

## **HINTS**

### **USO DEL MODELO ENTIDAD - RELACIÓN**

En la mayoría de los casos, basta pasar del Modelo Relacional a la implementación, obviando el Modelo Entidad - Relación, debido a que para el desarrollo de sistemas es éste el que aporta toda la información.

### **MODELADO DE UNA BASE DE DATOS**

Cuando se modela una base de datos es indispensable ir analizando los diseños, para así verificar si dan respuesta a todos los requerimientos funcionales.

### **USO DE VALORES NULOS**

En el modelo relacional, evitar que un atributo pueda contener valores nulos.

### **DEFINICIÓN DE DATOS**

En el modelo relacional definir bien los tipos de datos de cada atributo, sobre todo su capacidad, esto evitará iteraciones en el desarrollo.

### **USO DE VALORES POR DEFECTO**

Evitar definir valores por defecto en los atributos de cada entidad, y preferir que se solucione a nivel sistémico; esto evitará que se encubran fallas en el ingreso de los datos.

### **NORMALIZACIÓN EN LA VIDA REAL**

A medida que se adquiere experiencia, el proceso de normalización será cada vez más automatizado; por eso, las empresas tienden a no incluirlo en sus diseños. El estudiante debe acostumbrarse a realizar el proceso de normalización y optimización mientras esto no ocurra.

## ENTIDADES FUERTES Y DÉBILES

Cuando trabajamos con relaciones de entidades, podemos encontrar algunos términos peculiares, entre éstos: las entidades descritas como "fuertes" o "débiles". Este concepto depende únicamente de si una entidad tiene un ID o no.

Estos dos conceptos se pueden definir de la siguiente manera:

Una entidad **fuerte** no depende de ninguna otra entidad en el esquema, y siempre tendrá una primary key.

Por otra parte, una entidad **débil** depende de una entidad fuerte para asegurar su existencia; y a diferencia de una fuerte, ésta no tiene ninguna primary key.

## EL MODELO RELACIONAL Y SUS DIFERENCIAS CON EL MODELO CONCEPTUAL

La principal diferencia entre el modelo de datos conceptual y el relacional, es que el primero representa **entidades y sus relaciones**, mientras que el segundo proporciona **más detalles, incluidos atributos, claves primarias y externas**, además de entidades y relaciones.