

## PASAR DEL DER AL MR

- Si tengo dudas de la cardinalidad, preguntarme si lo que quiero es una foto instantánea o una visión histórica.
- Si me enumeran nombres de algo, preguntarme si es una entidad.
- Si una relación me queda con muchos atributos, preguntarme si es una entidad.
- Al terminar el ejercicio, dar una segunda leída al texto para saber que no me faltó nada.
- **No olvidarse la lista de claves foráneas!!! Tabla.FK -> Tabla.PK**

Reglas:

- 1) ENTIDADES FUERTES Cada una pasa a ser una tabla con el identificador como PK
- 2) RELACIONES N:1 o 1:1
  - BINARIA N:1 Del lado de la N, pongo la PK de la otra tabla como FK.
  - BINARIA 1:1 En sólo una de las dos tablas (cualquiera) agrego la PK de la otra como FK
  - UNARIA N:1 o 1:1 Agrego la PK de la tabla a la misma tabla pero como FK (sin que el nombre sea igual)
- 3) RELACIONES N:N Creo una nueva tabla. Las PK de cada tabla de origen serán PK y FK al mismo tiempo.
- 4) ATRIBUTOS DE RELACION Si la relación es N:N, se ponen como campos en la nueva tabla. Si es N:1 se ponen del lado de la N. Si es 1:1 se ponen de cualquier lado. Si el atributo es identificador, formará parte de la PK.
- 5) ATRIBUTOS CALCULADOS No se ponen :)
- 6) ATRIBUTOS AGRUPADORES No se ponen como campo. Sólo se usan como prefijo.
- 7) ATRIBUTOS MULTIVALUADOS Creo una tabla con el campo multivaluado. Se pone la PK de la entidad donde estaba antes. Esa va a ser la FK. **Todo forma la PK.**
- 8) ENTIDADES DEBILES Creo una tabla. Agrego la PK de la fuerte como PK y FK. **El discriminante también es PK.**
- 9) JERARQUIAS Creo una tabla para cada subentidad. La PK del papá va a ser PK y FK de cada subentidad. Si hay --- atributo de Tipo, se pone como campo en la **supraentidad**.
- 10) TERNARIAS Creo una tabla con las 3 PK de cada entidad que van a pasar a ser 3 FK. **La PK se define según las N** que haya. Elijo siempre todo lo que tiene N. Si no hay N, elijo el 1.

## PASAR DEL MR AL DER (INGENIERIA INVERSA)

- **Ir haciendo una marca al lado de cada tabla cuando ya la usé.**
- Si hay más de una opción posible con la cardinalidad, elegir la que tenga N porque es menos restrictiva.

Reglas:

- 1) ENTIDADES FUERTES Miro la PK. **Nada** de lo que es PK puede ser FK.
- 2) SUBENTIDADES Miro la PK. **Todo** tiene que ser FK. Toda la FK tiene que apuntar a una misma tabla.
- 3) ENTIDADES DEBILES Miro la PK. Debe tener más de un campo. **Sólo una parte** tiene que ser FK. La FK puede tener más de un campo pero deben apuntar a la misma tabla. Siempre es N:1
- 4) RELACIONES UNARIAS O BINARIAS 1:N Busco en las tablas que ya usé. Tiene que haber una FK que **no sea parte** de la PK. La N va del lado de la FK.
- 5) RELACIONES UNARIAS O BINARIAS N:N Busco en las tablas que no usé, tablas con **sólo 2 FK** (pueden ser compuestas). Las dos FK tiene que ser parte de la PK. Si la tabla tiene otros campos, esos son atributos de la relación. Si alguno de ellos además está en la PK, entonces es atributo identificador.
- 6) TERNARIAS Busco dentro de lo que no usé tablas con **sólo 3 FK** (pueden ser compuestas). Por lo menos 2 de las 3 tienen que estar en la PK. Las que están en la PK son las que llevan la N. Si la tabla tiene otros campos, esos son atributos de la relación. Si alguno de ellos además está en la PK, entonces es atributo identificador.

## **NORMALIZACION**

- 1) Claves primarias no numéricas -> Elegir una clave numérica. CLAVE SURROGADA  
Ej: numPatente -> idPatente (dejo el numPatente como atributo normal)
- 2) Claves primarias compuestas -> Elegir una clave simple.  
Ej: tipoDoc + numDoc -> id
- 3) Atributos: Cuando aparece un atributo que no puede estar escrito de diferentes maneras porque dificultaría las búsquedas. Si pienso a un atributo como si fuera un menú desplegable, entonces lo pongo en una entidad. Hay que evaluar el grado de repetición antes de normalizarlo. Si no se va a repetir mucho, entonces queda como atributo.  
Ej: Película(idPel, titulo, genero) -> Película(idPel, codGen)      Genero(codGen, descripcion)

## **DEFINICIONES TEORICAS**

**Base de datos:** Conjunto de datos almacenados en forma independiente de las aplicaciones que las utilizan. Poseen redundancia mínima y permiten el control de integridad. Nacen en los años 60. Son administradas por un Sistema de Gestión de Base de Datos (SGBD, o en inglés, DBMS).

**Entidad:** Abstracción del dominio (cosa, objeto o concepto) que se distingue del resto

**Relación:** Vínculo entre instancias de uno o más conjuntos de entidades

**Entidad débil:** Una entidad débil requiere de otra entidad para poder identificarse. Posee además un discriminante y puede tener otros atributos.

**Jerarquía:**

**Supraentidad:** Posee identificador y atributos comunes.

**Subentidad:** Tiene atributos específicos y hereda los de la supraentidad.

**Solapamiento:** Existe solapamiento cuando una instancia de una supraentidad puede ser a su vez una instancia de dos o más subentidades.

**Partición parcial:** Una instancia de una supraentidad puede no ser una subentidad.

**Partición total:** Toda instancia de una supraentidad es al menos una subentidad.

**Jerarquía tipo:** Partición total y sin solapamiento

**MR:**

Relación  $\equiv$  **Tabla**

Tupla  $\equiv$  Fila  $\equiv$  **Registro**

Atributo  $\equiv$  Columna  $\equiv$  **Campo**

Restricciones de integridad: Clave primaria (**PK**) y foránea (**FK**)

**Agregación:** La agregación es una abstracción en la cual una interrelación (junto con sus entidades vinculadas) es tratada como una entidad de alto nivel y puede participar de interrelaciones. Por supuesto, las entidades vinculadas a una agregación también pueden tener sus propias interrelaciones individualmente. El concepto de agregación se denota con un rectángulo conteniendo a la interrelación y sus entidades vinculadas.

**Normalización:** Mediante la normalización buscamos obtener un esquema de base de datos con redundancia mínima. El proceso requiere conocer las dependencias funcionales. La forma normal establece el nivel de redundancia requerido.

**Desnormalización:** Proceso por el cual se incorpora redundancia a un esquema de base de datos. Normalmente se efectúa sobre bases de datos estáticas (históricas). Comúnmente utilizado en sistemas ad-hoc.