# PRESENTACIÓN

**Fernando-**¡Hola que tal!

Nuestro equipo lo integran alumnos del grupo 2CM2 para la unidad de aprendizaje de Bases de Datos en la ESCOM, presentan:

Fernando Hernández Escobedo

**Luis-**Mi nombre es Luis Rojas Alvarado

**Gilberto-**Mi nombre es Gilberto Oledo Enríquez

**Juan-**Mi nombre es Juan Campos Leaños

Hoy presentaremos el tema de Dependencias Funcionales y Normalización

# INTRODUCCION:

**Luis-** El problema de la concepción de bases de datos relacionales se reduce a la descomposición sin pérdida de las relaciones universales con todos sus atributos, en sub-relaciones que no tengan anomalías.

Partimos de la idea de una Relación Universal en la que participan los atributos de la base de datos y se descomponen a través de tablas separadas o sub-relaciones. Vamos a combinar esa Relación Universal con un conjunto de reglas predefinidas para organizar datos basado en formas normales, las cuales las veremos más adelante. Con dichas reglas podemos separar atributos en diferentes tablas hasta que llega un punto en el que ya no hay relación entre los atributos de estas tablas, una vez encontradas estas últimas relaciones en las que ya no se puede separar, se dice que el conjunto de esas relaciones son mi base de datos.

**Gilberto-** Por ejemplo, pensando en una aplicación desarrollada para trabajar con un sistema en el cual la cantidad de información es enorme, tomaríamos todas las tablas y atributos que hay para formar la Relación Universal la cual sería verdaderamente grande, donde la posibilidad de generar diferentes errores es muy alta; Esto se puede solucionar a través del uso o aplicación de un conjunto de reglas predefinidas llamadas dependencias funcionales

**Fernando-** Las Dependencias Funcionales y la Normalización, son una estrategia muy útil en la base de datos ya que nos ayudan a evitar poner información de más, al separarla a lo máximo posible podremos evitar mucha redundancia, o mejor dicho como el profesor Demian Gutiérrez: **“Dependencias Funcionales y Normalización, o el arte de no redundar”**

# **DESARROLO:**

**Fernando-** La normalización de una base de datos es encontrar una forma de descomposición adecuada de la “relación universal” de la base de datos que nos permite cumplir con los criterios de eficacia, ausencia de redundancia, evolución, comprensión y flexibilidad.

Las técnicas formales para organizar datos a través de la normalización, son las siguientes, como se muestra en la imagen:

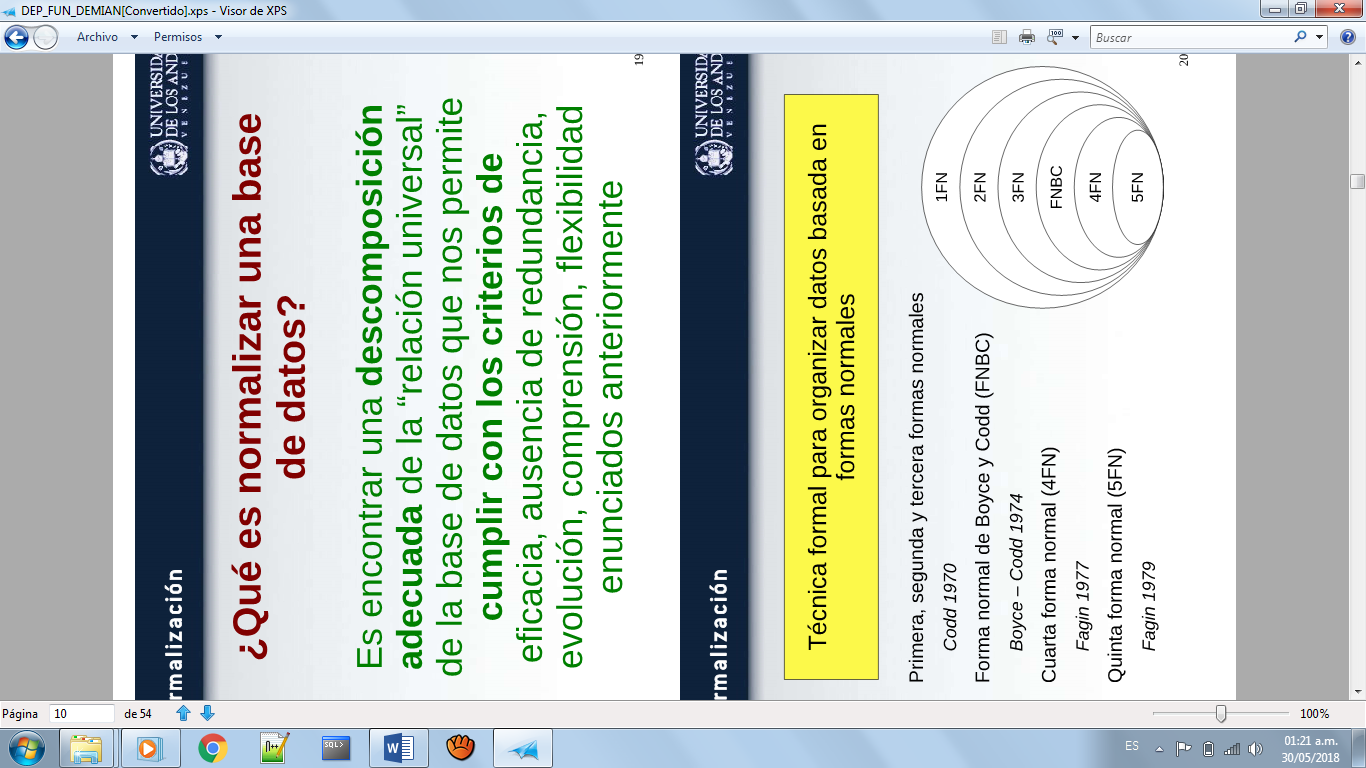
🡺Primera, segunda y tercera forma normal

🡺Forma Normal de Boyce y Codd

🡺Cuarta forma normal

🡺Quinta forma normal

\*Insertar imagen



Por tanto, tendríamos que llevar el esquema de la base de datos a alguna de las formas normales mencionadas.

Para eso, necesitamos conocer y comprender el concepto de “Dependencia Funcional”

* Son restricciones de integridad que permiten conocer que interrelaciones existen entre dos o más atributos del mundo real.
* Son propiedades inherentes al contenido semántico de los datos, que se han de cumplir para cualquier extensión del esquema de relación.

Es decir;

“Y **Depende Funcionalmente** de X si

cada valor de X tiene asociado

siempre el mismo valor de Y en

una relación R que contiene a

X y Y como atributos.”

ó

“X determina a Y”

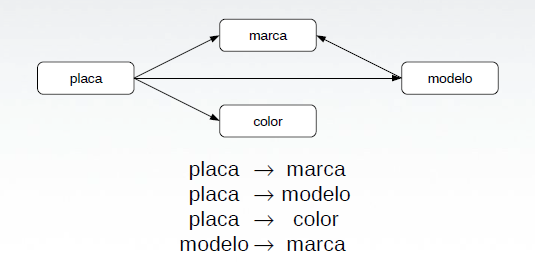
Una forma de representar las

dependencias funcionales es a través

de las redes de DF, como se muestra

en el siguiente ejemplo:

\*\*\*Insertar imagen:



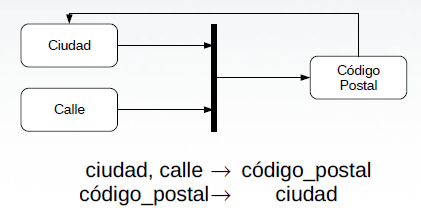
“Donde los nodos representan atributos y las aristas representan DF. Placa determina a marca, modelo y color, pero modelo determina marca”

Si hay más de un atributo del lado izquierdo se utiliza una línea que sirva para unir todos los atributos de la parte izquierda,

y de ella sale una línea al

atributo de la parte derecha:

\*\*\*Insertar imagen:

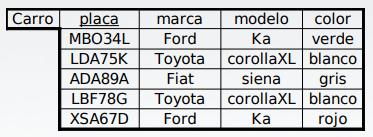


**Gilberto-**

Una vez que comprendimos la definición de dependencia funcional o dependencias funcionales de igual forma es importante conocer las propiedades de estas.

Se tiene la siguiente tabla llamada carro, conformada por 4 atributos, placa, marca, modelo y color  
es necesario recodar esta relación porque nos ayudara a ejemplificar cada una de las propiedades.

**Imagen:**

**Primera propiedad**

1. Reflexividad:

Si Y es un subconjunto de X entonces X implica a Y esto quiere decir que si recordamos la tabla llamada carro podemos decir que color implica color o bien la marca y el modelo implican la marca porque marca es un subconjunto de marca y modelo

**Imagen:**

****

Ej: color → color, ó bien: (marca, modelo) → marca

**Segunda propiedad.**

2. Aumento:

Quizá una forma de entender esto es tomando como ejemplo una ecuación, si multiplicamos por algo ambos lados de la ecuación esto no altera en lo absoluto su igualdad.

Por lo tanto podemos decir que si X implica a Y, X aumentado con Z implica a Y aumentado con Z

Por lo que modelo → marca ⇒ (modelo, color) → (marca, color)

**Imagen:**



**Tercera propiedad.**

3. Transitividad:

Decimos que si X implica a Y y Y implica a Z entonces X implica a Z

Es decir, si placa → modelo y modelo → marca ⇒ placa → marca

**Imagen:**

****

**Cuarta propiedad**

4.Descomposición:

Si X implica a Yy a Z en conjunto podemos decir que X implica a Y y X implica Z por separado

placa → (modelo, marca) ⇒ placa → modelo y placa → marca

**Imagen:**

****

**Quinta propiedad**

**5.Unión:**

**Por el contrario a la propiedad 4 ahora podemos decir lo siguiente:**

Si X implica a Y y X implica a Z por separado entonces X implica a YZ en conjunto

placa → modelo y placa → marca ⇒ placa → (modelo, marca)

**Imagen:**



**Sexta propiedad**

6.Pseudotransitividad:

Si tenemos una situación en la que X implica a Y y WY en conjunto implican a Z entonces podremos decir que WX implican a Z, es decir podemos sustituir Y ya que X implica a Y

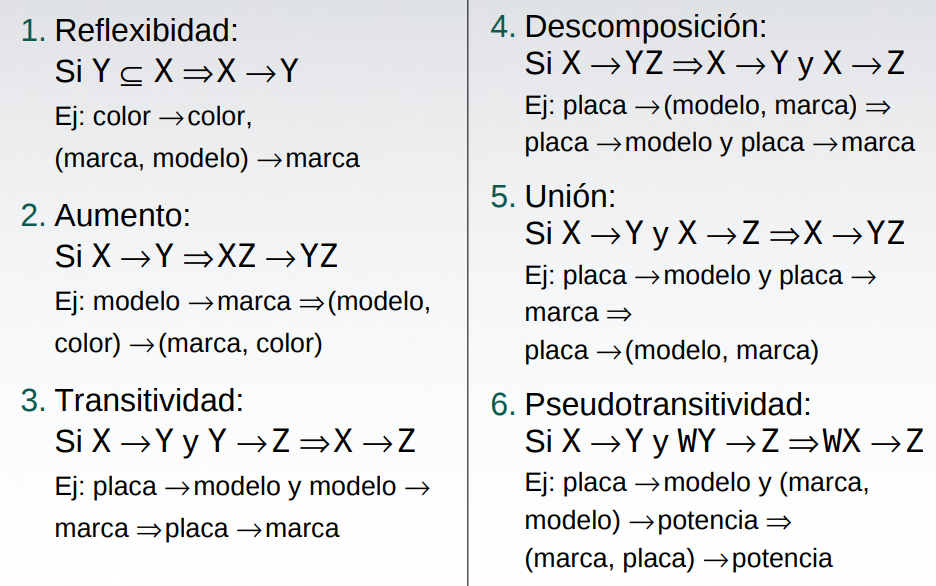
placa → modelo y (marca, modelo) → potencia ⇒

(marca, placa) → potencia

**Imagen:**



Aquí como conclusión mostramos un resumen porque a veces es necesario poder visualizar las propiedades en una sola hoja.

****

Ahora, otro concepto importante e interesante es el de Dependencias Funcionales Elementales por su abreviación (DF**E**)

Es una dependencia funcional de la forma X →A, donde A es un atributo único no incluido en X y donde no existe un subconjuto X' ⊆ X tal que X'→A

Ejemplos:

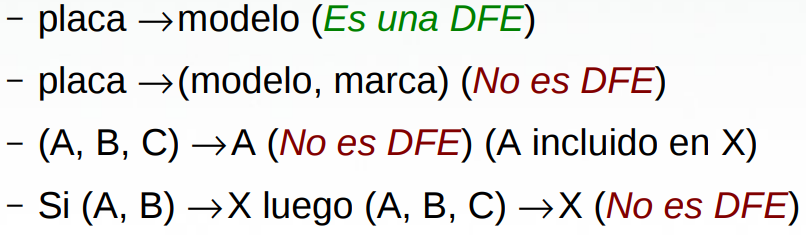
**– placa →modelo (Es una DFE)**

**– placa →(modelo, marca) (No es DFE)**

**– (A, B, C) → A (No es DFE) (A incluido en X)**

**– Si (A, B) → X luego (A, B, C) → X (No es DFE)**

**Imagen:**

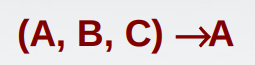
****

En parte, la importancia de las DFE es que no es posible simplificarlas más.

Por ejemplo la siguiente DFNE:

**(A, B, C) →A**

**Imagen:**

****

Se puede simplificar por reflexividad en la siguiente DFE:

**(B, C) →A**

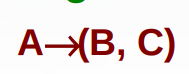
**Imagen:**

****

O bien la siguiente DFNE:

**A→(B, C)**

**Imagen:**

****

Se puede simplificar en las siguientes DFE:

**A→B y A→C**

**Imagen:**

****

Ahora bien, otra cosa que podemos calcular es el cierre transitivo de un conjunto de DF

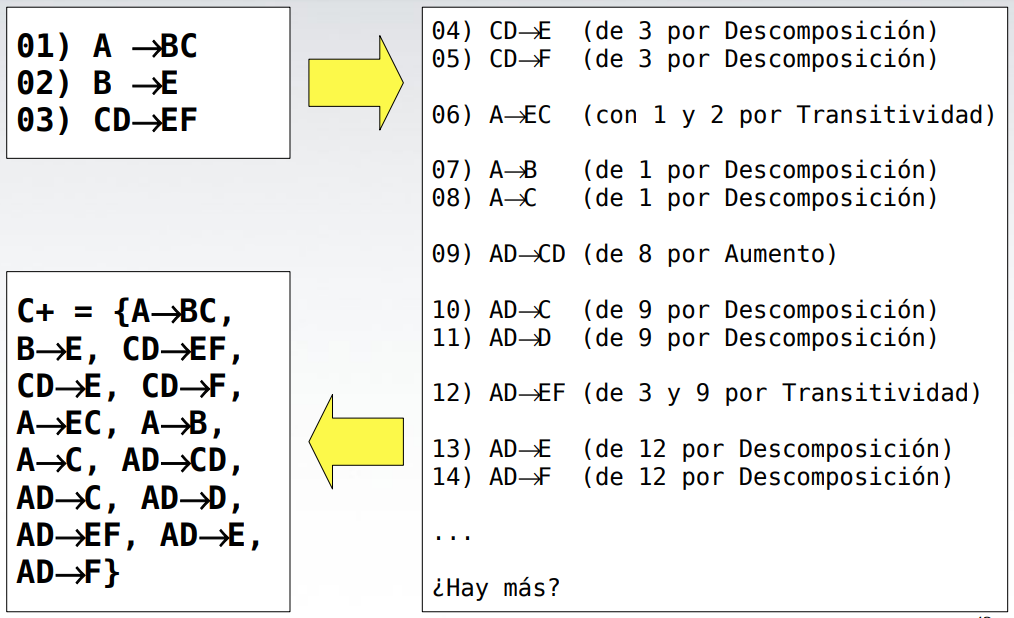
Dado un conjunto C de dependencias funcionales, el cierre transitivo de C, denotado por C+ , es un nuevo conjunto formado por las dependencias funcionales elementales más todas las calculadas aplicando los axiomas de Armstrong y sus reglas inferidas.

¿Y para que es útil? Puede suceder que tengamos dos conjuntos de dependencias funciones y queramos saber si son equivalentes.

Dos conjuntos de dependencias funcionales son equivalentes si tienen el mismo C+ . (o si a partir de los conjuntos iniciales se puede llegar a un mismo conjunto para ambos casos)

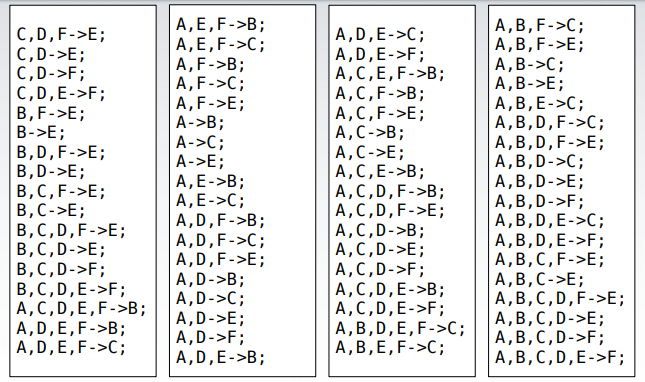
**POR EJEMPLO**

**Imagen:**



Sin embargo hacerlo por este método es hasta cierto punto complicado obtener en su totalidad el cierre transitivo

Resultado del cierre transitivo anterior



Aun calculándolo con un software nos damos cuenta que algunas dependencias NO son elementales y además puede suceder que algunas dependencias estén de más o repetidas ya que podemos inferir algunas cosas de otras y llegar a lo mismo.

TRUCO

Dos conjuntos de dependencias funcionales F y G son equivalentes si ambos tienen el mismo cierre transitivo, es decir, si: F+ = G+ (O si partiendo de F y G podemos obtener un mismo conjunto de dependencias funcionales H)

Dado un conjunto C de dependencias funcionales y un conjunto X de atributos, es posible determinar el conjunto X+ de todos los atributos que dependen funcionalmente de X. El conjunto X+ se conoce como el cierre de X bajo C.

resultado := X

mientras (cambios en resultado) hacer

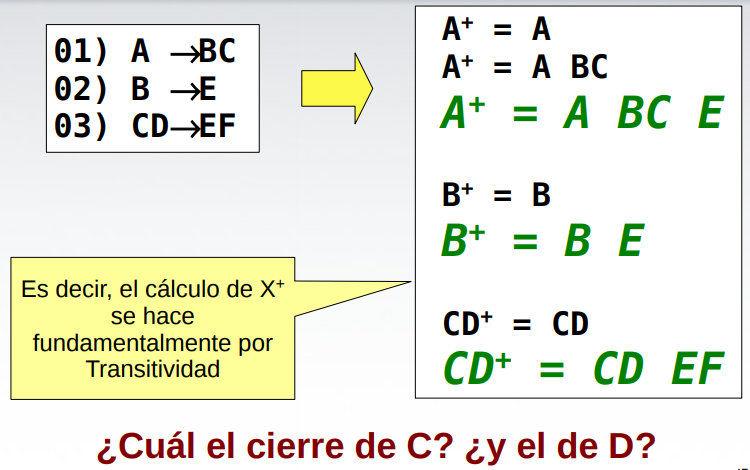
para cada dependencia funcional A→B en C hacer

si A ⊆ resultado entonces resultado := resultado U B

Ahora, dado el conjunto de dependencias funcionales que se muestra, podemos decir ¿Cuál es el cierre del atributo A?

Esto es equivalente a decir, ¿a que atributos puedo llegar a partir de A?

**Imagen:**



Por tanto el cierre del atributo A es A BC E

Existe un truco para poder decir su dos conjntos deDF son equivalentes podemos aplicar algun axioma a un conjunto a y obtenemos una nueva DF y al mismo tiempo aplicamos algu otro axioma al conjunto b y obtenemos la misma nueva dependencia y asi sucesivamente podemos decir que son equivalentes sin calcular todo el cierre

NORMALIZACION

**Luis-**

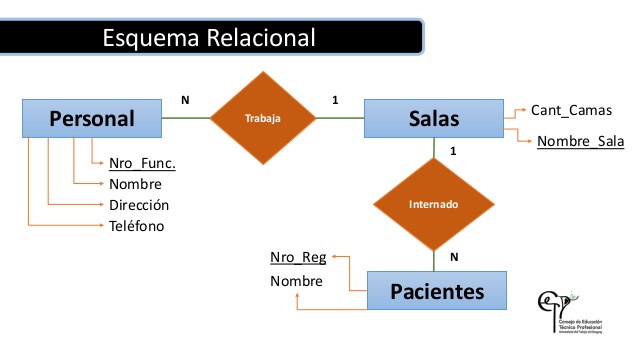
Relación

INSTANCIA

INSTANCIA

Esquema de interrelación entre atributos

Insertar imagen:



Interrelaciones existentes entre los atributos del mundo real.

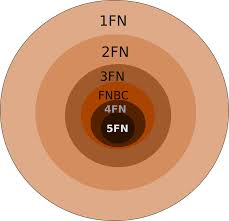
La normalización es una técnica formal para analizar las relaciones basándose en las claves primarias y las dependencias funcionales (codd 1972)

La técnica envuelve una serie de reglas que pueden ser utilizadas en una sola relación o en las tablas contenidas en una base de datos.

Cuando en una relación no se cumplen las reglas de normalización, las relaciones se deberán descomponerse en relaciones que satisfagan individualmente los requisitos de normalización.

Empezamos por definir que es una llave y una superllave,

Una superllave es es un conjunto de atributos dentro de una relación tal que no existen 2 tuplas iguales, una llave “k” es una superllave que cumple que si se le quita algunos de sus atributos deja de ser superllave. Y si una relación tiene más de una llave, cada una de ellas es una llave candidata, una de ellas es designada arbitrariamente como llave primaria, el resto son secundarias.



En la forma normal de BOYCE & CODD (BCNF) dice que en un esquema de relación R está en BCNF sí, siempre que una dependencia funcional “X” que determina a “A” se cumple en la relación R, entonces “X” es una superllave de la relación. (si está en 3FN y los determinantes son llaves).

EJEMPLLO

Sea R (nombre, teléfono, ocupación, dirección)

F = {nombre -> dirección,

nombre, teléfono, ocupación -> dirección}

**Juan**- El proceso de normalización de bases de datos consiste en aplicar una serie de reglas a las relaciones obtenidas tras el paso del modelo entidad-relación al modelo relacional.

Las bases de datos relacionales se normalizan para:

1. Evitar la redundancia de los datos.
2. Evitar problemas de actualización de los datos en las tablas.
3. Proteger la integridad de los datos

Cuando no existe normalización, se presentan anomalías en la base de datos. Estos problemas que ocasionan problemas al momento de insertar, modificar o eliminar datos.

|  |  |  |
| --- | --- | --- |
| La primera forma normal (1FN) | La Segunda Forma Normal (2FN) | La Tercera Forma Normal (3FN) |
| * Requiere que los datos sean atómicos. * En otras palabras prohíbe a un campo contener más de un dato de su dominio de columna. También exige que todas las tablas deben tener una llave primaria. * Por último, indica que una tabla no debe tener atributos que acepten valores nulos. | * Para que este en 2FN debe de estar antes en 1FN * Establece que todas las dependencias parciales se deben eliminar y separar dentro de sus propias tablas. Una dependencia parcial es un término que describe a aquellos datos que no dependen de la clave de la tabla para identificarlos. | * Para que la tabla esté en 3NF, se debe cumplir que la tabla ya esté en 2NF. Adicionalmente, ningun atributo no primario de la tabla debe ser dependiende transitivamente de una clave primaria. |

Para normalizar la base necesitamos conocer:

A

B

C

A

B

C

Dependencia funcional

Dependencia transitiva

Tenemos una base de datos donde se ve que se repiten varios atributos como el nombre de los alumnos, su dirección, teléfono y su carrera por lo que esta base no está normalizada, primero crearemos la tabla

Tabla alumno

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Matricula | Nombre | Dirección | Teléfono | Materia | Num Materia | Carrera |
| 1 | Sergio | Puebla22 | 56565656 | Base de datos | 123 | Sistemas |
| 1 | Sergio | Puebla22 | 56565656 | Programación web | 234 | Sistemas |
| 1 | Sergio | Puebla22 | 56565656 | Programación visual | 234 | Sistemas |
| 2 | Ana | Reforma1 | 23232323 | Base de datos | 123 | Mecatrónica |

**Primera forma normal**

Para la primera forma normal separamos en dos tablas para que no se repitan datos usando Matricula como la llave que conecta ambas tablas, la tabla a\_info y la tabla a\_materia, y aquí vemos que los alumnos ya no se repiten

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Matricula | Nombre | Dirección | Teléfono | Carrera |
| 1 | Sergio | Puebla22 | 56565656 | Sistemas |
| 2 | Ana | Reforma1 | 23232323 | Mecatrónica |

|  |  |  |
| --- | --- | --- |
| Matricula | Materia | Num Materia |
| 1 | Base de datos | 123 |
| 1 | Programación web | 234 |
| 1 | Programación visual | 234 |
| 2 | Base de datos | 123 |

**Segunda forma normal**

Para la segunda forma normal como Materia solo tiene dependencia con el Num de Materia se separa en otra tabla

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Matricula | Nombre | Dirección | Teléfono | Carrera |
| 1 | Sergio | Puebla22 | 56565656 | Sistemas |
| 2 | Ana | Reforma1 | 23232323 | Mecatrónica |

|  |  |
| --- | --- |
| Matricula | Num Materia |
| 1 | 123 |
| 1 | 234 |
| 1 | 234 |
| 2 | 123 |

|  |  |
| --- | --- |
| Materia | Num Materia |
| Base de datos | 123 |
| Programación web | 234 |
| Programación visual | 234 |
| Base de datos | 123 |

**Tercera forma normal**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Matricula | Nombre | Dirección | Teléfono | No Carrera |
| 1 | Sergio | Puebla22 | 56565656 | 1122 |
| 2 | Ana | Reforma1 | 23232323 | 3344 |

|  |  |
| --- | --- |
| No Carrera | Carrera |
| 1122 | Sistemas |
| 3344 | Mecatrónica |

|  |  |
| --- | --- |
| Matricula | Num Materia |
| 1 | 123 |
| 1 | 234 |
| 1 | 234 |
| 2 | 123 |

|  |  |
| --- | --- |
| Materia | Num Materia |
| Base de datos | 123 |
| Programación web | 234 |
| Programación visual | 234 |
| Base de datos | 123 |

La razón de ser de la carrera es por el alumno pero alumno no es ninguna llave primaria por lo que se crea una nueva llave No carrera con lo que se hace una nueva tabla

**Cuarta Forma Normal 4FN**

Una tabla está en 4NF si y solo si está en Tercera forma normal o en BCNF (Cualquiera de ambas) y no posee dependencias multivaluadas no triviales. La definición de la 4NF confía en la noción de una dependencia multivaluada.   
Una tabla con una dependencia multivaluada es una donde la existencia de dos o más relaciones independientes muchos a muchos causa redundancia; y es esta redundancia la que es suprimida por la cuarta forma normal

Ejemplos: Una relación está en 5FN si está en 4FN y además no existen restricciones impuestas por el creador de la DF. Una restricción de este tipo puede ser por ejemplo que una tabla se divida en subtablas. Esta forma normal tiene poca aplicación práctica porque dificulta el diseño ya que genera un número enorme de tablas.

**La** **quinta forma normal**(**5FN**), también conocida como **forma normal de proyección-unión** (**PJ/NF**)

[](http://3.bp.blogspot.com/-FgjxkQ1Wcpw/TmcD58DrOMI/AAAAAAAAADA/pfvdal5lBWs/s1600/4-1.png)Una relación está en 5FN si está en 4FN y además no existen restricciones impuestas por el creador de la DF. Una restricción de este tipo puede ser por ejemplo que una tabla se divida en subtablas. Esta forma normal tiene poca aplicación práctica porque dificulta el diseño ya que genera un número enorme de tablas.

