Revisiones del Modelo conceptual

Decisiones sobre el modelo conceptual:

* Conviene generar una entidad con un concepto nuevo o agregar un atributo a una entidad existente? la conveniencia es lo que el diseñador decide en base a lo que charla con el cliente. Por ejemplo, si yo dudo entre generar un nuevo atributo o una nueva entidad (agregando con una relación), me conviene MÁS generar una entidad, ya que a la larga me genera más beneficios.
* Cuando se debe usar una generalización y cuando el concepto representa una clasificación? Generalizo cuando puedo aplicar jerarquías, clasifico cuando armo una jerarquía y los hijos NO tienen atributos, por lo tánto para eso me basta una única entidad sin aplicar jerarquía.
* Convienen los atributos compuestos? O se deben generar atributos simples? Los compuestos son para describir más correctamente, pero no son obligatorios. Es a decisión del diseñador.

Compleción: un modelo es completo cuando representa todas las características del dominio de aplicación (análisis de requerimientos). 🡪revisar problemas, requerimientos, enunciados, y viendo que todo se cumple.

Corrección: usar bien las propiedades de conceptos E-I (Entidad-Relación = Entidad - Interrelación)

> **Sintáctica**: si los conceptos del E-I se usan correctamente.

> **Semántica**: los conceptos se usan de acuerdo a su definición, ósea, si no cometiste error de escritura o si no cometiste errores de interpretación a la hora de construir el modelo.

a) Usar atributos en lugar de entidades (pregunta anterior).

b) Olvidar una generalización (omitir jerarquías).

c) Olvidar una propiedad de herencia (casi imposible de que ocurra, puede que ocurran superposiciones 🡪 ej, definir un atributo en el padre y redefinirlo en el hijo).

d) Usar entidades en lugar de interrelaciones.

e) Olvidar un identificador de una entidad (no se puede olvidar, excepto si la entidad NO tiene un identificador posible).

f) Omitir cardinalidad (nada es obvio, siempre se debe definir necesariamente).

Minimalidad: cada aspecto aparece una sola vez en el esquema conceptual. Si hay información redundante ya hay problemas contra la minimalidad.

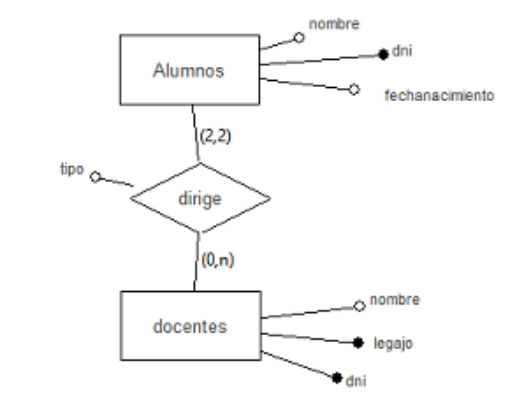
En el modelo LÓGICO la minimalidad NO debe cumplirse 🡪 debe solucionarse.

° Atributo derivado 🡪 atenta con la minimalidad.

° Ciclo de relaciones: como un gráfo, un ciclo de relaciones sería: A se relaciona con B, B se relaciona con C, C se relaciona con A. 🡪 puede representar información redundante, atenta a la minimalidad.

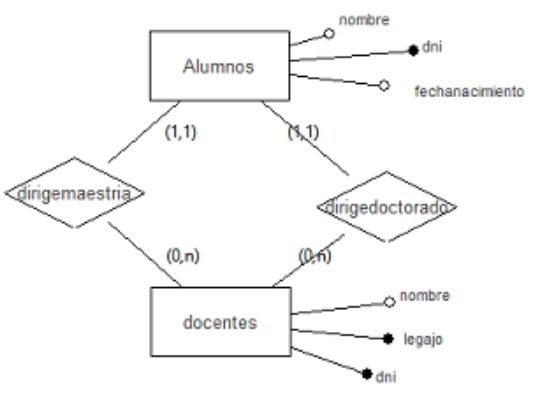
Expresividad: representa los requerimientos de manera natural y se puede entender con facilidad (todo esto en el modelo conceptual, sí podemos entender de muy buena forma el modelo, significa un modelo expresivo)

Ejemplo de un modelo no expresivo:



¿Qué es tipo? 🡪 no lo sabemos, por lo tanto carece de expresividad.

Ejemplo de modelo expresivo:



* Ahora “tipo” deja de ser algo ambiguo, para indicarnos que el problema implica maestría y doctorado. Un alumno puede ser dirigido por 1 docente que es el director de la maestría, y ese alumno también puede ser dirigido por 1 docente que es el director del doctorado. A su vez, un docente puede dirigir desde 0 hasta N alumnos.

Autoexplicación: un esquema conceptual se explica a sí mismo cuando puede representar un gran número de propiedades usando el mismo modelo conceptual, sin otros formalismos.

* Eliminar sub-entidades colgantes de la generalización.
* Eliminar entidades colgantes.
* Crear generalización: dos entidades similares crean una jerarquía de generalización.
* Crear subconjuntos.

Extensibilidad: un esquema se adapta fácilmente a requerimientos cambiantes cuando puede descomponerse en partes donde se realizan los cambios. (modularizar)

Cualquier cambio en el sistema implica un cambio en la BD.

-Como construir un modelo extensible?

> Siendo claro.

> Documentando el modelo.

> Haciendo las cosas por partes, sin cruzar mucho.

> Documentar todo lo hecho.

> Haciendo todos esos pasos logro el siguiente paso…

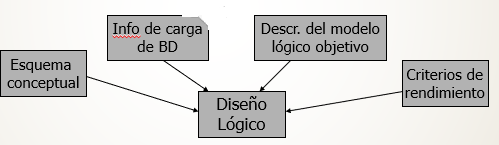
Legibilidad: es legible si se mira y SE ENTIENDE.

* Utilizar herramientas automatizadas (Caser).
* Estructuras simétricas.
* Se minimiza el número de cruces (padre arriba, hijos abajo).
* Generalización sobre los hijos.
* En resumen, haciéndolo BELLO dou.

Modelo Lógico

° Diseño lógico de alto nivel usando E-R. Una vez terminado el módelo conceptual, se llega al diseño lógico que contempla todas las necesidades del cliente anteriormente planteadas.

* Convertir el esquema conceptual en un esquema lógico.
* Enfoque global del diseño lógico.



1. Modelo lógico objetivo: es el acercamiento a la solución en computadora (se decide el modelo lógico objetivo para proceder) 🡪 consiste en decidir: orientado a objetos (se usa pero no a nivel mundial), relacional (se usa y vamos a trabajar éste), jerarquico (desuso), orientado a red (desuso). Para cada modelo lógico se llegan a distintas soluciones.
2. Información de carga de BD y Criterios de rendimiento: los criterios de rendimiento sirven de orientación a la hora de tomar decisiones que afecten al rendimiento de la BD (ej, una solución legible vs una solución rápida). La información de carga de la BD 🡪 es la suposición de cómo van a estar los datos.

° Decisiones: para generar un modelo lógico orientado al modelo RELACIONAL, debemos tomar decisiones sobre lo siguiente…

1. Atributos derivados (minimalidad atentada).
2. Atributos polivalentes (no se aceptan en modelo relacional).
3. Atributos compuestos (no se admiten atributos compuestos en relacional).
4. Ciclo de relaciones (minimalidad atentada).
5. Jerarquías (un modelo relacional no admite jerarquía, ni tiene concepto de herencia).

El modelo relacional necesita algunos cambios respecto de los puntos incompatibles. El modelo Orientado a Objetos si permite 3 puntos anteriores. Pero en la materia usaremos BD relacionales.

**Resoluciones para adecuarse al modelo RELACIONAL**:

Atributos derivados: atenta contra la minimalidad.

* Ventajas de dejar? Nos permite conseguir un valor X muy rápido, es más eficiente que ESTÉ en la BD. El tema con los atributos derivados es que hay que mantenerlos ACTUALIZADOS.
* Ventajas de sacar? Si lo saco debo calcular el valor X, por lo que implica ineficiencia.
* Resumen 🡪 cambia mucho, se usa poco, ENTONCES SACO. Cambia poco, se usa mucho, LO DEJO. (decisión extrema, pero es decisión nuestra)

Ciclo de entidades: atenta contra la minimalidad.

* Al fin y al cabo debemos tener una balanza para decidir si sacar la relación que provoca redundancia, o dejarla.

Ejemplo: yo quiero ir de Quilmes a Mar Del Plata, y quiero ir por ruta 2 pero está cortada, puedo ir por otra ruta y llegar a MDP?, la respuesta es SÍ, por lo tanto la ruta 2 es redundante.

* Si yo tengo más de un camino entre una entidad a otra, entonces hay redundancia que ATENTA CON LA MINIMALIDAD.
* El tema final es el siguiente, sí el camino adicional que atenta contra la minimalidad se trata de un camino que agiliza y que EFECTIVAMENTE es usado bastante entonces nos conviene tener dicha redundancia a mano.
* Puede ocurrir que exista un ciclo, pero que éste sea mínimo, es decir, no hay redundancia, y todas las relaciones son importantes: si saco alguna relación no se cumple la interpretación correcta del problema.

Atributos compuestos: no son permitidos en el modelo relacional.

* Opción A: generar un único atributo que se convierta en la concatenación de todos los atributos simples que contiene el compuesto (usada en el pasado).
* Opción B: definir los atributos simples sin un atributo compuesto que resuma. Aumenta cantidad de atributos pero permite definir cada dato de forma independiente (la que usa Bertone 🡪 apunta a esta).
* Opción C: generar una nueva entidad que va a representar un atributo compuesto, conformada por cada atributo simple que contiene. Dicha nueva entidad se relaciona con la entidad a la cual pertenecía el atributo compuesto .

Atributos polivalentes: no se aceptan en el modelo relacional.

* Un atributo en el modelo relacional debe ser necesariamente simple.
* Por lo tanto, **se genera una nueva entidad con ese atributo, y se genera una relación entre la nueva entidad y la entidad que contenía el atributo polivalente.**

Jerarquías: no se admiten en modelo relacional ni tampoco existe el concepto de herencia.

* Soluciones:
  + **Eliminar entidades hija 🡪 deja sólo la entidad padre, la cuál incorpora todos los atributos de sus hijos. Cada uno de éstos atributos deberá ser opcional (cardinalidad mínima en 0). Aparece un atributo monovalente obligatorio llamado “categoría”, el cuál nos va a decir a qué entidad hija corresponde el dato guardado en persona.**

**🡪resume las entidades y las relaciones. Pero saca más identificadores.**

* + **Eliminar entidades padre 🡪 deja sólo las especializaciones (hijas) con ésta solución los atributos del padre deberán incluirse en cada uno de los hijos. Con el tema identificadores, la herramienta CASER apuesta x una solución segura y, si en hija y padre hay identificador, se genera un identificador compuesto en la hija (es la solución segura, se puede modificar para tener identificadores simples).**

**🡪genera más relaciones y entidades. Conserva identificadores para futuras búsquedas.**

**🡪Si la cobertura de la jerarquía es parcial no se puede aplicar la solución para eliminar la entidad padre. Por qué? Algunos elementos del padre no están cubiertos por las hijas. Si se quita la entidad padre dichos elementos no tendrán más cabida en el modelo (se pierde información).**

* + **Conservar todo 🡪 convierte la jerarquía en relaciones 1 a 1 entre el padre y cada uno de los hijos, ésta solución permite que las entidades de la jerarquía mantengan sus atributos con una relación explícita entre padre e hijos. Es la solución más larga, pero es la que mejor capta el modelo conceptual. Hoy por hoy es la más eficiente y más usada. La pérdida de performance NO es significativa, es una solución más simple y ordenada 🡪 permite que el sistema sea más fácil de mantener a futuro.**