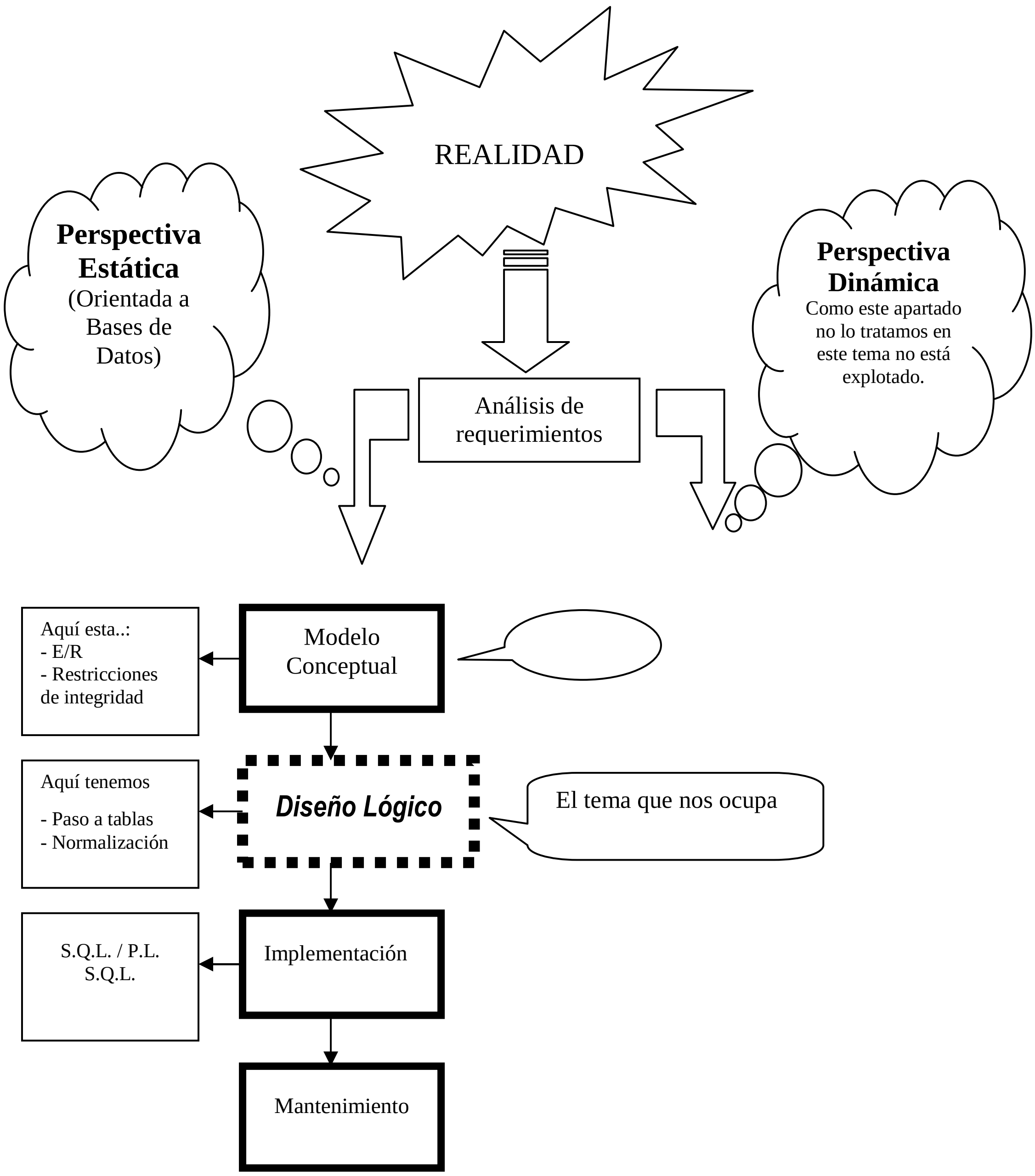
Disseny Lògic

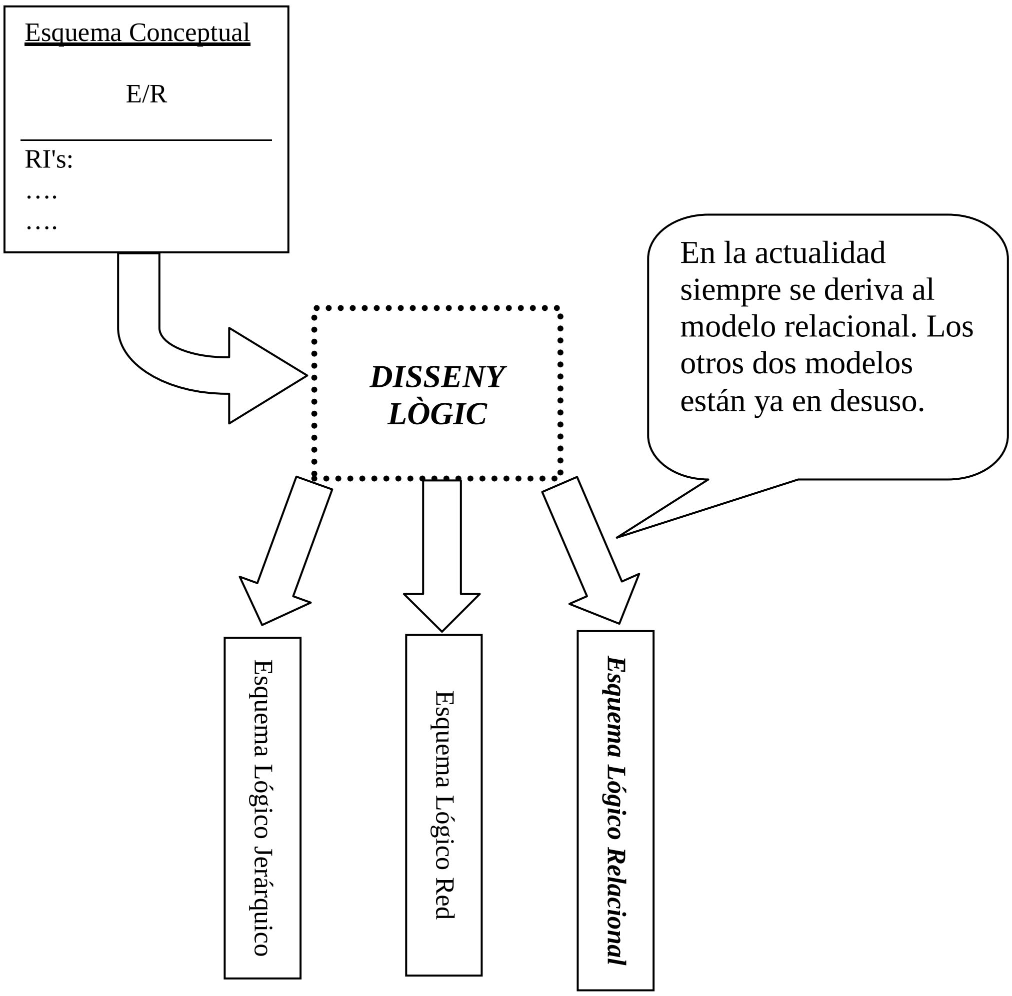
A. MOLL

1. INTRODUCCIÓN

El presente tema resuelve la incognita de qué hacer tras efectuar el análisis conceptual y disponer, bajo la perspectiva estática, del E/R con sus eventuales restricciones de integridad semánticas. Gráficamente:



Que des d'un altre angle tenim



2. PAS A TAULES

El proceso de obtención de un **Esquema Lógico Relacional** que presente adecuadamente toda la semántica del esquema conceptual (E/R + cjto de Restricciones de Integridad añadidas) se divide en dos fases:

1. En una **primera fase** se transforma el diagrama E/R a un Esquema Lógico Relacional aplicando un conjunto de reglas que dependiendo del objeto a transformar indica qué conjunto de tablas lo representan adecuadamente. También en esta fase tienen que ser tratadas las posibles restricciones del Esquema Conceptual traduciéndolas a expresiones equivalentes del Cálculo Relacional de Tuplas, de Dominios , Álgebra Relacional o simplemente especificarlas en Lenguaje Natural.

2. La **segunda fase** consiste en examinar cada una de las tablas obtenidas y comprobar que se encuentran adecuadamente "***normalizadas***". Lo aplazamos hasta el epígrafe 3.

Suponemos que el Sistema de Gestión de Base de Datos soporta:

- Clave Primaria (**CP**)
- Clave Alternativa (**CAlt**)
- Clave Ajena (**CAj**)
- Restricción de Valor no Nulo (**VNN**)

Además partimos de las siguientes definiciones:

Clave Candidata = UNICIDAD + MINIMALIDAD

Clave Principal o Primaria = "Una de las candidatas"

Clave Alternativa = Clave candidata no ppal

Obsérvese que con estas definiciones estamos permitiendo la introducción de valores nulos en las Claves Principales !!. El asunto se resuelve incorporando "*ad hoc*" al propio modelo de datos relacional la siguiente restricción ímplicita:

Integridad de la Clave Primaria = "No puede haber nulos en la CP"

Las claves alternativas en cambio sí admiten nulos. No obstante, conviene advertir que algunos autores realizan el diseño lógico con una definición menos "flexible " de clave alternativa.

Concretamente parten de una def de *Clave Candidata* = *Unicidad+Minimalidad+VNN*, lo cual conlleva algunos cambios en las transformaciones a efectuar del E/R, puesto que, en este contexto, las *alternativas* no admiten nulos.

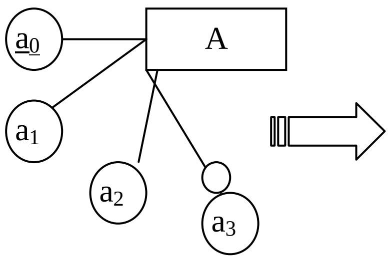
En definitiva no perdamos de vista el importante hecho de que para llevar a cabo el "paso a tablas" precisamos:

- Conocer, por un lado, las capacidades soportadas por el SGBD
- Y por otro, una def. de clave candidata que repercutirá, en resumidas cuentas, en saber si las claves alternativas admiten o no valores nulos.

Comencemos ya con los apuntes propios del paso a tablas.

2.1. ENTIDADES Y ATRIBUTOS

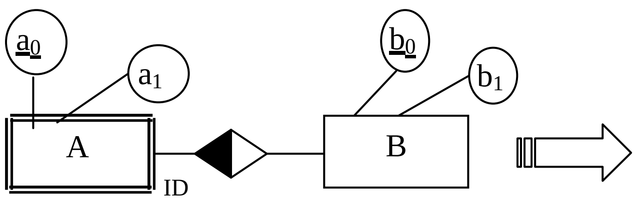
⇨ **ENTIDAD FUERTE**



- A = $\underline{a_0} + a_1 + a_2 + a_3$

  VNN { a_3 }

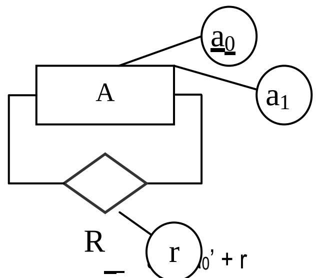
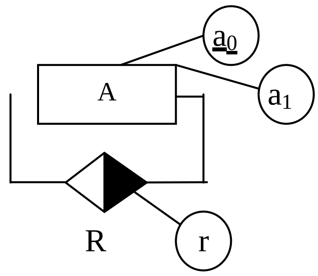
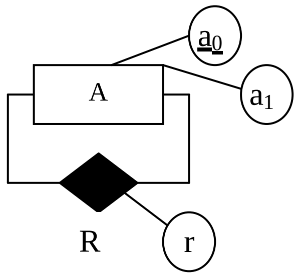
⇨ **ENTIDAD DÉBIL**



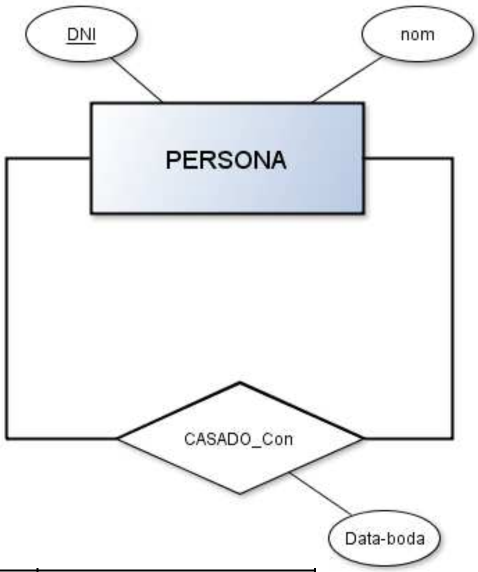
A = $\underline{a_0 + \mathbf{b_0}} + a_1$ CAj.{ $b_0 \rightarrow$ B}
B = $\underline{b_0} + b_1$

A modo de ejercicico recordatorio: ¿Cómo quedaría la transformación de los atributos compuestos?. ¿Y la de los multivaluados?.

2.2. RELACIONES

UNARIAS SIN R.E		
Uno a Uno (1:1)	**Uno a muchos (1:M)**	**Muchos a Muchos (M:M)**
 $_0$' + r CAlt.{ a_0'} ; CAj.{ a_0' ➔ A} **RI:** *nulo(a'$_0$)➔nulo(r)* *I d'altres RI's prou interessants que podem provocar l'aparició d'una taula per a R*	 A = $\underline{a_0}$ + a_1 +a_0' + r CAj.{ a_0' ➔ A} **RI:** *nulo(a'$_0$)➔nulo(r)*	 A = $\underline{a_0}$ + a_1 R = $\underline{a_0 + a_0'}$+ r CAj.{ a_0 ➔ A a_0' ➔ A }

2.2.1. Anàlisi concret de la unària 1-1



PERSONA= DNI+Nom+DniConyuge+ DataCasament

RI: Null (DniConyuge) → Null (DataCasament)

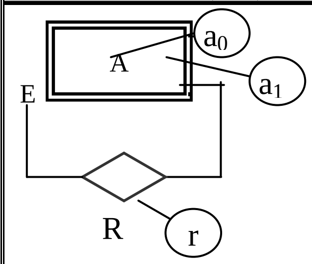
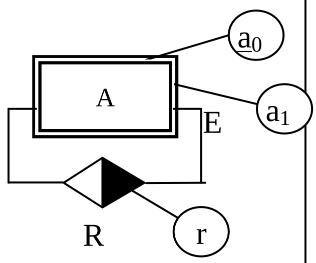
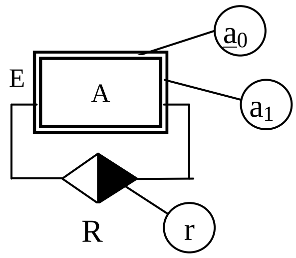
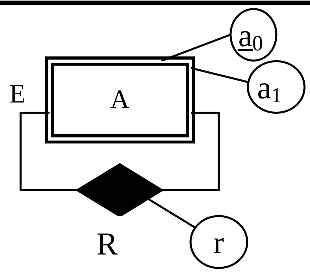
Hi pot haver solters. En ppi tot quasi ben captat però….

DNI	……	…….	Dni_Conyuge	Data_boda
Pepe	…….	…….	María	2005
……..	……..	……..	……..	……..
María			Antonio	2006

Exercici

Continueu amb l'anàlisi d'aquest problema sense descartar la possibilitat d'habilitar una taula nova per a la pròpia relació.

UNARIAS CON R.E.

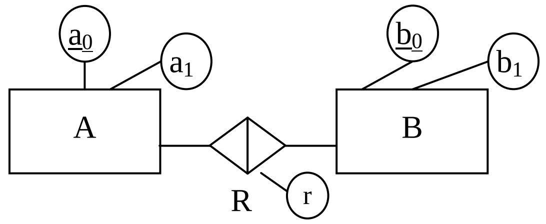
<table>
<tr><th>Uno a Uno (1:1)</th><th colspan="2">Uno a muchos (1:M)</th><th>Muchos a Muchos (M:M)</th></tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• A = $\underline{a_0} + a_1 + a_0' + r$
CAlt.{a_0'}; VNN{a_0'}
CAj.{ $a_0' \rightarrow A$}
Tenim alternatives interessants a esta solució estàndart. Recordem la relació casat_amb</td>
<td>• A = $\underline{a_0} + a_1 + a_0' + r$

VNN{a_0'}; CAj{$a_0' \rightarrow A$}
R.I:

$\exists!\ t\ (A(t) \rightarrow t.a_0 = t.a'_0)$</td>
<td>• A = $\underline{a_0} + a_1 + a_0' + r$
CAj.{ $a_0' \rightarrow A$}
R.I.: $A[a_0] \subset A[a_0']$
Recordemos que algunos autores consideran que este caso (y otros) no se pueden/deben transformar.</td>
<td>• A = $\underline{a_0} + a_1$
• R = $\underline{a_0 + a_0'} + r$
CAj.{ $a_0 \rightarrow A$
$a_0' \rightarrow A$ }
R.I.:
$A[a_0] \subset R[a_0']$</td>
</tr>
</table>

⇨ BINARIAS SIN R.E.

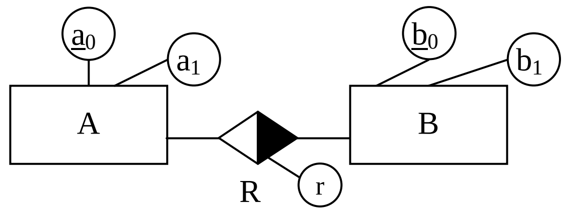
Uno a Uno (1:1)



- $A = a_0 + a_1$
- $B = b_0 + b_1 + a_0 + r$
  CAlt.{ a_0}
  CAj.{ a_0 ➔ A}

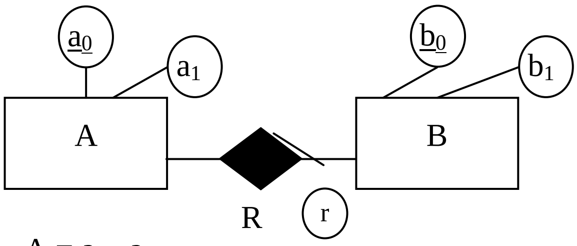
Nulo(a0) --> Nulo(r)

Uno a muchos (1:M)



- $A = \underline{a_0} + a_1$
- $B = \underline{b_0} + b_1 + a_0 + r$
  CAj.{ a_0 ➔ A} RI: *nulo(a_0)→nulo(r)*

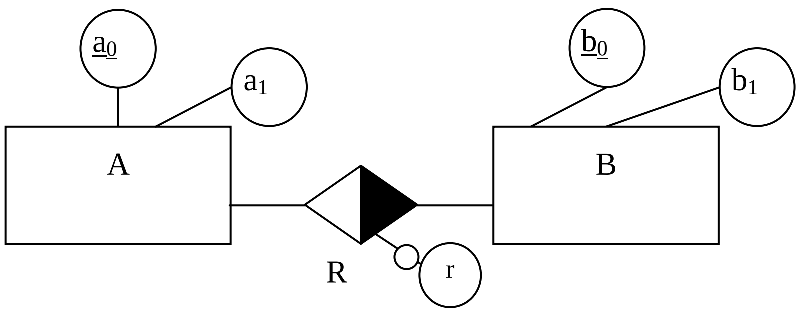
Muchos a Muchos (M:M)



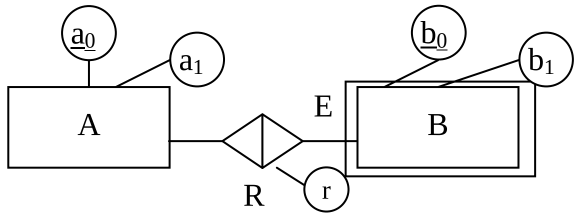
- $A = \underline{a_0} + a_1$
- $B = \underline{b_0} + b_1$

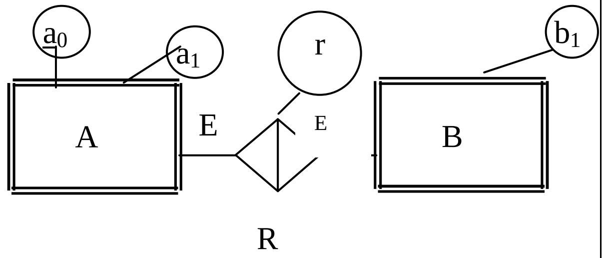
$R = \underline{a_0} + \underline{b_0} + r$ CAj.{ a_0 ➔ A; b_0 ➔ B}

Exercici: Transforma la següent binaria

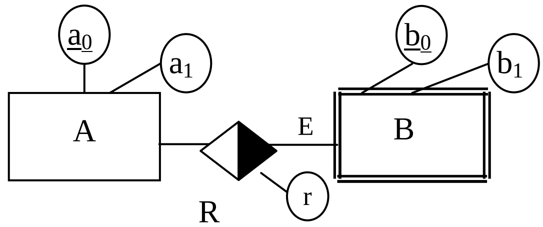


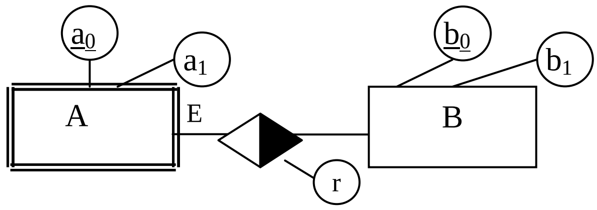
⇨ BINARIAS CON R.E.

<table>
<tr><th colspan="2">Uno a Uno (1:1)</th></tr>
<tr><td>


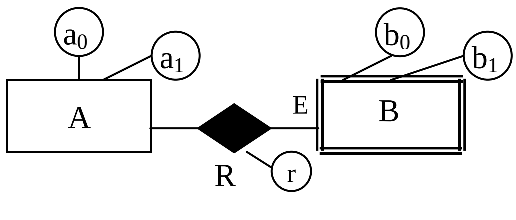
• A = $\underline{a_0} + a_1$
• B = $\underline{b_0} + b_1 + a_0 + r$
CAlt.{a_0} VNN(a_0) CAj.{$a_0 \rightarrow$ A}
</td><td>


• R ≅AB = $\underline{a_0} + a_1 + b_0 + b_1 + r$
CAlt.{ b_0 } VNN(b_0 })
L'únic cas en que tinc menys taules que entitats
</td></tr>
</table>

<table>
<tr><th colspan="2">Uno a muchos (1:M)</th></tr>
<tr><td>


A = $a_0 + a_1$
• B = $b_0 + b_1 + a_0 + r$
VNN{a_0} CAj.{$a_0 \rightarrow$ A}
</td><td>


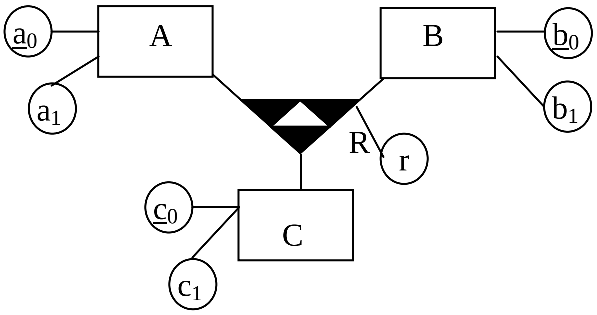
• A = $\underline{a_0} + a_1$
• B = $\underline{b_0} + b_1 + a_0 + r$
CAj.{$a_0 \rightarrow$ A}
R.I's.:
A[a_0] ⊂ B[a_0]; nulo(a_0)→nulo(r)
</td></tr>
</table>

<table>
<tr><th colspan="2">Muchos a Muchos (M:M)</th></tr>
<tr><td colspan="2">


</td></tr>
<tr><td>
• A = $a_0 + a_1$
• B = $b_0 + b_1$
• R = $\underline{a_0 + b_0} + r$
CAj.{$a_0 \rightarrow$ A
$b_0 \rightarrow$ B }
</td><td>
R.I.:
B[b_0] ⊂ R[b_0]
</td></tr>
</table>

⇨ TERNARIAS SIN R.E.

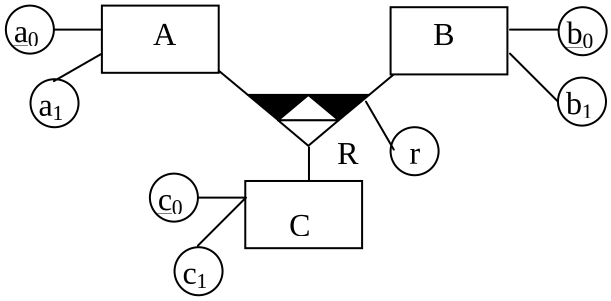
M:M:M



- $A = a_0 + a_1$
- $B = b_0 + b_1$
- $C = c_0 + c_1$
- $R = \underline{a_0 + b_0 + c_0} + r$

CAj.{$a_0 \rightarrow A$
$b_0 \rightarrow B$
$c_0 \rightarrow C$ }

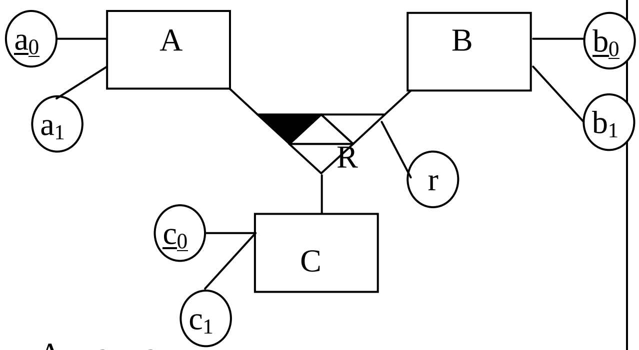
1:M:M



- $A= a_0 + a_1$
- $B = b_0 + b_1$
- $C = c_0 + c_1$
- $R = \underline{a_0 + b_0} + c_0 + r$

VNN{ c_0 }
CAj.{$a_0 \rightarrow A$
$b_0 \rightarrow B$
$c_0 \rightarrow C$ }

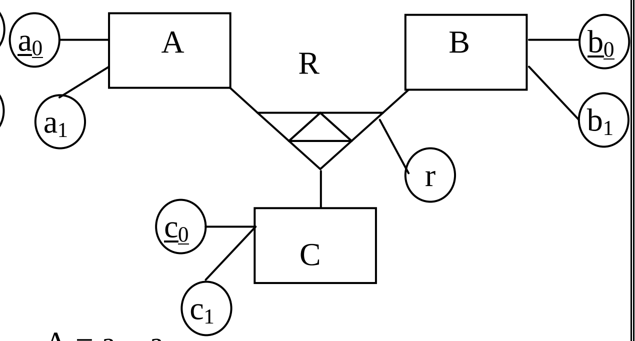
1:1:M



- $A = \underline{a_0} + a_1$
- $B = \underline{b_0} + b_1$
- $C = \underline{c_0} + c_1$
- $R = \underline{a_0 + b_0} + c_0 + r$

CAlt.{a_0, c_0} VNN {c_0}
CAj.{$a_0 \rightarrow A$
$b_0 \rightarrow B$
$c_0 \rightarrow C$ }

1:1:1

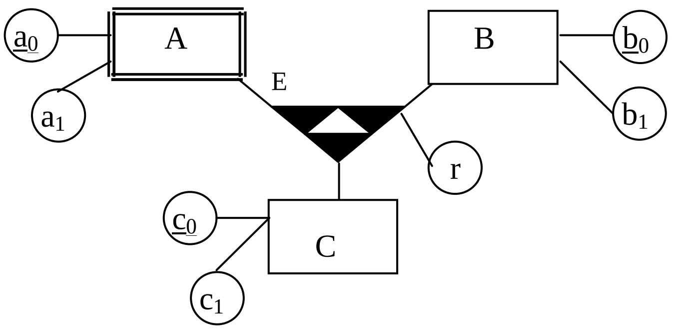


- $A = \underline{a_0} + a_1$
- $B = \underline{b_0} + b_1$
- $C = \underline{c_0} + c_1$
- $R = \underline{a_0 + b_0} + c_0 + r$

CAlt.{a_0, c_0} CAlt.{b_0, c_0} VNN {c_0}
CAj.{$a_0 \rightarrow A$
$b_0 \rightarrow B$
$c_0 \rightarrow C$}

⇨ **TERNARIAS CON R.E.**

M:M:M



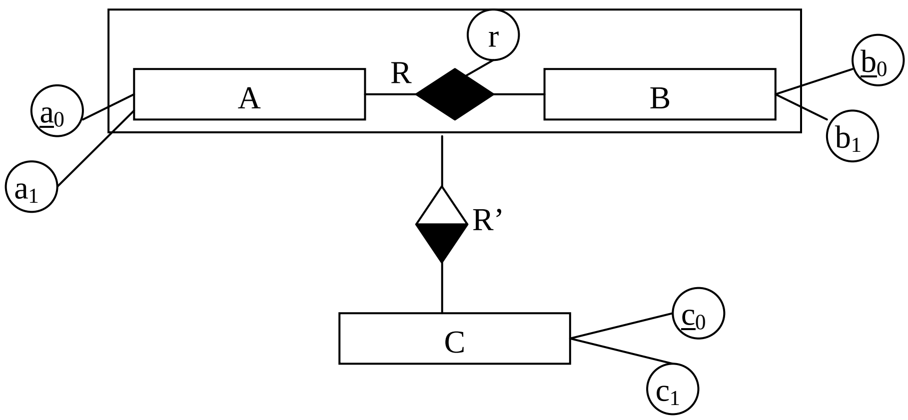
- A = $\underline{a_0} + a_1$
- B = $\underline{b_0} + b_1$
- C = $\underline{c_0} + c_1$
- R = $\underline{a_0 + b_0 + c_0} + r$

CAj.{$a_0 \rightarrow$ A
$b_0 \rightarrow$ B
$c_0 \rightarrow$ C }

R.I.:
$A[a_0] \subset R[a_0]$

Puesto que todas las ternarias generan tabla para la relación cualquier rest. de existencia que pongamos en cualesquiera entidades provocará la aparición de un RI como la del presente caso.

2.3. AGREGACIÓN

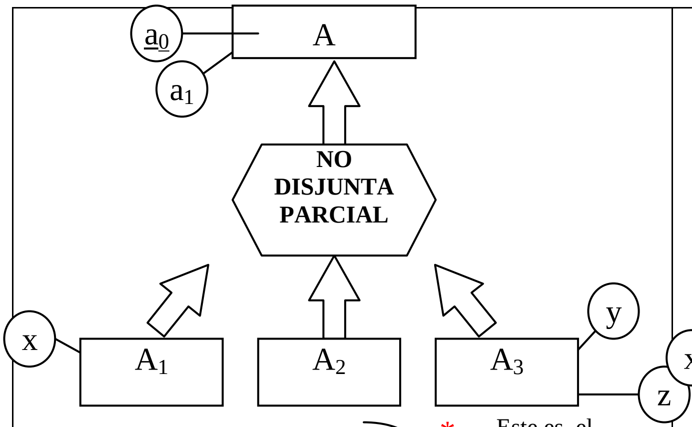


- A = $\underline{a_0} + a_1$
- B = $\underline{b_0} + b_1$
- R = $\underline{a_0 + b_0} + r$
  CAj.{ $a_0 \rightarrow$ A
  $b_0 \rightarrow$ B}
- C = $\underline{c_0} + c_1 + a_0 + b_0$
  CAj.{$(a_0, b_0) \rightarrow$ R}

<u>Pasos a seguir</u>:

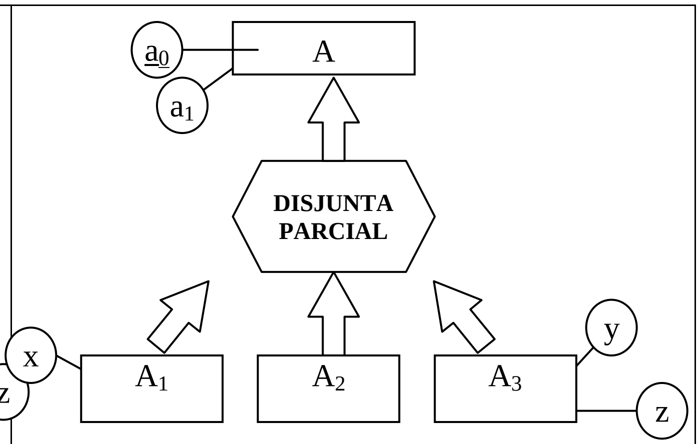
1) Resolver la agregación sin tener en cuenta el resto del Diagrama E/R.
2) Identificar entre las tablas obtenidas en el apartado anterior la "*representante*" de la agregación.
3) Continuar con el diseño del resto del Diagrama . Las participaciones del objeto agregado en relaciones se resolverán teniendo en cuenta el apartado anterior.

2.4. ESPECIALIZACIÓN

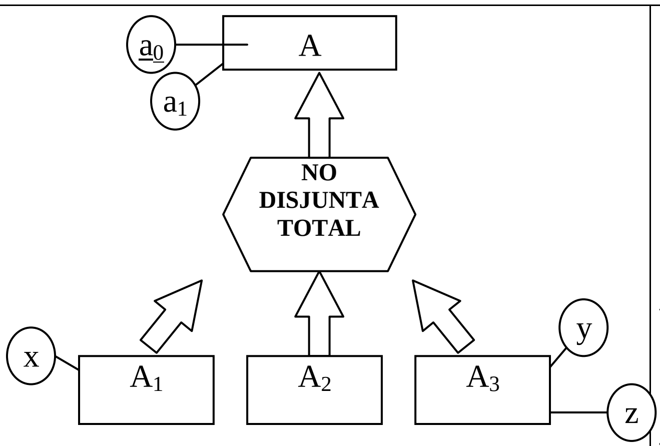


- $A = \underline{a_0} + a_1$
- $A1 = \underline{a_0} + x$ CAj.{a_0 ➔ A}
- $A2 = \underline{a_0}$ CAj.{a_0 ➔ A}
- $A3 = \underline{a_0} + y + z$ CAj.{a_0 ➔ A}

* Este es el **esquema lógico** que representa los distintos tipos de especializa-ción.



- **Esquema Lógico**: * i afegim la **R.I.** <u>DISJUNTA:</u>
  - o $A1[a_0] \cap A2[a_0] = \varnothing$
  - o $A1[a_0] \cap A3[a_0] = \varnothing$
  - o $A2[a_0] \cap A3[a_0] = \varnothing$



- **Esquema Lógico**: afegint una **R.I** per a <u>TOTAL:</u>

$A[a_0] = A1[a_0]$ U $A2[a_0]$ U $A3[a_0]$

La disjunta total tindrà l'esquema de la més "flexible" no disjunta parcial i a més, afegirem les restriccions d'integritat lògiques següents:

<u>DISJUNTA</u>
- o $A1[a_0] \cap A2[a_0] = \varnothing$
- o $A1[a_0] \cap A3[a_0] = \varnothing$
- o $A2[a_0] \cap A3[a_0] = \varnothing$

<u>TOTAL</u>
$A[a_0] = A1[a_0]$ U $A2[a_0]$ U $A3[a_0]$

3. NORMALITZACIÓ

La Normalización es la técnica de la *perspectiva estática,* en particular del modelo relacional, que consiste en aumentar gradualmente la “calidad” de las tablas.

Donde calidad = simplicidad + no redundancia.

La normalización se puede aplicar en tres ámbitos distintos, a saber :

- 1 Como una fase más del proceso de análisis y diseño de la perspectiva estática. En este ámbito la normalización tiene poco margen de maniobra porque las tablas obtenidas en el esquema lógico relacional ya están “altamente” normalizadas.
- 2 El “ámbito “by-Pass”. “Atacamos directamente” los requerimientos del sistema.
- 3 Exportarla a la perspectiva dinámica, “endulzando estáticamente” el análisis dinámico obtenido.

3.1. Procés de Normalització

1. Resolver especialización y/o opcionalidad.
2. Paso a 1FN
    - Optimización de claves
3. Paso a 2FN
4. Paso a 3FN
5. Paso a FNBC (Forma Normal Boyce Codd)
    - FNBC no redundante

Existe una 4FN, e incluso una 5FN. La mayoría de los libros llegan hasta la 3FN no redundante.

El proceso de normalización es *gradual*:

FNBC NO REDUNDANTE ➔ FNBC ➔ 3FN ➔ 2FN ➔ 1FN

Vamos a introducir unos conceptos previos necesarios para la posterior definición de las *Formas Normales.*

3.2. DEPENDENCIA FUNCIONAL

Sean X e Y dos atributos o subconjuntos de atributos de R. Diremos que **Y depende funcionalmente de X**, o lo que es lo mismo que **X determina a Y (X→ Y)** si dado un valor de X existe un único valor de Y asociado en esta tabla R.

PERSONA = **dni** + **nombre** + **edad** + **CP** + **localidad**

Localidad depende de CP. Edad depende de dni…
Si en dos tuplas se repite el CP (Código Postal) tb se repetirá la Localidad

3.3. AXIOMAS DE ARMSTRONG

- REFLEXIVIDAD

  Si $Y \subset X$ entonces $X \rightarrow Y$
  (El todo determina a la parte)

  edad + sexo → edad
  edad + sexo → sexo

- AMPLIFICACIÓN

  X → Y
  W es cualquiera
  $\Rightarrow$ WX→ WY

  En cambio sii WX→ WY entonces X→Y **FALSO**
  Contraejemplo:
  - dni + edad → dni + nombre CIERTO
  - edad → nombre FALSO

  No se puede, por tanto, simplificar la W como ocurre con las matemáticas.

- TRANSITIVIDAD

  X → Y
  Y → Z
  $\Rightarrow$ X → Z

- UNIÓN

  X → Y
  X → Z
  $\Rightarrow$ X → YZ

- DESCOMPOSICIÓN

  X → YZ ➔ X → Y, X → Z

- PSEUDOTRANSITIVIDAD

  WY → Z
  X → Y
  $\Rightarrow$ WX→ Z

  Fundamento teórico del paso a FNBC que más adelante veremos.

3.4. RESOLVER ESPECIALIZACIÓN Y/O OPCIONALIDAD

ESPECIALIZACIÓN

$A = \underline{a_0} + a_1 + [x|y] + a_2$

En notación de Marco convenimos en que se trata de una especialización disjunta y total

- Solución:

  - $A = \underline{a_0} + a_1 + a_2$ (Tabla maestra)
  - $AX = \underline{a_0} + x$ CAj.{a_0 → A}
  - $AY = \underline{a_0} + y$ CAj.{a_0 → A}

RI:

- $AX[a_0] \cap AY[a_0] = \emptyset$ (**Disjunta**)
- $A[a_0] \subset (AX[a_0] \cup AY[a_0])$ (**Total**)

OPCIONALIDAD

$A = a_0 + a_1 + (x) + a_2$

- Solución:

  - $A = \underline{a_0} + a_1 + a_2$
  - $A' = \underline{a_0} + x$ CAj.{a_0 → A} VNN{ x } ← Discutible!!

3.5. PRIMERA FORMA NORMAL (1FN)

Una tabla o estructura está en **1FN** si todos los dominios de sus atributos toman valores atómicos. Es decir dicha tabla o estructura no debe tener grupos repetitivos.

PASO A 1FN:

$A = \underline{a_0} + a_1 + a_2 + \{a_3\}$

- Solución:

  - $A = \underline{a_0} + a_1 + a_2$
  - $A' = \underline{a_0 + a_3}$ CAj.{a_0 → A}

Las claves ajenas las estamos poniendo como enriquecimiento del proceso de normalización estándar.

EJEMPLO:

PERSONA = <u>dni</u> + nombre + dir + {teléfono}

- PERSONA = <u>dni</u> + nombre + dir
- PERSONA_TLF = <u>dni + teléfono</u>
  CAj. {dni → PERSONA}

3.5.1. OPTIMIZACIÓN DE CLAVES

Consiste en ***minimizar*** las claves. Téngase en cuenta que al resolver el paso a 1FN podríamos, ante la dudad, haber “subrayado más campos de la cuenta”.

EJEMPLO:

$\mathbf{A = \underline{x + y} + a + b}$

Después de optimizar: $\mathbf{A = \underline{x} + y + a + b}$

3.6. SEGUNDA FORMA NORMAL (2FN)

> Una tabla está en **2FN** si y solo si está en 1FN y todo atributo no clave principal dependen de la totalidad de ésta y no de parte de ella. Sólo son sospechosas de no estar en 2FN aquellas tablas cuya clave principal sea compuesta.

PASO A 2FN:

$\mathbf{R = \underline{a_0 + b_0} + r_1 + r_2}$

- Solución:
  - $R = \underline{a_0 + b_0} + r_1$ CAj. { $\underline{b_0}$ → R’}
  - $R' = \underline{b_0} + r_2$

EJEMPLO:

NOTAS = <u>codAsig + nºExped</u> + nota + dirAlum

- NOTAS = <u>codAsig + nºExped</u> + nota CAj.{ <u>nºExped</u> → ALUMNO}
- ALUMNO = <u>nºExped</u> + dirAlum

¿Por qué es "bueno" pasar a 2FN?

3.7. TERCERA FORMA NORMAL (3FN)

Una tabla está en **3FN** si estando en 2FN todos los atributos (pueden ser compuestos) no clave (ya sea principal o alternativa) no dependen entre si.

PASO A 3FN:

$$\mathbf{A = \underline{a_0} + a_1 + a_2 + a_3}$$

- Solución:
  - $A = \underline{a_0} + a_1 + a_2$ CAj. { a_2 → A'}
  - $A' = \underline{a_2} + a_3$

EJEMPLO:

PERSONA = <u>dni</u> + nombre + dir + CP + localidad

- PERSONA = <u>dni</u> + nombre + dir + CP   CAj. {CP → LOCALIDADES}
- LOCALIDADES = <u>CP</u> + localidad

Al pasar a 3FN conseguimos que todos los atributos de la tabla dependan únicamente de la clave principal de ésta o de las alternativas. Con ello nos protegemos de introducir datos erróneos, evitamos duplicidades.

3.8. FORMA NORMAL BOYCE CODD (FNBC)

Concepto previo: **<u>DETERMINANTE</u>**

Un determinante es un atributo (simple o compuesto) del cual algún otro atributo depende funcionalmente de él. Es decir atributo(s) del que "salen flechas".

EJEMPLO:

PERSONA = <u>dni</u> + nombre + edad + dir + CP + localidad

Tenemos dos determinantes: **dni y CP**

Una tabla está en **FNBC** si estando en 3FN todo determinante es clave (principal o alternativa). Si la tabla está en 3FN y su clave principal es simple, ya está en FNBC.

PASO A FNBC:

$$\mathbf{R = \underline{a_0 + b_0} + x + r_1 + r_2 + r_3}$$

- Solución:
  - R = $\underline{a_0 + x} + r_1 + r_2 + r_3$ CAj. { x → X}
  - X = $\underline{x} + b_0$

3.9. FNBC NO REDUNDANTE

Se trata de eliminar más redundancias si cabe.

3.9.1. <u>REDUNDANCIA INTRATABLA</u>

Consiste en eliminar uno o varios campos que son redundantes unos con otros.

> **EJEMPLO**:
>
> T = ...+ Nota1 +Nota2 + ~~NotaMedia~~

3.9.2. <u>REDUNDANCIA INTERTABLA</u>

Se produce cuando los campos de una tabla son deducibles a partir de información almacenada en otras tablas. Es discutible la eliminación “sin más” de estos campos, ya que la eliminación de la redundancia intertabla suele provocar ralentización de las consultas y mantenimiento de la Base de Datos.

Se trata de buscar un compromiso entre pureza y eficiencia. Se estudiará cada caso particular.

3.9.3. FUSIONAR TABLAS QUE COMPARTAN LA MISMA CLAVE

Sólo fusionaremos si con ello no se rompe el resultado de la *opcionalidad* y *especialización* llevado a cabo en las primeras fases de la normalización.

EJEMPLO:

$$\left.\begin{array}{l} \bullet\ R = \underline{a} + b + x \\ \bullet\ R' = \underline{a} + b + y + z \end{array}\right\} \mathbf{R = \underline{a} + b + x + y + z}$$

Però després de la fusió caldrà fer una segona pasada en la eliminació de redundàncies tot i que poden apareixer, per exemple, noves redundàncies Intrataula. A més, podría generar-se alguna dependència entre atributs no clau ➔ 3FN altre cop!!