciales, algunos usuarios pueden estar autorizados sólo para devolver datos, mientras que otros pueden estar autorizados para devolver y actualizar. Sólo los administradores de la base de datos deben tener permitido ciertos privilegios, como crear nuevas cuentas o asignar permisos de acceso.

7. Compartir datos y sistema multiusuario  
  
Un sistema de base de datos multiusuario debe permitir el acceso de múltiples usuarios a la base de datos al mismo tiempo. Por lo tanto, el sistema gestor de la base de datos debe tener control del acceso al mismo tiempo que la información, garantizando que no se generen datos repetidos con diferente información, que siempre estén actualizados y que éstos sean correctos. A esta característica se le llama **“integridad de datos”.**

8. Backup y recuperación de datos  
  
Un Sistema de Base de Datos proporciona la capacidad de guardar los datos en un momento determinado y esta copia se guarda independiente del SGBD, a este proceso se le conoce como backup o copia de seguridad, lo cual se hace para prevenir la pérdida de datos por fallos en el Software, Hardware o errores del usuario al momento de interactuar con la base de datos.  
El sistema de recuperación y el backup son los responsables de la restauración del sistema. Si un sistema falla en medio de un programa complejo de actualización, el SGBD debe restaurar la base de datos al estado en el que estaba antes de que el programa comenzara a ejecutarse.

2.4. Base de datos relacional

Es un conjunto de datos organizados en tablas que se relacionan para permitir un almacenamiento práctico y útil de la información para la toma de decisiones.

* Tabla 🡪 Información Ej: Clientes 🡪 Número de identificación, Nombre 🡪 Registro o Tupla

Número\_Cuenta 🡪 Cliente y Cuenta es una relación: 1 a 1, muchos a 1 y muchos a muchos

Fr 🡪 Foreign Key

* 1. Actividad 2

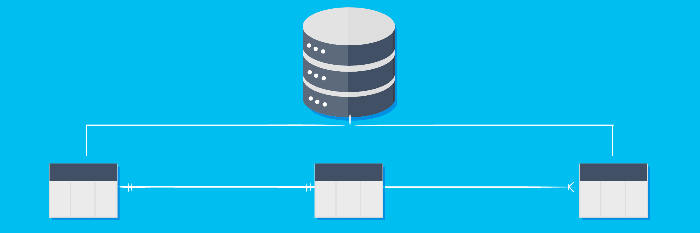
Tiempo de pensar

Recuerda que la tabla es la unidad fundamental de una base de datos relacional, sin embargo por sí sola no significa nada, pues no hay datos en ella. Según lo aprendido en la lección, una relación de uno a uno significa que un sólo elemento de una entidad o tabla se relaciona con un sólo elemento de otra.  
  
Instrucciones:  
  
Lee con atención la pregunta y, una vez lo tengas claro, selecciona dos opciones de respuesta.  
  
Pregunta:  
  
¿Cuáles de las siguientes opciones de respuesta plantean bases de datos relacionales uno a uno?

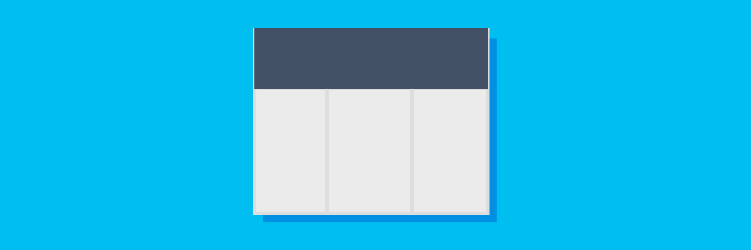


* 1. Lectura: Conceptos y restricciones

Bases de datos: Conceptos y restricciones



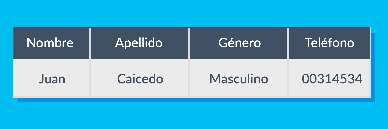
Recordemos que una base de datos relacional es una colección de elementos de datos organizados en múltiples tablas descritas detalladamente, desde las que podemos acceder y/o gestionar los datos de diversas maneras sin tener la necesidad de reorganizar la base de datos. Después de mencionar lo anterior, es momento de que veamos los conceptos y restricciones que nos van a garantizar el buen funcionamiento de nuestra base de datos.



**CONCEPTOS**  
  
**1. Tablas:** dentro de una base de datos relacional, las tablas son las estructuras que se encargan de almacenar y/o alojar la información de la base de datos. Las tablas son usadas para ordenar y entregar la información almacenada, éstas están compuestas por filas y columnas, las cuales pueden ser llenadas con cualquier tipo de datos.



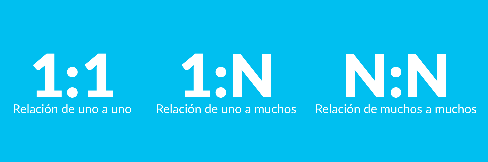
**2. Campos o atributos:** los campos o atributos de una tabla son cada una de las columnas que la componen, donde cada campo almacena un dato en concreto, por ejemplo, el campo **‘Nombre’.** A cada campo se le asigna el tipo de dato que almacenará, es decir, si el dato es un número, una cadena de texto, un dato booleano, etc., con el fin de darle un orden a la información que se está almacenando.



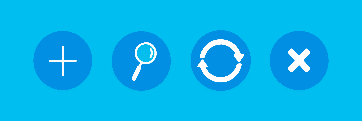
**3. Registros:** los registros son cada una de las filas que componen la tabla y su función es agrupar toda la información de un mismo elemento.



**4. Llaves o Claves:** una clave o llave es un campo en el cual su valor o contenido no puede estar duplicado dentro de la misma tabla, ya que éste nos permitirá identificar y diferenciar cada registro de los demás. Los principales tipos de llaves o claves que existen son:  
**- Llave o clave primaria (Primary Key - PK):** es un campo único e irrepetible que nos permite identificar y diferenciar la información de una tabla.  
**- Llave o clave foránea (Foreign Key - FK):** la clave foránea es una clave secundaria que se utiliza en una tabla, haciendo referencia a la clave primaria de la tabla con la cual está relacionada.  
**- Llave o clave candidata:** también conocida como clave única, esta llave es usada cuando aún no se define una clave o llave primaria. Se conoce como clave o llave candidata ya que es uno de los datos que no tiene tendencia a repetirse en otros registros.



**5. Relación:** dentro de una base de datos relacional, una relación es un vínculo que se establece entre dos o más tablas, el cual permite trabajar con todos los datos de éstas como si estuvieran en una sola tabla. Existen 3 tipos de relaciones entre tablas y son los siguientes:  
**- Relación de uno a uno:** es el tipo de relación menos frecuente entre tablas y sucede cuando un solo elemento de una entidad o tabla se relaciona con un solo elemento de otra. Un ejemplo de esta relación lo encontramos entre países y capitales: un país sólo puede tener una capital y una capital sólo puede estar en un país.  
**- Relación de uno a muchos:** es el tipo más frecuente de relación entre tablas. Una relación uno a muchos se da cuando un registro de una tabla se relaciona con más de un registro de otra. Por ejemplo, país y ciudad es una relación de uno a muchos porque un país tiene muchas ciudades, pero una ciudad sólo puede pertenecer a un país.  
**- Relación de muchos a muchos:** es aquella en la que los registros de dos tablas pueden relacionarse con más de un elemento de otra tabla. Por ejemplo, la relación entre personas y ciudades de residencia: una persona puede vivir en más de una ciudad durante su vida y, a su vez, una ciudad alberga más de una persona.



**6. Consultas:** en las bases de datos, una consulta es el método por medio del cual podemos acceder a los datos de nuestra base de datos; éstas nos permiten insertar, eliminar, modificar y borrar los datos de nuestras bases de datos a través de un lenguaje de consultas. En la actualidad el más utilizado es el SQL, cuyas siglas en inglés significan Structured Query Lenguage, y en español significa Lenguaje de Consulta Estructurada.  
  
Como vimos anteriormente, existen múltiples conceptos a tener presentes en el momento de trabajar con bases de datos relacionales, pero es necesario saber qué restricciones se presentan.  
  
**Restricciones:** las restricciones en las bases de datos relacionales son un conjunto de reglas y/o condiciones que se deben cumplir al pie de la letra para el manejo adecuado de los datos dentro de las tablas que la componen, con el fin de garantizar la integridad de los datos para realizar cualquier tipo de operación con ellos. Algunas de estas restricciones no son definidas por el usuario, ya que al trabajar con bases de datos relacionales, éstas ya tienen definidas ciertas restricciones que el usuario debe respetar, mientras que otras son definidas por el mismo usuario, por ejemplo: el tipo de dato de un campo, la longitud de caracteres, si el campo es obligatorio o no.

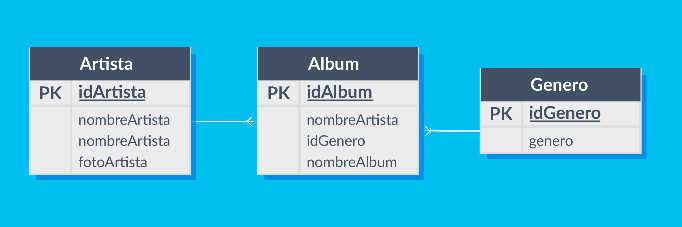
Estas son algunas de las restricciones más comunes:

1. Ninguna llave primaria puede ser un valor nulo.
2. Para que una llave foránea exista dentro de una tabla, debe ser delegada previamente como la llave primaria en la tabla con la cual se relaciona.
3. Los valores que se ingresan en una columna de la tabla deben corresponder al tipo de dato que se le asignó previamente al campo, es decir, si se creó el campo ‘Fecha’, el valor que éste debe almacenar no debe ser numérico o de texto, sino que debe ser de tipo fecha, por ejemplo: 11-11-1992 o 11/11/1992, según el formato que éste maneje.
4. Los campos que se marquen como obligatorios no pueden ser nulos o estar vacíos.
5. Los campos que forman una clave candidata deben tomar siempre valores distintos para cada posible registro.
   1. Lectura: Normalización

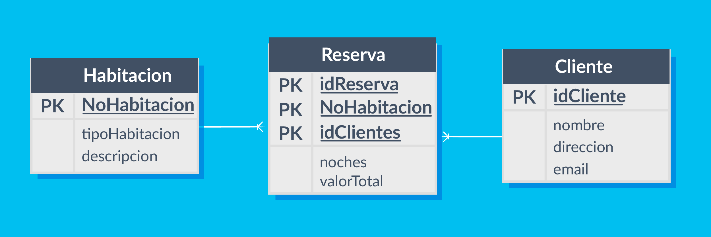
Normalización

Como mencionamos anteriormente, una base de datos relacional está compuesta de tablas que se relacionan una a otra a través de llaves. Las columnas de la base de datos se llaman atributos y las tablas se relacionan a través de una o varias llaves, esto depende de la base de datos que estemos trabajando. Cuando una llave primaria está compuesta de varios campos la llamamos **Llave Compuesta o Compuound Key.**

**Llave primaria**



**Llave Compuesta (Compound Key)**



**Llave Compuesta → idReserva+NoHabitacion+idCliente**

Pero para llegar a tener una base de datos en la que las tablas tengan información que no se repita, hay que seguir un proceso que se llama Normalización, que tiene como propósito organizar las columnas de las tablas con sus respectivas relaciones para eliminar la redundancia y mejorar la integridad de los datos.  
  
El proceso de Normalización está dividido en tres conjuntos de reglas:  
1. Primera Forma Normal (1FN)  
2. Segunda Forma Normal (2FN)  
3. Tercera Forma Normal (3FN)

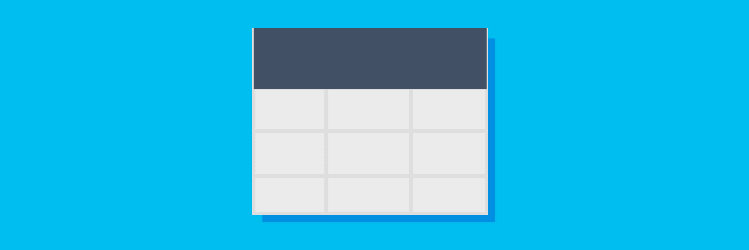


Según Edgar Frank Codd, cada uno de estos conjuntos de reglas tienen el propósito de solucionar o prevenir los siguientes problemas:

* Quitar de la colección de datos las dependencias no deseadas al momento de insertar, actualizar o eliminar datos.
* Reducir la necesidad de reestructurar las relaciones entre los datos cuando se añaden nuevos atributos o nuevas tablas a la base de datos.
* Hacer que la colección de relaciones sea neutral a las estadísticas de consulta, donde estas estadísticas pueden cambiar a lo largo del tiempo.
* Hacer que el modelo relacional proporcione más información a los usuarios de las bases de datos.

Aunque en la práctica no todas las reglas se cumplen, entre más se cumplan se diría que una base de datos es ‘más relacional’.  
Para entender cómo ejecutar el proceso de normalización, partiremos de la siguiente base de datos que está representada por esta tabla no normalizada.





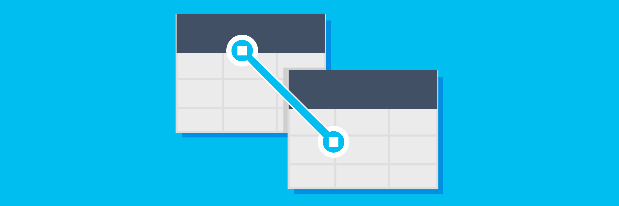
1. Primer nivel de normalización:

* Eliminar atributos repetidos dentro de una relación.
* Agrupar los elementos repetidos en una tabla.
* Identificar cada fila de la tabla con una columna.

Para nuestro ejemplo, esto se transforma en reducir los atributos “dirección1” y “dirección2”, que pertenecen al mismo contacto. Como resultado tenemos la siguiente tabla:



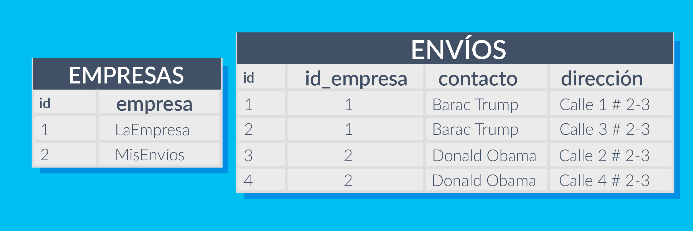
Por lo que podemos ver, para la relación “envíos” no existen campos repetidos pero hay filas que generan redundancia, como la repetición del nombre de un contacto o el nombre de la empresa. Para esto aplicaremos el segundo nivel de normalización.



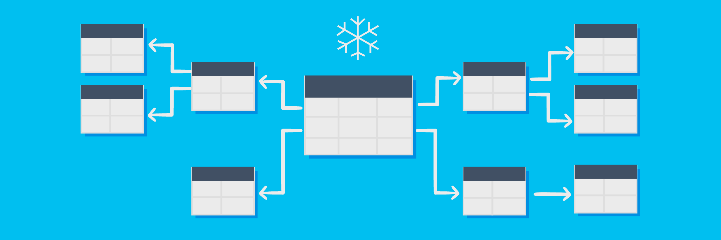
2. Segundo nivel de normalización:

* Separar en una tabla los datos que se repiten en varias filas que puedan generar errores en la integridad de los datos.
* Identificar la nueva tabla de relaciones con claves primarias.

Para el caso de la relación “envíos”, la columna “empresa” contiene datos que pueden corromper la integridad cuando se vaya a actualizar el nombre de una empresa. Esto genera como resultado una tabla de “empresas” y otra tabla de “envíos”.



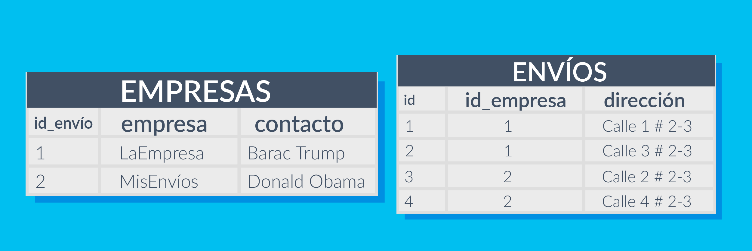
Vemos que se agregó la relación “empresas” y que se relaciona con la tabla principal de envíos mediante una clave foránea. Ahora veamos cómo aplicar el tercer nivel de normalización.



3. Tercer nivel de normalización:

* Eliminar los campos que pueden ser claves únicas y que no están relacionados con la clave primaria en tablas aparte.

En el caso de la base de datos de envíos, este campo sería el atributo “contacto” y puede ser agrupado en la relación existente entre las empresas, así quedamos con las siguientes tablas:



Estos tres niveles de normalización son los que se usan principalmente en el diseño de las bases de datos, pero puedes consultar por tu cuenta: la forma normal de Boyce-Codd, la cuarta Forma Normal (4FN) o la quinta Forma Normal (5FN).

1. Lección 2: Lenguaje SQL – DDL
   1. Instalación de PostgreSQL
   2. Actividad 1
   3. Diseño de Base de datos Relacional
   4. Actividad 2
   5. Actualizar registros
   6. Creación de relaciones y restricciones
   7. Ejercicio 1
   8. Ejercicio 2
2. Lección 3: Lenguaje SQL – DML
3. Lección 4: Qué es y cómo funciona una base de datos tipo No Relacional
4. Lección 5: Realizar consultas en base de datos No Relacional
5. 7. UNIDAD 1: Prueba