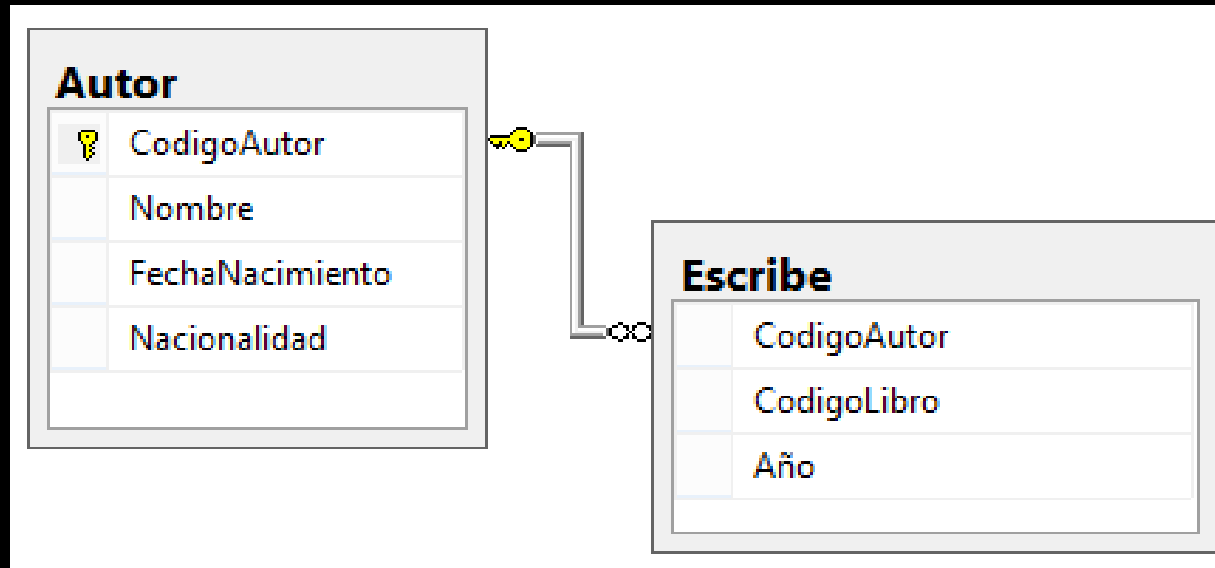


Modelo Relacional



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.

Modelo Relacional

- Su idea fundamental es el uso de "relaciones". Estas relaciones podrían considerarse en forma lógica como conjuntos de datos llamados "tuplas" (filas).
- 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.

Modelo Relacional

- 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 (fila) dentro de una tabla, campo es un dato en concreto (columna) 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:

<code>varchar(30)</code>	Texto con un tamaño máximo de 30 caracteres
<code>datetime</code>	Información de fecha y hora
<code>int</code>	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 (fila), cada atributo toma un solo valor. Se dice que las relaciones están *normalizadas*.
 - No hay dos atributos que se llamen igual.

Propiedades de las relaciones

- Las relaciones tienen las siguientes características:
 - El orden de los atributos no importa: los atributos no están ordenados.
 - Cada tupla (fila) 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

- 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.
- 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.

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:

- **Las columnas no admiten valores NULL.**

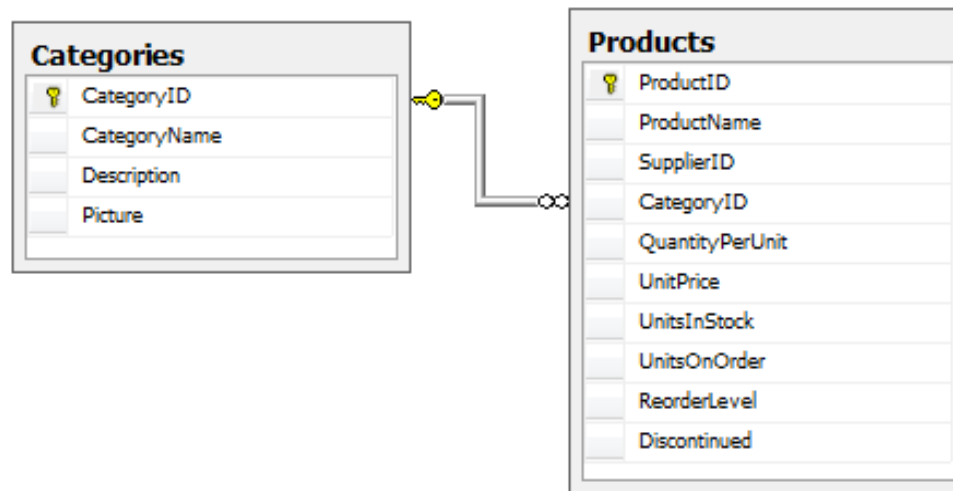
Las columnas de restricciones PRIMARY KEY que se especifiquen al crear la tabla se convierten implícitamente a NOT NULL. Una columna dispersa no se puede utilizar como parte de una clave principal porque las columnas dispersas deben permitir valores NULL.

- **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.