



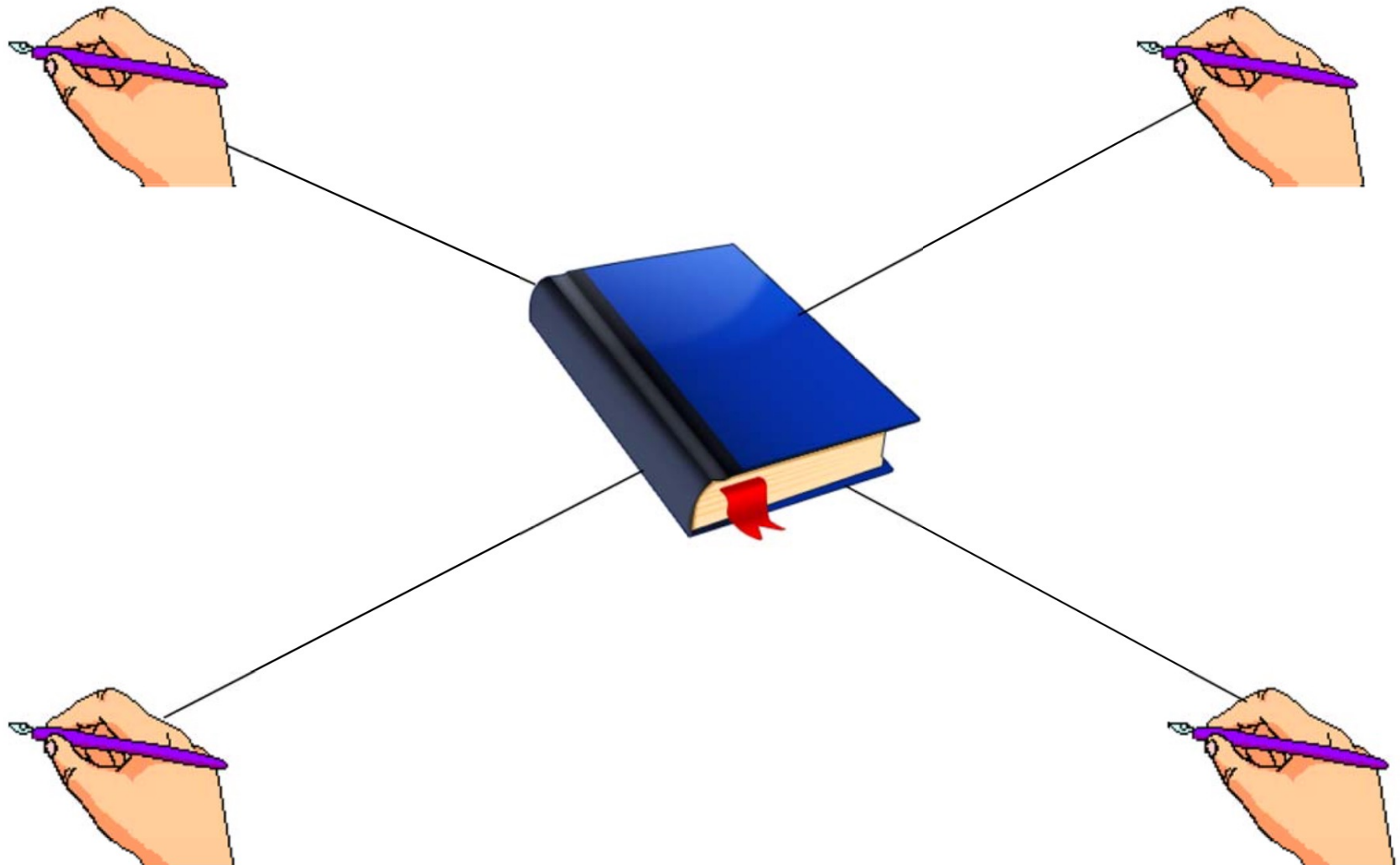
BASES DE DATOS

FES Aragón

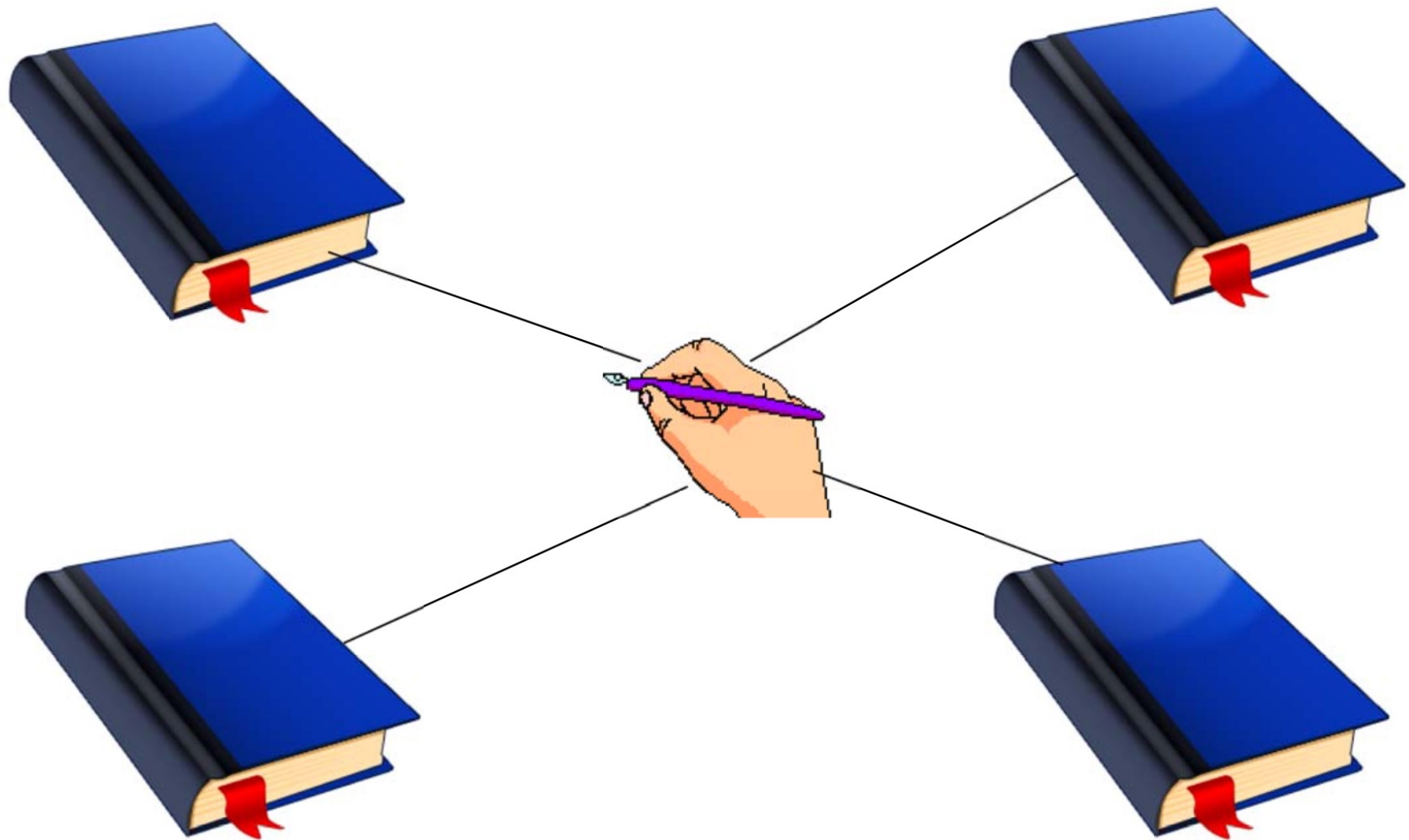
ICO

Dr. Omar Mendoza González

Relaciones



Relaciones



Modelo relacional

- Restricciones de integridad
 - Integridad de entidad
 - Integridad referencial (mediante el uso de claves).

Valor nulo (NULL)

- Marca para representar ***información desconocida o no aplicable***.
- El valor de un campo puede ser nulo por dos razones:
 - Existencia de registros con campos desconocidos en ese momento.
 - Existencia de campos inaplicables a ciertos registros.

Integridad de entidad

- Una relación cumple la **restricción de integridad de entidad** si *ningún campo que forme parte de la clave primaria de la relación puede tomar valores nulos*
- Para conseguir la integridad de entidad, basta declarar como '*Requerido*' (**NOT NULL**) todos los campos que formen parte de la PK de cada relación de la BD

Integridad referencial

- Si una relación $R1$ posee una clave foránea que la enlaza con la relación $R2$, entonces se dice que cumple la restricción de ***integridad referencial***
- Si todo valor de dicha clave foránea de $R1$ cumple una se las dos condiciones:
 - Coincide con algún valor de la clave primaria en la relación **$R2$**
 - Toma el valor nulo (NULL)

Integridad referencial

Relación ESCRITOR,
PK = (DNI:Texto)

DNI	Nombre	Fecha	País
67543198	Luis Ruíz	1/1/1965	Chile
89564123	Ana Pérez	2/7/1977	España

Relación OBRA,
PK = (Código:Texto)

Código	Título	Autor	Fecha
345	La huida	67543198	1993
111	El fin	33567900	1982
654	NULL	NULL	2001

FK = (Autor:Texto) (\rightarrow ESCRITOR)

Regla de integridad referencial

■ Inserción:

- El SGBD sólo permite insertar un nuevo registro en la relación R1 cuando el valor del campo C para ese registro coincida con algún valor de la PK de R2 que aparezca en la relación (o bien se deja nulo).

Regla de integridad referencial

■ **Modificación:**

- Actualización en **cascada**
- Si se modifica el valor de la PK de un registro de la relación padre R2, el SGBD modifica automáticamente dicho valor en todos los registros de la relación R1 que estén relacionados con él.

Regla de integridad referencial

■ Borrado:

- eliminación en **cascada**: si se elimina un registro de la relación padre R2, el SGBD elimina automáticamente todos los registros de la relación R1 que están relacionados con dicho registro.
- eliminación **restringida**: si se intenta eliminar un registro de la relación padre R2, el SGBD no permite si hay registros de la relación R1 que están relacionados con dicho registro.

Regla de integridad referencial

- Ningún componente de una clave foránea puede contener valores que no están presentes en la clave primaria (o alternativa) a la que referencia.
 - ¿Puede una clave foránea **aceptar nulos**?
 - ¿Cómo es el **dominio de una clave foránea** con respecto al dominio de la clave primaria (o alternativa) a la que referencia?
 - ¿Qué sucede **si se pretende borrar el valor al que referencia** (“padre”) una clave foránea?

Regla de integridad referencial

- Cascada.
- Restringido.
- Volver los valores nulos.
- Sin cambios.

Esquema Lógico Relacional

- Existe un **procedimiento general** para obtener el Esquema Relacional (**conjunto de Relaciones**, Tablas Relacionales)
- Entidades fuertes / débiles
- Interrelaciones
- Jerarquías de generalización/especialización

Entidades fuertes / débiles

- **Entidad fuerte** contiene una clave primaria.
 - Sí puede ser identificada unívocamente
- **Entidad débil** es aquella que no puede existir sin participar en la relación; no puede ser unívocamente identificada solamente por sus atributos.

Entidades fuertes / débiles

■ Dependencia por existencia

- Las ocurrencias de la entidad débil pueden identificarse mediante un atributo identificador clave sin necesidad de identificar la entidad fuerte relacionada.
- Se tratarán a nivel de diseño lógico como las entidades fuertes.

■ Dependencia por identidad

- La entidad débil no puede ser identificada sin la entidad fuerte relacionada
- Se creará una nueva relación con un campo por cada atributo de la entidad, y **se añaden los campos que forman la PK de la entidad *padre* de la que depende**

Interrelaciones

- La relación se tratará de forma distinta según el tipo de la interrelación:
- Muchos a muchos (N:M)
- Uno a muchos (1:N)
- Uno a uno (1:1)

Muchos a muchos (N:M)

- Se crea una nueva relación con los siguientes campos:
 - los campos de la PK de la entidad E1,
 - los campos de la PK de la entidad E2,
 - los campos correspondientes a los atributos propios de la interrelación (si los hubiese).
- Con respecto a las claves:
 - La clave primaria contendrá a las claves de E1 y E2
 - Podrá incluir más campos, dependiendo de la semántica.
 - Se añaden dos claves foráneas a la nueva relación
 - $FK=PK_{deE1}(\rightarrow E1)$
 - $FK=PK_{deE2}(\rightarrow E2)$

Uno a muchos (1:N)

- **NO** se creará ninguna relación nueva.
- Se modifica la relación asociada a la entidad que participa con cardinalidad máxima “muchas”.
- Suponiendo que E1 es quien participa con cardinalidad (1, n) o (0, n), modificar la relación asociada a la entidad E1 como sigue:
 - Añadir como campos nuevos los atributos que forman la PK de la entidad E2
 - Añadir los campos correspondientes a los atributos propios de la relación (si los hubiese)
 - Añadimos una nueva clave foránea:
 - FK = PK de E2 (\rightarrow E2)

Uno a uno (1:1)

- **NO** se creará ninguna **relación** nueva.
- Se tratan como las relaciones (1:N) en ese sentido.
- Puesto que las dos entidades participan con cardinalidad máxima 1, se tienen dos opciones:
 - añadir a la tabla asociada a E1 la PK de E2 y los atributos propios de la relación (1:1)
 - añadir a la tabla asociada a E2 la PK de E1 y los atributos propios de la relación (1:1).

Uno a uno (1:1)

- **dbdiagram.io** es una herramienta rápida y simple que utiliza DBML (database markup language) un lenguaje DSL (Domain-specific language) para definir estructuras de bases de datos.
- <https://dbdiagram.io/>
- <https://www.dbml.org/docs/>
- <https://dbdocs.io/>

Uno a uno (1:1)

```
table "escritor"{
  idEscritor int [pk, not null, increment]
  rfc varchar(13) [not null, unique]
  nombre varchar(50) [not null]
  ap_paterno varchar(50) [not null]
  ap_materno varchar(50)
  fecha date [note:'Fecha de nacimiento']
  nacionalidad varchar(30) [default:'Mexicana']
  idEstado int [not null]
  idPais int [not null]
  Indexes {
    (nombre, ap_paterno, ap_materno) [unique, name:'uk_nombre']
  }
}

table "pais" {
  idPais int [pk, not null]
  pais varchar(150) [not null, unique]
  codigoISO varchar(6) [not null, unique]
}

table "estado" {
  idEstado int [pk, not null, increment]
  estado varchar(150) [not null, unique]
  codigoISO varchar(6) [not null, unique]
}

Ref: escritor.idEscritor > pais.idPais // N:1
Ref: estado.idEstado < escritor.idEstado // 1:N

table "libro"{
  idLibro int [pk, not null, increment]
  titulo varchar(150) [not null]
  anio int
  isbn varchar(14) [not null, unique]
}
```