



BASES DE DATOS

FES Aragón

ICO

Dr. Omar Mendoza González

Modelo relacional

- Los **datos** se **estructuran lógicamente** en forma de **relaciones** (tablas).
- Mientras que en los Diagramas E/R se habla de entidades y relaciones, en el modelo relacional solamente se tiene un tipo de estructura, **todo son relaciones**.

Modelo relacional

- Se caracteriza por:
 - Ser sencillo y uniforme (colección de tablas y lenguajes declarativos)
 - Tener una sólida fundamentación teórica: el modelo está definido con rigor matemático
 - Ser independiente del almacenamiento físico y de las aplicaciones.

Modelo relacional

- Descripción de datos
 - Entidades y relaciones se representan en forma de tablas
 - Las tablas reciben el nombre de relación.
 - Las Filas (tuplas) contienen datos sobre cada entidad
 - Las columnas corresponden a los atributos de las entidades

Modelo relacional

■ RELACIÓN

- Es la estructura básica del modelo relacional. Se representa mediante una **tabla**.

■ DOMINIO

- Es el conjunto válido de valores que toma un **atributo**. Existen con independencia de cualquier otro elemento.

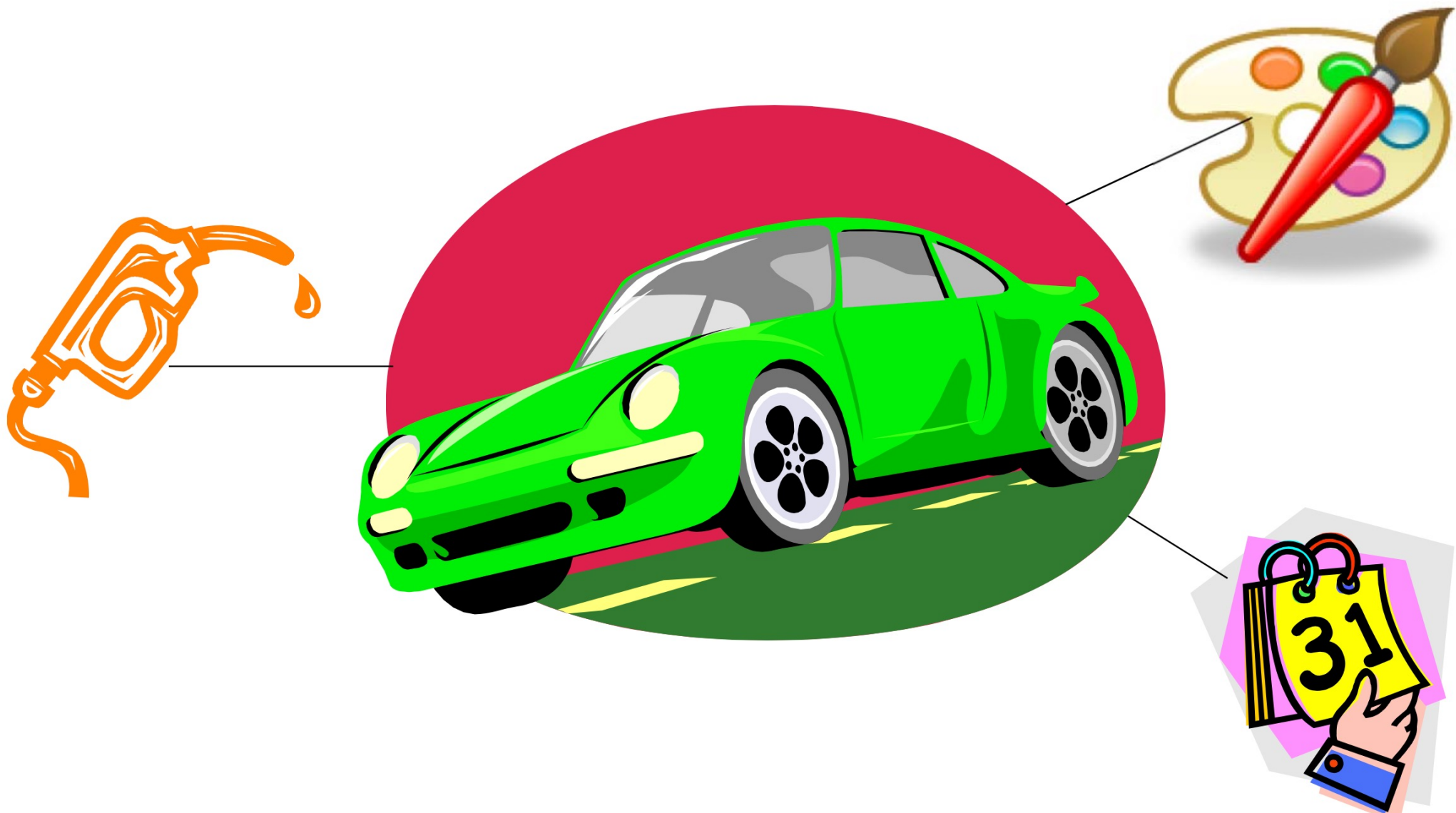
■ ATRIBUTO

- Representa las propiedades de la relación. Se representa mediante una **columna**.

■ TUPLA

- Es una ocurrencia de la relación. Se representa mediante una **fila**.

Entidades y Atributos



Ejercicio

- Casa
- NSS
- CURP
- #Teléfono
- Fecha
- Producto
- Computadora
- #Orden
- Ancho
- Nombre
- Factura
- Orden
- #Producto
- Estado
- Calle
- Teclado

Relaciones

- Una relación (matemáticamente) es un subconjunto del producto cartesiano de la lista de dominios $\{D_i\}$
- “un esquema de relación se compone de un nombre de relación R , un conjunto de n atributos $\{A_i\}$ y de un conjunto de n dominios (no necesariamente distintos) $\{D_i\}$ donde cada atributo será definido sobre un dominio”.

Relaciones

- Una relación consta de los siguientes elementos:
 - Nombre de la relación
 - Cabecera: conjunto de n pares atributo-dominio
 - Cuerpo: Conjunto de m tuplas
 - Esquema: constituido por el nombre de la relación y la cabecera
 - Estado: constituido por el esquema y cuerpo.

Relaciones

- El Universo de Discurso de una BD relacional está compuesto por un conjunto de dominios $\{D_i\}$ y de relaciones $\{R_i\}$ definidas sobre los dominios

The diagram shows a table with three columns and three rows. The columns are labeled *nombre*, *calle*, and *ciudad*. The rows contain data for three clients. Arrows point from the word **atributos** to the column headers, and from the word **tuplas** to the rows. The word *cliente* is centered below the table.

<i>nombre</i>	<i>calle</i>	<i>ciudad</i>
<i>Carmen</i>	Calvo Sotelo	Santander
<i>Ana</i>	Castellana	Madrid
<i>Pedro</i>	Torres Quevedo	Logroño
<i>Marie</i>	Eliseos	París

cliente

Relaciones

- Una relación de grado m consta de dos partes:
 - **Cabecera:** conjunto fijo de m campos.
 - Cada campo esta definido por su **Nombre** y su **Dominio** (que indica el tipo de valores que contendrá dicho campo).

$\{(Nombre_1 : Dominio_1), \dots, (Nombre_m : Dominio_m)\}$

Relaciones

- Una relación de grado m consta de dos partes:
 - **Cuerpo:** conjunto variable de registros (también denominados tuplas).
 - Cada registro es un conjunto de m valores:

$$Reg_1 \rightarrow \{(Nombre_1 : Valor_{1,1}), \dots, (Nombre_m : Valor_{1,m})\}$$

...

$$Reg_n \rightarrow \{(Nombre_1 : Valor_{n,1}), \dots, (Nombre_m : Valor_{n,m})\}$$

Relaciones

- Hay que diferenciar:

- **Esquema :**

- Conjunto de atributos $\{A_i\}$ junto con sus dominios (diseño lógico de la BD)

- **Instancia :**

- Conjunto de tuplas $r=\{t_1, \dots, t_n\}$ tal que $t_i=(x_1, \dots, x_n)$ con $x_j \in D_j$ (instancia del esquema, esto es, datos que en un momento determinado están en la BD)

Relaciones

- **Esquema:**

Persona {nombre:texto,calle:texto,ciudad:texto}

- **Instancia:**

(Carmen, Calvo Sotelo, CDMX),
(Luisa, Castellana, Monterrey),
(Pedro, Torres Quevedo, Leon),
(Marco, Eliseos, Guadalajara)

Relaciones

- Todos los registros del cuerpo una relación deben tener el *mismo número de campos*, aunque *alguno* este *vacío*.
 - En este caso, dicho campo vacío toma el valor *NULL*.
- Los **valores de los campos son atómicos**: fijado un registro, cada campo toma un único valor (no se admiten campos multivaluados).
- **No se admiten registros duplicados**. Dos registros de una relación deben diferir, al menos, en el valor de un campo.
- El orden de los registros en el cuerpo de una relación **no importa**.

Relaciones

DNI	Nombre	Dirección	Fecha
44345789	Ana Pérez	Sol, 17	9/5/1960
40876100	José Ruíz	Luna,1	1/1/1972
56123009	Luis Gómez	Feria,2	NULL

Dirección	Nombre	Fecha	DNI
Feria,2	Luis Gómez	NULL	56123009
Luna,1	José Ruíz	1/1/1972	40876100
Sol,17	Ana Pérez	9/5/1960	44345789

¿Es una buena definición de Relación?

Nombre y Apellido	Edad	Estudios
Juan Pérez	41	Lcdo. Química
Ana Sánchez	37	Lcdo. Medicina Lcdo. Física
Juan Pérez	41	Lcdo. Química
Félix González	32	NULL

¿Es una buena definición de Relación?

Multivaluado



Nombre y Apellido	Edad	Estudios
Juan Pérez	41	Lcdo. Química
Ana Sánchez	37	Lcdo. Medicina Lcdo. Física
Juan Pérez	41	Lcdo. Química
Félix González	32	NULL

Campos

- Cada campo debe poseer un **Nombre**
- Debe tener asociado un **Tipo** de dato.

Algunos tipos posibles (no los únicos) serían:

- **Texto**: cadenas de caracteres, ya sean letras, números con los que no realizar operaciones o símbolos.
- **Numérico**: números sobre los que tiene sentido realizar operaciones.
- **Fecha/hora**: almacena fechas, horas o ambas.
- **Sí/No**: datos que solo tengan dos posibilidades (verdadero-falso).
- **Autonumérico**: valor numérico (1,2,...) que el SGBD incrementa de modo automático cuando se añade un registro.

Campos

- Un campo puede poseer **opcionalmente** las siguientes propiedades:
 - **Descripción:** texto breve que aclara el contenido o la finalidad del campo.
 - **Tamaño:** indica el tamaño máximo permitido (aplicable a campos de texto o numéricos).
 - **Rango** de valores posibles, dentro de una lista de valores permitidos.
 - **Requerido** o **NOT NULL:** no se permiten valores nulos para dicho campo.
 - **Predeterminado:** se fija un valor por defecto para el campo.

Campos

- La cabecera de una relación gráficamente presentada así:

CURP	Nombre	Fecha	Nacionalidad	Direccion
------	--------	-------	--------------	-----------

- se podría describir como:

```
{  
  (CURP:Texto(18) NOT NULL),  
  (Nombre:Texto(50) NOT NULL),  
  (Fecha:Fecha Descripción= "Fecha de nacimiento del cliente"),  
  (Nacionalidad:Texto(20) Predeterminado="Mexicana"),  
  (Direccion:Texto)  
}
```

Claves y datos derivados

- Junto con la cabecera y los campos de las relaciones, se necesita indicar:
 - La **clave primaria**, y si se ha de generar automáticamente.
 - Las posibles claves alternativas.
 - Las **claves foráneas** y sus *reglas de comportamiento* ante el borrado y la modificación de la clave primaria a la que referencian.
 - Si alguna columna es un dato derivado (su valor se calcula a partir de otros datos de la base de datos) indicar cómo se obtiene su valor.

Tabla vs Relación

- Una **relación** es un **concepto abstracto** de origen matemático.
- Una **tabla** es una forma de **representar** (implementar) una relación (una estructura de datos).
 - Una tabla no tiene las restricciones inherentes de una relación como conjunto:
 - Puede haber dos filas iguales.
 - Las filas están ordenadas en el orden de grabación física por defecto o según el valor de la clave primaria.
 - Los atributos tienen un orden según se han definido en la tabla.
 - En cada celda de una tabla puede haber uno o varios valores.
 - Si bien en el segundo caso se puede obtener una tabla equivalente que cumple la regla de normalización.

Claves

■ Clave

- Conjunto de campos cuyos valores determinan unívocamente a cada registro de la relación.
- Dicho conjunto de campos debe ser minimal, *i.e.*, ningún subconjunto propio de la clave puede actuar también como clave.

Claves

- **Clave primaria (PK=Primary Key)**

- Es un campo o una combinación de campos que identifica de forma única a cada fila de una tabla.

- **Clave candidata**

- Cada uno de los campos o combinaciones de campos que pueden actuar como clave de la relación.

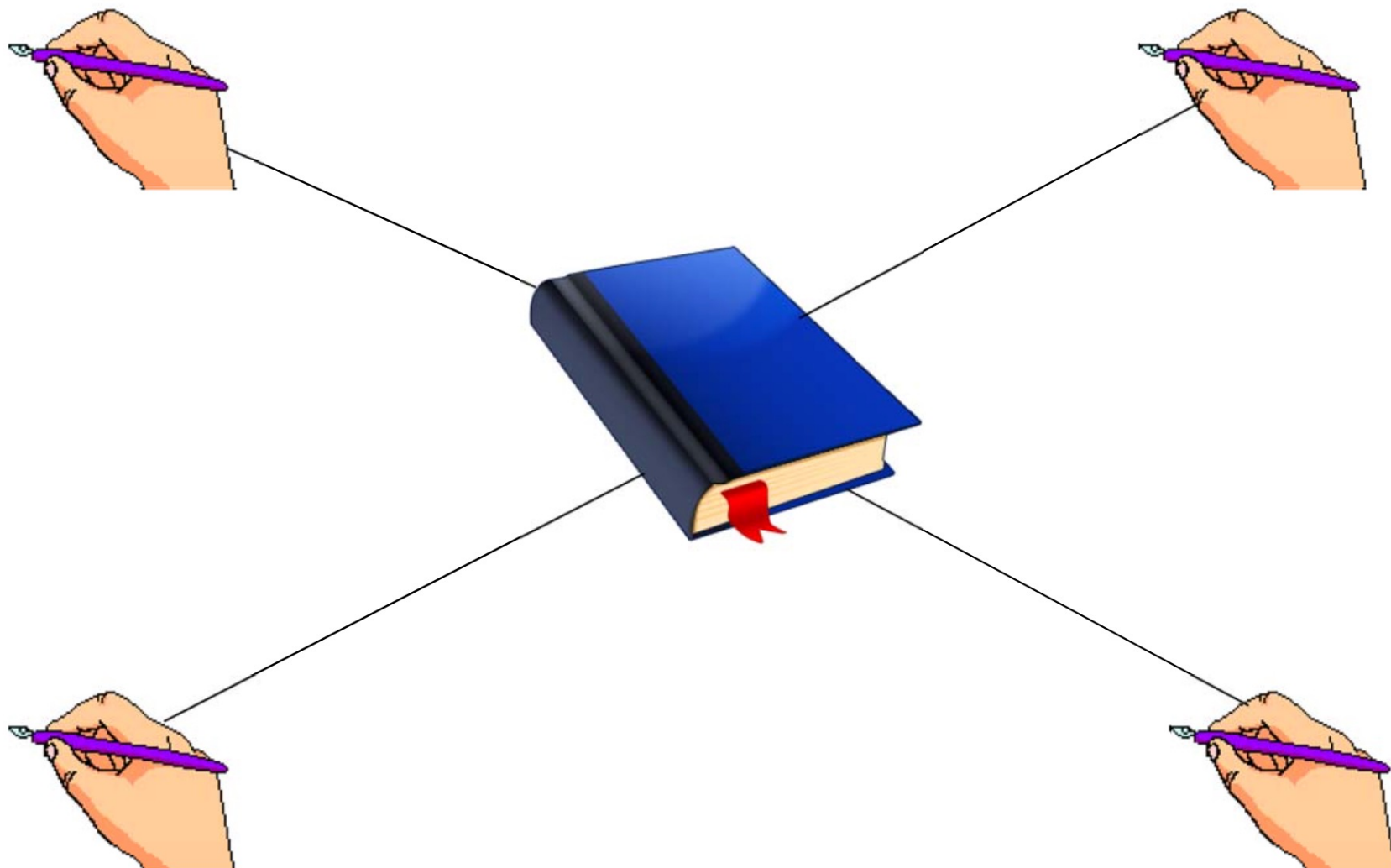
Claves

- **Clave foránea (FK=Foreign Key)**
 - campo o combinación de campos de una relación (relación hija) que funciona como clave primaria de otra relación de la BD (relación referenciada o relación padre).
- Las **claves foráneas** son **esenciales** en el Modelo Relacional, ya que permiten **enlazar las relaciones** de la BD.

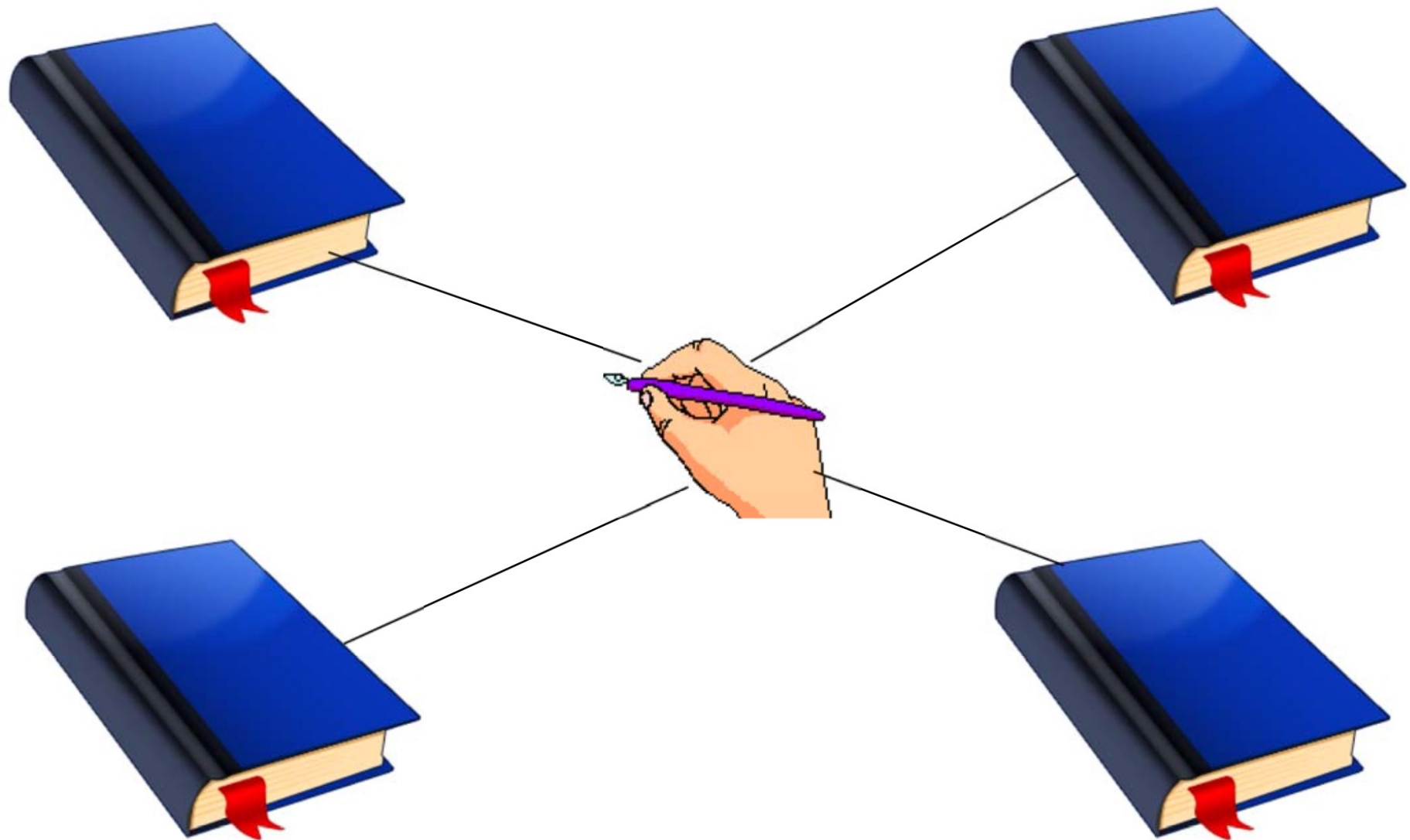
Claves

- Una clave foránea y la clave primaria de la relación referenciada asociada han de estar **definidas** sobre los **mismos dominios**.
- *Una relación puede poseer más de una clave foránea* (tendrá una clave foránea por cada relaciónn referenciada de la cual dependa).
- Una relación *puede no poseer ninguna clave foránea*.
- Una clave foránea *puede enlazar una relación consigo misma* (relaciones reflexivas).

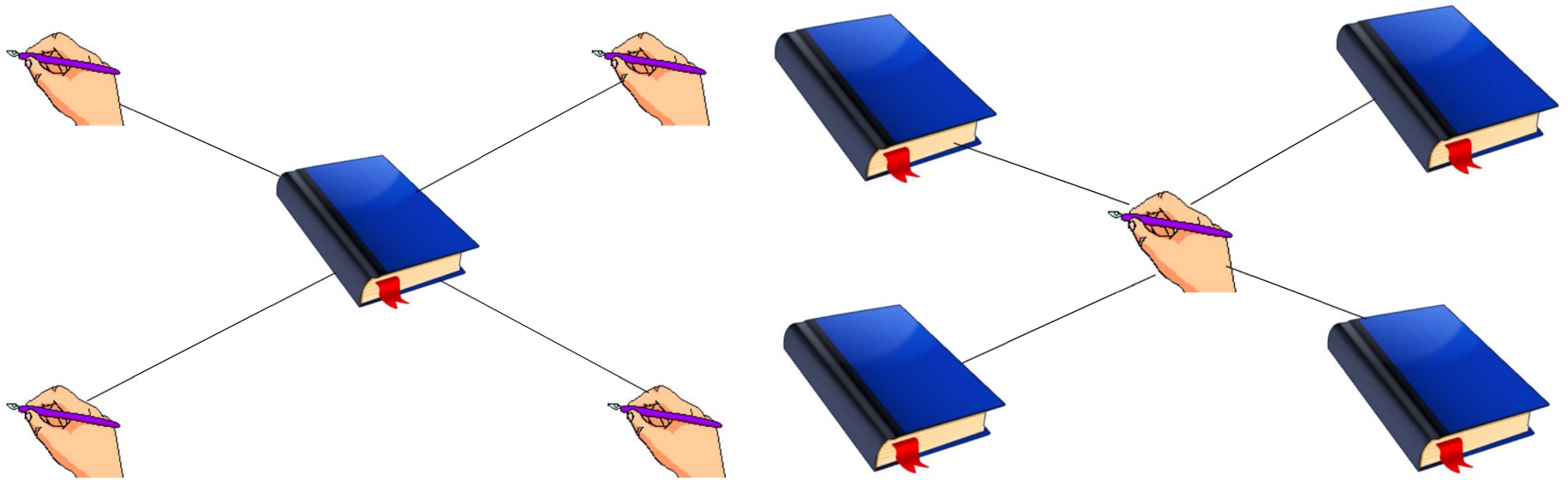
Relaciones



Relaciones



Relaciones



Ejemplo

■ Relación ESCRITOR:

RFC	Nombre	Fecha	Nacionalidad	Direccion
-----	--------	-------	--------------	-----------

Escritor

```
{  
    (RFC: Texto(13) NOT NULL),  
    (Nombre: Texto(50) NOT NULL),  
    (Fecha: Fecha Descripción= "Fecha de nacimiento del escritor"),  
    (Nacionalidad: Texto(20) Predeterminado="Mexicana"),  
    (Direccion: Texto),  
    PK = (RFC)  
}
```

Ejemplo

■ Relación EDITORIAL:

IdEditorial	Nombre	Ciudad	País	Direccion
-------------	--------	--------	------	-----------

Editorial

```
{  
  (IdEditorial: AI Numérico NOT NULL) PK,  
  (Nombre: Texto(150) NOT NULL),  
  (Ciudad: Texto(50) NOT NULL),  
  (Pais: Texto(3)),  
  (Direccion: Texto)  
}
```


Ejemplo

■ Relación LIBRO:

Codigo	Titulo	IdEditorial	RFCAutor	Año
--------	--------	-------------	----------	-----

Libro

```
{  
  (Codigo: Texto(10) NOT NULL) PK,  
  (Titulo: Texto(150) NOT NULL),  
  (IdEditorial: Numérico NOT NULL FK),  
  (RFCAutor: Texto(13) NOT NULL FK),  
  (Año: Numérico)  
}
```

Relaciones

■ Claves primarias

- Relación EDITORIAL: PK = (IdEditorial:Texto)
- Relación ESCRITOR: PK = (RFC:Texto)
- Relación LIBRO: PK = (Codigo:Texto)

■ Claves foráneas en LIBRO

FK = (IdEditorial) → (Editorial.IdEditorial)

FK = (RFCAutor) → (Escritor.RFC)

Relaciones

- Cardinalidad

- número de tuplas(m).

- Grado

- número de atributos(n).

- Dominio

- colección de valores permitidos en un atributo.