



# Diseño de Bases de Datos

Clase 2

# Agenda

## Revisiones del modelo conceptual

- Decisiones
- Transformaciones

## Modelo lógico

- Atributos derivados, compuestos y polivalentes
- Ciclos de entidades
- Jerarquías

## Modelo físico

- Conversión de entidades
- Conversión de relaciones

## Normalización

- Restricciones
- Dependencias
- Normalización

# Revisiones del modelo conceptual

## Decisiones

- Conviene generar una entidad con un concepto nuevo? O agregar un atributo a una entidad existente?
- Cuando se debe utilizar una generalización y cuando el concepto representa una clasificación?
- Convienen los atributos compuestos? O se deben generar atributos simples?

# Revisiones del modelo conceptual

**Compleción:** representa todas las características del domino de aplicación (análisis de requerimientos)

**Corrección:** usar con propiedad conceptos E-I

- Sintáctica: conceptos E-I se usan correctamente
- Semántica: conceptos se usan de acuerdo a su definición. Errores más frecuentes:
  - Usar atributos en lugar de entidades
  - Olvidar una generalización
  - Olvidar una propiedad de herencia
  - Usar entidades en lugar de interrelaciones
  - Olvidar un identificador de una entidad
  - Omitir cardinalidad

# Revisiones del modelo conceptual

**Minimalidad:** cada aspecto aparece una sola vez en el esquema.

- Ciclo de relaciones
- Atributos derivados

**Expresividad:** representa los requerimientos de manera natural y se puede entender con facilidad.

**Autoexplicación:** esquema se explica a si mismos cuando puede representarse un gran número de propiedades usando el modelo conceptual, sin otros formalismos.

- Eliminar sub-entidades colgantes de la generalización
- Eliminar entidades colgantes
- Crear generalización: dos entidades similares, crea una jerarquía de generalización
- Crear Subconjuntos

# Revisiones del modelo conceptual

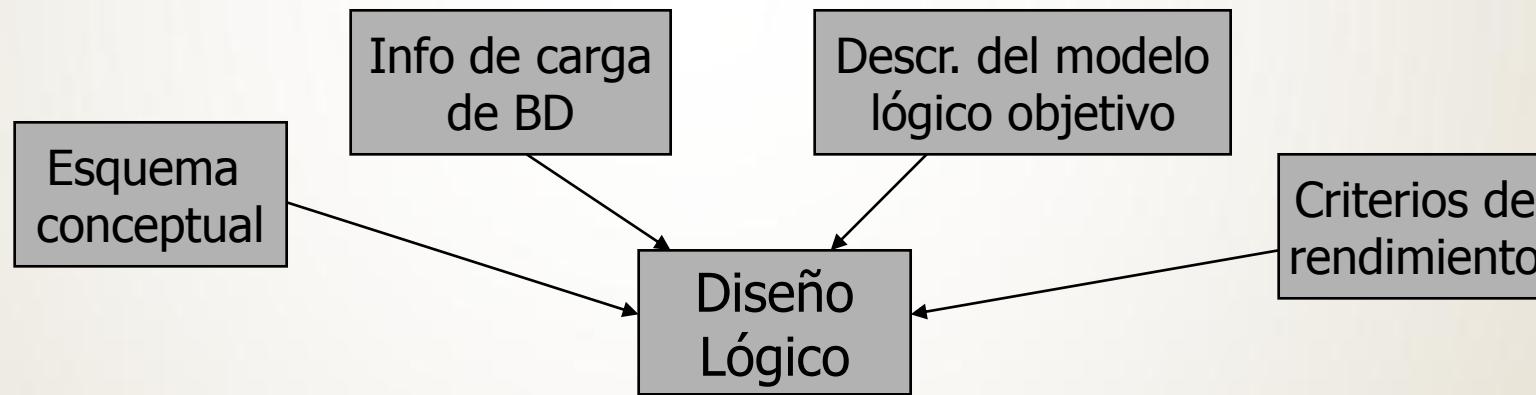
**Extensibilidad:** un esquema se adapta fácilmente a requerimientos cambiantes cuando puede descomponerse en partes, donde se hacen los cambios

**Legibilidad:**

- Utilizar herramientas automatizadas
- Estructuras simétricas
- Se minimiza el número de cruces
- Generalización sobre los hijos

# Modelo lógico

- ▶ Diseño lógico de alto nivel usando E-R
  - ▶ Convertir el esquema conceptual en un esquema lógico
  - ▶ Enfoque global del diseño lógico





# Modelo lógico

## Decisiones

- Atributos derivados
- Atributos polivalentes
- Atributos compuestos
- Ciclo de relaciones
- Jerarquías

# Modelo físico

El modelo físico (relacional) representa la BD como una colección de *relaciones*.

- En otros términos → cada relación se asemeja a una tabla de valores, o a un archivo plano de registros.
- Un registro o un elemento de una relación (tabla) se denomina *tupla*.
- Una atributo mantiene su nombre
- Cada tabla de valores resultante se denomina *relación*
  - Cada relación se obtiene a partir de una entidad o una relación ER.
- El tipo de datos que describe los tipo de valores de un atributo se denomina *dominio*.

# Modelo físico

## Pasos

- Eliminación de identificadores externos
- Selección de claves
  - Primaria
  - Candidata
- Conversión de entidades
- Relaciones

# Modelo físico

## Relaciones

- Cardinalidad Muchos a muchos
- Cardinalidad Uno a Muchos
  - **Clave foránea:** atributo/s de una tabla que en otra tabla es/son CP y que sirven para establecer un nexo entre ambas estructuras
  - Cobertura total
  - Cobertura Parcial
- Relaciones recursivas
- Relaciones ternarias

# Modelo físico

## Integridad referencial

- Propiedad deseable de las BD
- Asegura que un valor que aparece para un atributo en una tabla, aparezca además en otra tabla
- Tipos de IR:
  - Restringir la operación
  - Realizar la operación en cascada
  - Establecer la clave Foránea en nulo
  - No hacer nada

# Restricciones

## Restricciones de dominio

- Especifican que el valor de c/atributo A debe ser un valor atómico del dominio de A.

## Restricciones de clave

- Evita que el valor del atributo clave genere valores repetidos

## Restricciones sobre nulos

- Evita que un atributo tome nulo en caso de no ingresarle valor

# Restricciones

## Restricciones de integridad

- Ningún valor de la clave primaria puede ser nulo.

## Restricción de integridad referencial

- Se especifica entre dos relaciones y sirve para mantener la consistencia entre tuplas de las dos relaciones
- Establece que una tupla en una relación que haga referencia a otra relación deberá referirse a una tupla existente en esa relación
- Clave foránea está representada por un atributo de una relación que en otra es clave primaria.

# Restricciones

Las operaciones de Alta, Baja y Modificación (ABM) pueden generar violaciones a las restricciones anteriores.

- Alta
  - Puede violar: valor nulo para clave, repetición de la clave, integridad referencial, restricciones de dominio.
  - Si se viola la regla, la operación se rechaza
- Baja
  - Puede violar: integridad referencial (se procede como en el caso anterior)
- Modificación
  - Puede violar: cualquiera de las operaciones.

# Dependencias Funcionales

## Definición

- Una DF es una restricción entre dos conjuntos de atributos de la BD.
- Formalmente → una DF  $X \rightarrow Y$  entre dos conjuntos de atributos X e Y que son subconjuntos los atributos (R) de una relación (r), especifica una restricción sobre las posibles tuplas que podrían formar un estado de la relación r en R.
- La restricción indica que si t1 y t2 son dos tuplas cualesquiera en r y que si  $t1[X] = t2[X]$  entonces debe ocurrir que  $t1[Y] = t2[Y]$ .
- Esto significa que los valores del componente Y de una tupla de r dependen de los valores del componente X.

# Dependencias Funcionales

$X \rightarrow Y$

- El atributo Y depende del atributo X, ó
- El atributo X determina el valor único del valor Y, ó
- El valor del atributo Y está determinado por el valor del atributo X, ó
- Y depende funcionalmente de X.

Son todos sinónimos

En general

- si una restricción en R dice que no puede haber más de una tupla con un valor X en r (convirtiendo a X en **clave primaria**) entonces  $X \rightarrow Y$  para cualquier Y de R
- Si  $X \rightarrow Y$  en R, no se puede afirmar ni negar que  $Y \rightarrow X$ .      Cuando si y cuando no de esta afirmación???

# Dependencias Funcionales

## Ejemplo 1

- Departamento= (NroDpto, Nombre, #empleados)
  - NroDpto → nombre
  - NroDpto → #empleado
  - Nombre → #empleado ??
    - Cuando sí?
    - Cuando no?

## Ejemplo 2

- Empleado = (NroEmpl, Nombre, DNI, Sexo)
  - NroEmpl → nombre
  - NroEmpl → dni
  - NroEmpl → sexo
  - DNI → nroempl??
    - Cuando sí?
    - Que otras dependencias pueden surgir?

Que conclusiones obtenemos de estos ejemplos?

# Dependencias Funcionales

## Ejemplo 3

- Empl\_proyecto=(nro\_empl, nro\_proy, horasTrabajadas, nombre\_empleado, nombre\_proyecto)
  - (Nro\_empl, nro\_proy) → horastrabajadas
  - Nro\_empl → nombre\_empleado
  - Nro\_proy → nombre\_proyecto
  - Si continuamos en análisis de la transparencia anterior
    - (nro\_empl, nro\_proy) → nombre\_empleado ??
    - (nro\_empl, nro\_proy) → nombre\_proyecto ??
  - Que conclusión podemos obtener?

# Dependencias Funcionales

## Dependencia funcional completa

- Si A y B son atributos de una relación r, B depende funcionalmente de manera completa de A, si B depende de A pero de ningún subconjunto de A.
- En la transparencia anterior
  - $(\text{nro\_empl}, \text{nro\_proy}) \rightarrow \text{nombre\_empleado}$
  - $\text{Nro\_empl} \rightarrow \text{nombre\_empleado}$
  - Ambas funcionales, cual completa?
- $(\text{nro\_empl}, \text{nro\_proy}) \rightarrow \text{nombre\_proyecto}$
- $\text{Nro\_proy} \rightarrow \text{nombre\_proyecto}$
- Idem anterior

# Dependencias Funcionales

## Dependencia funcional parcial

- $A \rightarrow B$  es una dependencia funcional parcial si existe algún atributo que puede eliminarse de A y la dependencia continúa verificándose
- En la transparencia anterior
  - $(\text{nro\_empl}, \text{nro\_proy}) \rightarrow \text{nombre\_empleado}$
  - $\text{Nro\_empl} \rightarrow \text{nombre\_empleado}$
  - La primera es una dependencia PARCIAL
- $(\text{nro\_empl}, \text{nro\_proy}) \rightarrow \text{nombre\_proyecto}$
- $\text{Nro\_proy} \rightarrow \text{nombre\_proyecto}$
- Idem anterior

# Dependencias Funcionales

## Dependencia funcional transitiva

- Una condición en la que A, B y C son atributos de una relación tales que  $A \rightarrow B$  y  $B \rightarrow C$  entonces C depende transitivamente de A a través de B
- Ejemplo
  - Nro\_empleado  $\rightarrow$  nombre, posición, salario, nro\_dept, nombre\_dept
  - Nro\_dept  $\rightarrow$  nombre\_dept.
  - En este ejemplo
    - A = nro\_empleado
    - B = nro\_dept
    - C = nombre\_dept.

# Dependencias Funcionales

## Resumen

- Dependencia funcional
- Dependencia parcial
  - Parte\_clave → no\_clave
- Dependencia transitiva
- Dependencia Boyce Codd (explicada más adelante)
  - No\_clave → parte\_clave

# Normalización

## Definición:

- Técnica de diseño de BD que comienza examinando las relaciones que existen entre los atributos (dependencias funcionales). La normalización identifica el agrupamiento óptimo de estos atributos, con el fin de identificar un conjunto de relaciones que soporten adecuadamente los requisitos de datos de la organización.

## Propósito

- Producir un conjunto de relaciones (tablas) con una serie de propiedades deseables partiendo de los requisitos de datos de una organización.

# Normalización

La normalización es una técnica formal que puede utilizarse en cualquier etapa del diseño de BD.

La redundancia de datos en un modelo es la causa primaria de posibles inconsistencias.

Primer paso para un proceso de normalización

- Identificar la CP y las CC de cada relación (tabla) del modelo.

# Normalización

## Inicialmente (1972)

- Primera Forma Normal
- Segunda Forma Normal → sobre DF
- Tercera Forma Normal → sobre DF

## Se incorpora luego (1974)

- Forma Normal de Boyce Codd → sobre DF

## Luego 1977 y 1979

- Cuarta forma normal → sobre DM
- Quinta forma normal

# Normalización

Proceso → incremental → cada vez más restrictivo

- Comienza con BD en forma NO normal
- A medida que se avanza las relaciones (tablas) tiene un formato cada vez más restringido y son menos vulnerables a anomalías de actualización.
- En general, 1NF es muy restrictiva (se aplica siempre)
- El resto puede ser opcional, de hecho 2NF y 3NF normalmente se aplican siempre.

# Normalización

## Primera Forma Normal (1NF)

- Una tabla que contiene uno o más grupos repetitivos no está en 1FN, o sea una tabla que tenga atributos polivalentes.
- Un modelo estará en 1FN si para toda relación r del modelo (tabla) cada uno de los atributos que la forman es si y solo sí monovalente.
- Ej persona = (dni, nombre, sexo, títulos\*)
  - Se observa que el atributo títulos es polivalente
  - Solución
    - Persona = (dni, nombre, sexo)
    - títulos = (id, descripción)
    - Posee = (dni, id \_ título)

# Normalización

## Segunda forma normal (2NF)

- Una tabla que tenga atributos que dependan parcialmente de otro no está en 2NF
- Un modelo está en 2NF si y solo sí está en 1NF y para toda relación r del mismo (tabla) no existen dependencias parciales.
- Ej renta = (#cliente, #propiedad, nombrecliente, nombre propietario, monto renta, fecha inicio, duración)

# Normalización

## Dependencias

- #cliente, #propiedad → nombrecliente, nombrepropietario, monto renta, fecha inicio, duración (DF)
- #cliente → nombrecliente (DP)
- #propiedad → nombrepropietario (DP)

## Solución

- Cliente = (#cliente, nombre )
- Propiedad = (#propiedad,nombrepropietario)
- Renta = (#cliente, #propiedad, monto renta, fecha inicio, duración))

# Normalización

Ej 2: empleadoproyecto = (dniempleado, #proyecto, horas trabajadas, nombre empleado, nombre proyecto, fecha inicio proyecto, fecha inicio empleado proyecto)

- Dependencias funcionales
  - Dniempleado, #proyecto → horas trabajadas, nombre empleado, nombre proyecto, fecha inicio proyecto, fecha inicio empleado proyecto (DF)
  - Dniempleado → nombre empleado (DP)
  - #proyecto → nombre proyecto (DP)
- Solución
  - Empleados = (dniempleado, nombre empleado)
  - Proyectos = (#proyecto, nombre proyecto, fecha inicio proyecto)
  - Empleadoproyecto = (dniempleado, #proyecto, fecha incio empleado proyecto)

# Normalización

## Tercera forma normal (3NF)

- Una tabla que tenga atributos que dependan transitivamente de otro no está en 3NF
- Un modelo está en 3NF si y solo sí está en 2NF y para toda relación r del mismo (tabla) no existen dependencias transitivas.
- Ej empleado = (dniempleado, nombreempleado, #depto, nombredepto )
  - Dependencias
    - $\text{Dniempleado} \rightarrow \text{nombreempleado, } \#depto, \text{ nombre depto}$  (DF)

# Normalización

- ▶ #depto → nombredepto (DT)
- ▶ Solución
  - ▶ Empleado = (dniempleado, nombre, #depto )
  - ▶ Departamento = (#depto, nombredepto)
- ▶ Ej2 parcelas = (#propiedad, municipio, númeroparcela, area, precio, tasa fiscal)
- ▶ Dependencias
  - ▶ #propiedad → municipio, numeroparcela, area, precio, tasa fiscal (DF)
  - ▶ Municipio → tasa fiscal (DT)
  - ▶ Area → precio (DT)
- ▶ Solución
  - ▶ Parcela = (#propiedad, municipio, númeroparcela , area)
  - ▶ Municipio = ( municipio, tasa fiscal)
  - ▶ Areas = ( area, precio )

# Normalización

## Boyce Codd forma normal (BCNF)

- Una tabla que tenga atributos que dependan de acuerdo a la definición de Boyce Codd de otro no está en BCNF
- Un modelo está en BCNF si y solo sí está en 3NF y para toda relación r del mismo (tabla) no existen dependencia de Boyce Codd
- Algunos comentarios
  - Fue propuesta como una “suavización” de 3NF
  - Pero resultó ser más restrictiva

## Otra acepción de Boyce Codd

- Una relación (tabla) está en BCNF si y solo sí todo determinante es una clave candidata.

# Normalización

- ▶ Ejemplo entrevista = (#cliente, fechaentrevista, horaentrevista, empleado, lugarentrevista)
  - ▶ las DF existentes son:
    - ▶ #cliente, fechaentrevista → hora entrevista, empleado, lugarentrevista (CP)
    - ▶ Empleado, fechaentrevista, horaentrevista → #cliente (CC)
    - ▶ Lugarentrevista, fechaentrevista, horaentrevista → empleado, #cliente (CC)
    - ▶ Empleado, fechaentrevista → lugarentrevista
  - ▶ Como los tres primeros determinantes son CP o CC no generan inconvenientes.
  - ▶ Debemos, entonces, analizar la cuarta DF.
    - ▶ No hay problema con DP o DT
    - ▶ Pero el determinante no es CC o CP → no está en BCNF

# Normalización

- ▶ Veamos los problemas que pueden surgir

#cliente	fechaentrevista	Horaentrevista	Empleado	lugarentrevista
C123	12/12/2004	12:30 hs.	García	Aula 4
C332	12/12/2004	12:30 hs.	Perez	Aula 3
C340	15/12/2004	13:00 hs.	García	Aula 2
C124	12/12/2004	13:00 hs.	Perez	Aula 3

- ▶ Si el empleado Perez cambia su cita del día 12/12/2004 del aula 3 al aula 20, que pasa? Cuantos renglones hay que cambiar?
- ▶ Entonces es claramente visible que la información está repetida

# Normalización

- ▶ Como resolvemos el problema anterior
  - ▶ Entrevista= (#cliente, fechaentrevista, horaentrevista, empleado)
  - ▶ lugarezreunión = (empleado, fechaentrevista, lugarenrevista)
- ▶ En la conversión realizada
  - ▶ #cliente, fechaentrevista → hora\_entrevista, empleado (CP)
  - ▶ Empleado, fechaentrevista, horaentrevista → #cliente (CC)
  - ▶ Empleado, fecha entrevista → lugarentrevista (CP)
  - ▶ Pero se ha perdido una CC del problema
    - ▶ Lugarentrevista, fechaentrevista, horaentrevista → empleado, #cliente (CC)

# Normalización

## Entonces? Que hacer?

- La decisión de si es mejor detener el proceso en 3NF o llegar a BCNF depende de
  - la cantidad de redundancia que resulte de la presencia de una DF de Boyce Codd.
  - De la posibilidad de perder una CC con la cual se podrían realizar muchos más controles sobre los datos.

# Dependencias Multivaluadas

- ▶ La posible existencia de DM en una relación se debe a 1NF, que impide que una tupla tenga un conjunto de valores diferentes.
- ▶ Así, si una tabla tiene dos atributos multivaluados, es necesario repetir cada valor de uno de los atributos con cada uno de los valores del otro. Así se garantiza la coherencia en la BD.
- ▶ En general, una DM se da entre atributos A, B y C en una relación de modo que para cada valor de A hay un conjunto de valores se B y un conjunto de valores de C, sin embargo los conjuntos B y C no tienen nada entre sí.

# Dependencias Multivaluadas

- ▶ Ejemplo  $T1=(\text{nombre\_empleado}, \text{nombre\_propietario}, \text{sucursal})$

Sucursal	Empleado	propietario
Alfa	Gomez	Perez
Alfa	Gomez	García
Alfa	Rodriguez	Perez
Alfa	Rodriguez	García

- ▶ Supongamos que aparece el empleado Fernandez →

Sucursal	Empleado	propietario
Alfa	Gomez	Perez
Alfa	Gomez	García
Alfa	Rodriguez	Perez
Alfa	Rodriguez	García
Alfa	Fernandez	Perez
Alfa	Fernandez	García

# Dependencias Multivaluadas

► Supongamos que aparece el propietario Alvarez→

<b>Sucursal</b>	<b>Empleado</b>	<b>propietario</b>
Alfa	Gomez	Perez
Alfa	Gomez	García
Alfa	Rodriguez	Perez
Alfa	Rodriguez	García
Alfa	Fernandez	Perez
Alfa	Fernandez	García
Alfa	Gomez	Alvarez
Alfa	Rodriguez	Alvarez
Alfa	Fernandez	Alvarez

► La cantidad de información que se repite es muy alta

# Dependencia multivaluada

- Se dice que  $A \twoheadrightarrow B$  y que  $A \twoheadrightarrow C$
- Se lee A multidetermina B y A multidetermina C
- No es un problema el hecho que un atributo esté multideterminado.

# Normalización

## ► Cuarta forma Normal (4FN)

- Un modelo está en 4FN si y solo sí está en BCNF y para toda relación r del mismo (tabla) sólo existen dependencia multivaluadas triviales.
- Cuales DM son triviales?
  - $A \rightarrow\!\!\! \rightarrow B$
  - $A \rightarrow\!\!\! \rightarrow C$
  - Las anteriores son triviales.
- Volvamos al ejemplo anterior →
  - $(A,B) \rightarrow\!\!\! \rightarrow C$
  - $(A,C) \rightarrow\!\!\! \rightarrow B$

# Normalización

- ▶ Por que la afirmación anterior?
  - ▶ (sucursal, empleado) →→ propietario
  - ▶ (sucursal, propietario) →→ empleado
- ▶ Esto genera mucha repetición de información.
- ▶ Solución
  - ▶ t1= (sucursal, empleado)
  - ▶ t2= (sucursal, propietario)

# Normalización

- ▶ Quinta Forma Normal (5FN)
  - ▶ Un modelo está en 5FN si está en 4FN y no existen relaciones con dependencias de combinación
  - ▶ Una dependencia de combinación es una propiedad de la descomposición que garantiza que no se generen tuplas espurias al volver a combinar las relaciones mediante una operación del álgebra relacional.
  - ▶ En otras palabras ....→

# Normalización

- ▶ Supongamos que tenemos una tabla
  - ▶ Exporta = (compañía, país, producto )

Compañía	Pais	Producto
A	Alfa	Uno
A	Alfa	Dos
A	Beta	Uno
A	Beta	Tres
A	Gamma	Uno

- ▶ Que podemos decir?

- ▶ Supongamos ahora que la compañía A empieza a fabricar el producto cuatro y que lo compran el país Alfa y Gamma

Compañía	Pais	Producto
A	Alfa	Uno
A	Alfa	Dos
A	Beta	Uno
A	Beta	Tres
A	Gamma	Uno
A	Alfa	Cuatro
A	Gamma	Cuatro

# Normalización

- ▶ Se genera una gran repetición de información
- ▶ Caemos en 4FN?
- ▶ Por que?
- ▶ Solución
  - ▶ Exporta = (compañía, pais )
  - ▶ Fabrica = (Compañía, producto)
  - ▶ Compra = (país, producto)
- ▶ Hay mas formas normales?
  - ▶ Si algunos casos más..., no los veremos