



**Departamento de Informática**  
Universidad Técnica Federico Santa María



# Unidad III

## Diseño Lógico de Bases de Datos Relacionales

INF-239, ILI-239 Bases de Datos

Profesora Cecilia Reyes Covarrubias – Casa Central

Diapositivas realizadas con la colaboración Prof. J.Luis Martí – Campus San Joaquín



# TEMARIO UNIDAD III

3.1 Características de los Modelos de Datos Lógicos

3.2 Enfoque Metodológico para diseñar BDR

3.2.1 Enfoque Bottom-up

3.2.2 Enfoque Top-down

3.3 Otras Consideraciones de Diseño Lógico de BDR

3.4 Ejercicios de Diseño Lógico BDR (en clases)



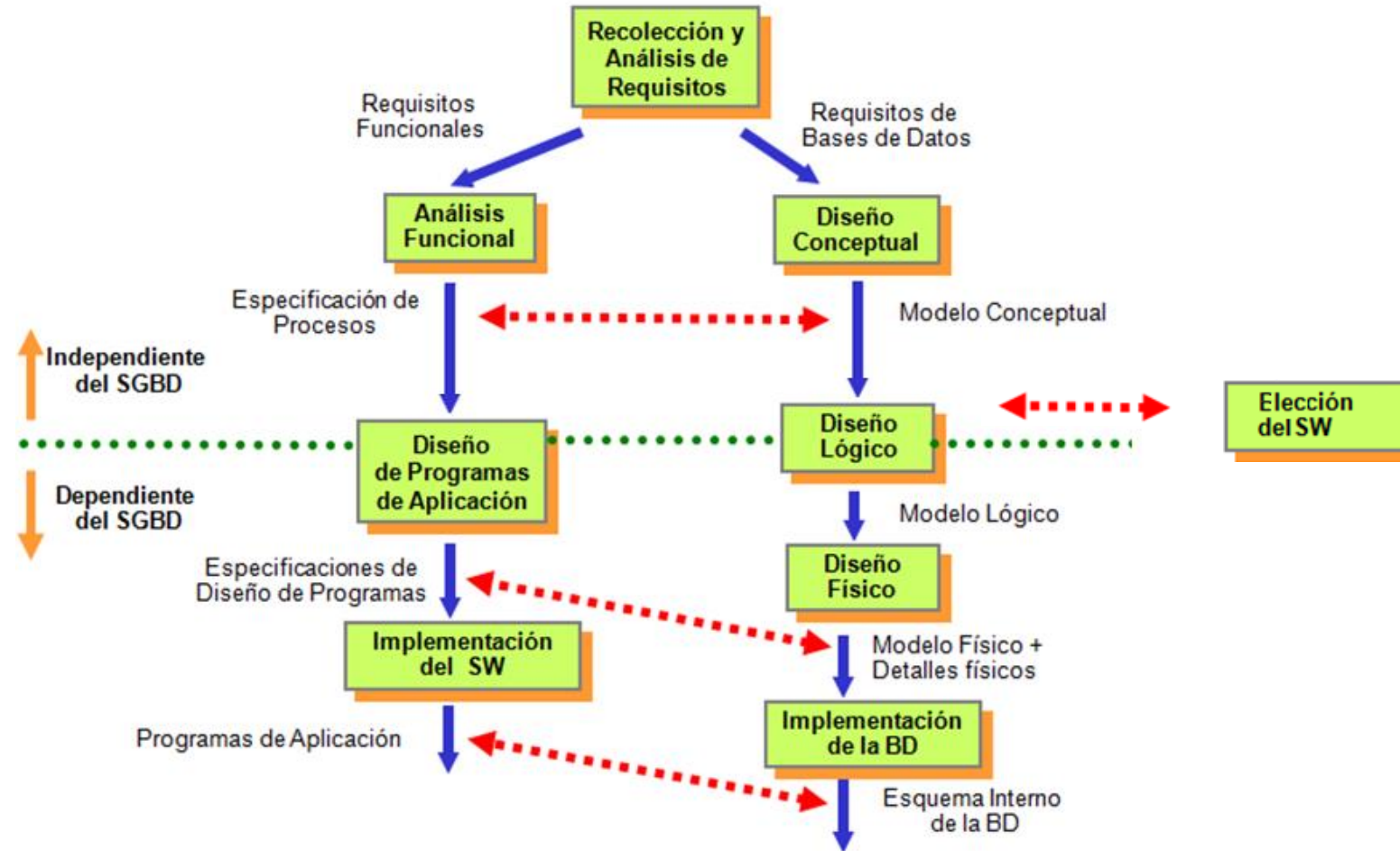
## 3.1 CARACTERÍSTICAS DE LOS MODELOS DE DATOS LÓGICOS



**Departamento de Informática**  
Universidad Técnica Federico Santa María



# ETAPAS PARA DISEÑO DE BD



# MODELO DE DATOS LÓGICO

- Un modelo de datos lógico es aquel modelo de datos que se genera al tomar uno conceptual e incorporarle las características propias de la clase de software (DBMS) sobre el cual se realizará la posterior implementación.
- Modelos de datos lógicos existentes serían:
  - Jerárquico (árbol)
  - Reticular (grafo)
  - **Relacional** (relación o tabla)
  - Orientado al Objeto (objeto)
  - Multidimensional (hipercubo o tabla multidimensional)
  - ...

# CARACTERÍSTICAS MODELO RELACIONAL

- Independencia (física) de los datos.
- Uso de claves lógicas.
- Estrategia de diseño basada en la Teoría de la Normalización (formas normales).
- Lenguaje de consultas de alto nivel (SQL).
- Se denomina relación a un archivo o tabla.
- Una relación corresponde a una tabla bidimensional formada por filas (tuplas o registros) y columnas (atributos o campos).

# CARACTERÍSTICAS MODELO RELACIONAL

- En una relación el orden de las filas y de las columnas es irrelevante.
- Cada columna contiene valores del mismo atributo, los que deben ser del mismo tipo y atómicos. Al conjunto de valores posibles para una columna se le llama **Dominio**.
- Cada fila es única y la unicidad está dada al manejar una columna (o conjunto de columnas) que no soporte valores repetidos, denominada clave primaria (PRIMARY KEY).
- **Cardinalidad** es el número de filas de una relación. Cambia cuando se insertan nuevas tuplas o se borran algunas de las existentes.
- **Grado** es el número de columnas de una relación. Es decir, cada fila debe contener tantos valores como indique su grado (suele no cambiar).

## 3.2 ENFOQUE METODOLÓGICO



**Departamento de Informática**  
Universidad Técnica Federico Santa María





# DISEÑO LÓGICO DE UNA BDR

- **Enfoque Bottom-Up:** mediante la integración de modelos parciales, obtenidos a partir de la normalización de las vistas (salidas, reportes) que componen un sistema.
- **Enfoque Top-Down:** construir un modelo conceptual y luego convertirlo en uno relacional, aplicando reglas de transformación formales.

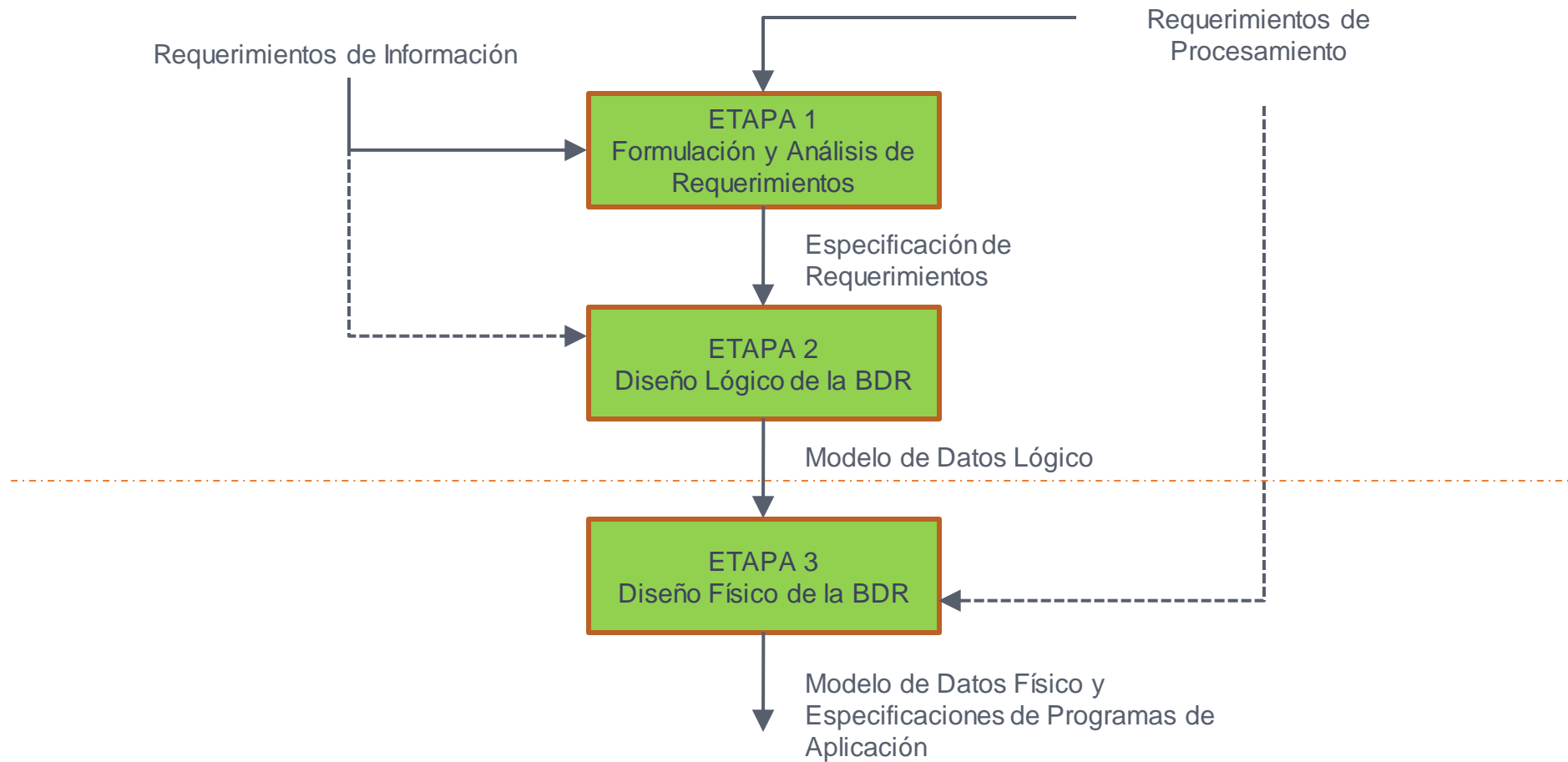
## 3.2.1 ENFOQUE METODOLÓGICO BOTTOM-UP



**Departamento de Informática**  
Universidad Técnica Federico Santa María



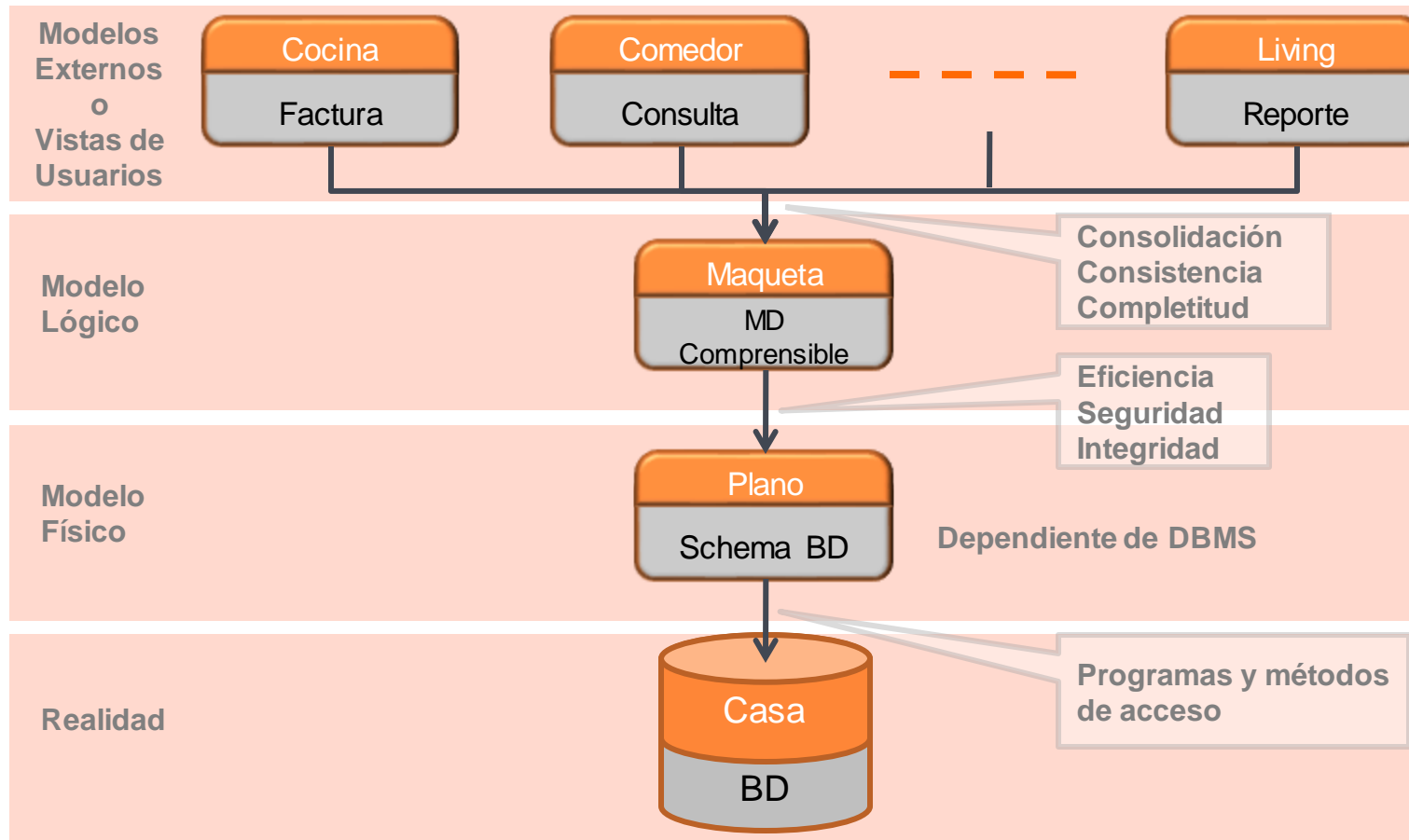
# ENFOQUE BOTTOM-UP



# Recordar Tipos de Modelos de Datos

Analogía con  
un Arquitecto

Arquitecto  
Informático



# ENFOQUE BOTTOM-UP – ETAPA 1

## FORMULACIÓN Y ANÁLISIS DE REQUERIMIENTOS

- Identificar el ámbito o área de la organización, incluyendo sus actores o usuarios.
- Recolectar los requerimientos de información o **vistas de usuarios** (documentos formales o informales), que utiliza el usuario para realizar su trabajo.
- Analizar las vistas para proponer mejoras a ellas y/o crear nuevas vistas.
- Establecer **requerimientos de procesamiento**.

# ENFOQUE BOTTOM-UP – ETAPA 1

## Ejemplo de Vista de Usuario

<b>Distribuidora ABC</b> RUT: 89.900.120-K Los Leones 180 Fono: 56-2-6677889 Santiago-Chile		<b>FACTURA</b>			Nº 4567890  FECHA: __/__/__
NOMBRE CLIENTE: _____ RUT: _____					
DIRECCIÓN: _____ CIUDAD: _____ FONO: _____					
COD-PROD	DESCRIPCIÓN	CANTIDAD	PRECIO UNITARIO	TOTAL LÍNEA	
			SUBTOTAL		
			IVA		
			TOTAL		

# ENFOQUE BOTTOM-UP – ETAPA 1

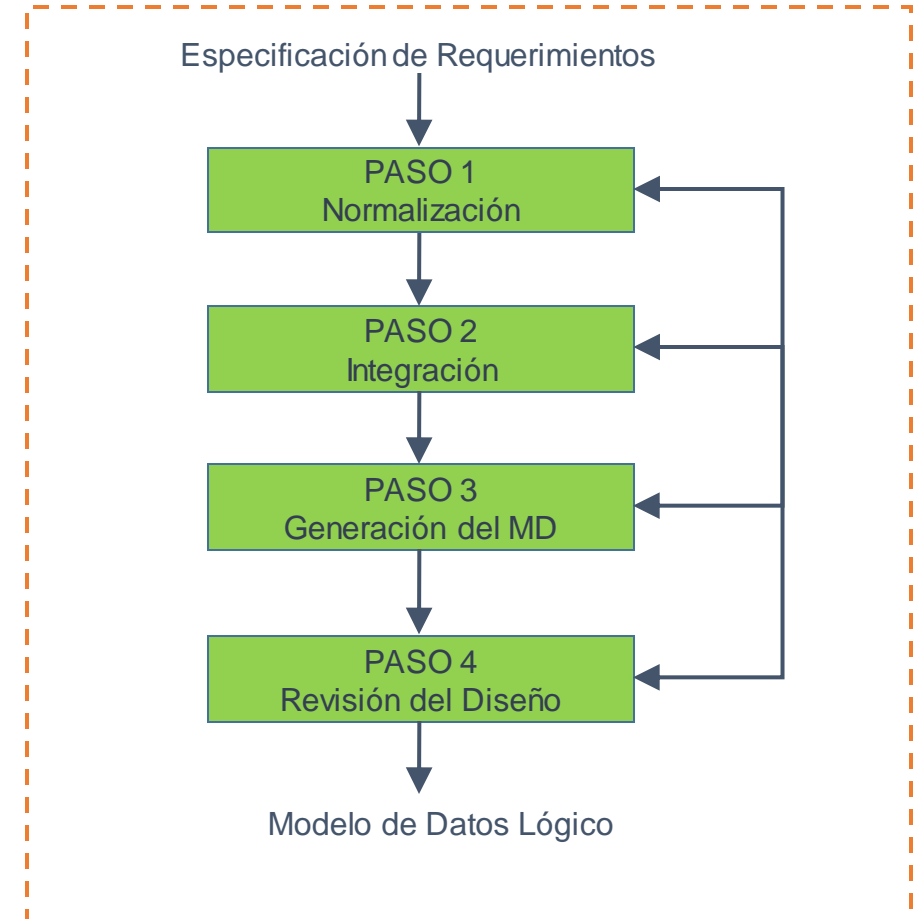
## Ejemplos de Requerimientos de Procesamiento

- **Privacidad:** ¿qué perfiles de accesos se necesitan para la vista?
- **Integridad:** ¿qué reglas de validación y de integridad referencial se debieran considerar para la vista?
- **Tiempo de Respuesta:** ¿la vista requiere ser obtenida en tiempo real?
- **Respaldo:** ¿cada cuánto tiempo conviene respaldar los datos de la vista?
- **Recuperación:** ¿cuánto tiempo se puede esperar para recuperar los datos de la vista en caso que se pierdan?
- **Archivamiento** (archiving): ¿cuándo un dato pasa a ser histórico?
- **Proyecciones Crecimiento:** ¿cuál es el crecimiento proyectado para la vista?

# ENFOQUE BOTTOM-UP – ETAPA 2

## DISEÑO LÓGICO DE BDR

1. Normalización
2. Integración de resultados de la normalización
3. Generación del modelo de datos lógico
4. Revisión del modelo diseñado (ver 3.3)





# ENFOQUE BOTTOM-UP – ETAPA 2

## DISEÑO LÓGICO DE BDR

### Teoría de la Normalización

- Su objetivo es agrupar atributos en relaciones (tablas) que se encuentren **bien estructuradas**, a través de un análisis de las llamadas **dependencias** entre dichos atributos.
- Se entiende por **relación bien estructurada** a aquella relación que minimiza la redundancia de información, evitando inconsistencias y anomalías en las típicas operaciones de inserción, eliminación y actualización sobre una base de datos.
- El resultado de la normalización son relaciones con **Dependencia Funcional** de la clave primaria respecto del resto de los atributos que la componen.

# ENFOQUE BOTTOM-UP – ETAPA 2

## DISEÑO LÓGICO DE BDR

### Teoría de la Normalización

- La **Dependencia Funcional** se obtiene cuando el valor de un atributo permite conocer en forma única, no ambigua y exacta, el valor de otro atributo. Es una asociación entre dos atributos que se define como:

“Para una relación R, el atributo B es funcionalmente dependiente del atributo A, si para cada instancia de A, el valor de A determina en forma única al valor de B”.

- Se escribe como:  $A \rightarrow B$  y se lee:

“A determina a B” o “B es funcionalmente dependiente de A”

- Ejemplos:  $RUT \rightarrow \text{nombre}$

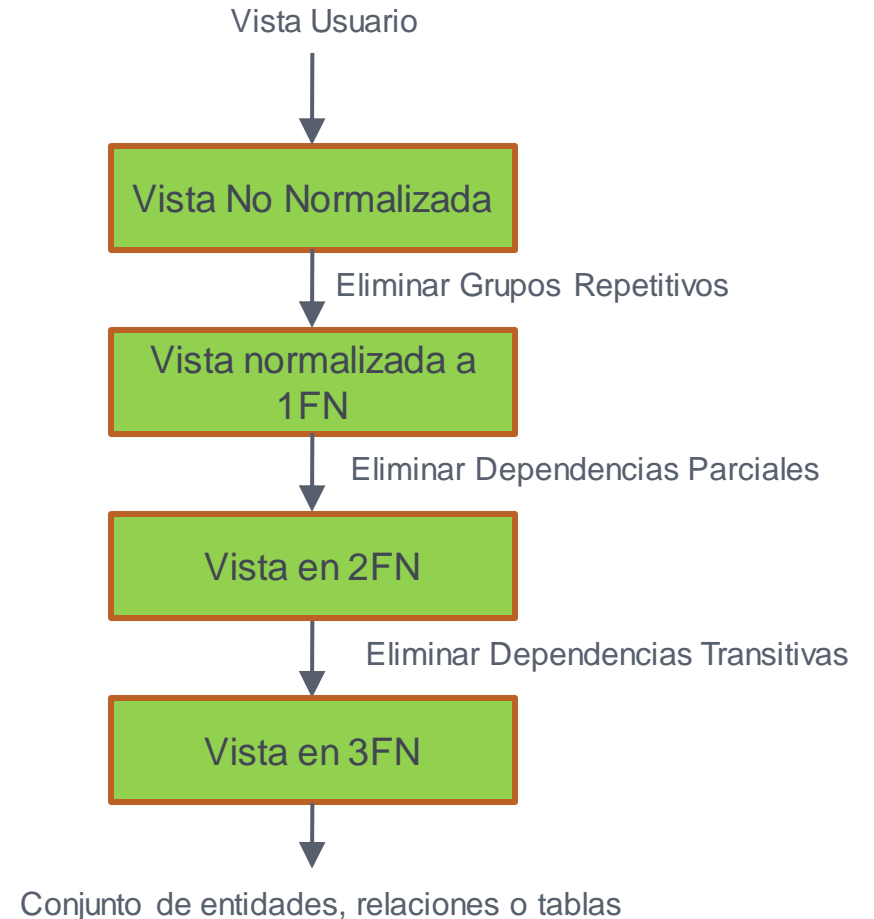
$NroFactura \rightarrow \text{total}$

# ENFOQUE BOTTOM-UP – ETAPA 2

## DISEÑO LÓGICO DE BDR

### Pasos de la Normalización

- Inicialmente, se tiene una Vista No Normalizada.
- Luego, se van analizando diversas dependencias anómalas entre los datos, que de presentarse se deben ir eliminando metódicamente para pasar a una siguiente Forma Normal (FN).
- “Dividir para reinar” es el lema.
- Aplicar hasta 3FN es, por lo general, suficiente para los casos reales, aunque no garantiza que todas las anomalías se hayan eliminado.

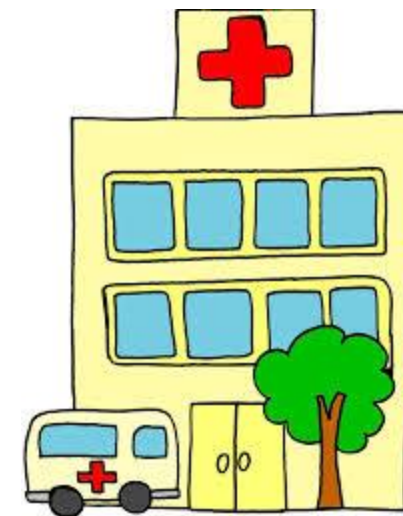


# ENFOQUE BOTTOM-UP – ETAPA 2

## DISEÑO LÓGICO DE BDR

### Ejercicio de la Normalización hasta 3FN: El Caso del Hospital

- Los requerimientos de información están expresados a través de las siguientes vistas:
  - Vista 1: Factura (usuario Cajero)
  - Vista 2: Informe Utilización Piezas (usuario Enfermeras)
  - Vista 3: Pantalla datos Paciente (usuario Recepcionista)
  - Vista 4: Informe para Médicos (usuario Médicos)
- ¿Qué mejoras se pueden realizar a las vistas existentes?
- ¿Qué nuevas vistas se pueden crear?
- En clases se aplicarán los pasos de la normalización para las 4 vistas.



# ENFOQUE BOTTOM-UP – ETAPA 2

## DISEÑO LÓGICO DE BDR

### Ejercicio de la Normalización: El Caso del Hospital

**Hospital Dr. Santana**  
RUT: 89.900.120-K  
Los Leones 180  
Fono: 56-2-6677889  
Santiago-Chile

**FACTURA**

NOMBRE PACIENTE: \_\_\_\_\_#PACIENTE: \_\_\_\_\_

DIRECCIÓN: \_\_\_\_\_CIUDAD: \_\_\_\_\_

COD-SISTEMA-SALUD: \_\_\_\_\_

#ITEM	NOMBRE ITEM	VALOR
SUBTOTAL		
%IMPUESTO		
TOTAL		

¿Qué mejoras se puede realizar a esta vista?

# ENFOQUE BOTTOM-UP – ETAPA 2

DISEÑO LÓGICO DE BDR

## Ejercicio de la Normalización: El Caso del Hospital



<b>Hospital Dr. Santana</b> RUT: 89.900.120-K Los Leones 180 Fono: 56-2-6677889 Santiago-Chile						<b>Informe Utilización Camas</b>	FECHA: __/__/__
#CAMA	#PIEZA	TIPO CAMA	NOMBRE PACIENTE	#PACIENTE	FECHA ALTA ESPERADA		

# ENFOQUE BOTTOM-UP – ETAPA 2

## DISEÑO LÓGICO DE BDR

### Ejercicio de la Normalización: El Caso del Hospital

#### Pantallazo Ingreso Pacientes

#Paciente:

Nombre Paciente:

Dirección:

Teléfono:

Celular:

Ubicación:

Citófono:

Sistema de Salud:

Fecha Admisión:

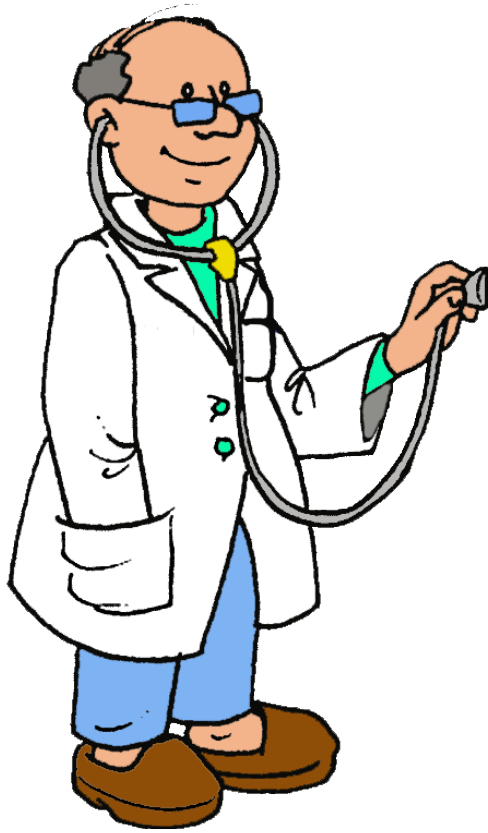
Fecha Alta:

The screenshot shows a web application titled "Sistema de Agenda y Gestión Médica". The main section is "Pacientes" with a search bar containing "ABARCA ROCHA GABRIEL". Below this are tabs for "Datos Personales", "Diagnóstico", "Exámenes", "Recetas", "Vacunas", and "Empresa o Institución". The "Datos Personales" tab is active, showing fields for "Apellido Paterno" (ABARCA), "Apellido Materno" (ROCHA), "Nombre" (GABRIEL), "Fecha", "Sexo", "Edad", "Domicilio" (Calle y No., Colonia, Delegación, C.P., Teléfono, R.F.C.), "Antecedentes" (Tipo, Caso, Peso, Talla, Última Consulta), "Diagnóstico", and "Recetas". There are buttons for "Guardar", "Eliminar", and "Nuevo". The bottom status bar shows "Usuario: Dr. Diego Hernández Santamaria".

# ENFOQUE BOTTOM-UP – ETAPA 2

## DISEÑO LÓGICO DE BDR

### Ejercicio de la Normalización: El Caso del Hospital



#### **Hospital Dr. Santana**

RUT: 89.900.120-K  
Los Leones 180  
Fono: 56-2-6677889  
Santiago-Chile

FECHA: \_\_/\_\_/\_\_

#### **Informe Médico**

#Médico:  
Nombre:  
Dirección:  
Teléfono:  
Especialidad:

#PACIENTE	NOMBRE PACIENTE	UBICACIÓN		DIAGNÓSTICO
		#CAMA	#PIEZA	

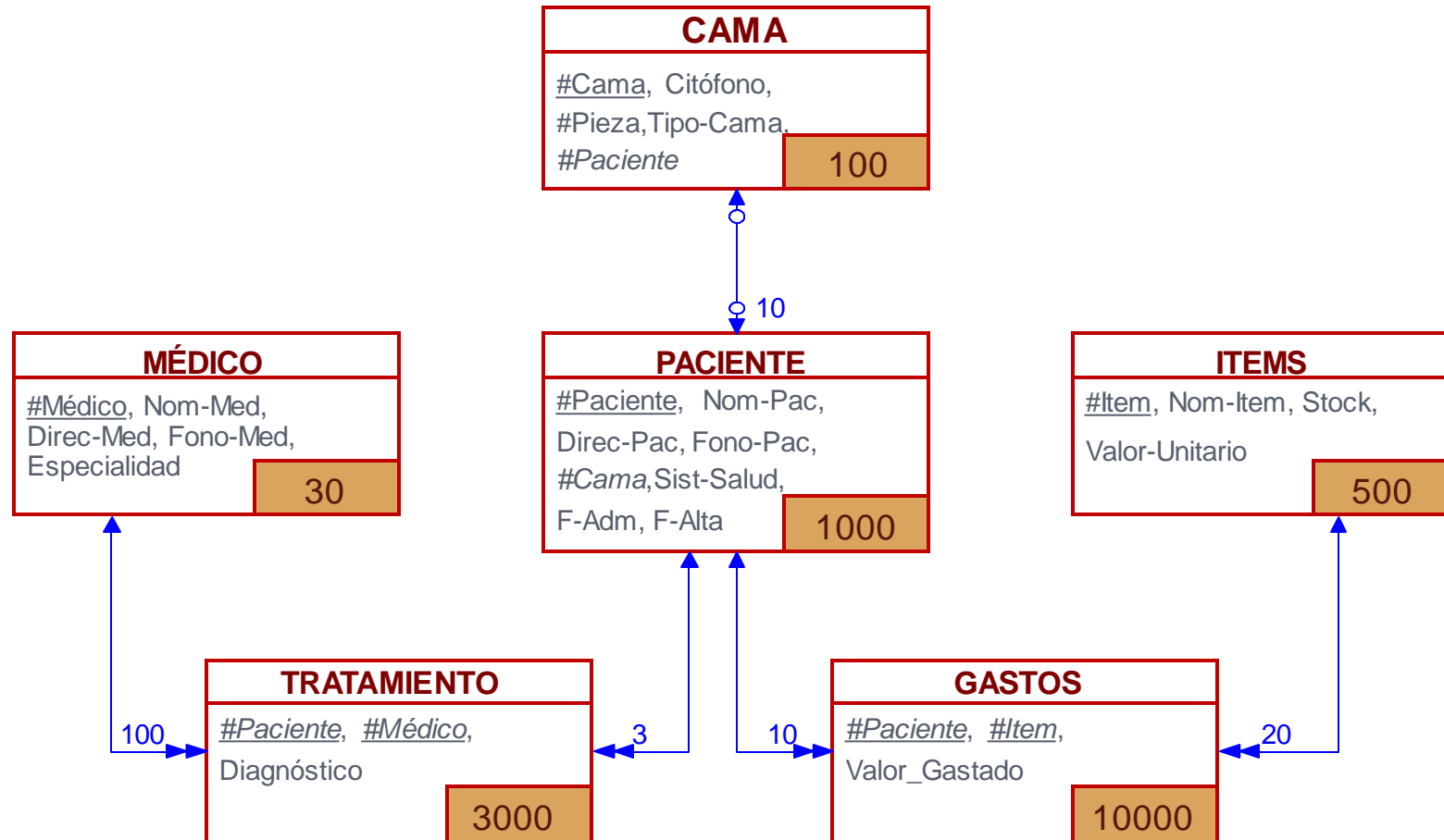


# ENFOQUE BOTTOM-UP – ETAPA 2

DISEÑO LÓGICO DE BDR

## El Caso del Hospital

## Ejercicio de Normalización, Integración y Generación Modelo Lógico



# ENFOQUE BOTTOM-UP – ETAPA 2

## DISEÑO LÓGICO DE BDR

### ¿Por qué normalizar a por lo menos 3FN?

- Para reducir la **redundancia de datos** y las **anomalías de mantención** (operaciones de inserción, eliminación y actualización sobre una base de datos).
- La aplicación del resto de las formas normales soluciona el problema de anomalías restantes, pero genera un número mayor de relaciones o tablas, por lo que reconstruir la vista original puede ser muy costoso.
- Por ende, un equilibrio entre normalización y eficiencia se logra aplicando el proceso hasta la 3FN.
- A través del siguiente ejemplo de normalización, se precisará esto.

# ENFOQUE BOTTOM-UP – ETAPA 2

## DISEÑO LÓGICO DE BDR

**Ejemplo de normalización:** considerar la siguiente vista simplificada de una factura...

RUT-Cliente		Número de Factura	
Razón Social Cliente		Fecha	
Teléfono Cliente			
Código Producto	Nombre Producto	Cantidad	Precio

asumiendo que el precio de un producto no cambia en el tiempo.

# ENFOQUE BOTTOM-UP – ETAPA 2

## DISEÑO LÓGICO DE BDR

La representación no normalizada de la factura es:

factura(#factura, fecha, RUT-cliente, nombre-cliente, teléfono-cliente,  
{código-producto, nombre-producto, precio, cantidad})

- Si ésta fuera una relación o tabla, a simple vista #factura sería clave única (PK), pero no lo es pues habrá que tener tantas filas como detalles de factura existan en ésta, debiéndose repetir dicha clave.
- Se genera un alto grado de redundancia pues habrán datos que se guardarán muchas veces.

# ENFOQUE BOTTOM-UP – ETAPA 2

## DISEÑO LÓGICO DE BDR

- Lo anterior se soluciona eliminando los llamados “grupos repetitivos”.
- Luego, una vista se encuentra en **primera forma normal (1FN)** si no presenta grupos repetitivos.
- Para la vista en desarrollo, su 1FN queda como:

factura(#factura, fecha, RUT-cliente, nombre-cliente, teléfono-cliente)

detalle(#factura, código-producto, nombre-producto, precio, cantidad)



# ENFOQUE BOTTOM-UP – ETAPA 2

## DISEÑO LÓGICO DE BDR



No obstante, esta vista aún presenta algunos problemas:

- **Redundancia de datos:** los datos de cada producto aparecen tantas veces según la cantidad de detalles a los cuales pertenezcan.
- **Anomalía de Inserción:** si se desea insertar los datos de un nuevo producto, no se podrá hacer hasta que no se haya vendido al menos una unidad, y pueda ser incluido en algún detalle.
- **Anomalía de Eliminación:** si hubiera sólo un detalle asociado a cierto producto, y este detalle se elimina (archiva), entonces también se borran los únicos datos del producto.
- **Anomalía de Actualización:** al actualizar el precio de un producto, deberá modificarse en todos los detalles asociados.

# ENFOQUE BOTTOM-UP – ETAPA 2

## DISEÑO LÓGICO DE BDR

- Lo anterior se soluciona eliminando los llamados “**dependencias parciales**”.
- Una dependencia parcial se puede representar como:  
$$a, b \rightarrow c$$
  
y una de las dependencias funcionales siguientes:  
$$\underline{b} \rightarrow c \qquad \underline{a} \rightarrow c$$
- Luego, una vista se encuentra en **segunda forma normal (2FN)** si está en primera forma normal y no presenta dependencias parciales.

# ENFOQUE BOTTOM-UP – ETAPA 2

## DISEÑO LÓGICO DE BDR

- Para la vista en desarrollo, las dependencias parciales identificadas son:

#factura, código-producto → nombre-producto

código-producto → nombre-producto

#factura, código-producto → precio

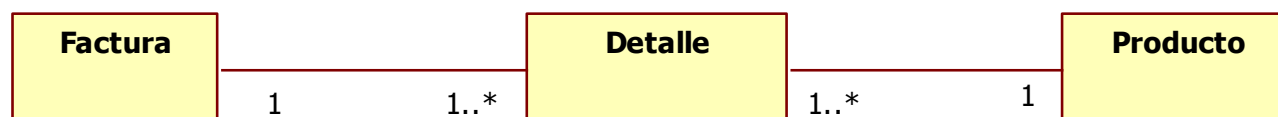
código-producto → precio

- Luego la vista en desarrollo en 2FN queda como:

factura(#factura, fecha, RUT-cliente, nombre-cliente, teléfono-cliente)

detalle(#factura, código-producto, cantidad)

producto(código-producto, nombre, precio)





# ENFOQUE BOTTOM-UP – ETAPA 2

## DISEÑO LÓGICO DE BDR



No obstante, esta vista aún presenta algunos problemas:

- **Redundancia de datos:** los datos de cada cliente aparecen tantas veces según la cantidad de facturas asociadas.
- **Anomalía de Inserción:** si se desea insertar los datos de un nuevo cliente, no se podrá hacer hasta que no tenga asociada al menos una factura que lo incluya.
- **Anomalía de Eliminación:** si hubiera sólo una factura asociada a cierto cliente, y ésta se elimina (archiva), entonces también se borran los datos del cliente.
- **Anomalía de Actualización:** al actualizar el teléfono de un cliente, deberá modificarse en todas las facturas asociadas.

# ENFOQUE BOTTOM-UP – ETAPA 2

## DISEÑO LÓGICO DE BDR

- Lo anterior se soluciona eliminando los llamados “**dependencias transitivas**”.
- Una dependencia transitiva se puede representar como:

$a \rightarrow b$

$a \rightarrow c$

$b \rightarrow c$

donde una de las dependencias funcionales es entre atributos no clave.

- Luego, una vista se encuentra en **tercera forma normal (3FN)** si está en segunda forma normal y no presenta dependencias transitivas.

# ENFOQUE BOTTOM-UP – ETAPA 2

## DISEÑO LÓGICO DE BDR

- Para la vista en desarrollo, las **dependencias transitivas** identificadas son:

#factura → RUT-cliente

#factura → nombre-cliente

RUT-cliente → nombre-cliente

#factura → RUT-cliente

#factura → teléfono-cliente

RUT-cliente → teléfono-cliente

# ENFOQUE BOTTOM-UP – ETAPA 2

## DISEÑO LÓGICO DE BDR

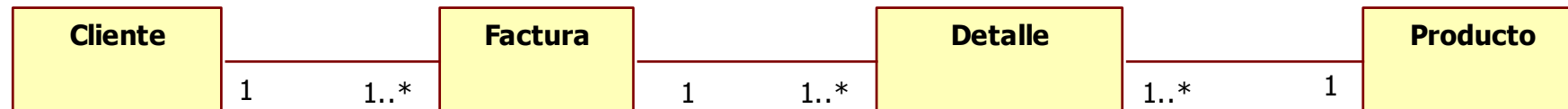
- Luego la vista en desarrollo en 3FN queda como:

cliente(RUT, nombre, teléfono)

factura(#factura, fecha, RUT)

detalle(#factura, código-producto, cantidad)

producto(código-producto, nombre, precio)



# ENFOQUE BOTTOM-UP – ETAPA 2

## DISEÑO LÓGICO DE BDR

### Consideraciones en la integración del resultado de la normalización

- La integración busca concentrar todas las relaciones (o tablas) obtenidas de la normalización de las vistas en un único modelo. Se realiza juntando las relaciones (o tablas) que tienen igualdad de clave primaria.
- Es importante al integrar tener en cuenta:
  1. Sinónimos y homónimos.
  2. Posibilidad que se produzcan dependencias transitivas
  3. Uso de generalización
- A través del siguiente ejemplo se precisarán estos 3 aspectos.

# ENFOQUE BOTTOM-UP – ETAPA 2

## DISEÑO LÓGICO DE BDR

### Consideraciones en la integración del resultado de la normalización

- **Sinónimos:** un mismo atributo es referenciado con nombres diferentes; por ejemplo, RUT y CódigoPersona.
- **Homónimos:** atributos diferentes son referenciados con el mismo nombre; por ejemplo, Nombre y Nombre-Completo.

# ENFOQUE BOTTOM-UP – ETAPA 2

## DISEÑO LÓGICO DE BDR

### Consideraciones en la integración del resultado de la normalización

- **Dependencias Transitivas:** hay situaciones en que al integrar dos relaciones (o tablas) en 3FN, se produce una dependencia transitiva, por ejemplo:

TRABAJADOR (RUT, nombre, código\_cargo)

TRABAJADOR (RUT, nombre, nombre\_cargo)

La integración de ambas relaciones o tablas resulta en:

TRABAJADOR (RUT, nombre, código\_cargo, nombre\_cargo)

donde:

RUT → código\_cargo

RUT → nombre\_cargo

código\_cargo → nombre\_cargo

situación que debe ser corregida, quedando:

TRABAJADOR (RUT, nombre, *código\_cargo*)

CARGO (código-cargo, nombre\_cargo)

# ENFOQUE BOTTOM-UP – ETAPA 2

## DISEÑO LÓGICO DE BDR

### Consideraciones en la integración del resultado de la normalización

- **Generalización:** la integración de dos relaciones (o tablas) con la misma clave primaria debe ser analizada pues puede corresponder a tipos de entidades diferentes.

PACIENTE (#paciente, nombre, ..., fecha\_atención, diagnósticoConsulta)

PACIENTE (#paciente, nombre, ..., fecha\_atención, #pieza)

En principio, deberían conformar una misma tabla:

PACIENTE (#paciente, nombre, ..., fecha\_atención, diagnósticoConsulta, #pieza)

...pero se pueden confundir los pacientes ambulatorios de los que se hospitalizan...



# ENFOQUE BOTTOM-UP – ETAPA 2

## DISEÑO LÓGICO DE BDR

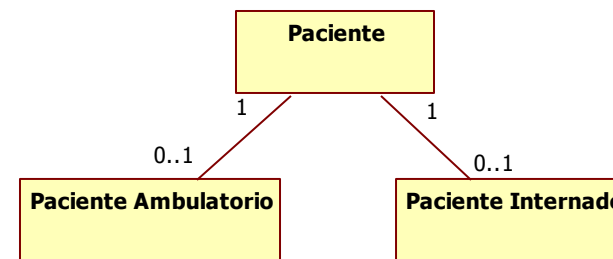
...por lo anterior, hay que chequear si alguna de las siguientes soluciones es más conveniente:

paciente(#paciente, nombre, ..., fecha\_atención, diagnósticoConsulta, #pieza, tipoPaciente)

paciente(#paciente, nombre, ..., fecha\_atención)

pacienteAmbulatorio(#paciente, diagnósticoConsulta)

pacienteInternado(#paciente, #pieza)



pacienteAmbulatorio(#paciente, nombre, ..., fecha\_atención, diagnósticoConsulta)

pacienteInternado(#paciente, nombre, ..., fecha\_atención, #pieza)

## 3.2.2 ENFOQUE METODOLÓGICO TOP-DOWN



**Departamento de Informática**  
Universidad Técnica Federico Santa María



# ENFOQUE TOP-DOWN

## DISEÑO LÓGICO DE BDR

- Transformación de un Modelo Conceptual a uno Relacional mediante la aplicación de reglas bien definidas.
- Se definen 8 pasos a llevar a cabo para lograr la transformación.
- Para efectos del curso, se hará uso de un modelo conceptual expresado en un diagrama de clases, para explicar el método ...

# ENFOQUE TOP-DOWN

## DISEÑO LÓGICO DE BDR

### PASOS

1. Tipos de Entidades “Fuertes”
2. Tipos de Entidades “Débiles”
3. Asociaciones 1:1
4. Asociaciones 1:N
5. Asociaciones M:N
6. Asociaciones n-arias ( $n \geq 3$ )
7. Herencia
8. Categorización (Interfaces o Herencia Selectiva)

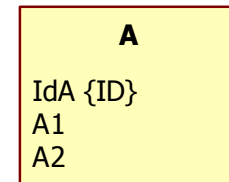
# ENFOQUE TOP-DOWN – PASO 1

## DISEÑO LÓGICO DE BDR

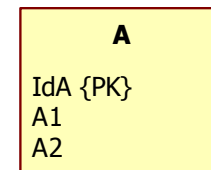
### Paso 1: Tipos de Entidades “Fuertes”

- Por cada tipo de entidad fuerte del modelo conceptual, crear una relación o tabla que incluya todos los atributos simples de ella. Incluir sólo los atributos atómicos de un atributo compuesto.
- Escoger uno de los atributos claves del tipo de entidad como clave primaria en la relación o tabla correspondiente. Si la clave escogida es compuesta, el conjunto de atributos simples que la conforman serán parte de la clave primaria de la relación o tabla.

### Modelo Conceptual



### Modelo Relacional



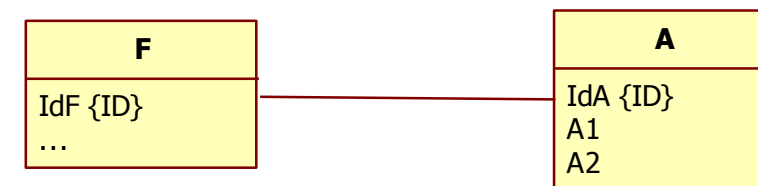
# ENFOQUE TOP-DOWN – PASO 2

## DISEÑO LÓGICO DE BDR

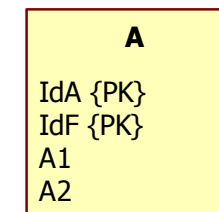
### Paso 2: Tipos de Entidades “Débiles”

- Por cada tipo de entidad débil del modelo conceptual, dependiente de una entidad fuerte, crear una relación o tabla que incluya los atributos simples, los atributos simples de los posibles atributos compuestos que existan, y los atributos que formen parte de la clave primaria de la entidad fuerte.
- Dejar como clave primaria de la relación o tabla, a la concatenación de la clave primaria de la entidad fuerte con el identificador del tipo de entidad débil.

### Modelo Conceptual



### Modelo Relacional



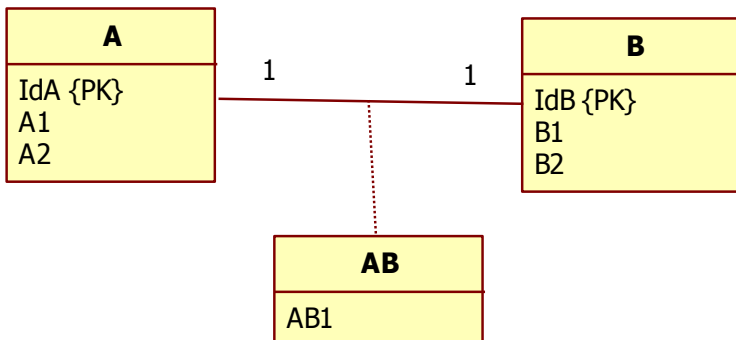
# ENFOQUE TOP-DOWN – PASO 3

## DISEÑO LÓGICO DE BDR

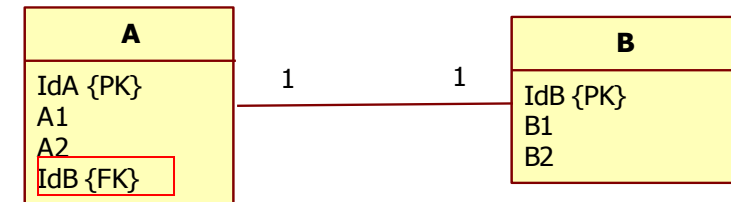
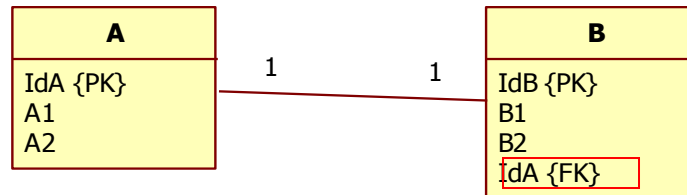
### Paso 3: Asociaciones 1:1

- Por cada asociación binaria 1:1, identificar las relaciones o tablas del modelo relacional correspondientes; escoger una de las relaciones o tablas e incluir su clave primaria como clave foránea en la otra relación.

#### Modelo Conceptual



#### Modelo Relacional: dos alternativas...



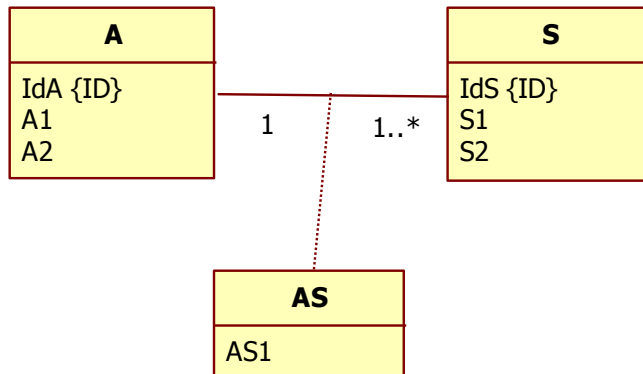
# ENFOQUE TOP-DOWN – PASO 4

## DISEÑO LÓGICO DE BDR

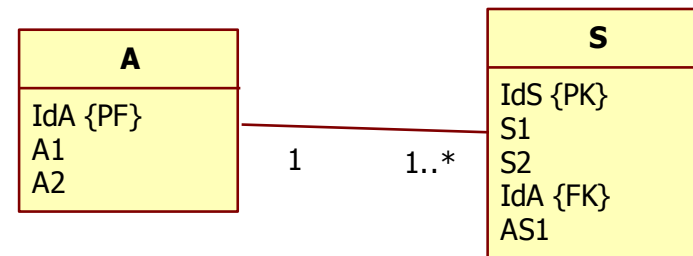
### Paso 4: Asociaciones 1:N

- Por cada asociación binaria 1:N (no débil), identificar la relación o tabla S que está en el lado de la cardinalidad “muchos”, e incluir en ésta como clave foránea, la clave primaria de la entidad con cardinalidad 1 (bajo esta asociación).
- Incluir cualquier atributo simple de la asociación como atributo de S.

Modelo Conceptual



Modelo Relacional





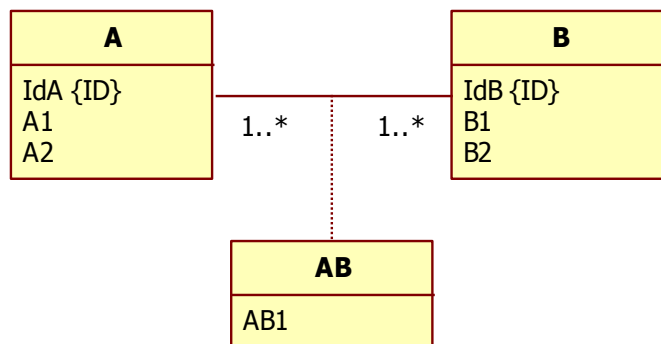
# ENFOQUE TOP-DOWN – PASO 5

## DISEÑO LÓGICO DE BDR

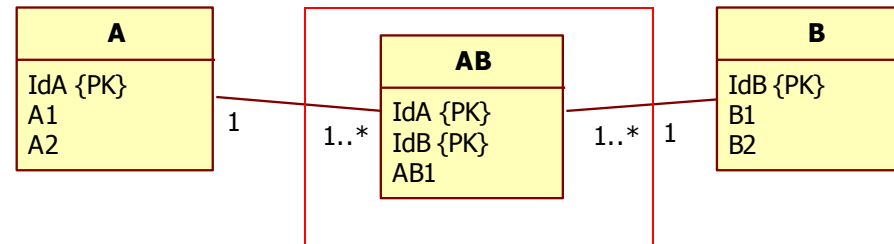
### Paso 5: Asociaciones M:N

- Por cada asociación binaria M:N crear una nueva relación o tabla que la represente. Incluir como claves foráneas a las claves primarias de las relaciones o tablas que participan en la asociación.
- Incluir, también, cualquier atributo simple de la asociación.

#### Modelo Conceptual



#### Modelo Relacional



# ENFOQUE TOP-DOWN – PASO 6

## DISEÑO LÓGICO DE BDR

### Paso 6: Asociaciones n-arias ( $n \geq 3$ )

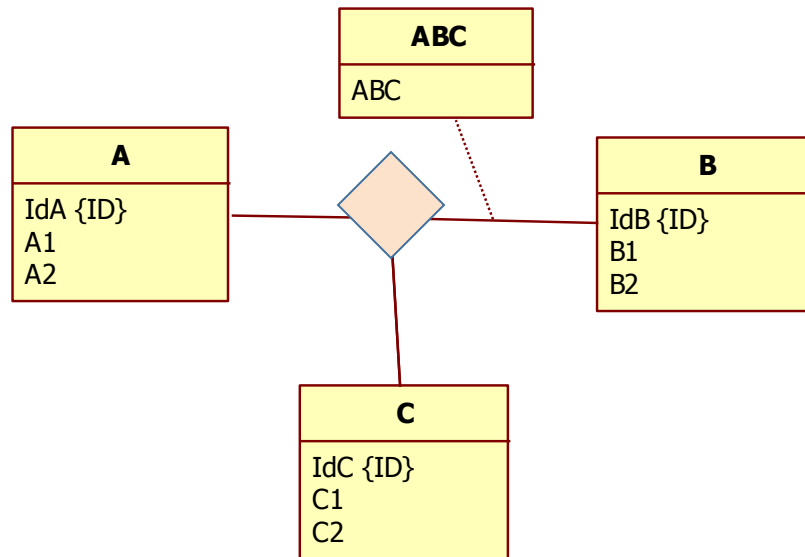
- Por cada asociación de grado 3 o más, crear una nueva relación o tabla para representarla.
- Incluir como claves foráneas a las claves primarias de las relaciones o tablas que representan a los tipos de entidades que participan en dicha asociación; normalmente, la concatenación de estas claves es la clave primaria de la nueva relación o tabla.

# ENFOQUE TOP-DOWN – PASO 6

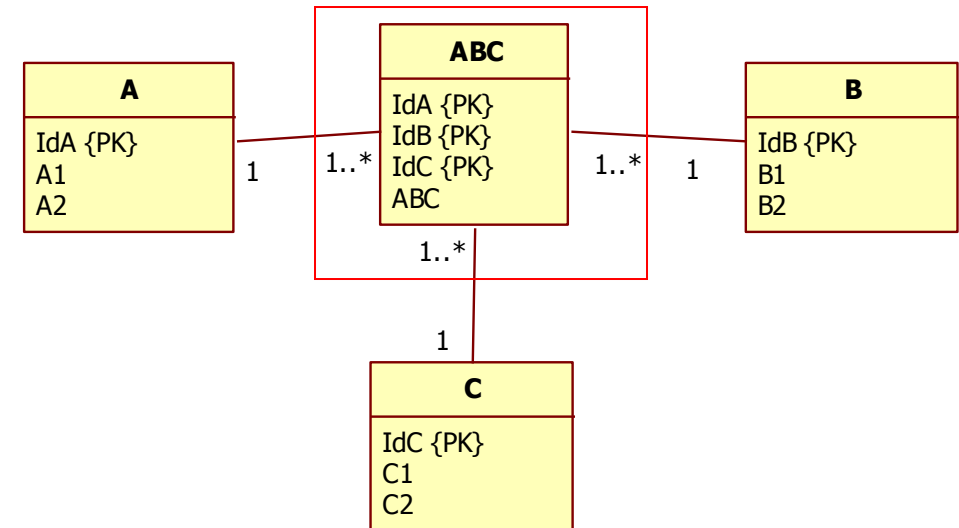
## DISEÑO LÓGICO DE BDR

### Paso 6: Asociaciones n-arias ( $n \geq 3$ )

Modelo Conceptual



Modelo Relacional

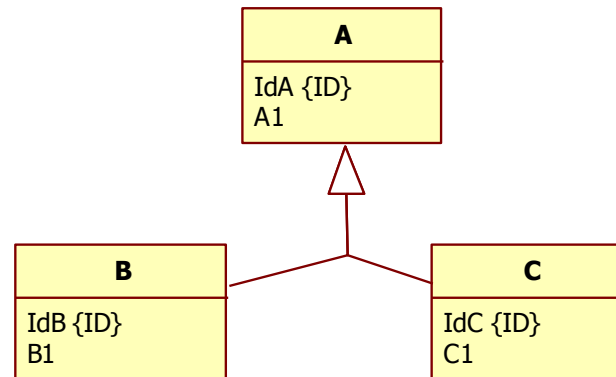


# ENFOQUE TOP-DOWN – PASO 7

## DISEÑO LÓGICO DE BDR

### Paso 7: Herencia

- Hay varias posibilidades (ya vistas en la integración de vistas normalizadas).
- Considerar la siguiente jerarquía:

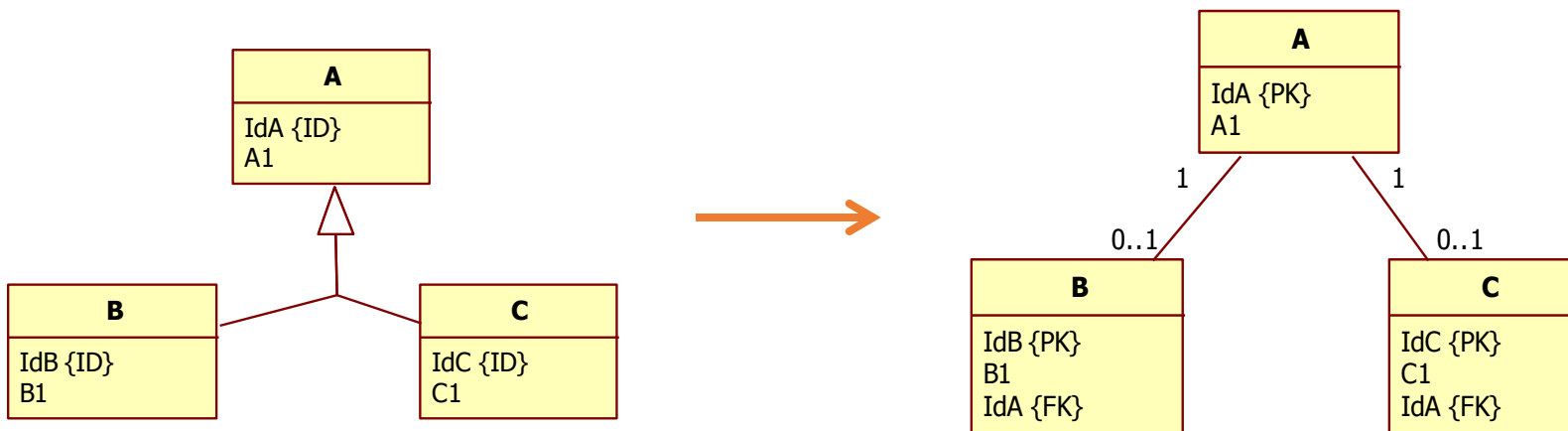


# ENFOQUE TOP-DOWN – PASO 7

## DISEÑO LÓGICO DE BDR

### Paso 7: Continuación Herencia

- Alternativa 1: crear una relación o tabla para la superclase con todos los atributos identificados en el modelo conceptual. Crear una relación o tabla por cada subclase, con todos los atributos propios de la subclase más el identificador de la superclase, quedando éste como la clave primaria de la relación.

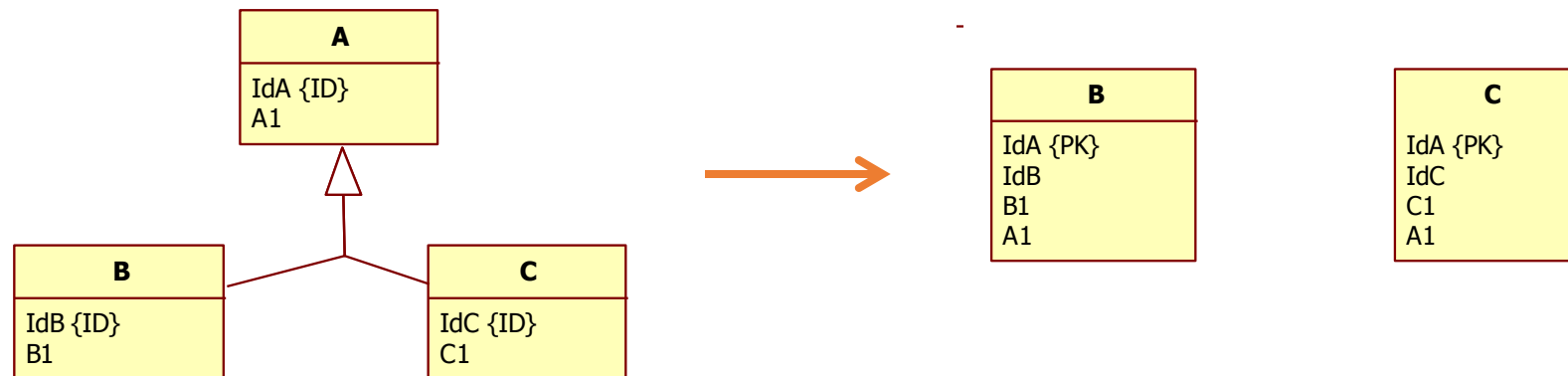


# ENFOQUE TOP-DOWN – PASO 7

## DISEÑO LÓGICO DE BDR

### Paso 7: Continuación Herencia

- Alternativa 2: crear una relación o tabla por cada subclase, incluyendo en cada una de ellas los atributos de la superclase y los atributos propios de la subclase correspondiente, dejando al identificador de la superclase como la clave primaria de la relación.

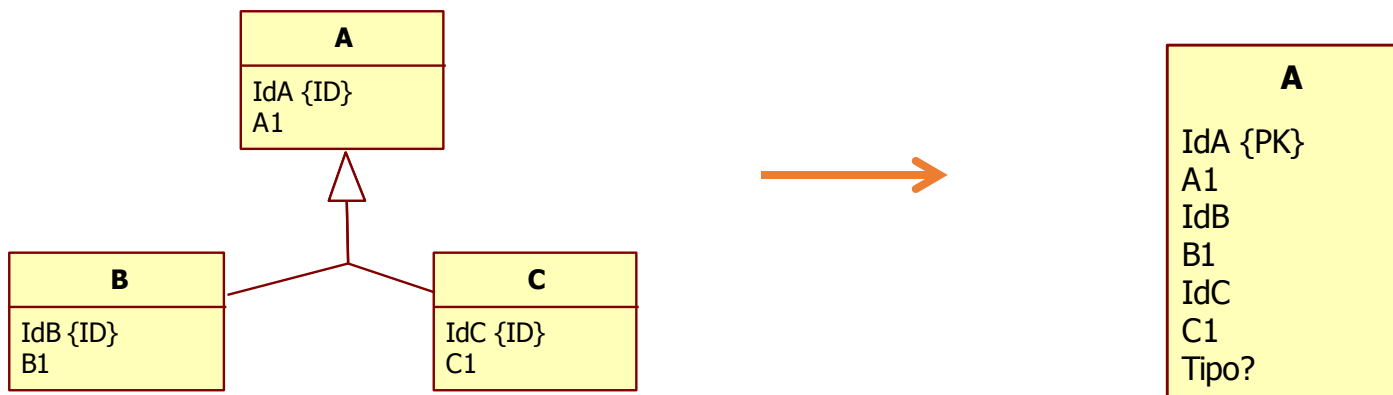


# ENFOQUE TOP-DOWN – PASO 7

## DISEÑO LÓGICO DE BDR

### Paso 7: Continuación Herencia

- Alternativa 3: crear una única relación o tabla que contenga todos los atributos de la superclase y todos los atributos pertenecientes a cada una de las subclases, más un atributo adicional que permita determinar a qué subclase pertenece cierta entidad. La clave primaria es el identificador de la superclase.

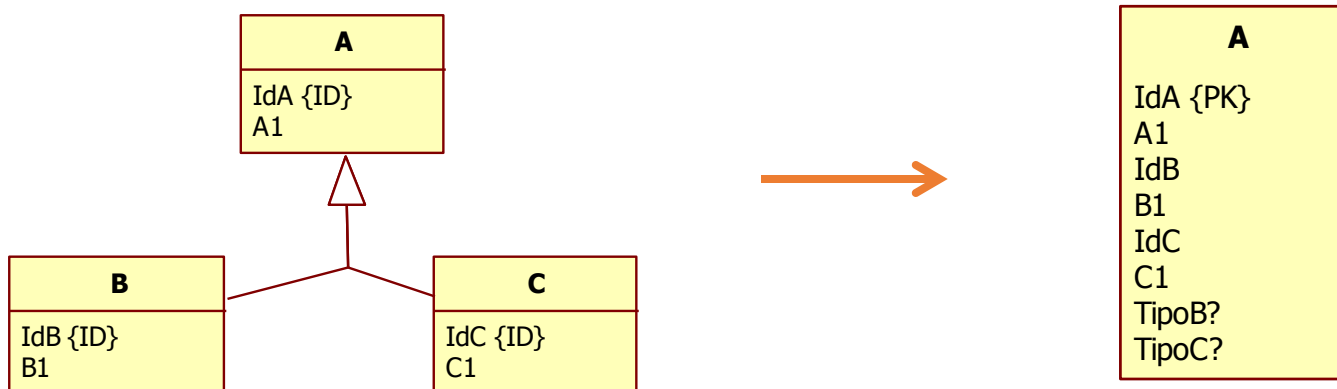


# ENFOQUE TOP-DOWN – PASO 7

## DISEÑO LÓGICO DE BDR

### Paso 7: Continuación Herencia

- Alternativa 4: crear una única relación o tabla que contenga todos los atributos de la superclase y todos los atributos pertenecientes a cada una de las subclases, más un conjunto de atributos adicionales (de tipo booleano) que permitan determinar a qué subclases pertenece cierta entidad.



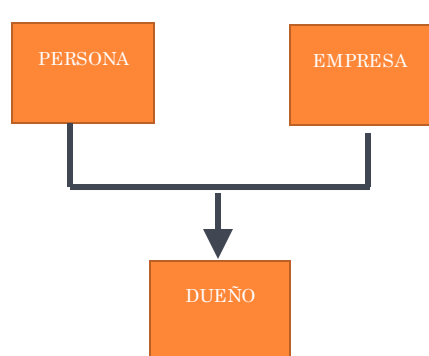


# ENFOQUE TOP-DOWN – PASO 8

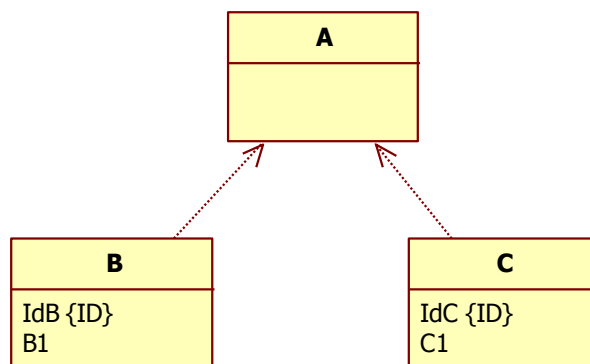
## DISEÑO LÓGICO DE BDR

### Paso 8: Categorización (Interfaces o Herencia Selectiva)

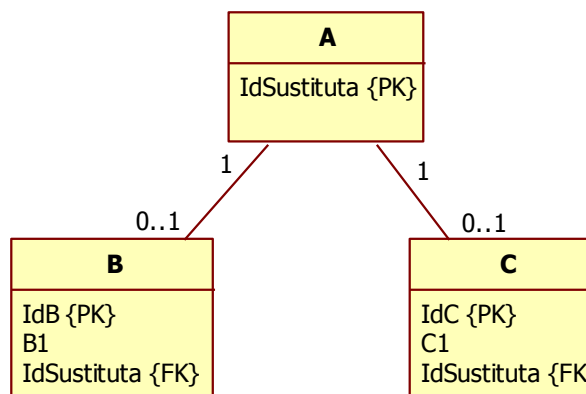
- Para el caso de que las superclases tengan diferentes identificadores, se debe crear una clave primaria especial, que se denomina **clave sustituta**, para la relación o tabla que representa la categoría; a ésta se le incluyen los atributos definidos en la categoría en el modelo conceptual.
- Finalmente, a cada relación o tabla asociada con una superclase de la categoría, se le agrega como clave foránea la clave sustituta.



### Modelo Conceptual



### Modelo Relacional



- Para el caso de que las superclases tengan el mismo identificador, no hay necesidad de la clave sustituta, y se usa algún esquema similar a los usados en el paso 7.

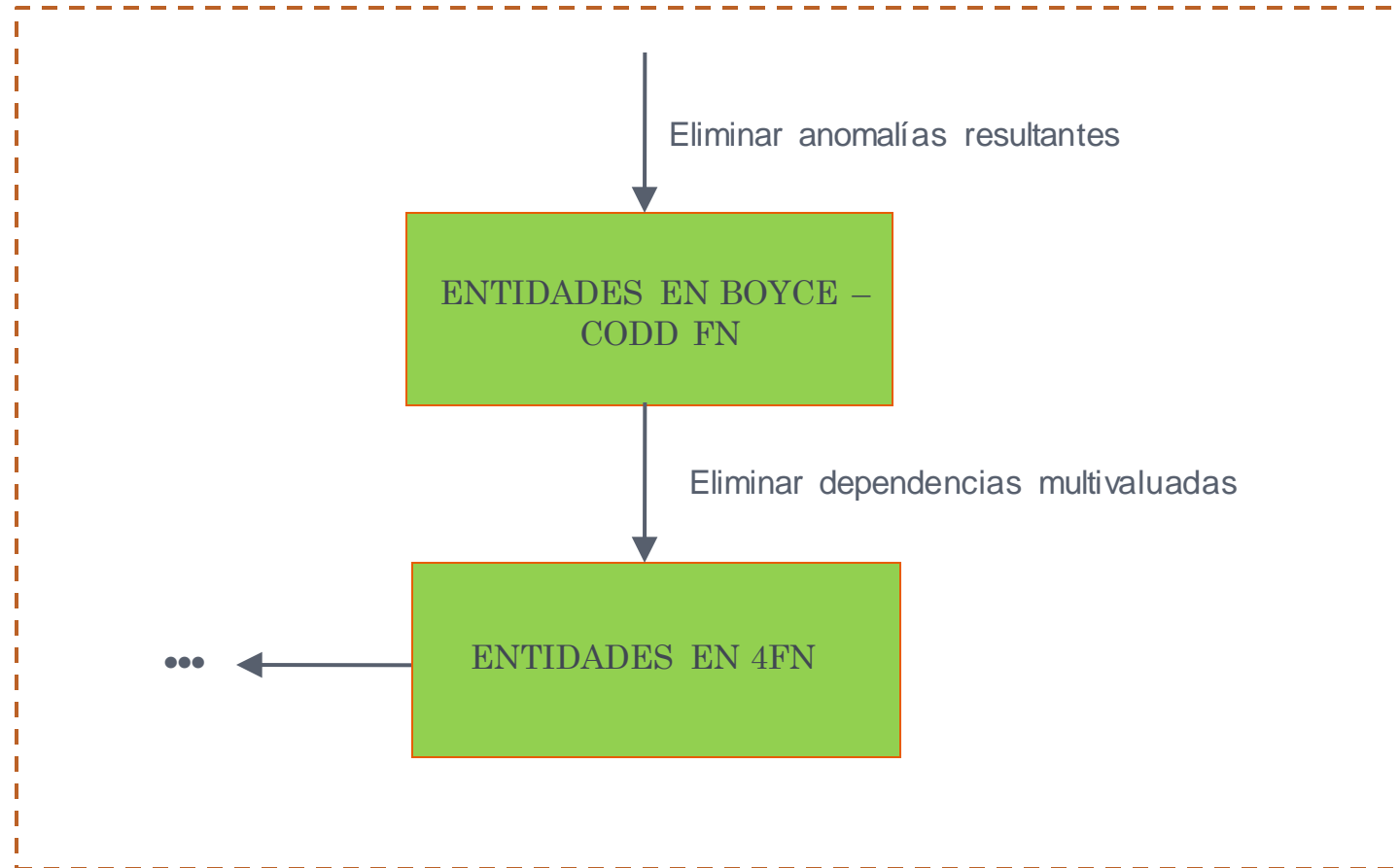
## 3.3 OTRAS CONSIDERACIONES DISEÑO LÓGICO BDR



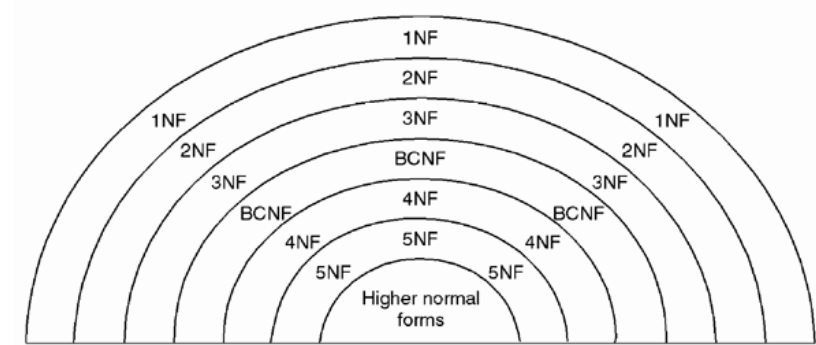
**Departamento de Informática**  
Universidad Técnica Federico Santa María



# OTRAS FORMAS NORMALES



# FORMA NORMAL BOYCE-CODD



- Cuando una entidad tiene más de una clave candidata, es posible que aún existan anomalías en 3FN (“overlapping de claves”).
- Es una situación rara, pero puede ocurrir.
- Boyce y Codd propone para eliminar esa anomalía: “Una Relación R está en BC FN, si y sólo si cada **determinante** es clave candidata”.
- Recordar definición de dependencia funcional y su escritura:

$A \rightarrow B$

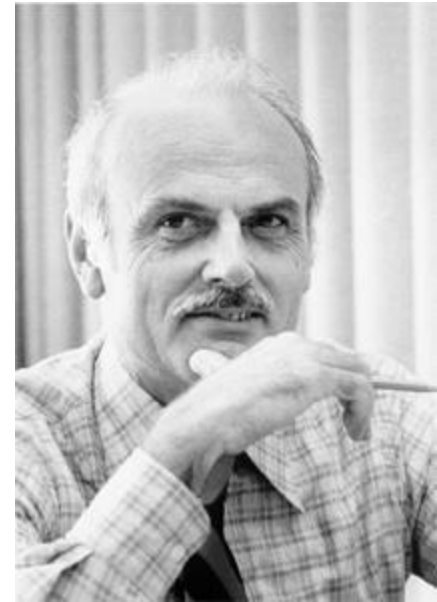
y se lee: “A determina a B” o “B es funcionalmente dependiente de A” siendo A un **determinante**.

# FORMA NORMAL BOYCE-CODD

Ejemplo B/C FN:

1. Cada alumno puede tener varias materias de interés.
2. Por cada materia, un alumno tiene sólo un tutor.
3. Cada materia tiene varios tutores.
4. Cada tutor orienta sólo en una materia.
5. Cada tutor orienta a varios estudiantes en una materia.

TUTORÍAS (Nro-Alumno, Materia, Tutor)



# FORMA NORMAL BOYCE-CODD

## CLAVES CANDIDATAS A PK:

TUTORÍAS (Nro-Alumno, Materia, Tutor)

TUTORÍAS (Nro-Alumno, Materia, Tutor)

Nro-Alumno	Materia	Tutor
123	Física	Einstein
123	Música	Mozart
456	Biología	Darwin
789	Física	Bohr
999	Física	Einstein

## DEPENDENCIAS FUNCIONALES:

Nro-Alumno, Tutor  $\rightarrow$  Materia

Nro-Alumno, Materia  $\rightarrow$  Tutor

Tutor  $\rightarrow$  Materia

## DETERMINANTES:

Nro-Alumno, Tutor (Si es clave candidata a PK)

Nro-Alumno, Materia (Si es clave candidata a PK)

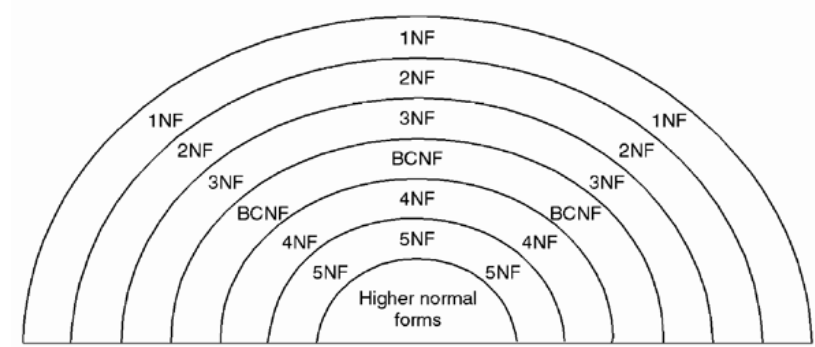
Tutor (No es clave candidata a PK)

a BCFN

<u>Nro-Alumno</u>	<u>Tutor</u>
123	Einstein
123	Mozart
456	Darwin
789	Bohr
999	Einstein

<u>Tutor</u>	Materia
Einstein	Física
Mozart	Música
Darwin	Biología
Bohr	Física

# 4º FORMA NORMAL



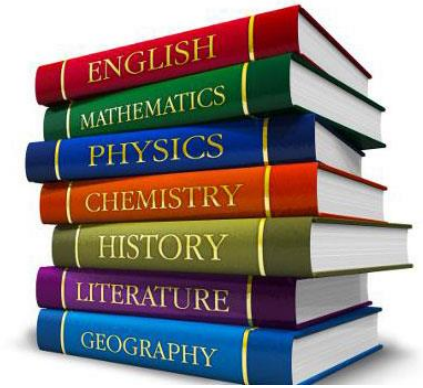
- Cuando una entidad está en B/C FN, puede tener anomalías con respecto a las dependencias multivaluadas (DM).
- Una DM existe cuando hay al menos tres atributos (A, B, C) en una relación R, y para cada valor de A existe un conjunto de valores de B y de C definidos, pero independientes entre sí.

# 4º FORMA NORMAL

Ejemplo 4 FN:

1. Cada asignatura puede tener varios profesores.
2. Cada asignatura utiliza varios textos.
3. El texto utilizado para una cierta asignatura es independiente del profesor.

ASIGNATURAS (Nom-Asignatura, Profesor, Libro)





# 4º FORMA NORMAL

Vista ASIGNATURA

Nom-Asignatura	Profesor	Libro
Administración	White Green Black	Drucker Peters
Finanzas	Gray	Weston Gilford



Tabla ASIGNATURA en BC FN

<u>Nom-Asignatura</u>	<u>Profesor</u>	<u>Libro</u>
Administración	White	Drucker
Administración	White	Peters
Administración	Green	Drucker
Administración	Green	Peters
Administración	Black	Drucker
Administración	Black	Peters
Finanzas	Gray	Weston
Finanzas	Gray	Gilford



ASIGNATURA-PROFESOR

<u>Nom-Asignatura</u>	<u>Profesor</u>
Administración	White
Administración	Green
Administración	Black
Finanzas	Gray

ASIGNATURA-LIBRO

<u>Nom-Asignatura</u>	<u>Libro</u>
Administración	Drucker
Administración	Peters
Finanzas	Weston
Finanzas	Gilford

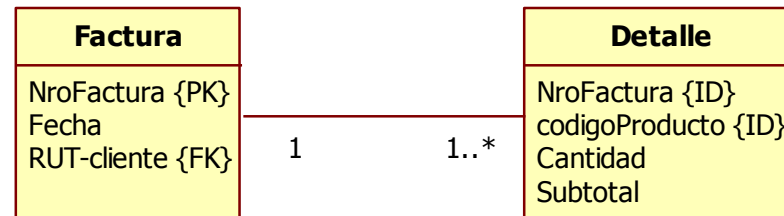
# DESNORMALIZACIÓN

- Conjunto de técnicas destinadas a mejorar los tiempos de respuesta, agregando redundancia a un modelo relacional.
- Normalmente, se tiende a evitar uno o más operaciones JOINS entre los archivos requeridos para responder alguna consulta.
- Algunos ejemplos:

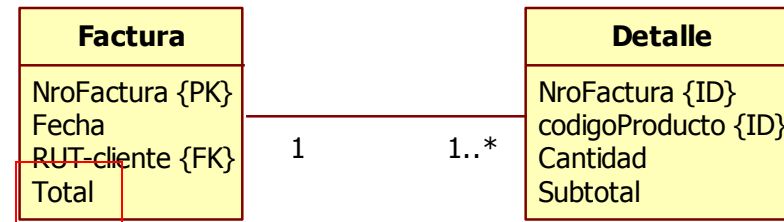
# DESNORMALIZACIÓN

- a) **Almacenando valores derivados:** agregar una columna extra para almacenar datos derivados a partir de otros ya presentes en el modelo de datos.

Modelo Relacional Normalizado



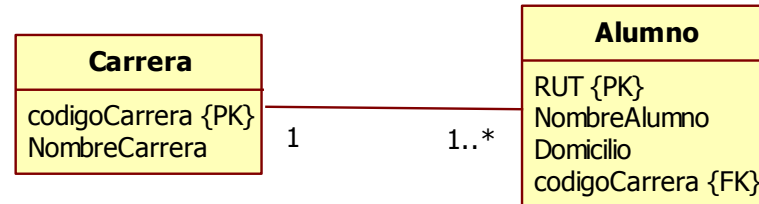
Modelo Relacional Desnormalizado



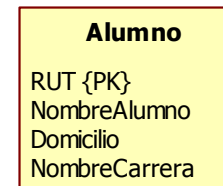
# DESNORMALIZACIÓN

- b) Tablas Prejoinizadas:** agregar una columna no clave a la tabla con la clave foránea (volver a 2FN).

Modelo Relacional Normalizado



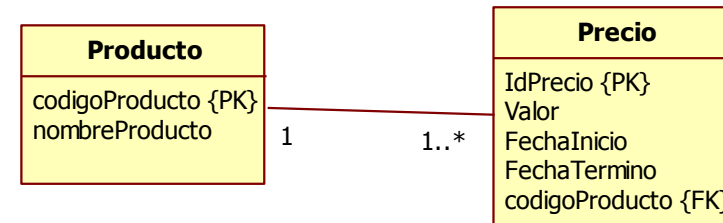
Modelo Relacional Desnormalizado



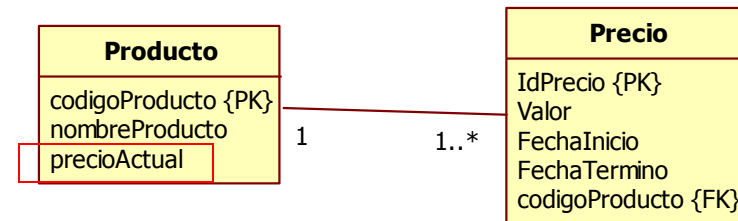
# DESNORMALIZACIÓN

- c) **Repitiendo el detalle en el maestro:** dada una asociación 1:N, agregar una columna en la tabla del extremo “uno” para registrar la ocurrencia más reciente del extremo “muchos” (valor actual v/s valor histórico).

Modelo Relacional Normalizado



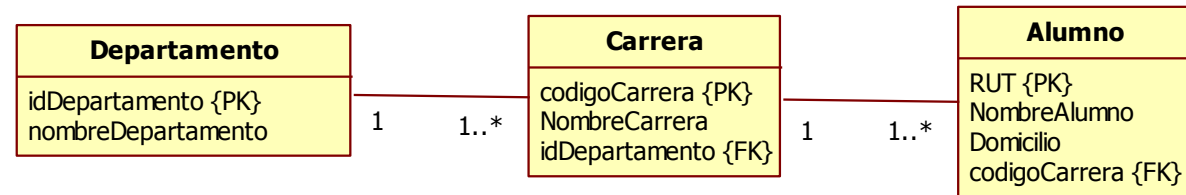
Modelo Relacional Desnormalizado



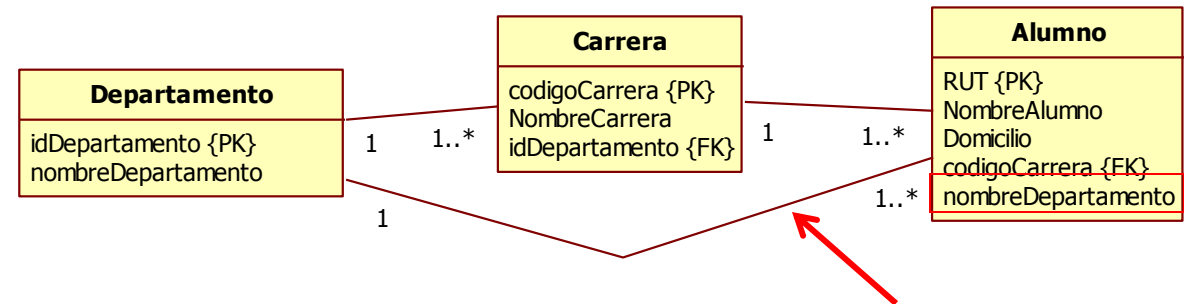
# DESNORMALIZACIÓN

- d) **Claves en “corto circuito”**: crear una nueva clave foránea desde el nivel de detalle más bajo al nivel más general.

Modelo Relacional Normalizado

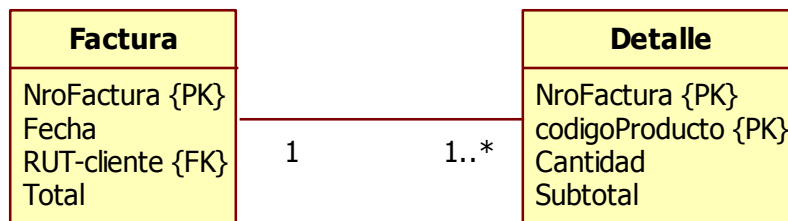


Modelo Relacional Desnormalizado



# COSTOS DESNORMALIZACIÓN

- Obliga a incorporar código adicional, para ayudar a mantener la integridad del diseño.
- Por lo general, se requiere un método para cada una de las operaciones DML típicas.
- Ejemplo: Almacenando valores derivados



- Si se agrega un nuevo Detalle, el Total de Factura debe aumentar en el valor del Subtotal correspondiente.
- Si se modifica el subTotal de un Detalle, en la misma cantidad debe variar el valor del Total.
- Si se elimina un Detalle, Total debe disminuir en la cantidad presente en el subTotal que desaparece.