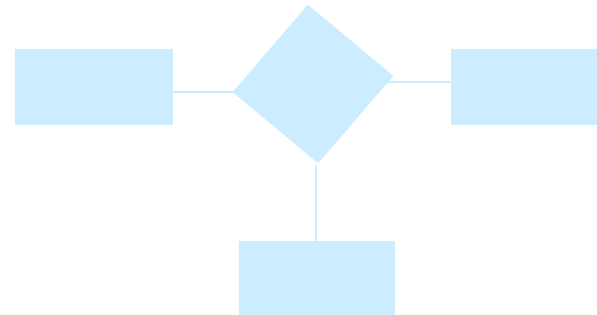


Bases de Datos y Sistemas de Información

Ernesto Ponsot Balaguer*

Universidad de Los Andes
Escuela de Estadística - FACES
Cátedra de Computación.

CONTINUACIÓN...



*: La Hechicera. Cub.28. Tlf.: 401122/401153.

E-mail: ernesto@faces.ula.ve

Relación...

Clave Primaria

dd/mm/aa

#-###-#

Bs.>0

> 0

Caracas,
Mérida, etc.

R

Fecha	N' de Parte	Precio	Cantidad	Ciudad
15/10/96	1-005-2	578.95	5	Mérida
22/10/96	1-006-2	758.25	6	Mérida
14/09/96	1-258-1	278.00	15	San Cristóbal
15/11/96	2-027-2	1596.25	3	Maracaibo
15/10/96	1-006-2	730.00	7	Caracas
12/01/97	2-027-2	1490.00	4	El Vigía

Tuplas

Cardi-
nalidad
m

Atributos

Grado *n*

Otra Representación...

- Toda relación puede representarse como una “tabla” de filas y columnas. Esta representación implica buscar una *instancia* de la relación (una fotografía de la relación, tomada en un instante fijo del tiempo), donde se listan tuplas ejemplo de la misma. Esta representación es explicativa pero un tanto aparatosa.

Otra Representación...

- Otra representación de una relación, que no muestra ejemplos de las tuplas, se consigue al usar la notación de conjuntos al explicitar únicamente los atributos de la cabecera y subrayar aquellos atributos que forman la clave primaria. Como ejemplo, la relación R anterior puede explicitarse:

Movimiento = { Fecha, Nº de Parte, Precio,
Cantidad, Ciudad }

Claves o Llaves...

- En la figura anterior, se muestran casi todos los elementos componentes de una relación, a la luz de la definición. El único elemento mostrado, ausente en la definición, es la “clave primaria”. Intuitivamente la palabra *clave* implica un identificador único para cada ítem de datos presente en la relación y lo de *primaria* presenta la idea de principal o preponderante.

Claves Candidatas...

- Formalmente, hay varias definiciones asociadas con este concepto:

Todo subconjunto K de los atributos de la cabecera de una relación R , que satisface las propiedades de **Unicidad** y **Minimalidad** se denomina una **Clave Candidata** de R .

Principio de Unicidad...

Dado un subconjunto K de los atributos de la cabecera de una relación R , se dice que K cumple la propiedad de **Unicidad** si y sólo si, en cualquier instante del tiempo, no existen entre las tuplas del cuerpo de R valores iguales de K . En este caso la igualdad se utiliza en el contexto de conjuntos. Hay que recordar que dos conjuntos son iguales, si y sólo si, están formados por los mismos elementos.

Principio de Minimalidad...

Dado un subconjunto K de los atributos de la cabecera de una relación R , que cumple la propiedad de unicidad, se dice que K cumple también la propiedad de **Minimalidad** si y sólo si, no es posible desechar alguno(s) de los elementos de K sin que el nuevo subconjunto K viole la propiedad de unicidad.

Claves Candidatas

- Son entonces los subconjuntos, con el menor número de elementos de la cabecera de la relación, cuyos correspondientes valores en el cuerpo de la relación, no se repiten. Esto es, el menor número de atributos que identifican de forma única a cualquier tupla de la relación.

Clave Primaria

- Sencillamente es aquella clave candidata seleccionada por el diseñador de la base de datos como la que identificará a las tuplas de la relación.
- Las claves candidatas no seleccionadas, se denominan claves alternativas y en realidad, pierden interés frente a la clave primaria.

Observaciones...

- En vista de que una relación está compuesta por un *conjunto* de tuplas, se puede asegurar que toda relación tiene al menos una clave candidata y por lo tanto puede definirse una clave primaria (ya que los conjuntos no contienen elementos repetidos, en este caso, el cuerpo de la relación es un conjunto de tuplas por lo cual las tuplas -que son los elementos del conjunto- no pueden repetirse).

Observaciones...

- La escogencia de la clave primaria se realiza con “sentido común” y en general debe escogerse aquel subconjunto de los atributos, que tenga el menor número de elementos o que intuitivamente represente mejor a las tuplas.

Observaciones...

- La búsqueda de claves candidatas se fundamenta en los supuestos que realice el diseñador y, en general, no necesita ser exhaustiva. Puede encontrarse una (la primera) y decidirse que ésta será la clave primaria, con lo que se detiene el proceso.

Ejemplos...

- Supóngase una relación que almacenará datos de los empleados de una empresa, de la forma siguiente:

Empleados={C.I., Apellidos, Nombres, Dirección,
Fecha de Nacimiento, Estado Civil}

Ejemplos...

- Se puede pensar en algunas claves de Empleados: $K_1=\{\text{C.I.}\}$, $K_2=\{\text{C.I., Apellidos}\}$, $K_3=\{\text{Apellidos, Nombres}\}$, $K_4=\{\text{Apellidos, Nombres, Dirección, Fecha de Nacimiento}\}$, sin embargo, puede verse:
 - ◆ K_1 es una clave candidata porque se supone que la cédula de identidad no se repite para ningún empleado y si se elimina este atributo el conjunto queda vacío (que por razones obvias no puede ser una clave candidata).

Ejemplos...

- ◆ K_2 cumple con el principio de unicidad (derivado de que C.I. es única) ya que ninguna combinación de C.I. y Apellidos se repite en la relación, pero no cumple con el principio de minimalidad ya que existe un subconjunto de K_2 más pequeño (K_1) que no viola el principio de unicidad.
- ◆ K_3 no puede ser una clave candidata ya que no cumple con el principio de unicidad: puede ser que en algún instante hayan dos empleados con iguales apellidos y nombres.

Ejemplos...

- ◆ K_4 si puede ser una clave candidata ya que puede suponerse que en ningún momento habrán dos empleados que tengan los mismos apellidos, nombres, dirección y fecha de nacimiento y además, si se retira cualquiera de estos atributos, ya no se puede garantizar la unicidad.
- Ahora bien, entre K_1 y K_4 , dos claves candidatas de Empleados, por lo anteriormente expuesto se debería seleccionar a K_1 como la clave primaria.

Ejemplos...

- De esta forma, la relación Empleados queda determinada así:

Empleados = { C.I., Apellidos, Nombres, Dirección,
Fecha de Nacimiento, Estado Civil }

- Quedando por expresar, los dominios asociados.

Claves Ajenas...

- Supóngase, la siguiente sección de una base de datos de empleados:
 - ◆ Empleados={C.I., Apellidos, Nombres, Código del Departamento}
 - ◆ Departamentos={Código, Nombre, C.I. del Gerente}

Claves Ajenas...

- El atributo Código del Departamento, en Empleados, obviamente está definido sobre el mismo dominio que el atributo Código en Departamentos. Más aún, no se esperaría que existiese en Empleados, un valor de Código del Departamento que no estuviese previamente en Departamentos (ya que esto implicaría la adscripción de un empleado a un departamento inexistente).

Claves Ajenas...

- De la misma forma, el atributo C.I. del Gerente, en Departamentos, está definido sobre el mismo dominio que el atributo C.I. en Empleados y no se esperaría que un Departamento tuviese asignado un gerente sin que éste estuviese registrado primero como empleado en Empleados.
- Por último, ambos atributos representan las claves primarias de las relaciones asociadas.

Claves Ajenas...

- Estos dos ejemplos ilustran lo que se conoce como una *clave ajena*.

Una **Clave Ajena** “en” una relación R_1 es cualquier subconjunto de atributos de R_1 que representen a una clave primaria de otra relación R_2 .

Claves Ajenas...

- Así, en el ejemplo, *Código del Departamento* es una clave ajena en Empleados ya que representa la clave primaria de Departamentos y *C.I. del Gerente* es una clave ajena en Departamentos ya que representa la clave primaria de Empleados.

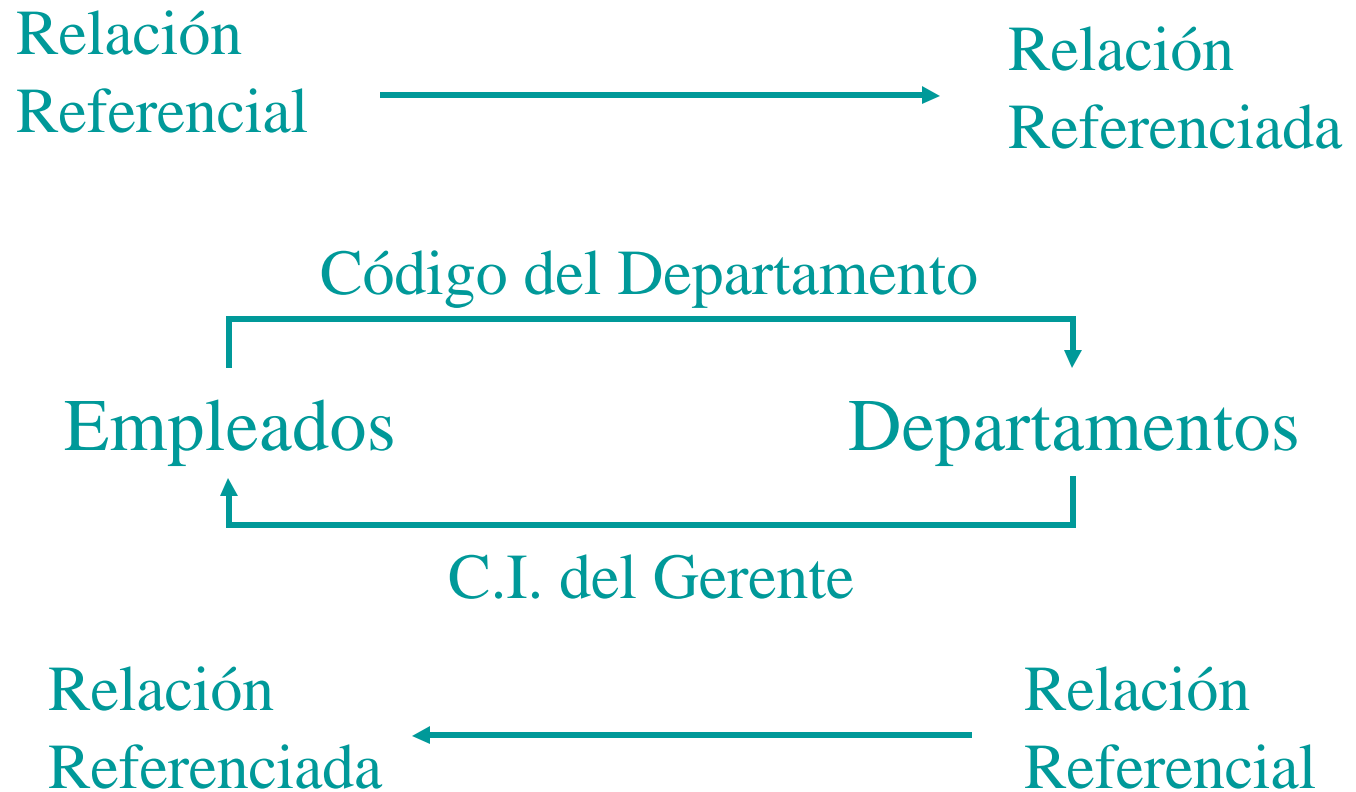
Claves Ajenas...Observaciones

- La clave ajena, es en realidad, la forma que tiene el modelo relacional de explicitar las asociaciones entre las distintas relaciones que lo componen.
- Como la clave primaria de una relación, identifica de manera única a cada tupla de la relación, al conocer el valor de la clave primaria de una tupla, se conocen inmediatamente los valores del resto de los atributos de la tupla (se conoce toda la tupla).

Diagrama Referencial...

- Conocidas las claves ajenas de todas las relaciones, puede construirse entonces un diagrama que muestre las distintas asociaciones existentes en la base de datos. Este diagrama se conoce con el nombre de **Diagrama Referencial**. En el ejemplo:

Diagrama Referencial.



Ejercicios

- En el ejercicio de la planta ensambladora de televisores, esquematice el modelo relacional, indicando claves primarias y claves ajenas.

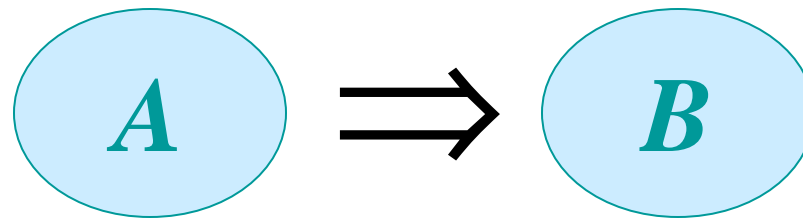
Normalización de Relaciones

- En la jerga de bases de datos, se denomina Normalización, al proceso mediante el cual se transforma un conjunto de relaciones dadas, en otro conjunto equivalente de relaciones “más deseable”.
- En este sentido, la normalización es un proceso que se fundamenta en el estudio crítico de todas y cada una de las relaciones incluidas en el modelo diseñado.

Normalización...

Dependencias Funcionales

Dos atributos A y B de una relación R (simples o compuestos) son tales que B es **dependiente funcional** de A si la ocurrencia de cualquier valor de A en R implica necesariamente la ocurrencia del mismo valor de B en R .



Normalización...

Dependencias Funcionales Completas...

Dados dos atributos A (en general compuesto) y B de una relación R , tales que B es dependiente funcional de A . Se dice que esta es una **dependencia funcional completa** si B no es dependiente funcional de ningún subconjunto de A de cardinalidad menor.

$$A = \{ A_1, A_2, \dots, A_k \} \Rightarrow B$$

$$A' = \{ A_1, A_2, \dots, A_{k-1} \} \not\Rightarrow B$$

$$A'' = \{ A_2, A_3, \dots, A_k \} \not\Rightarrow B$$

⋮

$$A^j = \{ A_l \} \not\Rightarrow B$$

Dependencias Funcionales... Observaciones

- Si B es dependiente funcional de A y A es un atributo simple, entonces necesariamente B es dependiente funcional completo de A .
- Si A es la clave primaria de la relación, entonces por definición, todos los atributos de la relación que no forman parte de A son dependientes funcionales de A .
- Nótese que la dependencia funcional se establece en una sólo dirección, sin embargo, entre dos claves candidatas de una relación cuya intersección es vacía, hay una dependencia funcional en ambas direcciones.

Primera Forma Normal (1FN)

Una relación está en 1FN si todos los dominios sobre los que se definen sus atributos, están formados por valores atómicos (escalares que no admiten descomposición). Es decir, si cumple estrictamente con la definición.

Segunda Forma Normal (2FN)

Una relación está en 2FN si está en 1FN y todos los atributos que no forman parte de la clave primaria, son dependientes funcionales completos de ella.

Tercera Forma Normal (3FN)

Una relación está en 3FN si está en 2FN y no existen dependencias funcionales transitivas.

- Dados tres atributos A , B y C , si B es Dep. Func. de A y C es Dep. Func. de B entonces, C es Dep. Func. de A . A esta última dependencia funcional, se le denomina transitiva (en palabras coloquiales, de “rebote”).

Normalización. Ejemplo.

- Estúdiese la normalización de la siguiente relación (que almacena información de los libros en una biblioteca):

Libros={ Código del Libro, Nombre del Libro, Nombres de los Autores, Tema Clave, Descripción del Tema, Ubicación={Estante, Repisa}, Proveedor del Libro, Precio del Libro }

Ejemplo: 1FN...

- La relación **Libros**, NO ESTÁ en 1FN ya que contiene atributos definidos sobre dominios cuyos elementos no son escalares. Tal es el caso de los atributos *Nombres de los Autores* y *Ubicación*, que en realidad se definen sobre una lista de valores.

Ejemplo: 1FN...

- Para solventar este problema, la primera tentación es sustituir el plural *Nombres de los Autores* por el singular *Nombre del Autor* y eliminar el atributo *Ubicación* colocando en su lugar dos atributos *Estante* y *Repisa*. Así:

Libros= { Código del Libro, Nombre del Libro, Tema Clave, Descripción del Tema, Proveedor del Libro, Precio del Libro, Nombre del Autor, Estante, Repisa }

Ejemplo: 1FN...

- Sin embargo, cuando se incluyó el plural *Nombres de los Autores*, lo que seguramente se pensó fue que un libro podría tener más de un autor. Luego, la transformación realizada ocasiona bien la pérdida de información (porque se tendría que restringir los autores a uno sólo) o bien la redundancia excesiva (pues de no restringirse el número de autores, habría que repetir toda la información del libro para cada autor -componiendo una tupla por autor-).

Ejemplo: 1FN...

- Por otra parte, cuando se incluyó el atributo *Ubicación* la idea fue identificar el par Estante-Repisa como una sola característica. Es claro que hay redundancia excesiva en la solución propuesta puesto que muchos libros pueden compartir el mismo estante y repisa por lo que será necesario repetir esta información (de dos atributos) demasiadas veces.

Ejemplo: 1FN...

- Entonces, la primera tentación resuelve el problema de la 1FN pero crea otros. Así, una solución mejorada es la creación de nuevas relaciones:

Autores={Código del Autor, Nombre del Autor}

Ubicaciones={Código de la Ubicación, Estante, Repisa}

Autores de Libros={Código del Libro, Nombre del Libro, Código del Autor}

Libros={Código del Libro, Nombre del Libro, Tema Clave, Descripción del Tema, Proveedor del Libro, Precio del Libro, Ubicación}

Ejemplo: 2FN...

- Ahora las nuevas relaciones “más deseables” están en 1FN, sin embargo, la relación Libros no está en 2FN pues los atributos que no forman parte de la clave primaria, NO SON dependientes funcionales completos de ella. En efecto, *Tema Clave*, *Descripción del Tema*, *Proveedor del Libro*, *Precio del Libro* y *Ubicación* son dependientes funcionales de {*Código del Libro*, *Nombre del Libro*} pero también lo son de {*Código del Libro*}.

Ejemplo: 2FN...

- Entonces, el atributo *Nombre del Libro* no debe formar parte de la clave primaria de la relación (había un error en la escogencia de la clave). El nuevo conjunto de relaciones es entonces:

Autores={Código del Autor, Nombre del Autor}

Ubicaciones={Código de la Ubicación, Estante, Repisa}

Autores de Libros={Código del Libro, Código del Autor}

Libros={Código del Libro, Nombre del Libro, Tema Clave, Descripción del Tema, Proveedor del Libro, Precio del Libro, Ubicación}

Ejemplo: 3FN...

- Ahora las relaciones están en 2FN, sin embargo, Libros no está en 3FN. Existen atributos en Libros que son dependientes de la clave primaria de forma transitiva. El atributo *Descripción del Tema* es directamente dependiente funcional de *Tema Clave* y como éste lo es de la clave primaria, *Descripción del Tema* termina dependiendo funcionalmente “de rebote” de la clave primaria (sin serlo realmente). Algo similar ocurre con *Precio del Libro* y *Proveedor*.

Ejemplo: 3FN...

- La solución definitiva:

Autores={Código del Autor, Nombre del Autor}

Ubicaciones={Código de la Ubicación, Estante, Repisa}

Autores de Libros={Código del Libro, Código del Autor}

Libros={Código del Libro, Nombre del Libro, Tema Clave, Código de la Ubicación}

Temas={Tema Clave, Descripción del Tema}

Proveedores de Libros={Código del Libro, Proveedor, Precio del Libro}

Ejercicio

- Realice el diagrama referencial de las soluciones parciales y de la definitiva. Compare los diagramas.