



DEPENDENCIAS FUNCIONALES Y NORMALIZACIÓN EN BD RELACIONALES

INTRODUCCIÓN

Hasta aquí se ha supuesto que los atributos se agrupan para formar un esquema de relación, empleando el sentido común del diseñador de base de datos o estableciendo una transformación de un esquema especificado en el modelo entidad-relación (ER) o en el modelo entidad-relación extendido (ERE); en un esquema relacional.

El modelo ER hace que el diseñador identifique tipos de entidad, sus atributos respectivos y tipos de relación; lo que da lugar a una agrupación natural y lógica de los atributos en relaciones cuando se siguen los procedimientos de transformación.

Sin embargo es preciso una medida formal de porqué una agrupación de atributos para formar un esquema de relación puede ser mejor que otra. Hasta ahora en el diseño conceptual y su transformación al modelo relacional, no se ha desarrollado ninguna medida de lo apropiado del diseño o de su calidad, que no fuera la mera intuición del diseñador.

En esta unidad se va a estudiar la teoría que se ha desarrollado para elegir buenos esquemas de relación, es decir, para medir formalmente las razones por las que una agrupación de atributos en esquemas de relación, es mejor que otra.

Existen dos niveles en los que se puede evaluar la bondad de los esquemas de relación:

- **Nivel lógico (o conceptual):** se refiere a la manera en que los usuarios interpretan los esquemas de relación y al significado de sus atributos. Esto ayuda a formular consultas correctamente.
- **Nivel de implementación (o de almacenamiento):** se refiere a como se almacenan y actualizan las tuplas de una relación base.

También se va a estudiar el concepto de Dependencia Funcional, que es la principal herramienta para medir formalmente la idoneidad de las agrupaciones de atributos en los esquemas de relación. Las dependencias funcionales se utilizan para agrupar atributos en esquemas de relación que estén en una forma normal. Un esquema de relación está en una forma normal cuando posee ciertas características deseables.

El proceso de Normalización consiste en analizar si las relaciones satisfacen una serie de requisitos cada vez más estrictos, que dan lugar a agrupamientos que van siendo cada vez mejores, o a formas normales superiores.

INTRODUCCION

El problema en el diseño de la Base de Datos (B.D.) consiste en elegir una estructura lógica conveniente para los mismos.

Los atributos se agrupan para formar un esquema de relación. Para ello se establece una transformación de un esquema especificado en el modelo Entidad-Relación (ER) en un esquema Relacional.

Sin embargo, es preciso una medida formal de porqué una agrupación de atributos para formar un esquema de relación, puede ser mejor que otro.



PAUTAS INFORMALES DE DISEÑO PARA LOS ESQUEMAS DE RELACIÓN

Existen cuatro medidas informales de calidad para el diseño de esquemas de relación:

- Semántica de los atributos de relación.
- Reducción de los valores redundantes en las tuplas.
- Reducción de los valores nulos en las tuplas.
- Eliminación de la posibilidad de tuplas espurias.

Semántica de los atributos de relación

Siempre que se agrupen atributos para formar un esquema de relación, se supone que existe un cierto significado asociado a los atributos.

Este significado o semántica, especifica como se han de interpretar los valores de los atributos almacenados en una tupla de relación. En otras palabras, qué relación existe entre los valores de los atributos de una tupla. En general, cuánto más fácil sea explicar la semántica de una relación, mejor será el diseño del esquema correspondiente.

Se debe diseñar un esquema de relación de modo que sea fácil de explicar su significado, teniendo en cuenta lo siguiente:

- No se deben combinar atributos de varios tipos de entidad, ni tipos de relación en una única relación.
- Si un esquema de relación corresponde a un tipo de entidad o a un tipo de relación, el significado tiende a ser claro.
- En caso contrario la relación tiende a ser una mezcla de múltiples entidades y relaciones; por lo que será semánticamente confusa.

Información redundante en las tuplas y anomalías de actualización

Uno de los objetivos en el diseño de esquemas es minimizar el espacio de almacenamiento que ocupan las relaciones base.

La agrupación de atributos en esquemas de relación, tiene un efecto significativo sobre el espacio de almacenamiento. El problema que surge al emplear las relaciones como relaciones base, es el de las anomalías de actualización (inserción, eliminación y modificación).

Se deben diseñar los esquemas de las relaciones base de modo que no existan anomalías de inserción, eliminación o modificación en las relaciones. Si existieran anomalías, se deben señalar con claridad, a fin de que los programas que actualicen la base de datos operen correctamente.

Valores nulos en las tuplas

Si muchos de los atributos no se aplican a las tuplas de la relación, se puede originar un considerable desperdicio de espacio en el nivel de almacenamiento y dificultar el entendimiento del significado de los atributos. Otro problema con los valores nulos es como manejarlos cuando se aplican funciones agregadas como COUNT o SUM.

Los valores nulos pueden tener múltiples interpretaciones como las siguientes:

- El atributo no se aplica a la tupla.
- Se desconoce el valor del atributo para la tupla.
- El valor del atributo se conoce pero está ausente, todavía no se ha registrado.

Siempre que sea posible, se debe evitar incluir en una relación base atributos cuyos valores puedan ser nulos. Si no es posible evitar los nulos, se debe asegurar de que se apliquen en casos excepcionales y no en la mayoría de las tuplas de una relación

Generación de tuplas espurias

Se deben diseñar los esquemas de relación de modo que puedan JUNTARSE mediante condiciones de igualdad sobre atributos que sean claves primarias y claves externas. Para garantizar que no se formarán tuplas espurias, se debe tener en cuenta lo siguiente:

- No se deben incluir relaciones que contengan atributos coincidentes que no sean combinaciones de clave primaria y clave externa.

- Si no se pueden evitar este tipo de relaciones, no se deben **JUNTAR** con dichos atributos porque la **JUNTA** puede dar lugar a tuplas espurias.

INTRODUCCION

Para ello existen dos herramientas importantes:

- **DEPENDENCIAS FUNCIONALES**
- **NORMALIZACION**

Contar con buenos esquemas de relación, ayuda a los usuarios a comprender con claridad el significado de los datos en las relaciones y por lo tanto a formular sus consultas correctamente.

DEPENDENCIAS FUNCIONALES

Las Dependencias Funcionales proporcionan la base para un tratamiento científico de diversos problemas prácticos, debido a que poseen un conjunto de propiedades formales que hacen posible tratar los problemas en cuestión de una manera formal rigurosa

1. DEPENDENCIAS FUNCIONALES

Una Dependencia Funcional (DF) es un vínculo muchos a uno que va de un conjunto de atributos a otro, dentro de un determinado esquema de relación.

Dado un esquema de relación **R**,

la DF: **A** \longrightarrow **B**

es válida para **R**, si y sólo si, siempre que 2 o más tuplas de **R** coinciden en su valor **A**, tienen también el mismo valor de **B**.

A y **B** son conjuntos del conjunto de atributos de **R**. Se dice que **A** determina funcionalmente a **B** o bien, que **B** es dependiente funcionalmente de **A**.

Ejemplo: teniendo en cuenta el diseño lógico de la Base de Datos de Proveedores y Partes, con fines didácticos se hará una modificación del esquema de relación **ENVIOS (VP)**, su nombre será **VPC** y mostrará los siguientes posibles valores:



VPC			
V#	Ciudad	P#	Cant
V1	Londres	P1	300
V1	Londres	P2	200
V1	Londres	P3	400
V1	Londres	P4	200
V1	Londres	P5	100
V1	Londres	P6	100
V2	París	P1	300
V2	París	P2	400
V3	París	P2	200
V4	Londres	P2	200
V4	Londres	P4	300
V4	Londres	P5	400

Dependencias Funcionales

existe una dependencia funcional en el conjunto de atributos $\{V\#, P\#\}$ (determinante) y el conjunto de atributos $\{CANT\}$ (dependiente). Esto significa lo siguiente:

- Para cualquier valor dado del par de atributos $V\#$ y $P\#$, sólo existe un valor correspondiente del atributo $CANT$, pero:
- Muchos valores distintos del par de atributos $V\#$ y $P\#$, pueden tener el mismo valor del atributo $CANT$.

Los valores incorporados en el esquema de relación VPC, satisfacen estas 2 propiedades, la Dependencia Funcional:

$$\{V\#\} \longrightarrow \{CIUDAD\}$$

también es satisfecha.

Tipos de Dependencias Funcionales

- **Completa:** B tiene dependencia funcional completa de A, si depende de A pero no de sus subconjuntos.
- **Trivial:** B tiene dependencia funcional trivial de A, si: $B \subset A$. No existe posibilidad de no ser satisfecha.
- **Elemental:** B tiene dependencia funcional elemental de A, si no es trivial y además posee un único atributo.

Dependencias Funcionales

1.1. TIPOS DE DFs

Dada la siguiente DF: $A \longrightarrow B$, se dice que la misma puede ser:

DF Completa: $\{V\#\} \longrightarrow \{CIUDAD\}$

DF Trivial: $\{V\#, P\#\} \longrightarrow \{V\#\}$

DF Elemental: $\{V\#, P\#\} \longrightarrow \{CANT\}$

Antes de definir los tipos transitivos de dependencia funcional, se debe tener en cuenta el concepto de descriptores equivalentes:

$$A \longrightarrow B \wedge B \longrightarrow A \equiv A \longleftrightarrow B$$

en este caso cuando el SGBD hace cumplir la dependencia funcional de **B**, también hará cumplir automáticamente la de **A**.

- **Transitiva:** dados 3 descriptores **A**, **B** y **C** del esquema de relación **R**; se dice que existe dependencia transitiva de **C** respecto de **A**, a través de **B**, si se cumple:

$$A \longrightarrow B \wedge B \longrightarrow C$$

A y **B** no deben ser descriptores equivalentes.

- **Transitiva estricta:** dados 3 descriptores **A**, **B** y **C** del esquema de relación **R**; se dice que existe dependencia transitiva estricta de **C** respecto de **A**, a través de **B**, si se cumple:

$$A \longrightarrow B \wedge B \longrightarrow C$$

No deben ser equivalentes **A** y **B**, ni **C** y **B**

Dependencias Funcionales

1.1. TIPOS DE DFs

DF Transitiva: $\{V\# \} \longrightarrow \{CIUDAD\}$
 $\{CIUDAD\} \longrightarrow \{CANT\}$
{V#} y {CIUDAD} no deben ser descriptores equivalentes, es decir: $\{V\# \} \longrightarrow \{CIUDAD\}$.
 Todos los proveedores de una ciudad, proveen siempre la misma cantidad y ésta NO podrá ser la misma en otra ciudad.

DF Transitiva estricta: $\{V\# \} \longrightarrow \{CIUDAD\}$
 $\{CIUDAD\} \longrightarrow \{CANT\}$
**No siendo equivalentes {V#} y {CIUDAD} ni {CANT} y {CIUDAD}, es decir: $\{V\# \} \longrightarrow \{CIUDAD\}$
 $\{CANT\} \longleftarrow \{CIUDAD\}$.**
 Todos los proveedores de una ciudad, proveen simple la misma cantidad y esta podrá ser la misma en otra ciudad.

Dependencias Funcionales

1.2. DF Irreducible

Sea S un conjunto de DF, será irreducible si y sólo si se satisfacen las siguientes 3 propiedades:

- La parte derecha de toda DF en **S** involucra un solo atributo.
- La parte izquierda de toda DF en **S** es a su vez irreducible (DF irreducible a izquierda).
- No es posible descartar de **S** ninguna DF, sin pérdida de información.

Los siguientes ejemplos son válidos para el esquema de relación P (Partes):

la DF: $\{P\# \} \longrightarrow \{PARTE\}$ $\{P\# \} \longrightarrow \{COLOR\}$
 $\{P\# \} \longrightarrow \{CIUDAD\}$ $\{P\# \} \longrightarrow \{PESO\}$

FORMAS NORMALES

Una vez estudiadas las dependencias funcionales y algunas de sus propiedades, se las puede utilizar como información sobre la semántica de los esquemas de relación.

Se supone que se da un conjunto de dependencias funcionales para cada relación y que cada relación tiene una clave primaria designada. Esta información combinada con los tests (condiciones) para las formas normales, conduce al proceso de normalización

2. NORMALIZACION

En cada esquema de relación se dan un conjunto de DF y una designación de clave 1º. Esta información combinada con los tests (condiciones) para las Formas Normales (FN), conduce al proceso de Normalización.

El proceso sigue un estilo descendente y evalúa cada relación respecto de los criterios de normas formales. Las relaciones se descomponen en la medida que sea necesario.

Introducción

En el proceso de normalización según la propuesta original de Codd (1972), se somete a un esquema de relación, a una serie de pruebas para certificar si pertenece o no, a una cierta forma normal.

El proceso sigue un estilo descendente, evaluando cada relación respecto a los criterios de normas formales y descomponiendo (**PROYECTAR**) las relaciones a medida que sea necesario. A este proceso se lo denomina diseño relacional por análisis.

La normalización de los datos puede considerarse como un proceso de análisis de los esquemas de relación basados en sus dependencias funcionales y claves primarias, para alcanzar las propiedades deseables de:

- Minimizar la redundancia.
- Minimizar las anomalías de inserción, eliminación y actualización.

Los esquemas de relación insatisfactorios que no cumplan determinadas condiciones de pruebas de formas normales, se descomponen (**PROYECTAR**) en esquemas de relación más pequeños que satisfagan dichas pruebas. Por lo que el procedimiento de normalización proporciona a los diseñadores los siguientes aspectos:

- Un marco formal para analizar los esquemas de relación basándose en sus claves y en las dependencias funcionales entre sus atributos.
- Una serie de pruebas normales que pueden efectuarse sobre esquemas de relación individuales, de modo que la base de datos relacional pueda normalizarse hasta el grado deseado.

El proceso de normalización por descomposición (**PROYECCIÓN**) debe confirmar también la existencia de propiedades adicionales, que los esquemas relacionales en conjunto deben poseer:

- Propiedad de reunión (**JUNTA**) sin pérdida o reunión (**JUNTA**) no aditiva, que garantiza que no se presentará el problema de las tuplas espurias, respecto a los esquemas de relación tras la descomposición (**PROYECCIÓN**).

- Propiedad de conservación de las dependencias, que asegura que todas las dependencias funcionales estén representadas en algunas de las relaciones individuales resultantes tras la descomposición (PROYECCIÓN).

Descomposición sin pérdida

Antes de definir este concepto, consideremos el siguiente esquema de relación de Proveedores (**V**):

V		
V#	Status	Ciudad
v3	30	París
v5	30	Atenas

sobre la relación de Proveedores (**V**) se harán dos proyecciones distintas (**a** y **b**):

a)

VST	
V#	Status
v3	30
v5	30

VC	
V#	Ciudad
v3	París
v5	Atenas

b)

VST	
V#	Status
v3	30
v5	30

VC	
Status	Ciudad
30	París
30	Atenas

al observar las dos descomposiciones (**a** y **b**) se observa que:

- En el caso **a**) no se pierde información. Los valores de los esquemas de relación de **VST** y **VC**, aún nos indican que un proveedor tiene un status dado y es de una ciudad dada.
- En el caso **b**) hay pérdida de información, aún se puede saber que un proveedor tiene un status dado, pero no se puede saber nada acerca de la ciudad del proveedor.

NORMALIZACIÓN

2.1. Descomposición sin Pérdida

El procedimiento de Descomposición implica:

- Separar una esquema de relación en otros.
- Reversibilidad del proceso (sin pérdida de información).

La cuestión de si una descomposición dada es sin pérdida, está íntimamente ligada con el concepto de la DF.

¿Porqué la 1º descomposición es sin pérdida y la 2º con pérdida?

- La descomposición es en realidad una proyección.
- Al aplicar la operación "junta", si se obtiene el esquema de relación original, NO hay pérdida.

NORMALIZACIÓN

2.1. Descomposición sin Pérdida

En otras palabras: la reversibilidad significa que el esquema de relación original, es igual a la junta de sus proyecciones.

Considerar: si R1 y R2 son proyecciones de cierto esquema de relación R, y R1 y R2 incluyen entre sí, a todos los atributos de R:

¿Qué condiciones se deben satisfacer para que al juntar R1 y R2, se obtenga la R original?

Aquí es donde entran a tallar las DF. Observar que el esquema de relación Proveedores (V) satisface el conjunto irreducible de DFs:

$\{V\# \} \longrightarrow \{STATUS\}$

$\{V\# \} \longrightarrow \{CIUDAD\}$

Debido al hecho que satisface estas dependencias funcionales, seguramente no puede ser una coincidencia que el esquema de relación V sea igual a la junta de sus proyecciones sobre:

$\{V\# \} \longrightarrow \{STATUS\} \wedge \{V\# \} \longrightarrow \{CIUDAD\}$

NORMALIZACIÓN

2.1. Descomposición sin Pérdida

Teorema de Heath:

Sea R{A, B, C} un esquema de relación (donde A, B y C son conjuntos de atributos).

Si R satisface la DF: $\{A\} \longrightarrow \{B\}$

entonces R es igual a la junta de sus proyecciones sobre $\{A,B\}$ y $\{A,C\}$

Recordemos que: $\{V\# \} \longrightarrow \{STATUS\}$

$\{V\# \} \longrightarrow \{CIUDAD\}$

son a la vez DF irreducibles e irreducibles a la izquierda. Conceptos importantes en la definición de la 2FN y 3FN.

NORMALIZACIÓN

2.1. Descomposición sin Pérdida

Las DFs son una noción Semántica:
las DFs son una clase especial de restricción de integridad y como tales, son una noción semántica. Por ejemplo, el hecho de que V satisfaga la DF: $\{V\} \longrightarrow \{CIUDAD\}$, significa:

- Existe una restricción que la B.D. representa: “cada proveedor está ubicado en una ciudad”
- Esta restricción debe ser observada de alguna manera en la B.D.
- La restricción debe ser especificada dentro de la definición de la B.D., para que el DBMS la haga cumplir.
- La forma de especificarla es mediante la declaración de la DF.

Importante: por razones de simplicidad se dará por hecho que cada esquema de relación, tiene exactamente una clave candidata, la cual es la clave primaria.

Atributo NO clave: es cualquier atributo que no participa en la clave primaria del esquema e relación respectivo.

Atributos mutuamente independientes: dos o más atributos son mutuamente independientes, si ninguno de ellos es dependiente funcionalmente de cualquier combinación de los otros. Esto implica que cada uno de dichos atributos, puede ser actualizado independientemente del resto.

Primera Forma Normal (1FN)

La 1FN se considera ahora parte de la definición formal de relación en el modelo relacional básico. Se definió para prohibir los atributos multivaluados, los atributos compuestos y sus combinaciones.

Establece que el dominio de un atributo debe incluir sólo valores atómicos (simples, indivisibles) y que el valor de cualquier atributo en una tupla debe ser un valor individual proveniente del dominio de ese atributo.

De esta forma la 1FN prohíbe las relaciones dentro de relaciones o las relaciones como atributos de tuplas. Por lo que los únicos valores de atributos que permite la 1FN, son los valores atómicos o indivisibles.

NORMALIZACIÓN

2.2. Primera Forma Normal (1FN)

Un esquema de relación está en 1FN, si y sólo si, en cada valor válido de ese esquema de relación, cada tupla contiene exactamente un valor para cada atributo.

De un modo más técnico: si todos los dominios subyacentes contienen valores atómicos.

Sin embargo un esquema de relación que solamente está en 1FN, tiene una estructura que es indeseable por diversas razones.

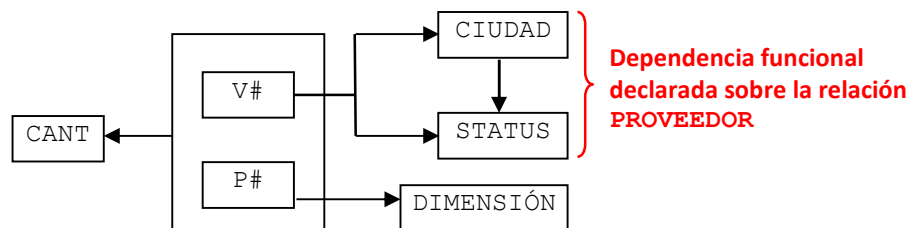
Ahora se ilustrará con un ejemplo los conceptos anteriores, dada la siguiente relación con su esquema correspondiente:

PROVEEDOR{V#, CIUDAD, STATUS, P#, DIMENSIÓN, CANT}
Primary Key{V#, P#}

y además se declara la siguiente dependencia funcional:

DF: CIUDAD → STATUS (“todos los proveedores de la misma ciudad tienen el mismo status”).

el diagrama de dependencias funcionales de PROVEEDOR es el siguiente:



el atributo DIMENSIÓN está multivaluado, debido a que una parte puede tener muchas dimensiones. Esto conduce a afirmar que la relación PROVEEDOR no sea considerada como relación en el Modelo Relacional debido a que no cumple la 1FN.

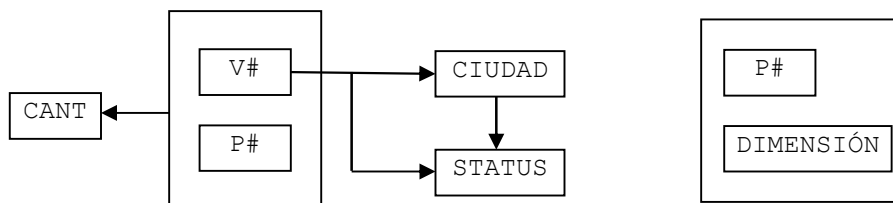
Solución del problema: se realiza una PROYECCIÓN de la relación PROVEEDOR en dos relaciones:

PRIMERA{V#, CIUDAD, STATUS, P#, CANT}
Primary Key{V#, P#}

y

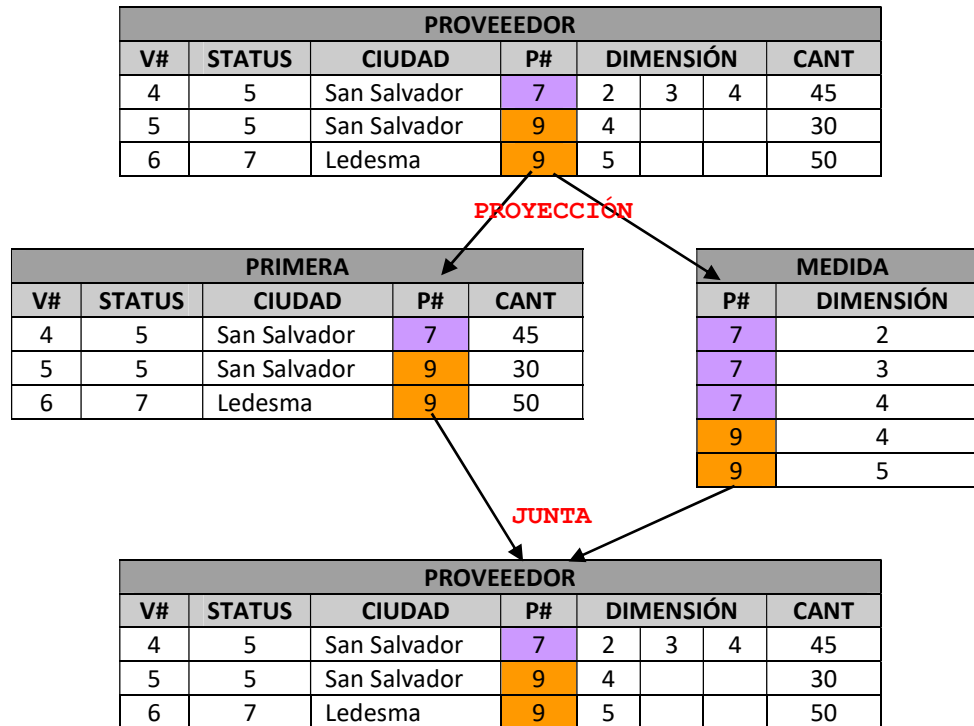
MEDIDA{P#, DIMENSIÓN}
Primary Key{P#, DIMENSIÓN}
Foreign Key{P#} references PRIMERA

Los diagramas de dependencias funcionales para las relaciones PRIMERA y MEDIDA son los siguientes:



Esta estructura modificada supera el problema del atributo multivaluado DIMENSIÓN al ubicarlo en la relación MEDIDA.

Las relaciones PRIMERA y MEDIDA son PROYECCIONES de PROVEEDOR y ésta relación se forma por la JUNTA de PRIMERA y MEDIDA a través de su atributo común P#. A continuación se le darán valores a los esquemas de relación:



Segunda Forma Normal (2FN)

Un esquema de relación está en 2FN, si y sólo si está en 1FN y todo atributo que no sea clave es dependiente irreduciblemente (dependencia funcional total) de la clave primaria.

Un esquema de relación que está en 1FN y no lo está en 2FN, siempre puede reducirse a un a una colección equivalente de esquemas de relación 2FN. El proceso de reducción consiste en reemplazar el esquema de relación 1FN por proyecciones convenientes. Siempre debe ser posible recuperar el esquema de relación original, juntando dichas proyecciones.

NORMALIZACIÓN

2.3. Segunda Forma Normal (2FN)

Un esquema de relación está en 2FN, si y sólo si, está en 1FN y todo atributo que no sea clave es dependiente irreduciblemente de la clave primaria.

Un esquema de relación que está en 1FN y no en 2FN, siempre puede reducirse a una colección equivalente de esquemas de relación 2FN.

El proceso de reducción consiste en reemplazar el esquema de relación 1FN por proyecciones convenientes. Siempre debe ser posible recuperar el esquema de relación original juntando dichas proyecciones.

La 2FN se basa en el concepto de dependencia funcional total. Dada la dependencia funcional:

$$A \longrightarrow B$$

es una dependencia funcional total, si la eliminación de un atributo del conjunto de atributos de A, hace que la dependencia funcional deje de ser válida

Ahora continuaremos con el ejemplo anterior, la relación MEDIDA no presenta problemas al estar en 2FN y en 3FN. Pero en la relación PRIMERA los atributos STATUS y CIUDAD no presenta una dependencia funcional total de la clave primaria {V#, P#}, por lo que dicho esquema de relación no está en 2FN. Recordando la definición de PRIMERA y su dependencia funcional:

```
PRIMERA{V#, STATUS, CIUDAD, P#, CANT}
      Primary Key{V#, P#}
DF: CIUDAD → STATUS
```

en el diagrama de dependencias funcionales de este esquema de relación, se observa que los atributos que no presentan una dependencia funcional total de su clave primaria (STATUS y CIUDAD), presentarán redundancias por cada envío del mismo proveedor. Estas redundancias conducen a una variedad de anomalías de actualización (inserción, borrado y modificación):

- **Inserción:** no se puede insertar un proveedor que está ubicado en una ciudad que tiene un status, hasta que ese proveedor suministre por lo menos una parte.
- **Borrado:** si se elimina la única tupla de un proveedor en particular, no sólo se elimina el envío que conecta al proveedor con una parte, sino que también se eliminan los datos propios del proveedor.
- **Modificación:** si a un proveedor se le cambia la ciudad en una tupla, se puede producir una inconsistencia debido a que el proveedor podrá tener para la misma ciudad dos status diferentes.

Solución del problema: se realiza una PROYECCIÓN de la relación PRIMERA en dos relaciones:

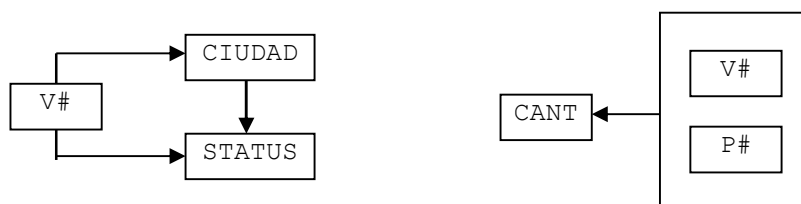
```
SEGUNDA{V#, STATUS, CIUDAD}
      Primary Key{V#}

y

VP{V#, P#, CANT}
      Primary Key{V#, P#}
      Foreign Key{V#} references SEGUNDA
```

El efecto de esta descomposición es ubicar los atributos CIUDAD y STATUS en otra relación donde tengan dependencia funcional total de la clave primaria {V#}, ya que en la relación PRIMERA tenían dependencia funcional parcial de la clave primaria {V#, P#}. El atributo CANT depende funcionalmente de forma total de la clave primaria {V#, P#}, debido a que en el envío de una cantidad se debe saber “quién” la envía y “qué” envía.

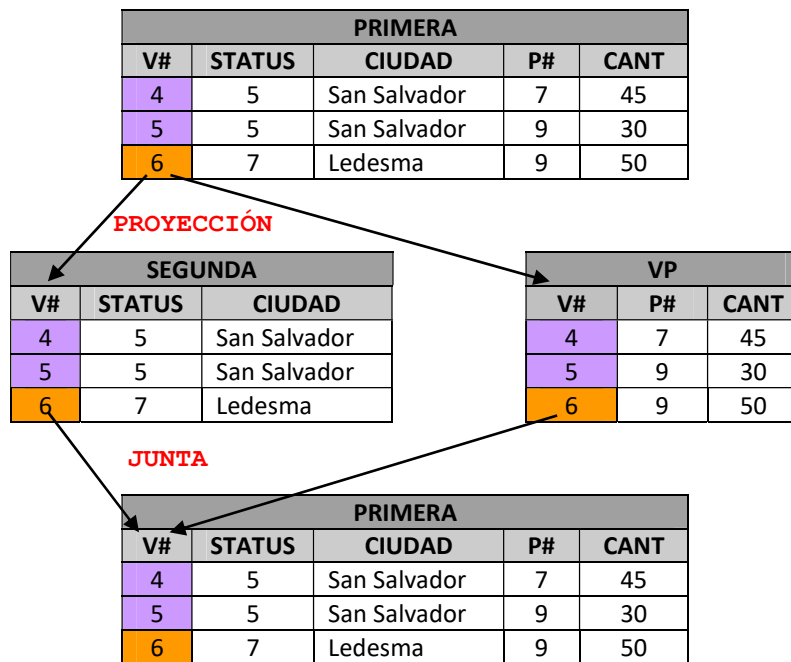
Los diagramas de dependencias funcionales para las relaciones SEGUNDA y VP son los siguientes:



Esta estructura modificada supera todos los problemas de anomalías de actualización de la estructura anterior (inserción, borrado y modificación):

- **Inserción:** ahora se puede insertar un proveedor que no haya suministrado envíos.
- **Borrado:** se puede eliminar un envío de un proveedor sin eliminar los datos personales del proveedor.
- **Modificación:** en esta estructura cada proveedor aparece una sola vez, por lo que si cambia de ciudad no habrá problemas.

Las relaciones SEGUNDA y VP son PROYECCIONES de PRIMERA y ésta relación se forma por la JUNTA de SEGUNDA y VP a través de su atributo común V#. A continuación se le darán valores a los esquemas de relación:



Tercera Forma Normal (3FN)

Un esquema de relación está en 3FN, si y sólo si está en 2FN y todos los atributos que no son clave son dependientes en forma NO transitiva de la clave primaria. Recordar que dependencias NO transitivas implica dependencias NO mutuas, o sea que cada atributo puede ser actualizado de forma independiente a los otros atributos. Entonces la normalización 3FN, consiste en tomar proyecciones para eliminar las dependencias transitivas.

NORMALIZACION

2.4. Tercera Forma Normal (3FN)

Un esquema de relación está en 3FN, si y sólo si, está en 2FN y todos los atributos que no son clave son dependientes en forma NO transitiva de la clave primaria.

Recordar: “dependencias NO transitivas” implica dependencias NO mutuas. Cada atributo puede ser actualizado de forma independiente a los otros atributos.

Entonces la normalización 3FN, consiste en tomar proyecciones para eliminar las dependencias transitivas.

La 3FN se base en el concepto de dependencia transitiva. Dada la dependencia funcional:

$$A \longrightarrow C$$

en un esquema de relación R, es una dependencia funcional transitiva, si existe un conjunto de atributos B que no sea un subconjunto de la clave de R y se cumplen tanto:

$$A \longrightarrow B \wedge B \longrightarrow C$$



Ahora continuaremos con el ejemplo anterior, la estructura VP no presenta problemas al estar ya en 3FN. Pero la estructura de SEGUNDA todavía ocasiona problemas, por la falta de independencia mutua entre sus atributos no clave. La dependencia funcional:

$$V\# \longrightarrow \text{STATUS}$$

aunque es una dependencia funcional total, es transitiva a través del atributo CIUDAD:

$$V\# \longrightarrow \text{CIUDAD} \wedge \text{CIUDAD} \longrightarrow \text{STATUS}$$

cada valor de V# determina un valor de CIUDAD y a su vez cada valor de CIUDAD determina el valor que asume STATUS. Esta situación conduce nuevamente a una variedad de anomalías de actualización (inserción, borrado y modificación) en el esquema de relación de SEGUNDA:

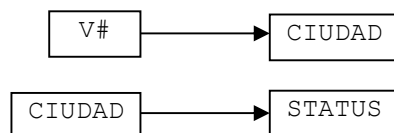
- **Inserción:** no se puede insertar el hecho de que una ciudad particular tiene un status particular, hasta que no se tenga ubicado un proveedor en dicha ciudad.
- **Borrado:** si se elimina la única tupla de una ciudad en particular, no sólo se elimina la información del proveedor respectivo, sino también la información del status de esa ciudad..
- **Modificación:** si a una ciudad se le cambia el status en una tupla, se produce una inconsistencia debido a que la misma ciudad tendrá un status diferente en otra u otras tuplas.

Solución del problema: se realiza una PROYECCIÓN de la relación SEGUNDA en dos relaciones:

```
CS{CIUDAD, STATUS}
    Primary Key{CIUDAD}
y
VC{V#, CIUDAD}
    Primary Key{V#}
    Foreign Key{CIUDAD} references CS
```

El efecto de esta descomposición es eliminar las dependencias transitivas de status sobre V# . Se puede decir de manera intuitiva que en el esquema de relación SEGUNDA, el atributo STATUS no describe la entidad identificada por la clave primaria V#; describe en su lugar la ciudad en la que el proveedor está ubicado.

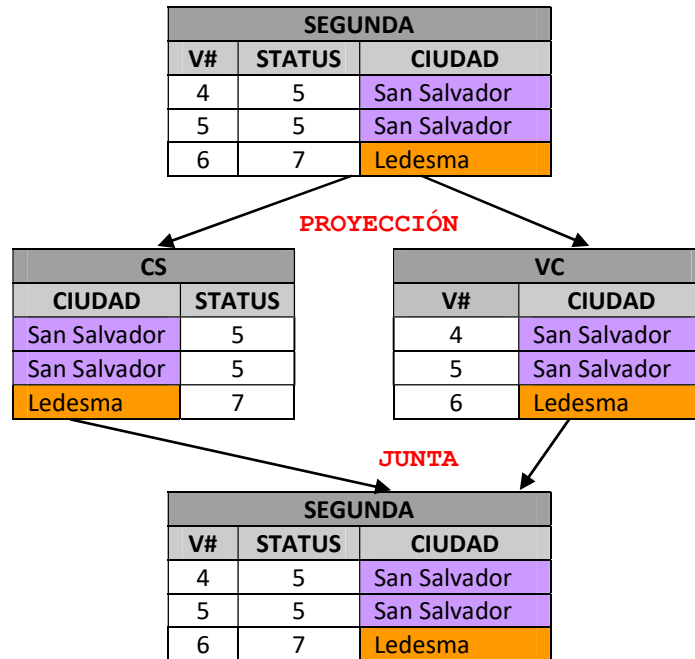
Los diagramas de dependencias funcionales para las relaciones CS y VC son los siguientes:



Esta estructura modificada supera todos los problemas de anomalías de actualización de la estructura anterior (inserción, borrado y modificación):

- **Inserción:** ahora se puede insertar el hecho de que una ciudad tiene un determinado status, sin necesidad de ubicar un proveedor en dicha ciudad; al registrar estos hechos en relaciones distintas.
- **Borrado:** ahora se puede eliminar un proveedor sin necesidad de eliminar los datos del status de una ciudad.
- **Modificación:** en la estructura CS se podrá cambiar el status de una tupla y no se producirán inconsistencias en cuanto a que la misma ciudad tenga más de un status

Las relaciones CS y VC son PROYECCIONES de SEGUNDA y ésta relación se forma por la JUNTA de CS y VC a través de su atributo común CIUDAD. A continuación se le darán valores a los esquemas de relación:



Resumen

Del esquema de relación original

PROVEEDOR{V#, STATUS, CIUDAD, P#, DIMENSIÓN, CANT}

Primary Key{V#, P#}

DF: CIUDAD → STATUS

que presentaba el problema del atributo multivaluado DIMENSIÓN, se fueron realizando sucesivas PROYECCIONES para ir corrigiendo los problemas de actualización hasta obtener como resultado los esquemas de relación siguientes:

MEDIDA{P#, DIMENSIÓN}

Primary Key{P#, DIMENSIÓN}

Foreign Key{P#} references VP

VP{V#, P#, CANT}

Primary Key{V#, P#}

Foreign Key{V#} references VC

VC{V#, CIUDAD}

Primary Key{V#}

Foreign Key{CIUDAD} references CS

CS{CIUDAD, STATUS}

Primary Key{CIUDAD}

Las relaciones entre las tablas se muestran en el siguiente diagrama:

