

The background of the slide features a pattern of overlapping, semi-transparent green hexagons of various sizes. In the upper right corner, there is a solid brown rectangular area. The main content is contained within a white rectangular box on the right side of the slide.

Normalización en Base de Datos

Ing. Víctor A. Fuentes T.

¿Qué es la normalización?

- Técnica de diseño de base de datos que examina las relaciones (dependencias funcionales) que existen entre los atributos.
- Evitar la redundancia de los datos.
- Evitar problemas de actualización de los datos en las tablas.
- Proteger la integridad de los datos.

Normalización

- Incluye la creación de tablas y el establecimiento de relaciones entre ellas según reglas diseñadas tanto para proteger los datos como para hacer que la base de datos sea más flexible.
- **Objetivo:** eliminar la redundancia y las dependencias incoherentes.

Dominios

- Todo atributo en una tabla tiene un dominio, el cual representa el conjunto de valores que el mismo puede tomar.
- **Dominio Simple:** Valores atómicos.
- **Dominio Compuesto:** Valor formado por diferentes dominios.
Ejemplo: Fecha (esta compuesto de día, mes y año)

Restricciones a cumplir

- El modelo relacional llama **tabla** a una relación, pero para que una tabla sea considerada como una relación tiene que cumplir algunas restricciones, como:
 - Cada tabla debe tener un nombre único.
 - No puede haber dos filas iguales.
 - No se permiten los duplicados.
 - Todos los datos en una columna deben ser del mismo tipo.

Dependencia Funcional

- Por definición, la Dependencia Funcional (DF) es un término que se deriva de la teoría matemática, tiene que ver con la dependencia de los valores de un atributo o conjunto de atributos; con los de otro atributo o conjunto de atributos.
- Es decir, un conjunto de atributos B es funcionalmente dependiente de un conjunto de atributos A, si un determinado conjunto de valores para cada atributo en A, determina un valor único para el conjunto de atributos en B.



**La notación $A \rightarrow B$
se usa para denotar que B es funcionalmente dependiente de A.**

Dependencia Funcional

EJEMPLOS:

FechaDeNacimiento \rightarrow Edad

Cedula \rightarrow Nombre, Apellido

Se dice entonces que un atributo B es funcionalmente dependiente de un atributo A, si dado un valor de A, existe solo un valor de B



Dependencia Funcional

Dependencia Funcional TOTAL: $\{A,B,C,\dots\} \rightarrow X$ es una dependencia funcional total si y solo si, al remover cualquiera de los atributos de $\{A,B,C,\dots\}$ la dependencia funcional **NO** se mantiene.

Dependencia Funcional PARCIAL: $\{A,B,C,\dots\} \rightarrow X$ es una dependencia funcional parcial si y solo si, al remover cualquiera de los atributos de $\{A,B,C,\dots\}$ la dependencia funcional **SÍ** se mantiene.

FORMAS NORMALES (FN)

Reglas para Normalizar

- Cada regla se denomina una "forma normal".
- Si se cumplen la tres primeras reglas se dice que la base de datos está en la "Tercera forma normal".
- Aunque son posibles otros niveles de normalización, la tercera forma normal se considera el máximo nivel necesario para la mayor parte de las aplicaciones.

Primera Forma Normal (1FN)

- Una relación está en primera forma normal (1FN) cuando dicha relación no tiene atributos multivaluados y ningún atributo compuesto.
 - ❑ Cada atributo es un valor único para cada tupla.
 - ❑ Esto significa que cada atributo en cada columna o celda de la tabla contiene un solo valor.
 - ❑ No es permitido repetir campos o tener grupos.

Primera Forma Normal (1NF)

- Es una relación en la que la intersección de toda fila y columna contiene un valor y sólo un valor.
- ClientRental* (clientNo, propertyNo, cName, pAddress, rentStart, rentFinish, rent, ownerNo, oName)
- Note que la tabla ClientRental está en 1NF.

<u>clientNo</u>	<u>propertyNo</u>	cName	pAddress	rentStart	rentFinish	rent	ownerNo	oName
CR76	PG4	John Kay	6Lawrence	1-Jul-09	31-Aug-11	350	CO40	Tina Musphy
CR76	PG16	John Kay	5 Novar Dr	1-sep-11	1-sep-12	450	CO93	Tony Shay
CR56	PG4	Aline Stew	6Lawrence	1-sep-11	10-dec-12	350	CO40	Tina Musphy
CR56	PG36	Aline Stew	2 Manor Rd	10-oct-12	12-jun-13	375	CO93	Tony Shaw
CR56	PG16	Aline Stew	5 Novar Dr	1-Nov-05	10-Aug-06	450	CO93	Tony Shaw

Primera Forma Normal

- Observe que el ejemplo anterior contiene información repetida entre tuplas o registros que describen a los clientes, los inmuebles alquilados y los propietarios.
- Es importante analizar si existen riesgos de una implementación de este tipo, en particular al momento de actualizar la información que contiene cada registro.
- Note que si desea cambiar el nombre del cliente (Client Name), aunque sea el mismo, debe ir varias veces a cada uno de los registros para cambiarlo. Lo mismo puede ocurrir al campo de dueños (Owner Name).
- En este caso se requiere el uso de una llave compuesta para evitar la repetición.

Segunda Forma Normal (2NF)

- Es una relación que está en primera forma normal (1FN), y en la que todo atributo que no sea de clave principal depende **funcionalmente** de manera completa de la clave principal.

Observación:

- *Hay una dependencia funcional completa si A y B son atributos de una relación donde B depende funcionalmente de manera completa de A si B depende funcionalmente de A pero no de ningún subconjunto propio de A.*

Segunda Forma Normal (2NF)

- La segunda forma normal se aplica a las relaciones con claves compuestas (de dos o más atributos).
- Una relación con una con una clave principal de un único atributo está automáticamente en 2NF.
- Si una tabla no está en 2FN se debe descomponer. Estos atributos son desplazados en relaciones separadas, con sus atributos dependientes.

Segunda Forma Normal (2NF)

- Esta relación está en 1FN.
- Analice los atributos de la tabla y determine si hay dependencia funcional total.

<u>clientNo</u>	<u>propertyNo</u>	cName	pAddress	rentStart	rentFinish	rent	ownerNo	oName
CR76	PG4	John Kay	6Lawrence	1-Jul-09	31-Aug-11	350	CO40	Tina Musphy
CR76	PG16	John Kay	5 Novar Dr	1-sep-11	1-sep-12	450	CO93	Tony Shay
CR56	PG4	Aline Stew	6Lawrence	1-sep-11	10-dec-12	350	CO40	Tina Musphy
CR56	PG36	Aline Stew	2 Manor Rd	10-oct-12	12-jun-13	375	CO93	Tony Shaw
CR56	PG16	Aline Stew	5 Novar Dr	1-Nov-05	10-Aug-06	450	CO93	Tony Shaw

Segunda Forma Normal (2NF)

- Lo anterior nos permite concluir que se pueden obtener las siguientes relaciones:

Client (clientNo, cName)

Rental (clientNo, propertyNo, rentStar, rentFinish)

PropertyOwner (propertyNo, pAddress, rent, ownerNo, oName)

<u>clientNo</u>	<u>propertyNo</u>	cName	pAddress	rentStart	rentFinish	rent	ownerNo	oName
CR76	PG4	John Kay	6Lawrence	1-Jul-09	31-Aug-11	350	CO40	Tina Musphy
CR76	PG16	John Kay	5 Novar Dr	1-sep-11	1-sep-12	450	CO93	Tony Shay
CR56	PG4	Aline Stew	6Lawrence	1-sep-11	10-dec-12	350	CO40	Tina Musphy
CR56	PG36	Aline Stew	2 Manor Rd	10-oct-12	12-jun-13	375	CO93	Tony Shaw
CR56	PG16	Aline Stew	5 Novar Dr	1-Nov-05	10-Aug-06	450	CO93	Tony Shaw

Segunda Forma Normal

Las tres relaciones mostradas están en 2NF, ya que todo atributo que no es clave principal depende funcionalmente por completo de la clave principal de la relación.

CLIENT	<u>clientNo</u>	cName
	CR76	John Kay
	CR56	Aline Stew

RENTAL	<u>clientNo</u>	<u>propertyNo</u>	rentStar	rentFinish
	CR76	PG4	1-Jul-09	31-Aug-11
	CR76	PG16	1-sep-11	1-sep-12
	CR56	PG4	1-sep-11	10-dec-12
	CR56	PG36	10-oct-12	12-jun-13
	CR56	PG16	1-Nov-05	10-Aug-06

PROPERTY OWNER	<u>propertyNo</u>	pAddress	rent	ownerNo	oName
	PG4	6Lawrence	350	CO40	Tina Murphy
	PG16	5 Novar Dr	450	CO93	Tony Shaw
	PG36	2 Manor Rd	375	CO93	Tony Shaw

Tercera Forma Normal (3NF)

- Es una relación que está en primera y segunda forma normal y en la que ningún atributo que no sea clave principal depende **transitivamente** de la clave principal.
- En otras palabras, un atributo que no es llave, no puede depender de otro que no sea llave.

Observación

- *La dependencia transitiva es aquella en la que A, B y C son atributos de una relación tales que si $A \rightarrow B$ y $B \rightarrow C$, entonces C depende transitivamente de A a través de B (supuesto que A no sea funcionalmente dependiente de B o C)*

Tercera Forma Normal (3NF)

- Los resultados de la 2NF pueden seguir teniendo redundancia.
- Por ejemplo, si desea actualizar el nombre de un propietario tal como Tony Shaw (CO93), se tienen que actualizar dos tuplas en la relación PropertyOwner. Si ello no se hace la información de la Base de Datos tendría información incoherente.

Esta dificultad está causada por una dependencia transitiva y por ende puede corregirse usando la 3NF.

<u>propertyNo</u>	pAddress	rent	ownerNo	oName
PG4	6Lawrence	350	CO40	Tina Murphy
PG16	5 Novar Dr	450	CO93	Tony Shaw
PG36	2 Manor Rd	375	CO93	Tony Shaw

Tercera Forma Normal (3NF)

Note la transitividad de la dependencia:

propertyNo \rightarrow pAdress, rent, ownerNo, oName

ownerNo \rightarrow oName (dependencia transitiva)

- Todos los atributos que no son clave principal dentro de la relación PropertyOwner dependen funcionalmente de la clave principal, con la excepción de oName que depende transitivamente de ownerNo.
- En la tercera forma normal debe romperse esta transitividad. Las otras relaciones ya están entonces en una 3NF.

Tercera Forma Normal (3NF)

De lo anterior se puede concluir que las relaciones en su 3NF serían las siguientes:

Client (clientNo, cName)

Rental (clientNo, propertyNo, rentStar, rentFinish)

PropertyforRent (propertyNo, pAddress, rent, ownerNo)

Owner (ownerNo, oName)

Tablas en Tercera Forma Normal

CLIENT

<u>clientNo</u>	cName
CR76	John Kay
CR56	Aline Stew

RENTAL

<u>clientNo</u>	<u>propertyNo</u>	rentStar	rentFinish
CR76	PG4	1-Jul-09	31-Aug-11
CR76	PG16	1-sep-11	1-sep-12
CR56	PG4	1-sep-11	10-dec-12
CR56	PG36	10-oct-12	12-jun-13
CR56	PG16	1-Nov-05	10-Aug-06

PROPERTY FOR RENT

<u>propertyNo</u>	pAddress	rent	ownerNo
PG4	6Lawrence	350	CO40
PG16	5 Novar Dr	450	CO93
PG36	2 Manor Rd	375	CO93

OWNER

<u>ownerNo</u>	oName
CO40	Tina Murphy
CO93	Tony Shaw

Referencia

Connolly, Thomas & Begg, Carolyn (2005). *Sistemas de bases de datos*. Pearson Education S.A.: Madrid.