

PROGRAMA DE ESTUDIOS

DESARROLLO DE SISTEMAS DE INFORMACIÓN

**DISEÑO E IMPLEMENTACIÓN
DE BASE DE DATOS**

Tema

MODELOS RELACIONAL

1. Características del modelo relacional de base de datos

- **Composición.** Todos los datos se representan conceptualmente como una disposición ordenada de datos en filas y columnas, llamada relación o tabla.
- **Encabezado y cuerpo.** Cada tabla debe tener un encabezado y un cuerpo. El encabezado es simplemente la lista de columnas. El cuerpo es el conjunto de datos que llena la tabla, organizado en filas.
- **Valores.** Todos los valores son escalares. Es decir, en cualquier posición dada de fila/columna en la tabla, hay solo un valor único.
- **Reglas.** El modelo relacional de base de datos debe elaborarse basado en unas reglas, llamadas de integridad, que garantizan la identificación de las filas y columnas.

Elementos del modelo relacional de base de datos

La siguiente figura muestra una tabla con los nombres de sus elementos básicos, que conforman una estructura completa.

| Atributos o Campos | Codigo_Emp | Nombre | Año |
|--------------------|------------|------------|-----|
| Tuplas | 21130 | Amar Jain | 1 |
| | 30745 | Kuldeep | 3 |
| | 41894 | Manoj | 2 |
| | 51207 | Rita bajaj | 6 |

- **Tupla.** Cada fila de datos es una tupla, conocida también como registro. Cada fila es una n-tupla, pero la “n-” generalmente se descarta.
- **Columna.** Cada columna de una tupla se llama atributo o campo. La columna representa el conjunto de valores que puede tener un atributo específico.
- **Clave.** Cada fila tiene una o más columnas denominadas clave de la tabla. Este valor combinado es único para todas las filas de una tabla. Mediante esta clave se identificará cada tupla de forma única. Es decir, *la clave no puede estar duplicada*. Se le llama clave primaria. Por otro lado, una clave externa o secundaria es el campo

de una tabla que se refiere a la clave primaria de alguna otra tabla. Se utiliza para referenciar a la tabla primaria.

2. **Reglas de integridad del modelo relacional de base de datos al diseñar el modelo relacional, se definen algunas condiciones que deben cumplirse en la base de datos, denominadas reglas de integridad.**

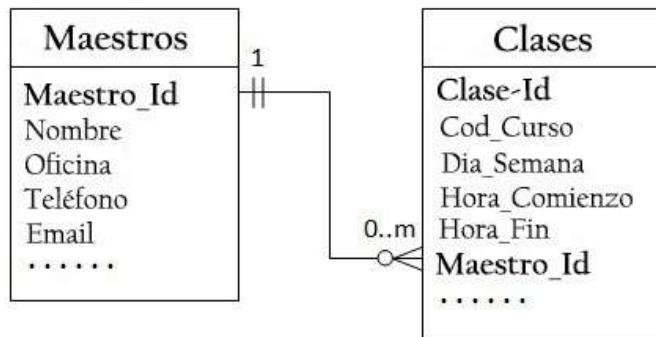
- **Integridad de la clave.** La clave primaria debe ser única para todas las tuplas y no puede tener el valor nulo (NULL). De lo contrario, no podrá identificar la fila de forma exclusiva. Para una clave compuesta por varias columnas, ninguna de esas columnas puede contener NULL.
- **Integridad referencial.** Cada valor de una clave externa debe coincidir con un valor de la clave primaria de la tabla referenciada o primaria. En la tabla secundaria solo se podrá insertar una fila con una clave externa si ese valor existe en una tabla primaria. Si el valor de la clave cambia en la tabla primaria, por actualizarse o eliminarse la fila, entonces todas las filas en las tablas secundarias con esta clave externa deben actualizarse o eliminarse.

3. **¿Cómo hacer un modelo relacional de base de datos?**

- **Recopilar datos.** Se deben recopilar los datos necesarios para almacenarlos en la base de datos. Estos datos se dividen en diferentes tablas. Se debe elegir un tipo de datos apropiado para cada columna. Por ejemplo: números enteros, números de punto flotante, texto, fecha, etc.
- **Definir claves primarias.** Para cada tabla se debe elegir una columna (o pocas columnas) como clave primaria, que identificará de forma única cada fila de la tabla. La clave primaria también se utiliza para hacer referencia a otras tablas.
- **Crear relaciones entre tablas.** Una base de datos que consista en tablas independientes y no relacionadas tiene poco propósito. El aspecto más importante en el diseño de una base de datos relacional es identificar las relaciones entre las tablas. Los tipos de relación son:
 - **Uno a muchos.** En una base de datos “Listado de clases”, un maestro puede enseñar en cero o más clases, mientras que una clase es impartida por un

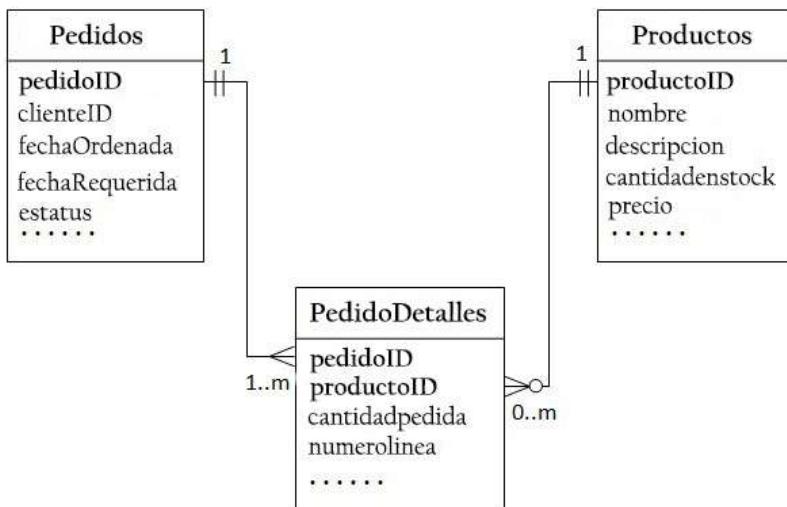
solo maestro. Este tipo de relación se conoce como uno a muchos. Esta relación no se puede representar en una sola tabla. En la base de datos “Listado de clases” se puede tener una tabla llamada Maestros, que almacena información sobre los maestros. Para almacenar las clases impartidas por cada maestro, se podrían crear columnas adicionales, pero se enfrentaría un problema: cuántas columnas crear. Por otro lado, si se tiene una tabla llamada Clases, que almacena información sobre una clase, se podrían crear columnas adicionales para almacenar la información sobre el maestro. Sin embargo, como un maestro puede enseñar en muchas clases, sus datos se duplicarían en muchas filas de la tabla Clases.

- **Diseñar dos tablas.** Por tanto, se necesitan diseñar dos tablas: una tabla Clases, para almacenar información sobre las clases, con Clase_Id como clave principal, y una tabla Maestros, para almacenar información sobre los maestros, con Maestro_Id como clave principal. Luego se puede crear la relación uno a muchos almacenando la clave primaria de la tabla Maestro (Maestro_Id) en la tabla Clases, como se ilustra a continuación.



- **Diseñar c/tablas.** La columna Maestro_Id en la tabla Clases es conocida como clave externa o clave secundaria. Para cada valor Maestro_Id en la tabla Maestro, puede haber cero o más filas en la tabla Clases. Para cada valor Clase_Id en la tabla Clases, hay solo una fila en la tabla Maestros.

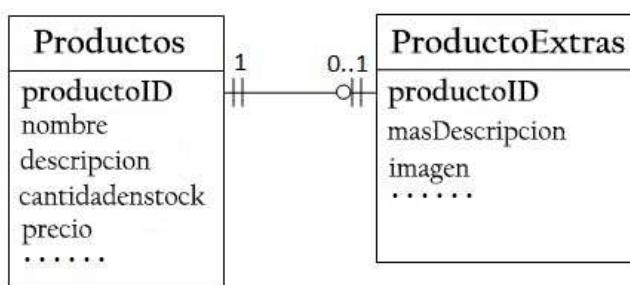
- **Muchos a muchos.** En una base de datos “Venta de productos”, el pedido de un cliente puede contener varios productos, y un producto puede aparecer en varios pedidos. Este tipo de relación se conoce como muchos a muchos. Se puede comenzar la base de datos “Venta de productos” con dos tablas: Productos y Pedidos. La tabla Productos contiene información sobre los productos, con productID como clave primaria. Por otro lado, la tabla Pedidos contiene los pedidos del cliente, con pedidID como clave primaria. No se pueden almacenar los productos pedidos dentro de la tabla Pedidos, ya que no se sabe cuántas columnas reservar para los productos. Tampoco se pueden almacenar los pedidos en la tabla Productos por la misma razón. Para admitir una relación muchos a muchos, se necesita crear una tercera tabla, conocida como tabla de unión (PedidoDetalles), donde cada fila representa un elemento de un pedido particular. Para la tabla PedidoDetalles, la clave primaria consta de dos columnas: pedidID y productoID, identificando de forma única cada fila. Las columnas pedidID y productoID en la tabla PedidoDetalles se utilizan para referenciar a las tablas Pedidos y Productos. Por tanto, también son claves externas en la tabla PedidoDetalles.



- **Uno a uno.** En la base de datos “Venta de productos”, un producto puede tener información opcional, como descripción adicional y su imagen.

Mantenerla dentro de la tabla Productos generaría muchos espacios vacíos.

Por tanto, se puede crear otra tabla (ProductoExtras) para almacenar los datos opcionales. Solo se creará un registro para los productos con datos opcionales. Las dos tablas, Productos y ProductoExtras, tienen una relación uno a uno. Para cada fila en la tabla Productos hay máximo una fila en la tabla ProductoExtras. Se debe usar el mismo productoID como clave principal para ambas tablas.



4. Ejemplo de modelo relacional de base de datos

Supongamos una base de datos que consta de las tablas Suplidores, Piezas y Envíos. La estructura de las tablas y algunos registros de muestra se exponen a continuación:

| Suplidores | | | | |
|------------|--------|---------|----------|--|
| Sno | Nombre | Estatus | Ciudad | |
| S1 | Suneet | 20 | Qadian | |
| S2 | Ankit | 10 | Amritsar | |
| S3 | Amit | 10 | Qadian | |

| Piezas | | | | |
|--------|----------|-------|------|-------|
| Pno | Nombre | Color | Peso | Stock |
| P1 | Tuerca | Rojo | 12 | 100 |
| P2 | Perno | Verde | 17 | 220 |
| P3 | Tornillo | Azul | 17 | 110 |
| P4 | Tornillo | Rojo | 14 | 680 |

| Envíos | | |
|--------|-----|----------|
| Sno | Pno | Cantidad |
| S1 | P1 | 250 |
| S1 | P2 | 300 |
| S1 | P3 | 500 |
| S2 | P1 | 250 |
| S2 | P2 | 500 |
| S3 | P2 | 300 |

Cada fila en la tabla Suplidores se identifica mediante un número de suplidor (SNo) único, identificando de manera única cada fila de la tabla. Igualmente, cada pieza tiene un número de pieza (PNo) único.

Además, no puede existir más de un envío para una combinación dada Suplidor/Pieza en la tabla Envíos, ya que esta combinación es la clave primaria de Envíos, que funge como tabla de unión, por ser una relación muchos a muchos.

La relación de las tablas Piezas y Envíos viene dada por tener en común el campo PNo (número de pieza) y la relación entre Suplidores y Envíos surge por tener en común el campo SNo (número de suplidor).

Analizando la tabla Envíos se puede obtener como información que se está enviando un total de 500 tuercas desde los suplidores Suneet y Ankit, 250 cada uno.

Igualmente, se enviaron 1.100 pernos en total desde tres suplidores diferentes. Se enviaron 500 tornillos azules desde el suplidor Suneet. No hay envíos de tornillos rojos.

Bibliografía:

- Fundamentos de Sistemas de Bases de Datos- Ramez Elmasri, Shamkant B. Navathe
- [Modelo relacional de base de datos: qué es, características, ejemplo](#)

