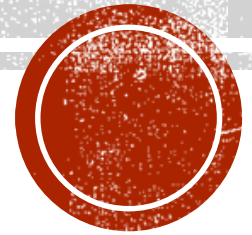


DEPENDENCIA FUNCIONAL Y FORMAS NORMALES



REGLAS DE NORMALIZACIÓN

¿Qué es normalizar?

- Es eliminar la **REDUNDANCIA (repetición)** y las **INCONSISTENCIAS** de dependencia en el **DISEÑO** de las tablas.
- Es el proceso de organizar los datos de una base de datos, se incluye la creación de tablas y el establecimiento de relaciones entre ellas según reglas diseñadas tanto para proteger los datos como para hacer que la base de datos sea más flexible



REGLAS DE NORMALIZACIÓN

- Hay algunas reglas en la normalización de una base de datos, las cuales se denominan como **"forma normal"**.
- Existen 5 formas normales en donde cada una, determina las características que debe ir adquiriendo la base de datos.
- Si se cumple la primera regla, se dice que la base de datos está en la "primera forma normal". Si se cumplen las tres primeras reglas, la base de datos se considera que está en la "tercera forma normal". La tercera forma normal se considera el máximo nivel necesario para la mayor parte de las aplicaciones.



El Problema

Crear una base de datos que almacene la información de los clientes de la empresa COMERCIAL “SAN LUIS” que permita comunicarles novedades de los productos, promociones, por correo electrónico.



LISTADO DE LA INFORMACIÓN DE LOS CLIENTES

- Nombre del cliente
- Nombre de la Empresa
- Dirección empresa
- E-mail (todos los que pueda tener)



PRIMERA FORMA NORMAL

Tabla usuarios				
Nombre	Empresa	Dir_empresa	Email1	Email2
Carlos Pérez	Telecom	Col. Buenavista #502	carlospe@telecom.com	carlitos@hotmail.com
María Castro	Digital Co	Ave. Olímpica #30	maricastro@digital.sv	Maria1984@yahoo.com

¿Qué hacer si necesitamos colocar otra e-mail?



Creamos un campo nuevo email3?

No!

Debemos crear un sistema de base de datos funcional que pueda crecer y adaptarse a los nuevos requisitos.



PRIMERA FORMA NORMAL - 1NF

1. Elimine los grupos de columnas repetidos de las tablas.
2. Identifique cada conjunto de datos relacionados con una clave principal.



LA TABLA EN PRIMERA FORMA NORMAL

Usuario				
Código	Nombre	Empresa	Dir_empresa	Email
CP0203	Carlos Pérez	Telecom	Col. Buenavista #502	carlospe@telecom.com
CP0203	Carlos Pérez	Telecom	Col. Buenavista #502	carlitos@hotmail.com
MC0204	María Castro	Digital Co	Ave. Olímpica #30	maricastro@digital.sv
MC0204	María Castro	Digital Co	Ave. Olímpica #30	Maria1984@yahoo.com

Definimos clave primaria

Eliminamos grupos repetitivos



SEGUNDA FORMA NORMAL – 2NF

1. Crear tablas separadas para aquellos grupos de datos que se aplican a varios registros.
2. Relacionar éstas tablas mediante una clave externa.



LA TABLA EN SEGUNDA FORMA NORMAL

Usuario			
Código	Nombre	Empresa	Dir_empresa
CP0203	Carlos Pérez	Telecom	Col. Buenavista #502
MC0204	María Castro	Digital Co	Ave. Olímpica #30

Relacionamos
las tablas

Grupos
de datos

E-mails		
Numero	Código	Email
1	CP0203	carlospe@telecom.com
2	CP0203	carlitos@hotmail.com
3	MC0204	maricastro@digital.sv
4	MC0204	Maria1984@yahoo.com



TERCERA FORMA NORMAL – 3NF

1. Eliminar aquellos campos que no dependan de la clave primaria.



LAS TABLAS EN TERCERA FORMA NORMAL

Usuario		
Código	Nombre	Cod_empresa
CP0203	Carlos Pérez	Tele01
MC0204	María Castro	Digi02

Los datos de la empresa no dependen del código del usuario

Relacionamos las tablas

Empresa		
Cod_empresa	Empresa	Dir_empresa
Tele01	Telecom	Col. Buenavista #502
Digi02	Digital Co	Ave. Olímpica #30

E-mails		
Numero	Código	Email
1	CP0203	carlospe@telecom.com
2	CP0203	carlitos@hotmail.com
3	MC0204	maricastro@digital.sv
4	MC0204	Maria1984@yahoo.com



RELACIONES ENTRE LOS DATOS

Generalmente encontraremos relaciones uno a varios

- Uno a uno
- Uno a varios
- Varios a varios



INTEGRIDAD DE LOS DATOS

- La exigencia de integridad de los datos garantiza la calidad de los datos de la base de datos.
- Por ejemplo, si se especifica para un empleado el valor de código **C1037**, la base de datos no debe permitir que otro empleado tenga el mismo valor como código.



INTEGRIDAD DE LOS DATOS

- Si en la tabla hay una columna **dept_id** en la que se almacena el número de departamento que se encuentra entre **1 y 5**, la base de datos sólo debe permitir valores que correspondan a los números de departamento de la empresa ya definidos.



La función de la normalización es favorecer la integridad de los datos, sin importar la actividad que se desarrolle sobre la base de datos. Trata de evitar lo máximo posible la posibilidad de introducir datos que no sean correctos. Dentro del proceso de normalización podemos distinguir cuatro tipo de integridades:

- **Integridad de entidad**
- **Integridad de dominio**
- **Integridad referencial**
- **Integridad definida por el usuario**



INTEGRIDAD DE ENTIDAD

- La integridad de entidad define una fila como entidad única para una tabla determinada. La integridad de entidad exige la integridad de las columnas de los identificadores o la clave principal de una tabla, mediante índices y restricciones UNIQUE, o restricciones PRIMARY KEY o propiedades de IDENTITY.



INTEGRIDAD DE DOMINIO

- La integridad de dominio viene dada por la validez de las entradas para una columna determinada.
- Puede exigir la integridad de dominio para restringir el tipo mediante los TIPOS DE DATOS, el formato mediante reglas y restricciones como CHECK, o el intervalo de valores posibles mediante restricciones FOREIGN KEY, definiciones DEFAULT, y definiciones NOT NULL .



INTEGRIDAD REFERENCIAL

- La integridad referencial protege las relaciones definidas entre las tablas cuando se crean o se eliminan filas.
- En SQL Server la integridad referencial se basa en las relaciones entre claves externas y claves principales mediante restricciones FOREIGN KEY y CHECK.
- La integridad referencial garantiza que los valores de clave sean coherentes en las distintas tablas.



La integridad referencial también vigila que se cumplan las siguientes reglas:

- No se podrá introducir un valor en la tabla relacionada si antes no ha sido introducida en la tabla principal.
- No se puede eliminar un registro de una tabla principal si existen registros coincidentes en la tabla relacionada.
- No se puede cambiar un valor de la clave principal (Primary key) en la tabla principal si el registro tiene registros relacionados.



INTEGRIDAD DEFINIDA POR EL USUARIO

- La integridad definida por el usuario permite definir reglas de empresa específicas que no pertenecen a ninguna otra categoría de integridad. Todas las categorías de integridad admiten la integridad definida por el usuario. Esto incluye todas las restricciones de nivel de columna y nivel de tabla en CREATE TABLE, procedimientos almacenados y desencadenadores.



RESTRICCIONES

- Una restricción es una limitación . Situada en cualquier columna o tabla, una restricción asegura que nuestros datos satisfacen determinadas **reglas de integridad de datos**.



UNIQUE

- Esta restricción obliga a que todos los valores de una determinada columna no estén repetidos en otros registros. Si tenemos varias restricciones UNIQUE en una misma tabla, todas deben ser cumplidas a la vez para cada registro.
- Con la restricción UNIQUE aseguramos la integridad de identidad de la tabla, ya que cumplimos con la norma de que cada registro es diferente al resto. Si aplicamos claves principales a una tabla, automáticamente se asigna esta restricción a esa columna.



DEFAULT

- Como su propio nombre indica, esta restricción introduce un valor por defecto en una columna cuando no se indica ningún valor para insertar.
- Con esta restricción aseguramos la integridad de dominio, ya que aseguramos valores válidos para nuevos registros que se inserten.



CHECK

- Esta restricción evalúa por medio de expresiones los valores que se insertan en una columna.
- Esta expresión, una vez que se evalúa devuelve un resultado, en función de si el dato es válido (Verdadero) o no (Falso), por lo tanto devuelve un valor booleano que indica si el dato tendrá permiso para ser ingresado o no.



PRIMARY KEY

- La clave principal (PRIMARY KEY) nos permite asegurar la integridad de entidad (puesto que es única en cada registro) y por otro lado nos garantiza la estabilidad de las relaciones con otras tablas.

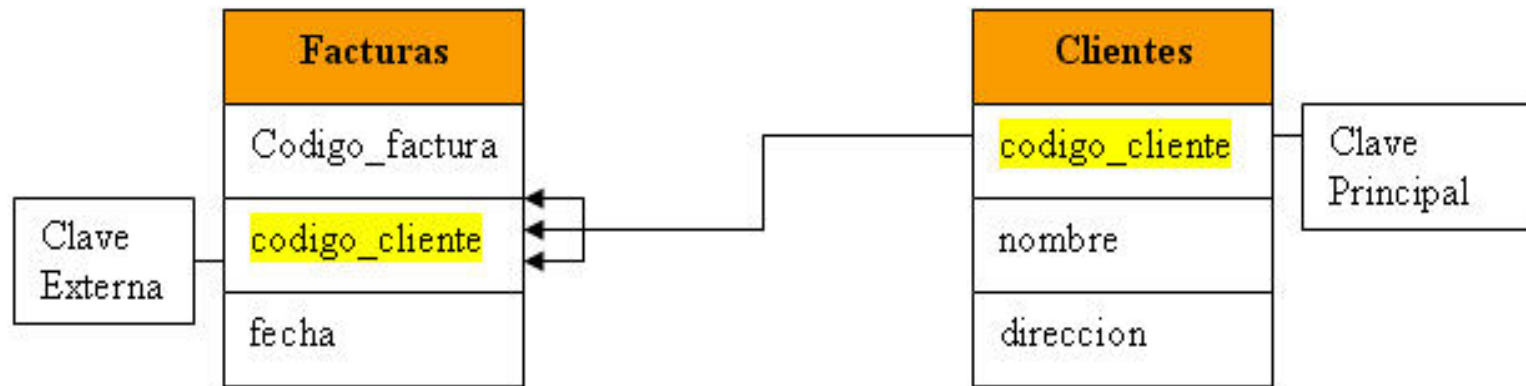


FOREIGN KEY

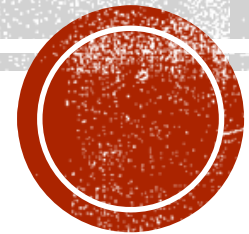
- La restricción FOREIGN KEY (clave externa o foránea), esta se le conoce como la pareja de la restricción PRIMARY KEY, y juntas cumplen con la integridad referencial.
- Una clave externa es una copia de la clave principal de la tabla principal, se inserta en la tabla que se pretende enlazar y con esto creamos la relación entre un par de tablas. Las claves externas pueden ser varias en una misma tabla, mientras que las principales deben ser únicas.



EJEMPLO DE RESTRICCIONES



MODELO RELACIONAL



MODELO RELACIONAL

Este es el modelo más utilizado en la actualidad para modelar problemas reales y administrar datos dinámicamente.

Tras ser postulados sus fundamentos en 1970 por Edgar Frank Codd, de los laboratorios IBM en San José (California), no tardó en consolidarse como un nuevo paradigma en los modelos de base de datos.



- Su idea fundamental es el uso de "relaciones". Estas relaciones podrían considerarse en forma lógica como conjuntos de datos llamados "tuplas".
- En este modelo, el lugar y la forma en que se almacenen los datos no tienen relevancia (a diferencia de otros modelos como el jerárquico y el de red). Esto tiene la considerable ventaja de que es más fácil de entender y de utilizar para un usuario eventual de la base de datos.



- La información puede ser recuperada o almacenada mediante "consultas" que ofrecen una amplia flexibilidad y poder para administrar la información.
- El lenguaje más habitual para construir las consultas a bases de datos relacionales es SQL, Structured Query Language o Lenguaje Estructurado de Consultas, un estándar implementado por los principales motores o sistemas de gestión de bases de datos relacionales.
- Durante su diseño, una base de datos relacional pasa por un proceso al que se le conoce como normalización de una base de datos.



BASES DE DATOS RELACIONALES

- Una base de datos relacional almacena los datos en varios lugares denominados tablas y además controla la manera en que esas tablas se relacionan entre sí.
- En ocasiones encontrará las siglas RDBMS, del inglés Relational Database Management System, para hacer referencia a una base de datos relacional.



TABLAS

- Las tablas son los objetos que almacenan los datos, una de las normas básicas para la base de datos es que cada tabla debería almacenar información sobre una entidad en concreto esto se conoce como regla de normalización.
- Gran parte del trabajo que se realiza con una base de datos gira en torno a las tablas, cada base de datos admite estas cuatro operaciones básicas:
 1. Añadir información a una tabla
 2. Actualizar la información que ya existe en una tabla
 3. Eliminar información que ya existe en una tabla
 4. Ver la información contenida en una tabla



REGISTROS, CAMPOS Y VALORES

- Cada tabla se compone de registros y campos, registro es toda la información sobre una de las entidades dentro de una tabla, campo es un dato en concreto almacenado en una tabla.

	CategoryID	CategoryName	Description	Picture
1	1	Beverages	Soft drinks, coffees, teas, beers, and ales	0x151C2F00020000000D000E0014002100FFFFFFFF
2	2	Condiments	Sweet and savory sauces, relishes, spreads, and ...	0x151C2F00020000000D000E0014002100FFFFFFFF
3	3	Confections	Desserts, candies, and sweet breads	0x151C2F00020000000D000E0014002100FFFFFFFF
4	4	Dairy Products	Cheeses	0x151C2F00020000000D000E0014002100FFFFFFFF
5	5	Grains/Cereals	Breads, crackers, pasta, and cereal	0x151C2F00020000000D000E0014002100FFFFFFFF
6	6	Meat/Poultry	Prepared meats	0x151C2F00020000000D000E0014002100FFFFFFFF
7	7	Produce	Dried fruit and bean curd	0x151C2F00020000000D000E0014002100FFFFFFFF
8	8	Seafood	Seaweed and fish	0x151C2F00020000000D000E0014002100FFFFFFFF

En la tabla anterior se puede identificar como:

Campo: CategoryID

Registro: 1, Bevarages, Soft drinks, coffees, teas etc.



PROPIEDADES DE CAMPO

- Una base de datos completa como SQL Server le permite crear las diferencias entre cada campo especificando propiedades de campo:

	Column Name	Data Type	Allow Nulls
▶	CategoryID	int	<input type="checkbox"/>
	CategoryName	nvarchar(15)	<input type="checkbox"/>
	Description	ntext	<input checked="" type="checkbox"/>
	Picture	image	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Esta vista muestra la información del esquema de la tabla Categories en lugar de los datos, el esquema de una base de datos es una manera de hacer referencia a toda la información de diseño que determina lo que se puede almacenar en esa base de datos.



PROPIEDADES DE CAMPO

- Se muestran las tres propiedades más importantes de cada campo de la tabla:
 1. El nombre de la columna (Column Name)
 2. El tipo de datos (Data Type)
 3. Si se permiten los valores nulos (Allow Nulls)
-
- Por medio del nombre de la columna de un campo (o columna) se puede hacer referencia a ese campo en la tabla, por lo general se asignaran nombres significativos a los campos.



PROPIEDADES DE CAMPO

- El tipo de dato de un campo determina los datos que se pueden almacenar en él, por ejemplo:

varchar(30)	Texto con un tamaño máximo de 30 caracteres
datetime	Información de fecha y hora
int	Números enteros

La propiedad de longitud de un campo especifica la cantidad máxima de datos que se puede almacenar en él.

La propiedad permitir valores nulos muestra si se permiten valores nulos en ese campo, si un campo no permite valores nulos, debe ingresarse un valor no nulo para ese campo en cada registro antes de poder guardar el registro.




PROPIEDADES DE LAS RELACIONES

Las relaciones tienen las siguientes características:

- Cada relación tiene un nombre y éste es distinto del nombre de todas las demás.
- Los valores de los atributos o campos son atómicos: en cada tupla, cada atributo toma un solo valor. Se dice que las relaciones están *normalizadas*.
- No hay dos atributos que se llamen igual.
- El orden de los atributos no importa: los atributos no están ordenados.
- Cada tupla es distinta de las demás: no hay tuplas duplicadas.
- El orden de las tuplas no importa: las tuplas no están ordenadas.



CLAVES Y RESTRICCIONES

	Column Name	Data Type	Allow Nulls
	CategoryID	int	<input type="checkbox"/>
	CategoryName	nvarchar(15)	<input type="checkbox"/>
	Description	ntext	<input checked="" type="checkbox"/>
	Picture	image	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

En la imagen anterior se observará que al lado izquierdo del campo CategoryID posee un símbolo de llave, esta indica que esa columna es la clave principal de la tabla.

Una **clave principal** es una información de identificación exclusiva que le permite buscar un registro determinado dentro de una tabla, en la misma tabla no puede haber dos registros con el mismo valor en el campo de la clave principal, la clave principal podría estar compuesta por un solo campo o por varios



PRIMARY KEY

- Al crear una tabla puede crear una sola restricción PRIMARY KEY como parte de la definición de tabla. Si la tabla ya existe, puede agregar una restricción PRIMARY KEY, siempre que no exista ya otra restricción PRIMARY KEY. Una tabla puede contener una sola restricción PRIMARY KEY.
- Cuando se agrega una restricción PRIMARY KEY a una o varias columnas de la tabla, el Motor de base de datos examina los datos y metadatos de columnas existentes para garantizar que se cumplen las siguientes reglas en las claves principales:

1. Las columnas no admiten valores NULL.

Las columnas de restricciones PRIMARY KEY que se especifiquen al crear la tabla se convierten implícitamente a NOT NULL. Una columna dispersa no se puede utilizar como parte de una clave principal porque las columnas dispersas deben permitir valores NULL.

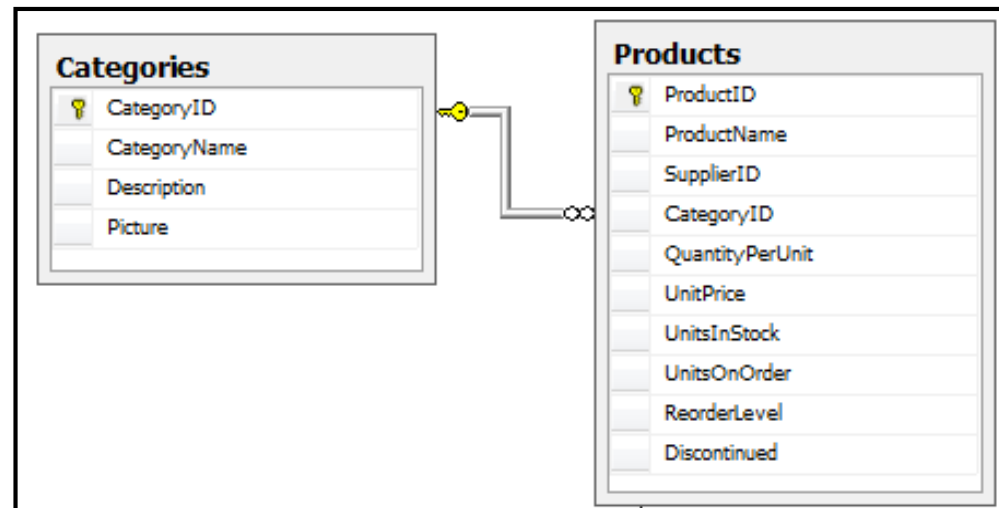
2. No puede haber valores duplicados.

Si se agrega una restricción PRIMARY KEY a una columna que tiene valores duplicados o permite valores NULL, el Motor de base de datos devuelve un error y no agrega la restricción.



CLAVES EXTERNAS O FORÁNEAS

- Una clave externa es un campo (o campos) que señala la clave principal de otra tabla. El propósito de la clave externa es asegurar la integridad referencial de los datos. En otras palabras, sólo se permiten los valores que se esperan que aparezcan en la base de datos.
- La integridad referencial asegura que se mantengan las referencias entre las claves principales y las externas. Por ejemplo:



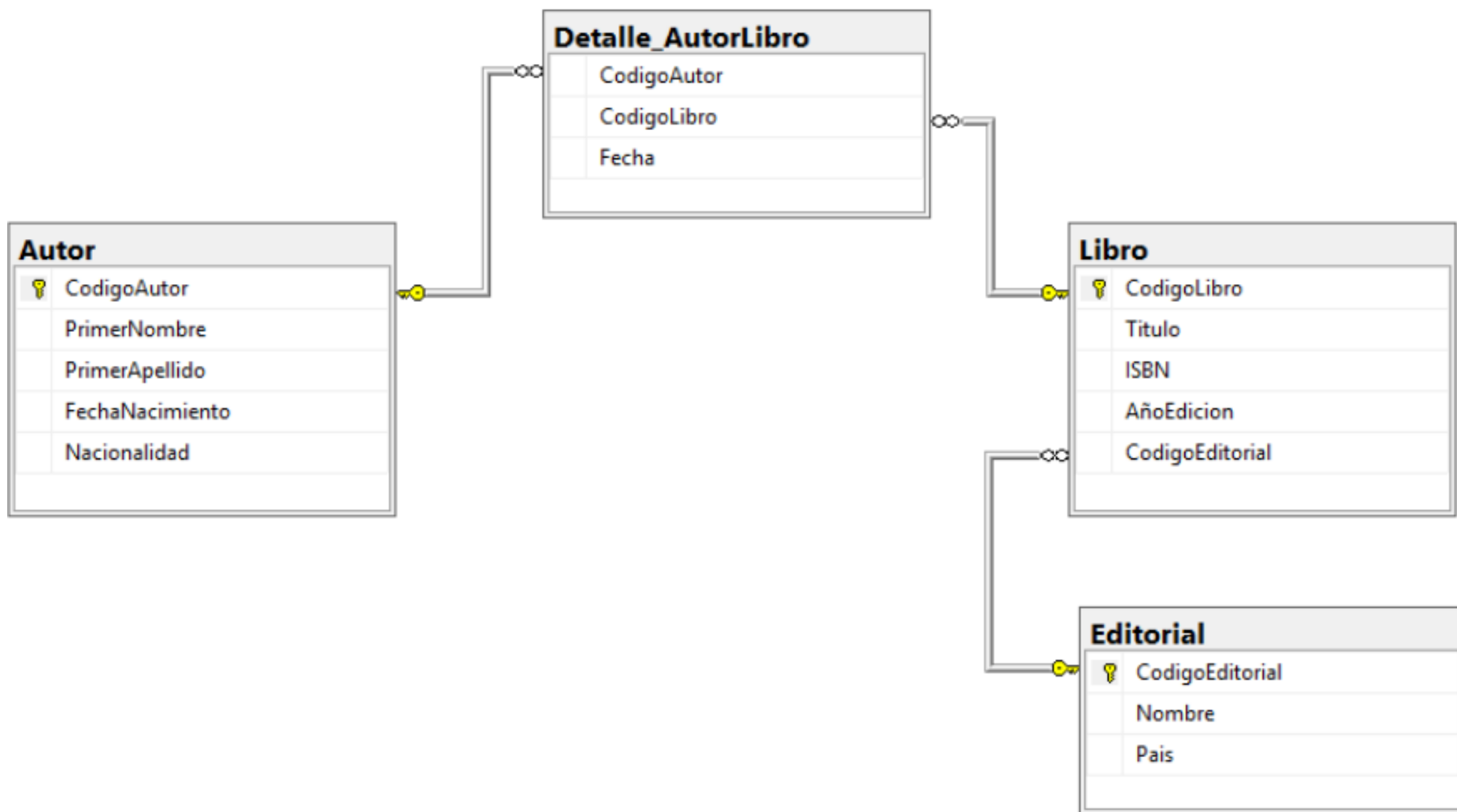
Se controla que si se agrega un código de categoría (CategoryID) en la tabla "Products", tal código exista previamente en la tabla "Categories".



FOREIGN KEY

- Cuando se crea una tabla, se puede crear una restricción FOREIGN KEY como parte de la definición de esa tabla. Si ya existe una tabla, puede agregar una restricción FOREIGN KEY, siempre que dicha restricción esté vinculada a una restricción PRIMARY KEY, una tabla puede contener varias restricciones FOREIGN KEY.
- Elimine una restricción FOREIGN KEY para quitar el requisito de integridad referencial entre las columnas de clave externa y las columnas de la clave principal.





Preguntas:

1. ¿Qué es una dependencia funcional?
2. ¿Qué es un diseño de base datos?

