

FICHEROS Y BASES DE DATOS (E44)

3º INGENIERÍA EN INFORMÁTICA

Tema 13.

Diseño Lógico

- 1.- Introducción.
- 2.- Diseño Lógico en el Modelo Relacional.
 - 2.1.- Correspondencia de Esquemas.
 - 2.2.- Normalización.
 - 2.3.- Reglas de Integridad.

(Capítulos del 11 y 12 del Batini)

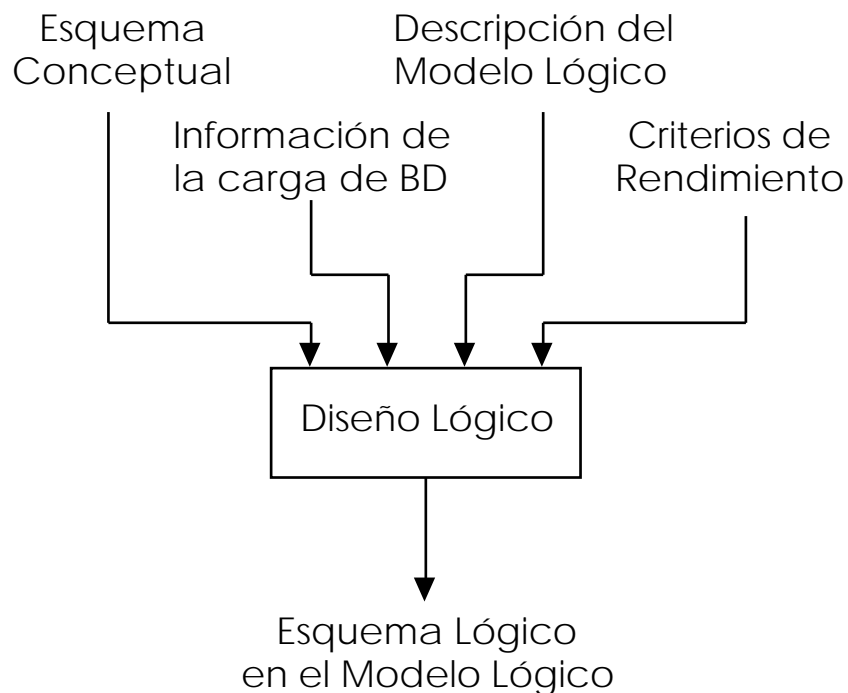
(Capítulo 21 del Date)

INTRODUCCIÓN

Comentarios Iniciales

- El objetivo del diseño lógico es convertir un esquema conceptual en un esquema lógico que se ajuste al sistema de gestión de base de datos a utilizar.
- El diseño conceptual pretendía generar un esquema lo más completo y expresivo posible.
- Con el diseño lógico se pretende obtener un esquema que utilice de la forma más eficiente los recursos del modelo lógico para estructurar datos y modelar restricciones.
- Este planteamiento es común para cualquier modelo lógico, aunque en este caso sólo se va a considerar el modelo relacional.

Representación del Diseño Lógico



INTRODUCCIÓN

Entradas y Salidas del Diseño Lógico

- El diseño lógico es un proceso que dadas unas informaciones iniciales genera el esquema lógico de la aplicación.
 - Una de las entradas a este proceso es el Esquema Conceptual a transformar.
 - Por supuesto, también se debe especificar el Modelo Lógico a utilizar.
 - Otras informaciones que ayudan en el diseño es la Información sobre la Carga de la Base de Datos,
 - Población de la Base de Datos.
 - Consultas y Transacciones más utilizadas.
 - Frecuencia de las operaciones.
- y los Criterios de Rendimiento que permitirán analizar la eficiencia de la implementación, y que utilizan los siguientes parámetros,
- Tiempo de Respuesta, máximo y promedio de las operaciones sobre la base de datos.
 - Almacenamiento ocupado por la base de datos.
 - Utilización de la CPU o tiempo de E/S.
- Los rendimientos se analizan mediante el estudio de la estructura de las operaciones, su frecuencia, y el número de casos de entidades e interrelaciones visitadas.

DISEÑO LÓG. EN EL MOD. RELACIONAL

Comentarios Iniciales

- En este caso, el objetivo es transformar un Esquema Entidad/Relación en un Esquema Lógico Relacional.
- Para posibilitar esta tarea es necesario realizar una correspondencia entre los dos esquemas que participan en esta fase.
- La primera tarea es definir la Correspondencia de los Esquemas inicial y final, mediante la transformación de las estructuras utilizadas en los modelos asociados.
- Además se pretende que el esquema lógico cumpla ciertas características,
 - Las relaciones deben de estar en tercera normal.
 - Se deben definir las claves primarias y ajenas de todas las relaciones.
 - Es necesario incluir en el esquema las reglas de integridad necesarias.
- Por lo que el proceso de diseño también debe contemplar aspectos relacionados con,
 - La Normalización de las relaciones del esquema.
 - Definición de las Reglas de Integridad útiles dentro del esquema.

DISEÑO LÓG. EN EL MOD. RELACIONAL

Transf. de Jerarquías de Generalización

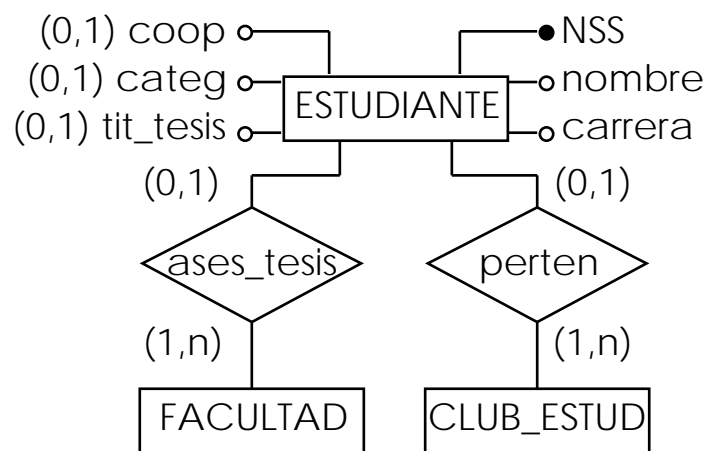
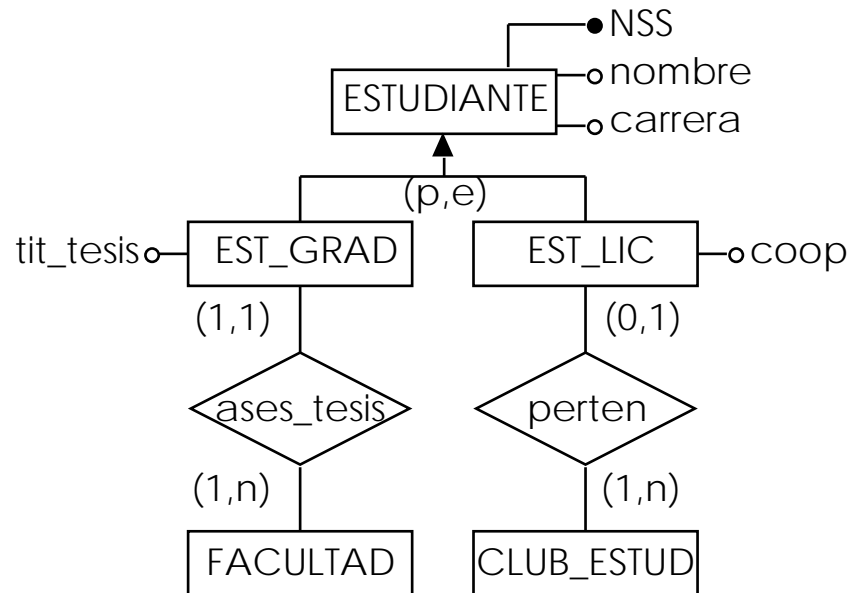
- El modelo relacional no puede representar jerarquías de generalización.
- Por esta razón resulta aconsejable modificar el esquema entidad/relación antes de estudiar la correspondencia.
- Existen tres posibles alternativas.
 - Integración de las entidades subconjunto.
 - Eliminación de la entidad genérica.
 - No eliminar ninguna de las entidades.

Integración de las Entidades Subconjunto

- Los atributos de las entidades subconjunto se añaden a los de la entidad genérica con cardinalidad mínima cero.
- Se añade un atributo discriminativo en la entidad genérica, que indica el caso al que pertenece la entidad considerada, con
 - Cardinalidad Mínima cero si la jerarquía es parcial, o uno si fuera total.
 - Cardinalidad Máxima uno si la jerarquía es exclusiva, o n si fuera superpuesta.
- Es aplicable a cualquier tipo de jerarquía, pero puede generar la aparición de muchos valores nulos.
- Todas las operaciones se deben aplicar sobre la entidad genérica.

DISEÑO LÓG. EN EL MOD. RELACIONAL

Ejemplo de Integración de Ent. Subconjunto



DISEÑO LÓG. EN EL MOD. RELACIONAL

Eliminación de la Entidad Genérica

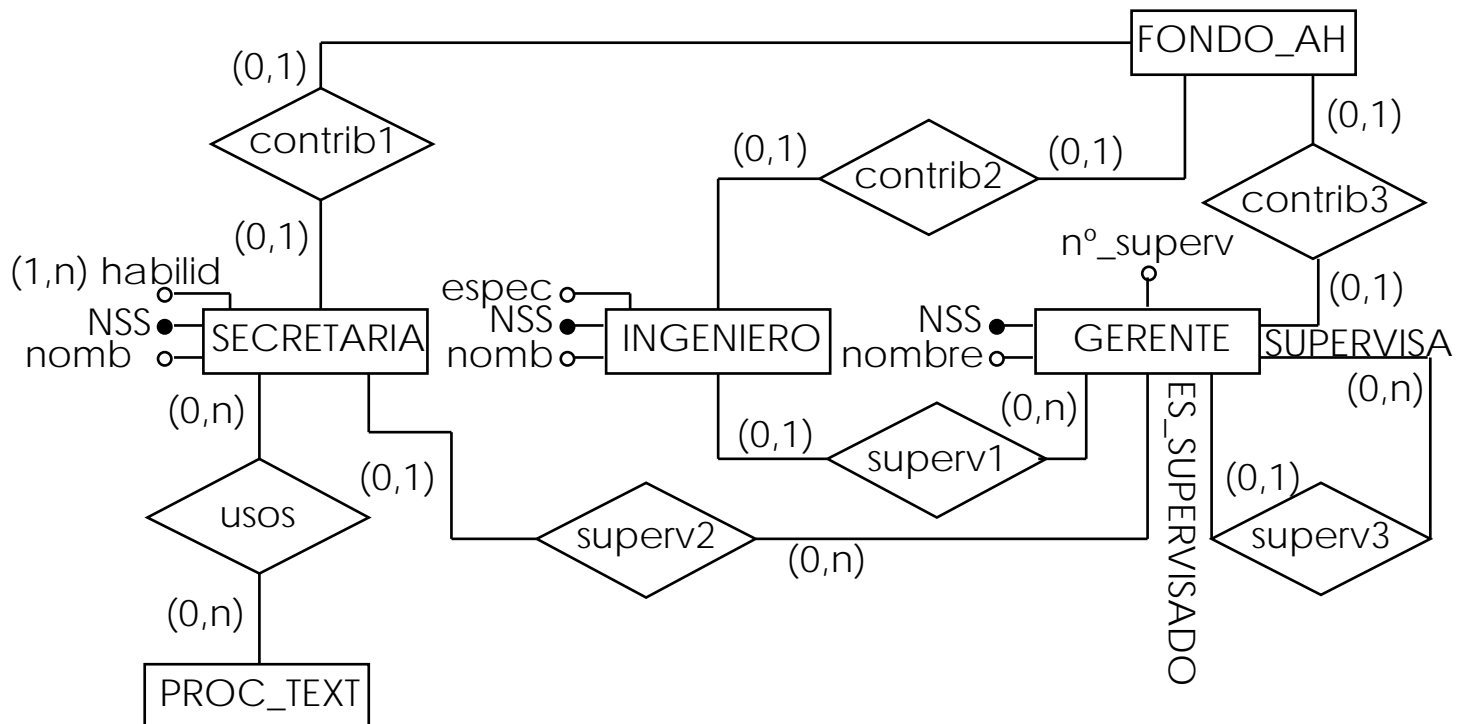
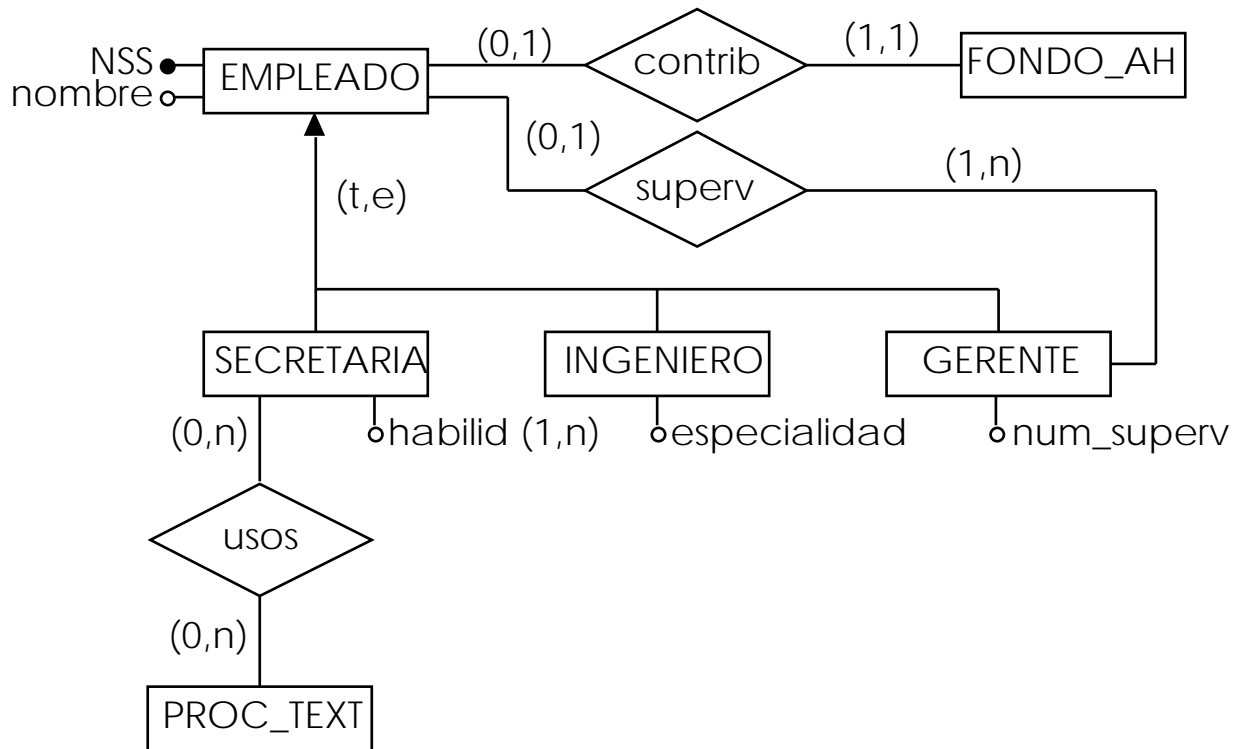
- Los atributos de la entidad genérica son heredados por todas las entidades subconjunto.
- Sólo es útil para jerarquías totales y exclusivas, ya que de otro modo los casos aparecerían en más de una entidad subconjunto.
- Se pierde el concepto de inclusión de las entidades subconjunto en la entidad genérica.
- Las operaciones sobre la entidad genérica deben aplicarse sobre todas las entidades subconjunto.

No Eliminar Ninguna de las Entidades

- En este caso la inclusión de las entidades subconjunto sobre la entidad genérica se modeliza mediante la interrelación ES_UN.
- La cardinalidad de la entidad genérica sobre la interrelación es (0,1), mientras que en las entidades subconjunto es (1,1).
- Aparece un identificador externo asociado a cada entidad subconjunto definido sobre la interrelación que la conecta con la entidad genérica.
- Es la alternativa más flexible, y además las operaciones se siguen realizando sobre las mismas entidades.

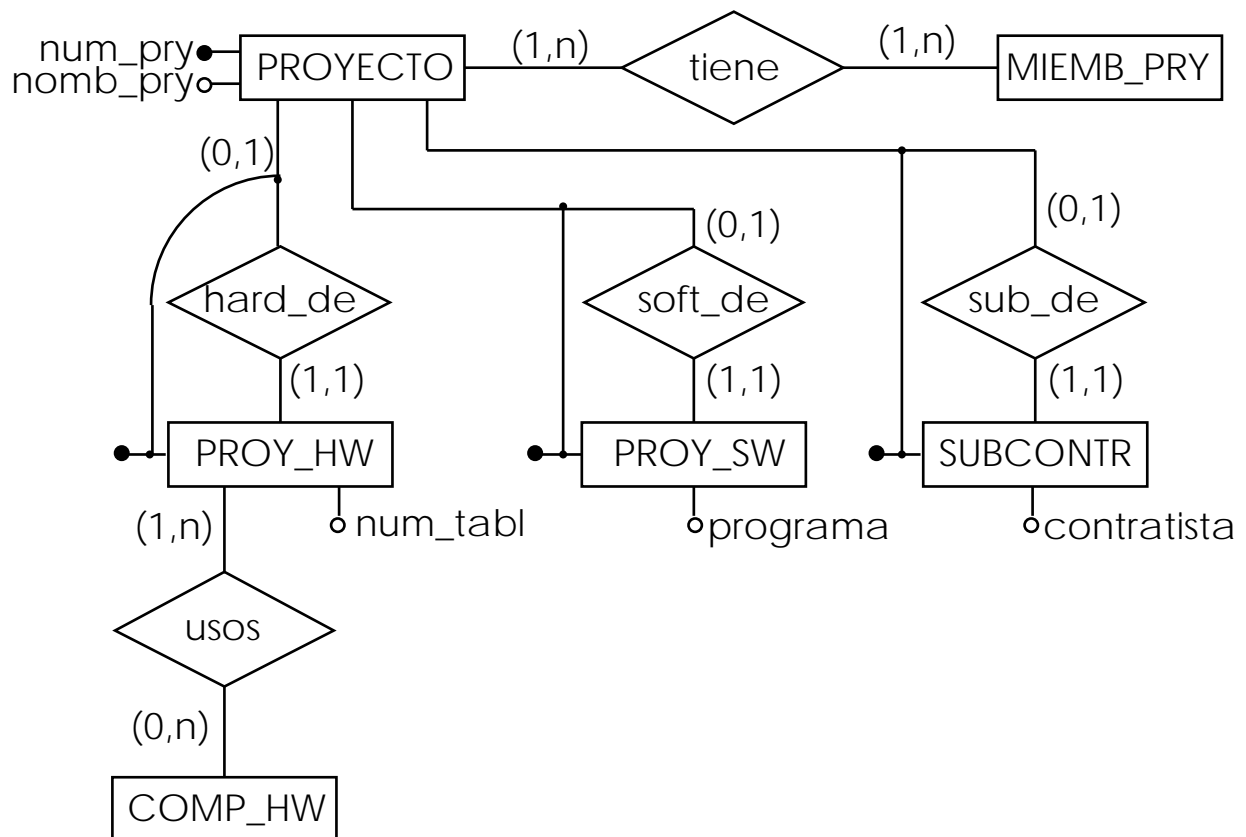
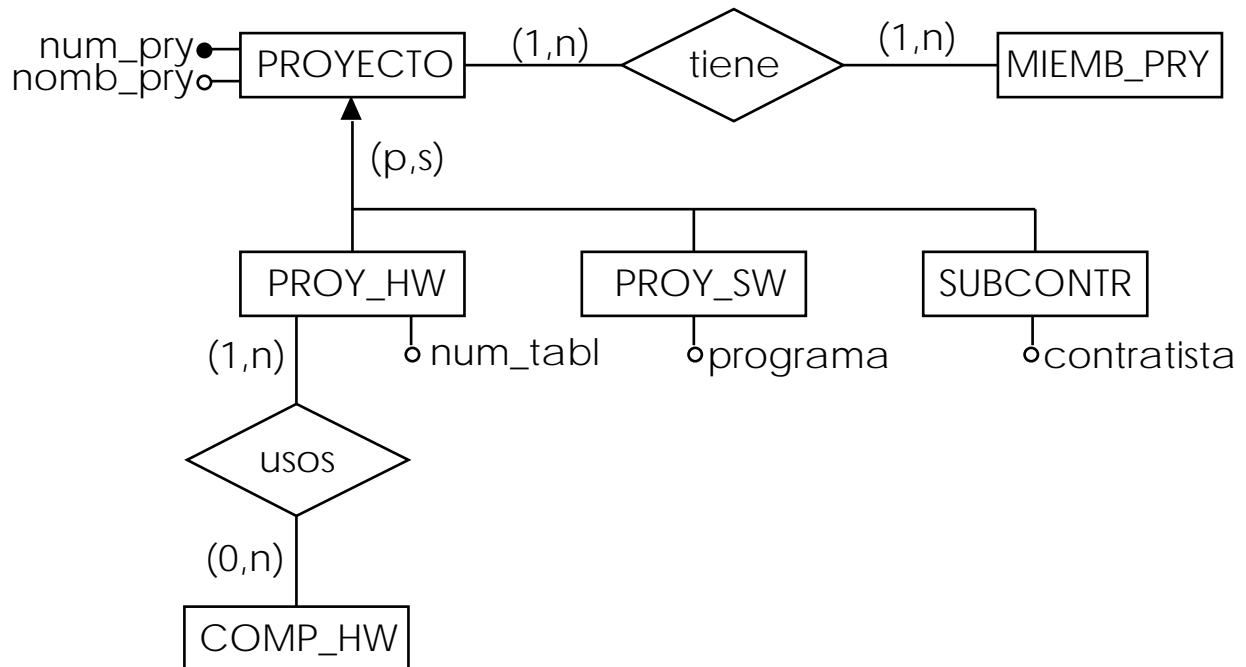
DISEÑO LÓG. EN EL MOD. RELACIONAL

Ejemplo de Elimin. de la Entidad Genérica



DISEÑO LÓG. EN EL MOD. RELACIONAL

Ejemplo de No Eliminación



DISEÑO LÓG. EN EL MOD. RELACIONAL

Transf. de Identificadores Externos

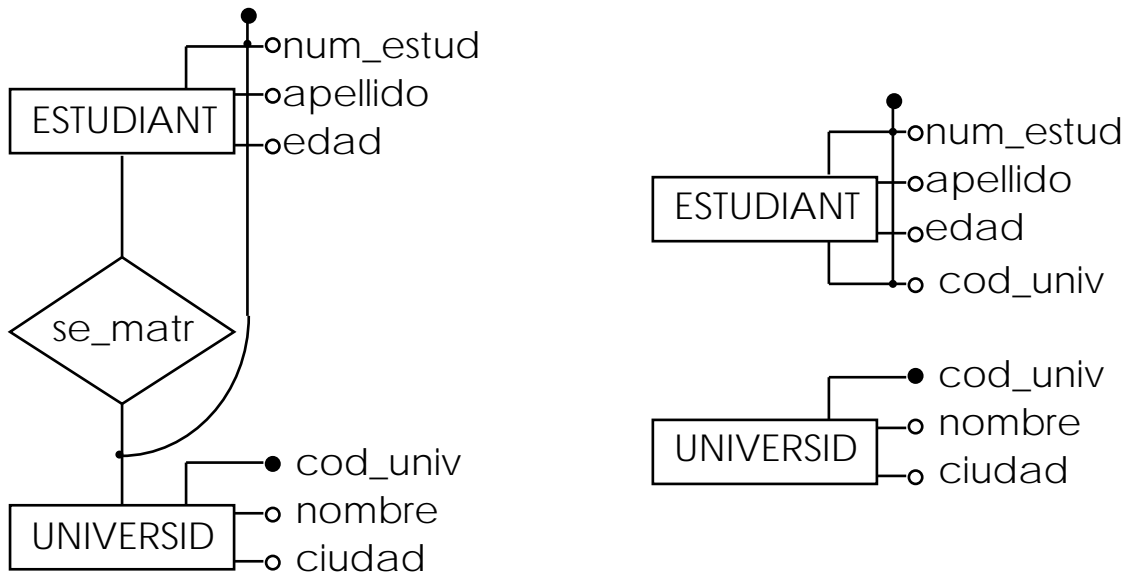
- En el modelo relacional no se puede utilizar identificadores externos, por lo que hay que transformarlos en identificadores internos.
- Una posibilidad es importar a la entidad los atributos asociados al identificador de las entidades que forman parte del identificador externo.
- Las interrelaciones asociadas al identificador externo puede eliminarse, ya que la relación entre las entidades ya se incorpora al modelo.
- Este proceso se inicia en las entidades con identificadores internos, y continua con las entidades vecinas.

Transf. de Atributos Compuestos

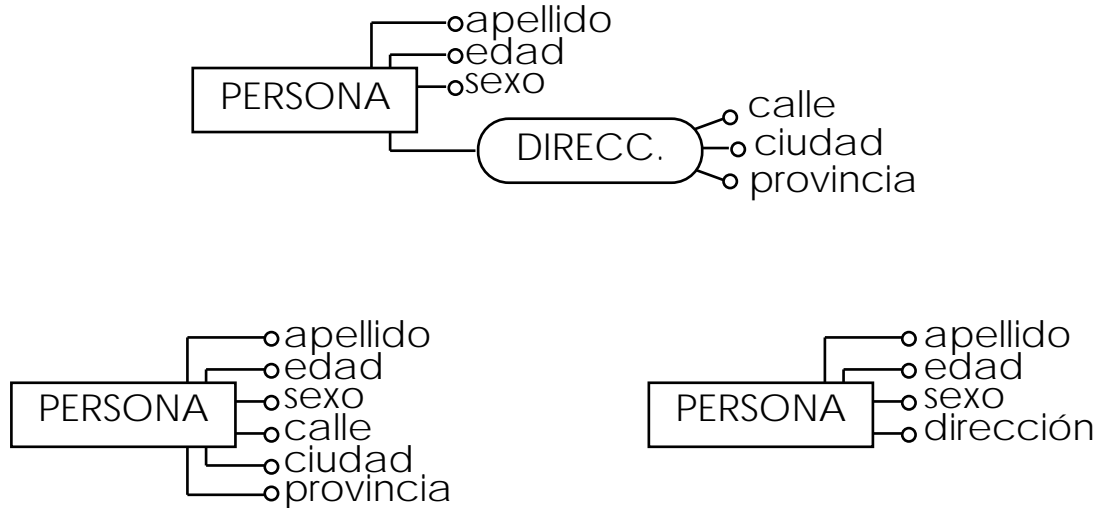
- El modelo relacional no permite representar atributos compuestos, por lo que se busca una alternativa.
- Transformar los atributos compuestos, como
 - Considerar el atributo compuesto como un atributo simple.
 - Eliminar el atributo compuesto y considerar cada uno de sus atributos como atributo simple de la entidad.
- En el primer caso no es posible descomponer el atributo en sus elementos, mientras que en el segundo se pierde su significado conjunto.

DISEÑO LÓG. EN EL MOD. RELACIONAL

Ejemplo de Transf. de Identificadores Externos



Ejemplo de Transf. de Atributos Compuestos



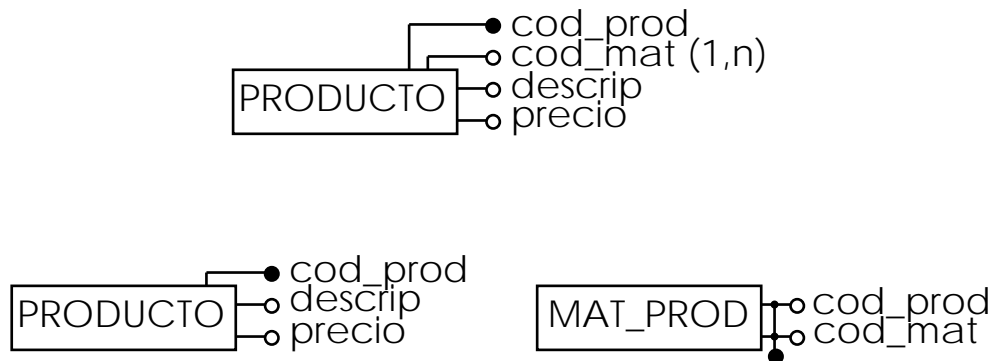
DISEÑO LÓG. EN EL MOD. RELACIONAL

Transf. de Atributos Polivalentes

- El modelo relacional no permite la gestión de atributos polivalentes, ya que sólo permite los atributos atómicos.
- Cada atributo polivalente de una entidad requiere la inclusión de una nueva entidad, donde se cumple que,
 - El atributo polivalente se convierte en un atributo monovalente.
 - La entidad hereda los atributos asociados al identificador de la entidad original.
 - El identificador de esta nueva entidad está compuesto por todos sus atributos.
- Si el atributo polivalente se asocia a una interrelación, también es necesario crear una nueva entidad, pero la herencia de atributos depende del tipo de interrelación,
 - Si fuera uno a uno hereda el identificador de una de las entidades.
 - Si fuera uno a muchos sólo hereda el identificador del lado "muchos".
 - Si fuera muchos a muchos hereda los dos identificadores.

DISEÑO LÓG. EN EL MOD. RELACIONAL

Ejemplo de Transf. de Atributos Polivalentes



DISEÑO LÓG. EN EL MOD. RELACIONAL

Transf. de Entidades

- Las entidades del esquema conceptual se transforman en relaciones base dentro del esquema lógico.
- Los atributos de las entidades se convierten en atributos de la relación base asociada.
- La clave primaria de la relación base es uno de los identificadores de la entidad, y siempre que sea posible un identificador simple.



EMPLEADO (NSS, empleado, apellido, salario)

Transf. de Interrelaciones

- Las interrelaciones deben ser transformadas, ya que no son soportadas por el modelo relacional.
- La transformación es diferente para cada tipo de interrelación, por lo que se consideran por separado.
- La cardinalidad de las entidades asociadas a la interrelación también influye en el modo de aplicar la transformación.

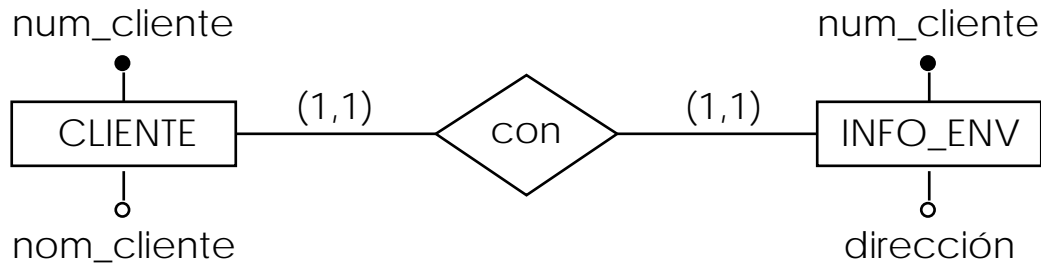
DISEÑO LÓG. EN EL MOD. RELACIONAL

Transf. de Interrelaciones Uno a Uno

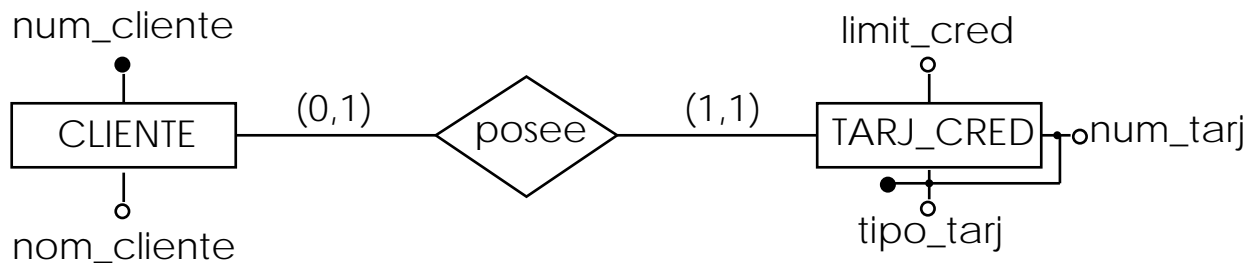
- Cuando la cardinalidad mínima de ambas entidades es igual a uno, las dos entidades y la interrelación se Integran en una Relación.
- La relación resultante presenta los atributos que no son clave primaria de las relaciones asociadas a las entidades.
- Por lo que respecta a la clave primaria, se pueden presentar dos circunstancias.
 - Si la clave primaria coincide en ambas relaciones se incluye una única vez en la nueva relación, y será su clave primaria.
 - En caso contrario, se incluyen las dos claves primarias y se elige una como clave primaria.
- Si alguna de las entidades tiene cardinalidad mínima igual a cero, la interrelación se transforma en una relación,
 - Hereda como atributos las claves primarias de las relaciones anteriores, y una de ellas será la clave primaria.
 - Aparecen como atributos, los atributos de la interrelación.
- Si sólo una de las entidades tuviera una cardinalidad igual a cero, también se podrían haber integrado las relaciones en una nueva relación.
- En ese caso, la clave primaria debería ser la asociada con la entidad con cardinalidad mínima igual a cero.

DISEÑO LÓG. EN EL MOD. RELACIONAL

Ejemplo de Transf. de Interrelac. Uno a Uno



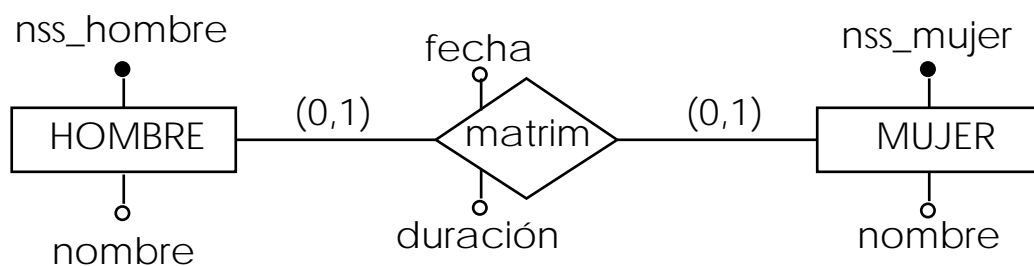
ENVIO_CLIENTE (num_cliente, nombre_cliente, dirección)



CLIENTE (num_cliente, nom_cliente)

TARJ_CRED (num_tarj, tipo_tarj, limit_cred)

POSEE_TARJ (num_tarj, tipo_tarj, num_cliente)



HOMBRE (nss_hombre, nombre)

MUJER (nss_mujer, nombre)

MATRIM (nss_hombre, nss_mujer, fecha, duración)

DISEÑO LÓG. EN EL MOD. RELACIONAL

Transf. de Interrelaciones Uno a Muchos

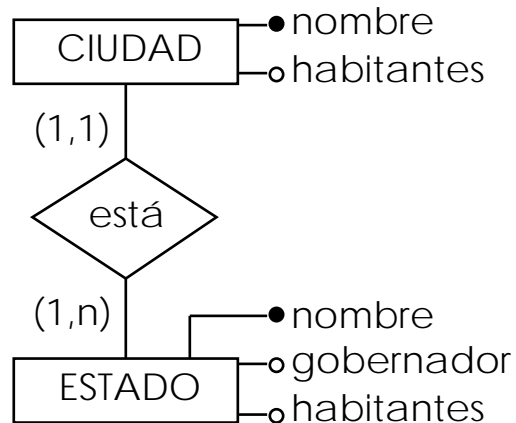
- Estas interrelaciones se pueden transformar de dos modos diferentes,
 - Añadiendo la clave primaria de la relación asociada a la entidad de la parte uno y los atributos de la interrelación en la relación asociada a la entidad de la parte muchos.
 - Generando una nueva relación que incluye los atributos de la interrelación y las claves primarias de las entidades. La clave primaria corresponde con la de la parte muchos.
- Cuando la cardinalidad mínima de la entidad muchos es igual a uno, se suele utilizar la primera opción.
- En cambio, si la cardinalidad mínima fuera cero la primera opción podría generar la aparición de valores nulos en las tuplas de la relación.
- Para evitar estos problemas se suele utilizar la segunda alternativa.

Transf. de Interrelaciones Muchos a Muchos

- En este caso la cardinalidad mínima no tiene ninguna influencia, y la transformación siempre se realiza del mismo modo.
- Se genera una nueva relación que incluye los atributos de la interrelación y las claves primarias de las dos entidades.
- La clave primaria depende de cada caso.

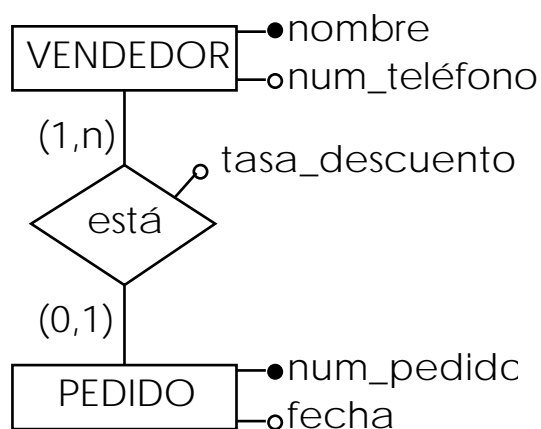
DISEÑO LÓG. EN EL MOD. RELACIONAL

Ejemplo de Transf. de Interr. Uno a Muchos



CIUDAD (nombre_ciudad, nombre_estado, habitantes)

ESTADO (nombre_estado, gobernador, habitantes)



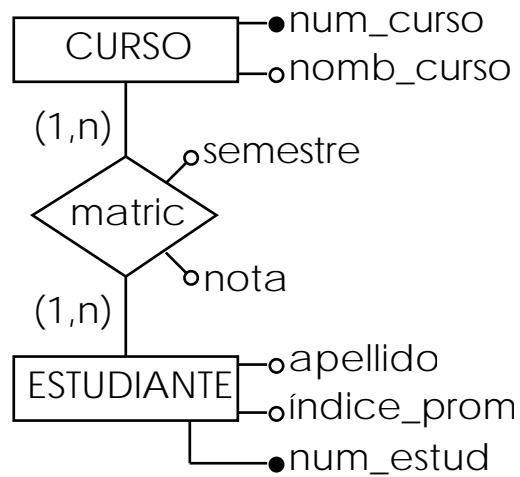
VENDEDOR (nombre_vend, num_teléfono)

PEDIDO (num_pedido, fecha)

PEDIDO_VENTAS (num_pedido, nombre_vend, tasa_descuento)

DISEÑO LÓG. EN EL MOD. RELACIONAL

Ejemplo de Transf. de Interr. Muchos a Muchos



CURSO (num_curso, nomb_curso)

ESTUDIANTE (num_estud, apellido, índice_prom)

MATRIC (num_curso, num_estud, semestre, nota)

DISEÑO LÓG. EN EL MOD. RELACIONAL

Transf. de Interrelaciones n-arias

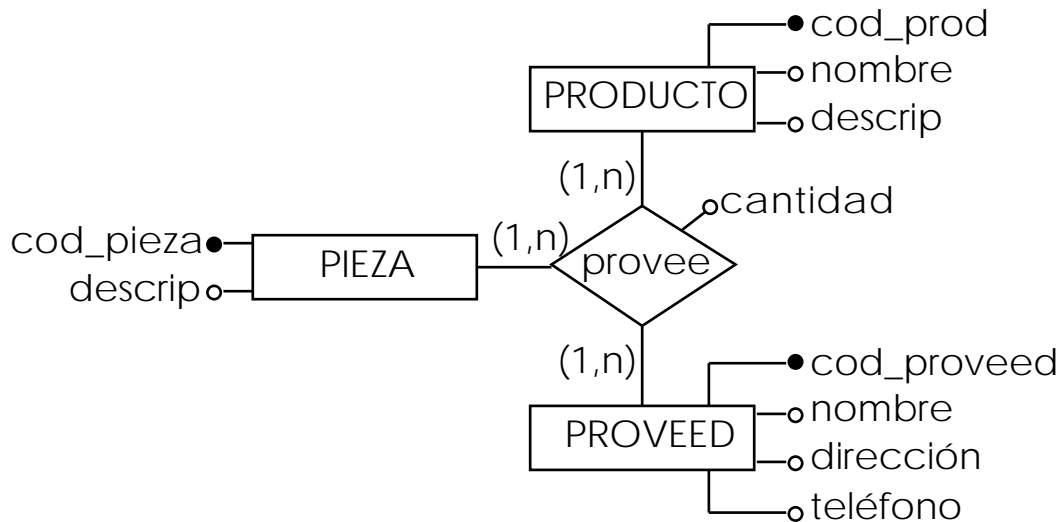
- Las interrelaciones n-arias se transforman de modo similar a las interrelaciones binarias de muchos a muchos.
- Según esto, se genera una única relación que hereda las claves primarias de las relaciones asociadas a las entidades que conecta.
- La clave primaria de la nueva relación es un subconjunto de la unión de las claves primarias heredadas y los atributos de la interrelación..
- Los atributos de la interrelación se incluyen como atributos de la relación resultante.

Transf. de Interrelaciones Recursivas

- Las transformaciones a aplicar son semejantes a las analizadas para las relaciones binarias.
- La única diferencia es que en este caso las dos relaciones asociadas son la misma.
- Por esta razón, cuando se debe crear una nueva relación, la clave primaria de la relación asociada se hereda dos veces, lógicamente, con nombre diferente.

DISEÑO LÓG. EN EL MOD. RELACIONAL

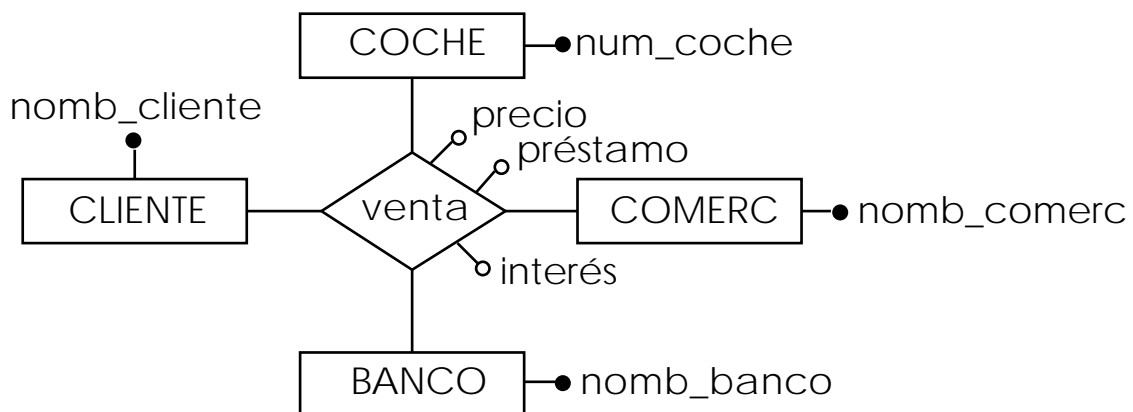
Ejemplo de Transf. de Interr. n-arias



PRODUCTO (cod_prod, nombre, descrip) PIEZA (cod_pieza, descrip)

PROVEED (cod_proveed, nombre, dirección, teléfono)

PROVEE (cod_prod, cod_pieza, cod_proveed, cantidad)



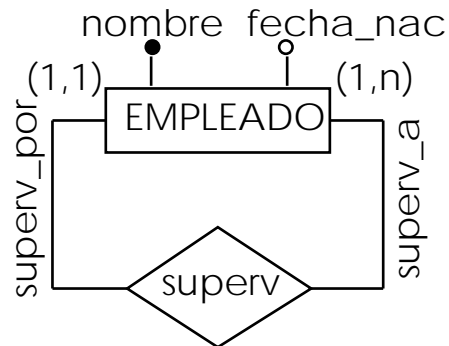
COCHE (num_coche) CLIENTE (nomb_cliente)

COMERC (nomb_comerc) BANCO (nomb_banco)

VENTA (num_coche, nomb_cliente, nomb_comerc, nomb_banco,
precio, préstamo, interés)

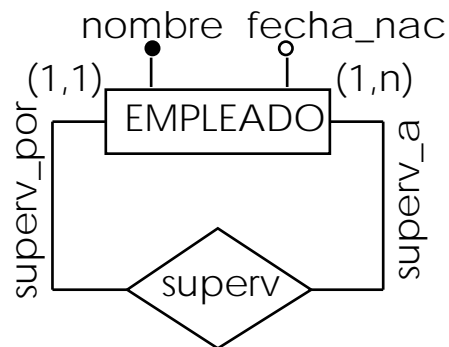
DISEÑO LÓG. EN EL MOD. RELACIONAL

Ejemplo de Transf. de Interr. Recursivas

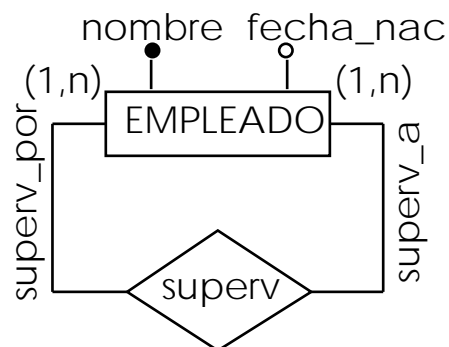


EMPLEADO (nombre, fecha_nac)

SUPERV_POR (nombre_supervisado, nombre_supervisor)



EMPLEADO (nombre, fecha_nac, nombre_supervisor)



EMPLEADO (nombre, fecha_nac)

SUPERV_POR (nombre_supervisado, nombre_supervisor)

NORMALIZACIÓN DE RELACIONES

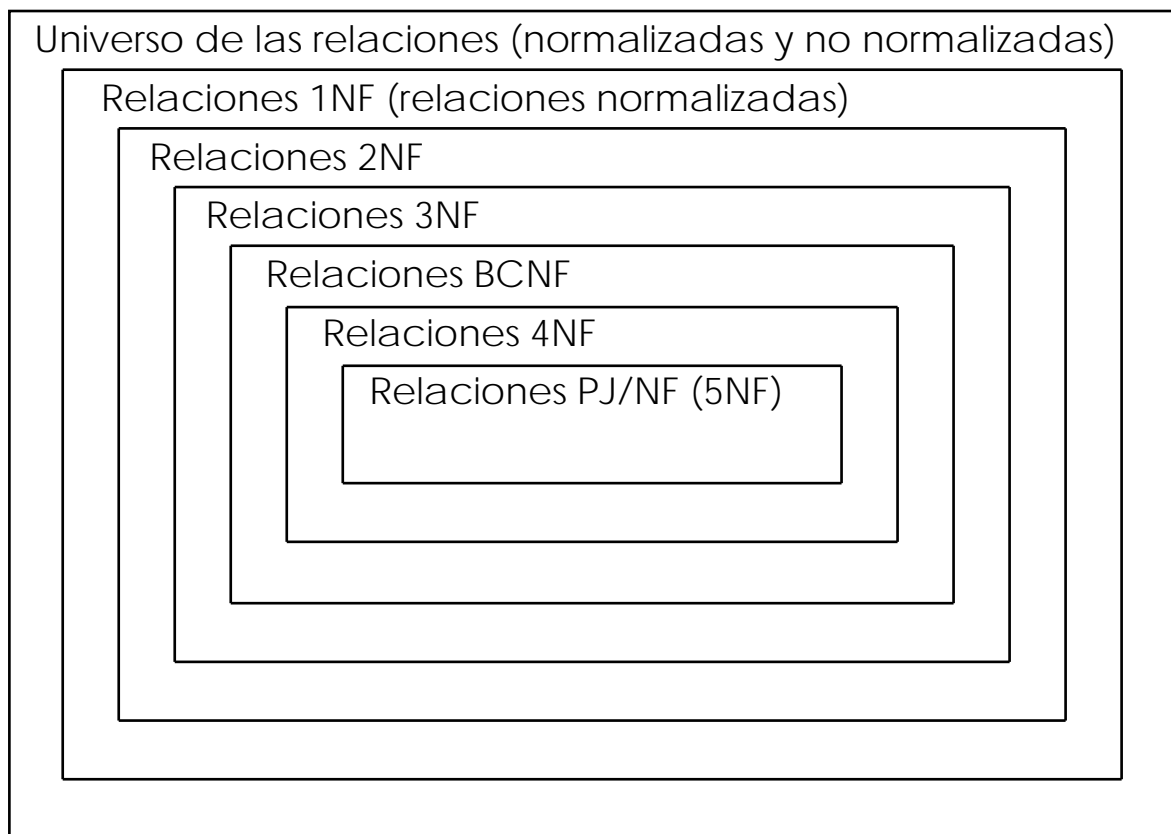
Comentarios Iniciales

- La teoría de la normalización, desarrollada por Codd en 1972, permite mejorar el diseño lógico de un sistema de información.
- Se fundamenta en las Formas Normales, que son un conjunto de restricciones que deben de cumplir las relaciones,
 - Una relación está en primera forma normal, si satisface que sus dominios simples sólo tienen valores atómicos.
- Por la definición anterior, se cumple que las relaciones de una base de datos siempre están normalizadas, pero pueden presentar algunas propiedades no deseables.
- Estas propiedades suelen asociarse con la dependencia funcional de los atributos de las relaciones.
- El objetivo de este proceso es convertir las relaciones de una base de datos a una forma más deseable.
- Entre las ventajas de la normalización se encuentran las siguientes,
 - Evita dependencias de datos en inserciones, borrados y modificaciones.
 - Mejora la independencia de los datos.
 - No establece restricciones artificiales en la estructura de los datos.

NORMALIZACIÓN DE RELACIONES

Formas Normales

- Inicialmente se definieron tres formas normales, aunque con posterioridad se han incluyendo más formas normales.
- Todas ellas normalizaciones se incluyen en una jerarquía, de modo que una relación que está en una forma normal, también está en la forma normal anterior.
- También se define un Procedimiento de Normaliación que permite pasar un conjunto de relaciones de una forma normal a una forma más deseable.
- El procedimiento es reversible, con lo cual se asegura que no existe pérdida de información en su aplicación.



NORMALIZACIÓN DE RELACIONES

Dependencia Funcional

- Dada una relación R, el atributo Y de R depende funcionalmente del atributo X de R,

$$R.X \rightarrow R.Y$$

si y sólo si un valor Y en R está asociado a cada valor X en R, donde tanto X como Y puede ser atributos compuestos.

- Si el atributo X es una clave candidata de R, entonces todos los atributos Y de la relación dependen funcionalmente de X, por la definición de clave candidata.
- Si el atributo Y es por completo dependiente funcionalmente de X, entonces no depende funcionalmente de ningún subconjunto propio de X .
- En los estudios posteriores se trabaja con la dependencia funcional completa, aunque no se especifique explícitamente.
- La dependencia funcional es un concepto semántico, ya que forma parte del proceso de entender el significado de los datos.
- También se puede entender como una clase especial de restricción de integridad.
- Y en cualquier caso define una relación muchos a uno entre los atributos.
- Las restricciones referenciales se pueden considerar como dependencias funcionales entre relaciones.

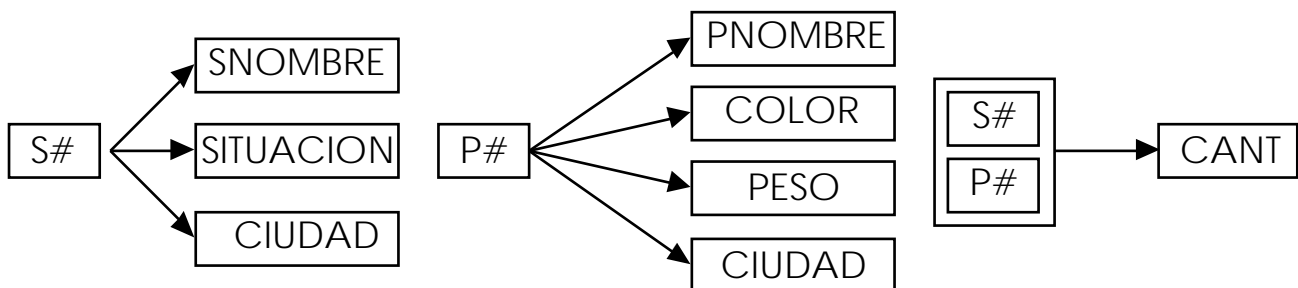
NORMALIZACIÓN DE RELACIONES

Representación de las Depend. Funcion.

- Las dependencias funcionales en una relación se representan mediante un diagrama de dependencias funcionales.
- El análisis de estos diagramas es fundamental en el proceso de normalización.
- Las dependencias funcionales que no se asocien con ninguna clave candidata pueden producir algún tipo de problemática.
- Por tanto, el objetivo de la normalización es eliminar estas dependencias funcionales.

Tercera Forma Normal

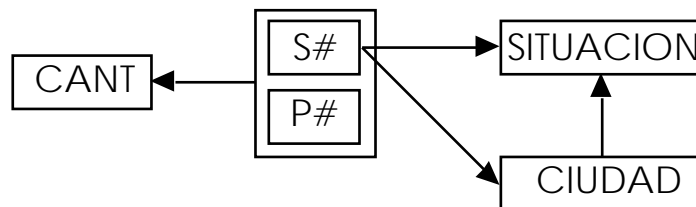
- El objetivo del diseño lógico es obtener relaciones en tercera forma normal.
- Una relación está en tercera forma normal si y sólo si sus atributos no clave son,
 - Mutuamente Independientes. No existe un atributo que dependa funcionalmente de alguna combinación del resto de atributos.
 - Por completo dependientes funcionalmente de la clave primaria.



NORMALIZACIÓN DE RELACIONES

Primera y Segunda Forma Normal

- Una relación se encuentra en primera forma normal si todos sus atributos son atómicos.
- En esta definición no se incluye ninguna restricción sobre la dependencia funcional de los atributos de la relación.
- Por lo tanto se pueden presentar diagramas funcionales como el siguiente,



- La relación representada en el diagrama,

PRIMERA (S#, P#, CANT, SITUACION, CIUDAD)

puede presentar redundancias que generan anomalías de actualización,

- Inserción. No es posible conocer la ciudad asociada a un proveedor hasta que éste no suministre un envío.
- Borrado. El borrado de los envíos de un proveedor borra de la base de datos la ciudad asociada.
- Actualización. La ciudad debe actualizarse en todos los envíos de un proveedor.
- Los problemas comentados se producen porque no todos los atributos dependen de la clave primaria.

NORMALIZACIÓN DE RELACIONES

Primera y Segunda Forma Normal

- Una relación se encuentra en segunda forma normal si está en primera forma normal y todos los atributos no clave dependen de la clave primaria.
- Para pasar de la primera a la segunda forma normal se aplica un proceso de reducción de la relación definido por una proyección.
- La operación inversa se realiza mediante la realización de una reunión.
- Desde un punto de vista formal la reducción se realiza como sigue,
 - Dada una relación como la siguiente,

$R(A, B, C, D)$, PRIMARY KEY (A, B) , $R.A \multimap R.D$

- Se obtienen las siguientes relaciones,

$R_1(A, D)$, PRIMARY KEY (A)

$R_2(A, B, C)$, PRIMARY KEY (A, B) , FOREIGN KEY (A) REFERENCES R1

- Aplicando el proceso de reducción sobre la relación PRIMERA se obtendrían las relaciones SP y SEGUNDA,

$SP(\underline{S\#}, \underline{P\#}, CANT)$, $SP \multimap SEGUNDA$
 $SEGUNDA(\underline{S\#}, CIUDAD, SITUACION)$

con los siguientes diagramas funcionales,



NORMALIZACIÓN DE RELACIONES

Segunda y Tercera Forma Normal

- La relación SEGUNDA todavía presenta algún problema que deben de resolverse.
- Estos problemas se fundamentan en la falta de independencia de sus atributos no clave.
- Dicha falta de independencia produce la aparición de dependencias funcionales de tipo transitivo,

$R.A \rightarrow R.B$, $R.B \rightarrow R.C$, $R.A \rightarrow R.C$

- Las dependencias transitivas también generan redundancias, y por tanto, anomalías de actualización,
 - Inserción. Hasta que no exista un proveedor de una ciudad no sabemos su situación.
 - Borrado. Al borrar los proveedores de una ciudad borramos de la base de datos su situación.
 - Actualización. El valor de situación se debe actualizar en todos los proveedores de una ciudad.
- Todos estos problemas se resolverán mediante la transformación de esta relación a la tercera forma normal.

NORMALIZACIÓN DE RELACIONES

Segunda y Tercera Forma Normal

- Una relación está en tercera forma normal, si está en segunda forma normal y además cada atributo no clave no depende de modo transitivo de la clave primaria.
- La eliminación de las dependencias transitivas también se apoya en la aplicación de una proyección de la relación asociada.
- El operador reunión permite reconstruir la relación original.
- Desde un punto de vista formal, la reducción sigue el siguiente planteamiento,

- Dada la relación R,

$R(A, B, C)$, PRIMARY KEY (A) , $R.B \dashv\dashv\dashv R.C$

- Se proyecta para obtener,

$R_1(B, C)$, PRIMARY KEY (B)

$R_2(A, B)$, PRIMARY KEY (A) , FOREIGN KEY (B) REFERENCES R1

- La aplicación de esta descomposición sobre SEGUNDA permite obtener las relaciones,

$SC(\underline{S\#}, CIUDAD)$, $SC \dashv\dashv\dashv CS$

$CS(\underline{CIUDAD}, SITUACION)$

cuyo diagrama funcional se muestra a continuación,



NORMALIZACIÓN DE RELACIONES

Buenas y Malas Descomposiciones

- Normalmente, el proceso de reducción puede realizarse de diferentes maneras, debiéndose elegir siempre la más adecuada.
- Una buena descomposición es aquella que genera relaciones independientes.
- Rissanen definió en 1977 la independencia entre dos proyecciones de una relación.
- Dos proyecciones, R1 y R2, de la relación R son independientes si y sólo si cumplen que,
 - Toda dependencia funcional en R se puede deducir de las dependencias funcionales definidas en R1 y R2.
 - Los atributos comunes de R1 y R2 son una clave candidata de al menos una de las dos proyecciones.
- En el ejemplo anterior se ha elegido una buena descomposición, pero se podían haber elegido otras alternativas,
 - Una mala descomposición por que una de las dependencias funcionales se convierte en una restricción interrelacional.

SC (S#, CIUDAD)
SS (S#, SITUACION)

- Una mala descomposición por producir una pérdida de información.

SS (S#, SITUACION)
CS (CIUDAD, SITUACION)