



CAPÍTULO 12º: FORMAS NORMALES

1 Normalización .-

El proceso de normalización consiste en seguir una serie de pasos o normas, al definir una Base de Datos Relacional que, tras ser aplicadas, se obtienen los datos agrupados en diferentes relaciones, de forma tal que la estructura obtenida es óptima para su implementación, gestión y aplicación desde diferentes aplicaciones futuras.

Se dice que una relación está en una forma normal cuando satisface un conjunto de restricciones impuestas por dicha norma.

El proceso de normalización parte de las formas normales definidas por E. F. Codd, creador de las Bases de Datos Relacionales, en 1970. Inicialmente se definieron tres formas normales (1FN, 2FN, 3FN). Debido a ciertas anomalías detectadas en ciertas bases de datos, se definió una forma normal mas completa (Forma de Boyce y Codd) y, posteriormente Fagin definió la 4FN y la 5FN.

La normalización se basa en que *"los datos son independientes de las aplicaciones que los gestionar"* y su objetivo es *"obtener el mayor número de relaciones posible, dejando en cada una de ellas los atributos imprescindibles para representar a la entidad (objeto) o a la relación entre entidades a la que hace referencia la relación mediante la conexión de sus claves"* .

Las ventajas que se obtienen tras la normalización de los datos para su eficaz gestión son:

- *Facilidad de uso.*- Los datos están agrupados en relaciones que identifican claramente un objeto o una relación.
- *Flexibilidad.*- La información que necesitan los usuarios puede obtenerse de las relaciones relacionales o las relaciones mediante operaciones de álgebra relacional, uniendo relaciones, seleccionando sus valores, proyectándolos, etc.
- *Precisión.*- Las interrelaciones entre relaciones consiguen mantener información diferente relacionada con toda exactitud.
- *Seguridad.*- Los controles de acceso para consultar o actualizar información (tanto en relaciones como en atributos) son mucho mas sencillos de implementar.
- *Independencia de datos.*- Los programas no están ligados a las estructuras, con lo que se consigue aumentar la base de datos añadiendo nuevos atributos o nuevas relaciones sin que afecten a los programas que las usan.
- *Claridad.*- La representación de la información es clara y sencilla para un usuario: son relaciones simples.



- *Facilidad de gestión.*- Los lenguajes manipulan la información en forma sencilla, al estar los datos basados en el álgebra y cálculo relacional.
- *Mínima redundancia.*- La información no estará duplicada innecesariamente dentro de las estructuras.
- *Máximo rendimiento de las aplicaciones.*- Sólo se trata aquella información que va a ser de utilidad en cada aplicación concreta.

El proceso de normalización se realiza después del análisis detallado del mundo real objeto de la base de datos, estableciendo las interrelaciones y las restricciones existentes entre los datos, sus agrupaciones y el modelo Entidad - Relación subsiguiente.

2 Primera Forma Normal (1FN).-

Se dice que una relación está en 1FN (Primera Forma Normal) si y sólo si los valores que componen cada atributo de una tupla son atómicos. Esto es: Se dice que una relación está en 1FN sí y sólo si cada atributo de la relación toma un único valor del dominio correspondiente.

3 Dependencia Funcional .-

Las dependencias funcionales determinan una manera de definir restricciones en un esquema relacional.

Dada una relación **R** que contiene los atributos **X** e **Y** se dice que **Y** *depende funcionalmente de X* ($X \rightarrow Y$) sí y sólo sí en todo momento cada valor de **X** tiene asociado un solo valor de **Y**. Esto es lo mismo que decir que si dos tuplas de **R** tienen el mismo valor para su atributo **X** forzosamente han de tener el mismo valor para el atributo **Y**.

A veces, para determinar el valor de un atributo es preciso conocer el valor de varios atributos, así puede ser que si el atributo **Y** no depende funcionalmente del atributo **X** ($X \not\rightarrow Y$) y simultáneamente el atributo **Y** tampoco depende funcionalmente del atributo **Z** ($Z \not\rightarrow Y$) sin embargo **Y** dependa funcionalmente de la composición de los atributos **X** y **Z** ($X \cdot Z \rightarrow Y$).

Dependencias completa y parcial: Dado un atributo compuesto **X** formado por los atributos **X₁** y **X₂** se dice que el atributo **Y** tiene una *dependencia funcional completa* con respecto a **X** si:

$$X_1 \not\rightarrow Y ; X_2 \not\rightarrow Y ; Y \not\rightarrow X \quad \text{pero } X \rightarrow Y$$

Por el contrario, dado un atributo compuesto **X** formado por los atributos **X₁** y **X₂** se dice que el atributo **Y** tiene una *dependencia funcional parcial* con respecto a **X** si:

$$X_1 \rightarrow Y \quad \text{o} \quad X_2 \rightarrow Y \quad \text{y} \quad X \rightarrow Y$$



Axiomas de Armstrong. Son un conjunto de reglas que permiten deducir implicaciones lógicas en un conjunto de dependencias funcionales.

Si se considera una relación relacional o una relación T , de la cual son atributos X, Y, Z, W :

Reflexividad: Si los valores del conjunto de atributos Y están incluidos o son iguales a un conjunto de atributos X se verifica que Y depende funcionalmente de X :

$$\text{Si } Y \subset X \Rightarrow X \rightarrow Y$$

Aumentatividad: Si el conjunto de atributos Y depende funcionalmente de X , la dependencia se mantiene si se añade a ambos conjuntos el mismo atributo:

$$\text{Si } X \rightarrow Y \Rightarrow X \bullet Z \rightarrow Y \bullet Z$$

Dependencia transitiva: Si en la relación T existen las siguientes dependencias funcionales

$$X \rightarrow Y ; Y \rightarrow Z ; Y \not\rightarrow X$$

Se dice que Z tiene una dependencia transitiva respecto X a través de Y :

$$X \rightarrow Z$$

Unión o aditividad: Si Y depende de X y además Z también depende de X la composición de Y y Z depende también de X :

$$\text{Si } X \rightarrow Y \text{ y } X \rightarrow Z \Rightarrow X \rightarrow Y \bullet Z$$

Pseudo-transitividad: Si el atributo Y depende funcionalmente de X y el atributo Z depende funcionalmente del atributo compuesto $W \bullet Y$ se verifica entonces que Z depende funcionalmente del atributo compuesto $W \bullet X$:

$$\text{Si } X \rightarrow Y \text{ y } W \bullet Y \rightarrow Z \Rightarrow W \bullet X \rightarrow Z$$

Descomposición o proyectividad: Si Y depende funcionalmente de X y los valores de Z están incluidos en Y entonces Z depende funcionalmente de X :

$$X \rightarrow Y \text{ si } Z \subset Y \Rightarrow X \rightarrow Z$$

4 Segunda Forma Normal (2FN) .-

Se dice que una relación se encuentra en 2FN si y sólo si cumple las condiciones siguientes:

- Se encuentra en la Primera Forma Normal.



- Todo atributo secundario (es decir, todo atributo de la relación que no pertenece a la clave principal) tiene una dependencia funcional completa de la clave principal completa.

Dado que la clave principal puede ser una clave compuesta, una relación no estará en la 2FN si algún atributo de ella depende funcionalmente de una parte de la clave principal pero no de la clave completa

5 Teorema de la 2FN.-

Sea una Relación formada por los atributos **A, B, C, D** con clave primaria compuesta por los atributos **A** y **B**. Si se cumple que:

$$A \rightarrow D$$

Entonces la Relación puede descomponerse en dos Relaciones Relación1 y Relación2 con los atributos respectivos:

RELACIÓN1 (A, D)

RELACIÓN2 (A, B, C)

6 Dependencia Funcional Transitiva.-

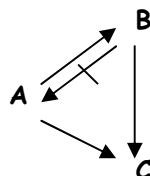
Anteriormente, dentro de los Axiomas de Armstrong, se ha establecido la dependencia funcional transitiva cuya definición formal es: Dados tres subconjuntos distintos de atributos **A, B** y **C** pertenecientes a una relación **T**, tales que cumplen las condiciones:

$$A \rightarrow B \quad \text{y} \quad B \not\rightarrow A$$

Se dice que **C** tiene una *dependencia funcional transitiva* con **A** o que es *transitivamente dependiente* de **A** si se cumple:

$$B \rightarrow C$$

La representación gráfica de dicha dependencia funcional transitiva será:





Por lo tanto el atributo (o subconjunto de atributos) **C** será transitivamente dependiente del atributo (o subconjunto de atributos) **A** si puede determinarse por diferentes caminos, en particular uno directamente, y otro, mediante un atributo (o subconjunto de atributos) **B**.

7 Tercera Forma Normal (3FN) .-

Se dice que una relación está en 3FN sí y sólo si cumple las siguientes condiciones:

- Se encuentra en la Segunda Forma Normal.
- Ningún atributo no primario es transitivamente dependiente de cada posible clave (primaria o candidatas) de la relación.

Esto quiere decir que no existe ningún atributo no principal que dependa transitivamente de alguna de las claves de la relación.

8 Teorema de la 3FN .-

Sea una Relación formada por los atributos **A**, **B**, **C** con clave primaria formada por el atributo **A**. Si se cumple que:

$$B \rightarrow C$$

Entonces la Relación puede descomponerse en dos Relaciones Relación1 y Relación2 con los atributos respectivos:

RELACIÓN1 (A, B)

RELACIÓN2 (B, C)

9 Descomposición de relaciones .-

La transformación de una relación que se encuentra en una determinada forma normal en otra relación cuya forma normal es superior se realiza por medio del operador *proyección* del álgebra relacional .

Así, por ejemplo la relación:

RELACIÓN(Campo1, Campo2, Campo3)

es tal que se encuentra en 1FN por que su único atributo no principal (Campo3) no depende totalmente de la clave (agregación de Campo1 y de Campo2), sino de parte de ella (por ejemplo, Campo3 solamente depende de Campo2), puede llevarse a una forma normal más avanzada descomponiéndola mediante proyecciones, obteniendo así varias relaciones:



RELACIÓN1 = $\Pi_{\text{Campo1, Campo2}}$ (**RELACIÓN**)

RELACIÓN2 = $\Pi_{\text{Campo2, Campo3}}$ (**RELACIÓN**)

Estando ambas relaciones en una forma normal superior. En concreto en 3FN ya que la combinación natural **RELACIÓN1** * **RELACIÓN2** mediante el atributo común Campo2 devuelve la relación original **RELACIÓN**

9.1 *Descomposición sin pérdida de información .-*

Se dice que una descomposición se ha realizado sin pérdida de información cuando la combinación natural de las proyecciones resultantes devuelve la relación original.

La condición necesaria y suficiente para que una descomposición se produzca sin pérdida de información es que el *atributo común de las dos relaciones sea clave, al menos en una de ellas.*

9.2 *Descomposición sin pérdida de dependencias funcionales .-*

Las dependencias funcionales recogen la semántica del mundo real por lo que es conveniente conservarlas en el proceso de descomposición

9.3 *Descomposición en proyecciones independientes .-*

La descomposición de una relación R en un conjunto de relaciones {R_i} se dice que se ha realizado en proyecciones independientes si no ha habido pérdida de información ni pérdida de dependencias funcionales.

Se trata de la mejor descomposición ya que las relaciones resultantes son equivalentes a la relación original y, en ellas se han eliminado las anomalías de inserción, modificación y borrado.

Toda relación en 3FN puede descomponerse sin pérdida de información ni de dependencias funcionales.

10 Forma Normal de Boyce - Codd .-

Se define como *determinante* en una Relación a un atributo del cual depende funcionalmente de manera completa cualquier otro atributo de la Relación

Se dice que una Relación está en la Forma Normal de Boyce - Codd (FNBC) si y sólo si todo determinante de ella es una clave candidata.

En la práctica, muy pocas Relaciones que se encuentren en la 3FN no están en FNBC. Este tipo de tablas son aquellas en las que se dan las siguientes circunstancias:



- Existen varias claves candidatas
- Las claves candidatas son compuestas
- Las claves candidatas se solapan, esto es, tienen por lo menos un atributo común.

11 Teorema de Boyce - Codd .-

Sea una Relación formada por los atributos **A, B, C, D** con claves candidatas compuestas (**A, B**) y (**B, C**) tal que:

$$A \leftrightarrow C$$

Entonces la Relación puede descomponerse en cualquiera de las dos siguientes maneras:

RELACIÓN1 (A, C)

RELACIÓN2 (B, C, D)

O bien:

RELACIÓN1 (A, C)

RELACIÓN2 (A, B, D)

12 Dependencias Multivaluadas .-

Dada una relación con los atributos **A, B, C** se dice que se cumple en ella una dependencia multivaluada

$$A \rightarrow B$$

Si y sólo si el conjunto de valores correspondiente a un par dado (valor de **A**, valor de **C**) en la Relación depende sólo del valor de **A** y es independiente del valor de **C**.

La dependencia multivaluada **A → B** se cumple sí y sólo sí también se cumple **A → C**

13 Cuarta Forma Normal (4FN) .-

Una Relación se encuentra en 4FN sí y sólo sí:

- Está en FNBC
- No existen dependencias multivaluadas



14 Teorema de Fagin .-

Dada la relación formada por los atributos A, B, C con las siguientes dependencias multivaluadas:

$$A \rightarrow B$$

$$A \rightarrow C$$

Entonces la relación puede descomponerse en dos relaciones:

RELACIÓN1 (A, B)

RELACIÓN2 (A, C)

15 Dependencias de Reunión .-

Una *Dependencia de Reunión* es una restricción en una Relación. Se dice que una Relación satisface la dependencia de reunión (X, Y, ... , Z) sí y sólo sí la Relación es igual a la reunión de sus proyecciones según X, Y, ... , Z . Donde X, Y, ... , Z son subconjuntos del conjunto de atributos de la Relación.

16 Quinta Forma Normal (5FN) .-

Una Relación se encuentra en 5FN sí y sólo sí toda dependencia de reunión en la Relación es una consecuencia de las claves candidatas, esto es, la relación estará en 5FN si está en 4FN y no existen restricciones impuestas por el creador de la Base de Datos.

17 Metodología de diseño de una Base de Datos .-

Como ejemplo de aplicación se establece con un ejemplo la metodología a seguir en el diseño de una base de datos:

La Comunidad de Madrid desea guardar información sobre los alojamientos rurales que existen en dicha comunidad. Para ello decide crear una base de datos que recoja las siguientes consideraciones:

Un alojamiento rural se identifica por un nombre ("Villa Aurora", "Las Rosas", etc..), tiene una dirección, un teléfono y una persona de contacto que pertenece al personal del alojamiento.

En cada alojamiento trabajan una serie de personas que se identifican por un código de personal. Se requiere conocer el nombre completo, la dirección y el NIF. Aunque en el alojamiento trabajen varias personas, una persona sólo puede trabajar en un alojamiento.



Los alojamientos se alquilan por habitaciones y se desea conocer cuántas habitaciones componen el alojamiento y de qué tipo (individuales, dobles, triples) es cada una de estas habitaciones, si poseen cuarto de baño y el precio.

En alguno de estos alojamientos se realizan actividades multiaventura organizadas para huéspedes (senderismo, bicicleta de montaña, etc.). Estas actividades se identifican por un código. Es de interés saber el nombre de la actividad, la descripción y el nivel de dificultad de dicha actividad (de 1 a 10).

Estas actividades se realizan un día a la semana, por ejemplo, en la casa "Villa Aurora" se practica el senderismo los jueves y se desea guardar esta información. Pero puede haber algún día en el que no se practique ninguna actividad.

Diseñar la Base de Datos indicada de forma normalizada.

1º DISEÑO CONCEPTUAL: MODELO ENTIDAD /RELACIÓN

1 Paso: *Elaborar las listas de conceptos candidatos a ser entidades e interrelaciones indicando los conceptos que no se sabe como catalogar.* Un análisis del enunciado indica:

Entidades:	Interrelaciones:
ALOJAMIENTOS	TRABAJA_EN
PERSONAL	ALQUILAN
HABITACIONES	
ACTIVIDADES	REALIZAN

¿DIAS_SEMANA?

Las entidades e interrelaciones suelen estar explícitamente indicadas en el universo definido por el cliente.

2 Paso: *Construir una matriz ENTIDADES / ENTIDADES que representen las relaciones junto con el tipo de correspondencia.* Para ello se analizan los supuestos indicados en el enunciado, así como los supuestos implícitos o de sentido común:

a) Supuestos del enunciado:

En un ALOJAMIENTO pueden TRABAJAR varias PERSONAS
Cada ALOJAMIENTO puede ALQUILAR varias HABITACIONES
Cada ALOJAMIENTO puede REALIZAR varias ACTIVIDADES

b) Supuestos implícitos

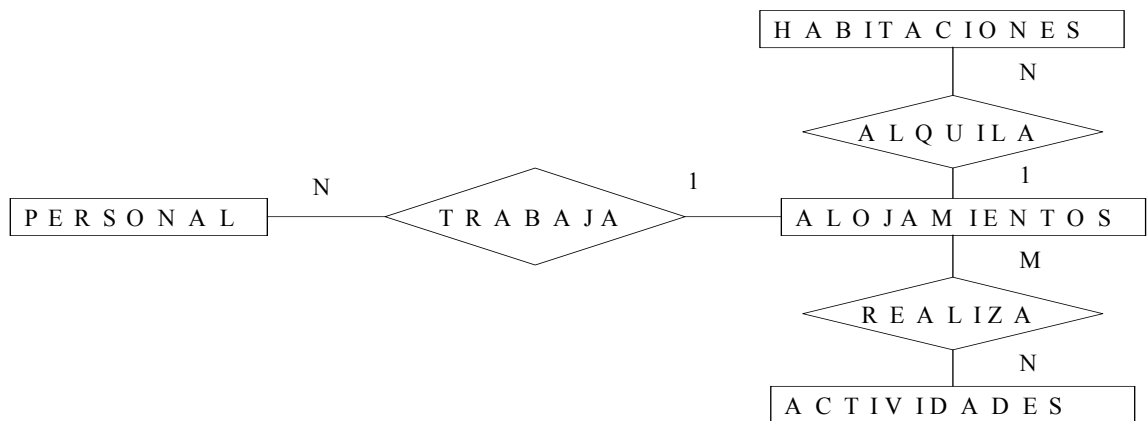
Ninguno

La matriz obtenida será:



	ALOJAMIENTOS	PERSONAL	HABITACIONES	ACTIVIDADES
ALOJAMIENTOS	-	X	ALQUILAN (1:N)	REALIZAN (M:N)
PERSONAL	TRABAJA EN (1:N)	-	X	X
HABITACIONES	X	X	-	X
ACTIVIDADES	X		X	-

3 Paso Realizar una versión preliminar del diagrama ENTIDAD / RELACIÓN:



4 Paso Análisis de las Cardinalidades Mínimas. Anteriormente se han definido las cardinalidades máximas. En cuanto a las cardinalidades mínimas se tiene:

- TRABAJAN: Una PERSONA sólo puede trabajar en un ALOJAMIENTO
- ALQUILAN : Una HABITACIÓN sólo puede estar ALQUILADA (pertenece) a un ALOJAMIENTO.
- REALIZAN Una ACTIVIDAD puede realizarse en varios ALOJAMIENTOS.

5 Paso Análisis de redundancias. En el ejemplo anterior al no existir ciclos en el diagrama ENTIDAD / RELACION no existen problemas de redundancias.

Por tanto puede indicarse que el diagrama ENTIDAD / RELACION establecido es válido.

2º DISEÑO LÓGICO: MODELO RELACIONAL

1 Paso: Identificación de las relaciones básicas. De la lectura del enunciado



Un alojamiento rural se identifica por un nombre ("Villa Aurora", "Las Rosas", etc.), tiene una dirección, un teléfono y una persona de contacto que pertenece al personal del alojamiento.

ALOJAMIENTOS (Nombre-A, Dirección, Teléfono, Contacto*)

En principio Nombre-A identifica la clave primaria y Contacto se señala con * indicando que puede tener valores nulos, permitiendo la posibilidad de que, cuando se dé de alta un nuevo alojamiento no esté definida todavía la persona asignada.

En cada alojamiento trabajan una serie de personas que se identifican por un código de personal. Se requiere conocer el nombre completo, la dirección y el NIF. Aunque en el alojamiento trabajen varias personas, una persona sólo puede trabajar en un alojamiento.

PERSONAL (Código-P, Nombre-P, Dirección, NIF)

Se asigna como clave primaria a Código-P y como clave alternativa a NIF.

Los alojamientos se alquilan por habitaciones y se desea conocer cuántas habitaciones componen el alojamiento y de qué tipo (individuales, dobles, triples) es cada una de estas habitaciones, si poseen cuarto de baño y el precio.

HABITACIONES (Nombre-A, Nº-Hab, Tipo, Baño*, Precio*)

La clave primaria será el número de habitación "Nº-Hab" junto con el alojamiento "Nombre-A" donde está ubicada. Como puede darse el caso de que no se suministre información sobre si tiene baño o no y sobre su precio, ambos atributos pueden ser nulos. Además,

se desea conocer cuántas habitaciones componen el alojamiento

por lo que será preciso ampliar la relación ALOJAMIENTOS con el atributo Número-H que indica el número de habitaciones que hay en el alojamiento:

ALOJAMIENTOS (Nombre-A, Dirección, Teléfono, Contacto*, Número-H)

En alguno de estos alojamientos se realizan actividades multiaventura organizadas para huéspedes (senderismo, bicicleta de montaña, etc.). Estas actividades se identifican por un código. Es de interés saber el nombre de la actividad, la descripción y el nivel de dificultad de dicha actividad (de 1 a 10).

Este párrafo establece la necesidad de la relación

ACTIVIDADES (Código-Ac, Nombre-Ac, Descripción, Nivel)

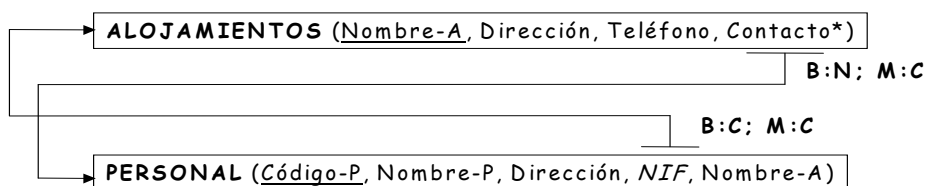
Restricciones semánticas.



Un alojamiento rural se identifica por un nombre ("Villa Aurora", "Las Rosas", etc..), tiene una dirección, un teléfono y una persona de contacto que pertenece al personal del alojamiento.

En cada alojamiento trabajan una serie de personas pero una misma persona sólo puede trabajar en un alojamiento por lo que debe introducirse el atributo Nombre-A en la relación PERSONAL que será clave ajena de esta relación y referenciará a la relación ALOJAMIENTOS.

Además la persona de contacto pertenece al alojamiento por lo que "Contacto" será en la relación ALOJAMIENTOS clave ajena que referenciará a la relación PERSONAL



Opciones de Borrado y Modificación.-

Si se elimina un alojamiento de la base de datos, automáticamente ha de eliminarse todo el personal que trabaja en él, el borrado es por tanto en Cascada (B:C) y lo mismo ocurre con las modificaciones (M:C)

Sin embargo si se elimina una persona, la relación ALOJAMIENTOS no debe verse afectada y si la persona eliminada es la de contacto, como el atributo contacto admite valores nulos, puede considerarse el borrado como puesta a nulos (B:N) . La modificación sin embargo debe hacerse en cascada, pues si se modifica la persona de contacto en PERSONAL, debe modificarse también la relación ALOJAMIENTOS (M:C)

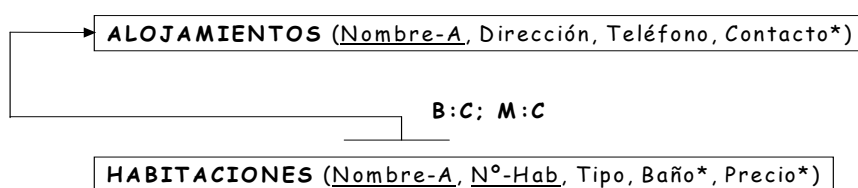


Otras reglas no contempladas.-

Debe controlarse en la definición de la clave el hecho de que en un alojamiento trabaje al menos una persona.

Los alojamientos se alquilan por habitaciones y se desea conocer cuántas habitaciones componen el alojamiento

Los alojamientos disponen de una serie de habitaciones por lo que éstas dependen de los alojamientos en el sentido de que para hacer referencia a una habitación es preciso conocer el nombre del alojamiento donde está incluida. Nombre-A es por tanto la clave ajena de la relación HABITACIONES que referencia a la relación ALOJAMIENTOS:



Opciones de borrado y modificación.-

Si se elimina o modifica un alojamiento deben eliminarse o modificarse todas las habitaciones de dicho alojamiento por lo que tanto el borrado como la modificación han de ser en cascada (B:C; M:C).

Otras reglas no contempladas.-

Como se desea conocer el número de habitaciones que tiene cada alojamiento debe contemplarse el hecho de que cada vez que se dé de alta o se elimine una habitación, automáticamente se incremente o decremente en uno el valor del atributo Número-H.



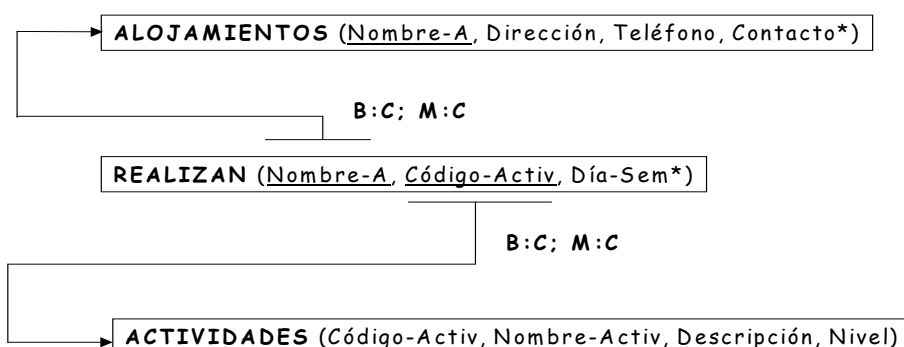
En alguno de estos alojamientos se realizan actividades multiaventura organizadas para huéspedes (senderismo, bicicleta de montaña, etc..).

Estas actividades se realizan un día a la semana, por ejemplo, en la casa "Villa Aurora" se practica el senderismo los jueves y se desea guardar esta información. Pero puede haber algún día en el que no se practique ninguna actividad.

Dado que cada alojamiento puede proporcionar varias actividades multiaventura y además una misma actividad puede desarrollarse en varios alojamientos, existirá una nueva relación: REALIZAN cuya clave primaria será la composición de las claves primarias de ALOJAMIENTOS y de ACTIVIDADES. Cada una a su vez será clave ajena que referencia a la relación de que proceden. Además

Estas actividades se realizan un día a la semana

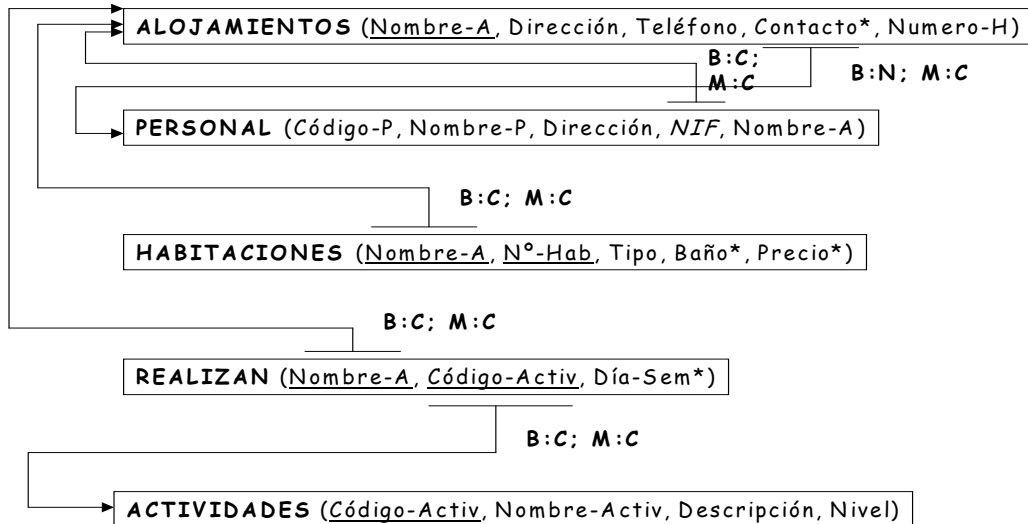
Debe introducirse el atributo "Día-Sem" en la relación REALIZAN este atributo puede tener valores nulos para contemplar el hecho de que una actividad propuesta en un cierto alojamiento todavía no esté configurada.



Opciones de borrado y modificación.-

Cada vez que se elimine o se modifique una actividad se debe eliminar o modificar la tupla correspondiente de la relación intermedia, por lo tanto, tanto el borrado como la modificación será en cascada.

En resumen, el esquema relacional será el siguiente:



3º NORMALIZACION DE LAS RELACIONES

1 Primera Forma Normal

Las cinco relaciones se encuentran en la 1FN dado que los valores de todos sus respectivos atributos son atómicos.

2 Dependencias funcionales.

ALOJAMIENTOS (Nombre-A, Dirección, Teléfono, Contacto*, Número-H)

Nombre-A → Dirección
Nombre-A → Teléfono
Nombre-A → Contacto
Nombre-A → Número-H

Todos los atributos dependen directamente de la clave primaria

PERSONAL (Código-P, Nombre-P, Dirección, NIF, Nombre-A)

Código-P → Nombre-P
Código-P → Dirección
Código-P ↔ NIF
Código-P → Nombre-A

Todos los atributos dependen directamente de la clave primaria



HABITACIONES (Nombre-A, Nº-Hab, Tipo, Baño*, Precio*)

(Nombre-A, Nº-Hab) → Tipo

(Nombre-A, Nº-Hab) → Baño

(Nombre-A, Nº-Hab) → Precio

Todos los atributos dependen directamente de la clave primaria, la cual es compuesta

REALIZAN (Nombre-A, Código-Activ, Día-Sem*)

(Nombre-A, Código-Activ) → Sia-Sem

Todos los atributos dependen directamente de la clave primaria, la cual es compuesta

ACTIVIDADES (Código-Activ, Nombre-Activ, Descripción, Nivel)

Código-Activ → Nombre-Activ

Código-Activ → Descripción

Código-Activ → Nivel

Todos los atributos dependen directamente de la clave primaria

3 Recubrimiento Mínimo

Para cada relación debe determinarse que el conjunto de dependencias del apartado anterior son dependencias elementales, no existe ningún atributo extraño en ninguna de ellas y no existen dependencias redundantes. Para ello

Dependencias elementales

Debe comprobarse que:

- En todas las dependencias se cumple que tienen un único atributo implicado
- Todas las dependencias son plenas, esto es, ningún subconjunto del atributo implicante (determinante) puede implicar al atributo implicado
- Ninguna dependencia es trivial, es decir, ningún atributo del determinante aparece como implicado

Todas las dependencias son elementales.

Atributos Extraños

Se dice que el atributo A es extraño en la dependencia $(X,A) \rightarrow Y$ cuando se cumple que $X \rightarrow Y$ sin que A tome parte de la dependencia.



Ninguna dependencia de las indicadas anteriormente tiene atributos extraños.

Dependencias Redundantes

Para cada una de las relaciones indicadas no existen dependencias redundantes

Por tanto, el recubrimiento mínimo de cada relación es el conjunto de las dependencias indicadas anteriormente.

4 Claves Primarias

Para cada relación se han definido ya las claves primarias

5 Segunda Forma Normal

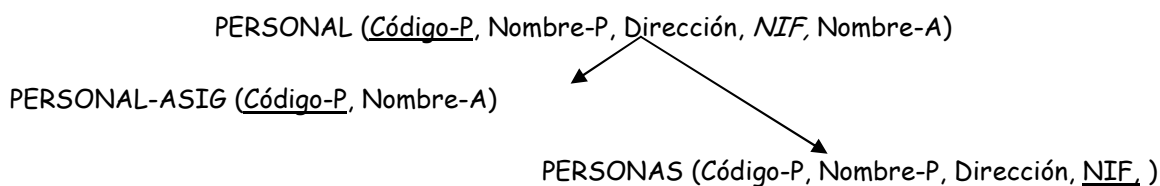
Para cada relación todos los atributos tiene una dependencia funcional completa de su respectiva clave, por lo que las cinco relaciones están en 2FN

6 Tercera Forma Normal

Para cada relación no existen atributos que tengan una dependencia funcional transitiva de su respectiva clave, por lo que las cinco relaciones están en 3FN

7 Forma Normal de Boyce Codd

Unicamente la relación PERSONAL no está en la FNBC ya que el atributo NIF determinante que no forma parte de la clave primaria, por lo que, en virtud del teorema de Boyce Codd puede descomponerse en dos recomponerse en dos relaciones:



4º IMPLEMENTACION DE LA BASE DE DATOS

1 Creación de dominios.

```
CREATE DOMAIN Nombre_Valido CHAR(30);
CREATE DOMAIN Tipo_Codigo CHAR(3);
CREATE DOMAIN Tipo_Bano CHAR(2);
CHECK (VALUE IN ("SI","NO"));
```

2 Creación de tablas



```
CREATE TABLE Alojamientos (  
Nombre_A Nombre_Valido,  
Direccion CHAR(25) NOT NULL,  
Telefono CHAR(9) NOT NULL,  
Contacto Tipo_Codigo,  
Numero_H INTEGER NOT NULL,  
PRIMARY KEY (Nombre_A)  
);
```

```
CREATE TABLE Personal_Asig (  
Codigo_P Tipo_Codigo,  
Nombre_A Nombre_Valido NOT NULL  
PRIMARY KEY (Codigo_P),  
FOREIGN KEY (Nombre_A) REFERENCES Alojamiento  
ON UPDATE CASCADE  
ON DELETE CASCADE  
);
```

```
CREATE TABLE Personas (  
Codigo_P Tipo_Codigo,  
Nombre_P Nombre_Valido NOT NULL,  
Direccion CHAR(25) NOT NULL,  
NIF CHAR(10) NOT NULL  
PRIMARY KEY (NIF),  
FOREIGN KEY (Nombre_P) REFERENCES Personal_Asig  
ON UPDATE CASCADE  
ON DELETE CASCADE  
);
```

```
ALTER TABLE Alojamientos  
ADD FOREIGN KEY (Contacto) REFERENCES Personal_Asig  
ON DELETE SET NULL  
ON UPDATE CASCADE;
```

```
CREATE TABLE Habitaciones (  
Nombre_A Nombre_Valido,  
No_Hab Integer(3) NOT NULL,  
Tipo Char(1) NOT NULL,  
Bano Tipo_Bano,  
Precio INTEGER,  
PRIMARY KEY (Nombre_A, No_Hab),  
FOREIGN KEY (Nombre_A) REFERENCES Alojamiento  
ON UPDATE CASCADE  
ON DELETE CASCADE
```



);

```
CREATE TABLE Actividades (  
Codigo_Activ Tipo_Codigo,  
Nombre_Activ Nombre_Valido NOT NULL,  
Descripcion CHAR(50) NOT NULL,  
Nivel CHAR(2) NOT NULL,  
PRIMARY KEY (Codigo_Activ),  
CHECK (Nivel BETWEEN 1 AND 10)  
);
```

```
CREATE TABLE Realizan (  
Nombre_A Nombre_Valido,  
Codigo_Activ Tipo_Codigo,  
Dia_Sem CHAR(10)  
PRIMARY KEY (Codigo_Activ, Nombre_A),  
FOREIGN KEY (Codigo_Activ) REFERENCES Actividades  
ON UPDATE CASCADE  
ON DELETE CASCADE,  
FOREIGN KEY (Nombre_A) REFERENCES Alojamientos  
ON UPDATE CASCADE  
ON DELETE CASCADE  
);
```

2 Consultas Principales:

Nombre y descripción de las actividades que se realizan en el alojamiento denominado "La Huerta"

SQL

```
SELECT A.Nombre_Activ, A.Descripcion  
FROM Actividades A, Realizan B  
WHERE Nombre_A = "La Huerta" AND A.Codigo_Activ = B.Codigo_Activ
```

Algebra Relacional

$$\Pi_{\text{Nombre_Activ, Descripcion}}((\text{ACTIVIDADES}) * \Pi_{\text{Codigo_Activ}}(\sigma_{\text{Nombre_Activ} = \text{"La Huerta"}}(\text{REALIZAN})))$$

Nombre de los alojamientos que tienen habitaciones dobles y realizan actividades de senderismo

SQL

```
SELECT DISTINCT Nombre_A
```



FROM Actividades A, Realizan B, Habitaciones C
WHERE A.Nombre_A = "Senderismo" AND C.Tipo = "Doble" AND
(B.Codigo_Activ = A.Codigo_Activ AND B.Nombre.A = C.Nombre_A)

Algebra Relacional

$\Pi_{\text{Nombre_A}}(\sigma_{\text{Tipo}=\text{"Doble"}}(\text{HABITACIONES})) * (\Pi_{\text{Nombre_A}}(\text{REALIZAN}) * \Pi_{\text{Codigo_Activ}}(\sigma_{\text{Nombre_Activ}=\text{"Senderismo"}}(\text{ACTIVIDADES})))$
