



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO  
FACULTAD DE CIENCIAS  
FUNDAMENTOS DE BASES DE DATOS

# Normalización de Bases de Datos

**Gerardo Avilés Rosas**  
gar@ciencias.unam.mx

- Uno de los principales problemas que se presentan cuando se convierten directamente diseños de BDs del **modelo E/R** → **Modelo Relacional** es la **redundancia**.
- La **redundancia** consiste en que un *hecho* se repita en más de una **tupla**.
- Una de las causas más común en redundancia es intentar agrupar en una relación las propiedades **multivaluadas** y **univaluadas** de un objeto.



Vamos a suponer que tenemos la siguiente relación:

Sucursal	Alcaldía	Activo	Cliente	NumPrestamo	Importe
<b>Centro</b>	Cuauhtémoc	\$1,800 M	Santos	<b>P-17</b>	\$200,000
<b>Copilco</b>	Coyoacán	\$420 M	Gómez	<b>P-23</b>	\$400,000
<b>Viveros</b>	Coyoacán	\$340 M	López	<b>P-15</b>	\$300,000
<b>Centro</b>	Cuauhtémoc	\$1,800 M	Toledo	<b>P-14</b>	\$300,000
<b>Eugenia</b>	Benito Juárez	\$80 M	Santos	<b>P-93</b>	\$100,000
<b>Zapata</b>	Benito Juárez	\$1,600 M	Pérez	<b>P-11</b>	\$180,000
<b>San Ángel</b>	Álvaro Obregón	\$60 M	Vázquez	<b>P-29</b>	\$240,000
<b>San Fernando</b>	Tlalpan	\$740 M	López	<b>P-16</b>	\$260,000
<b>Centro</b>	Cuauhtémoc	\$1,800 M	González	<b>P-23</b>	\$400,000
<b>Viveros</b>	Coyoacán	\$340 M	Rodríguez	<b>P-25</b>	\$500,000

¿Existe algún tipo de redundancia?

- **Redundancia de información:**

Agregar un nuevo préstamo o cambiar la dirección de una sucursal (duplicar información).

- **Incapacidad para representar cierta información:**

¿Cómo dar de alta una nueva sucursal?

¿Qué sucede si no hubiera préstamos?

- **Eliminar información de más:**

¿Qué puede suceder al borrar algún préstamo?

¿Qué sucede si fuera el único préstamo otorgado en la sucursal?

Es decir, se pueden presentar problemas (**anomalías**) en la base de datos.

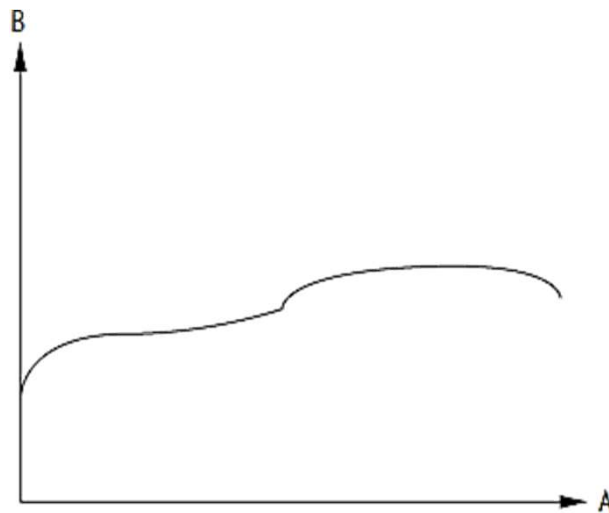
Una **anomalía** es un problema que surge en una Base de Datos. Las principales anomalías que se pueden encontrar son:

- **Redundancia:** La información puede repetirse innecesariamente en varias *tuplas*.
- **Anomalías de actualización:** Podemos cambiar información en una *tupla* y dejarla inalterada en otra.
- **Anomalías de eliminación:** Si un conjunto de valores queda vacío, podemos perder información adicional como efecto secundario.

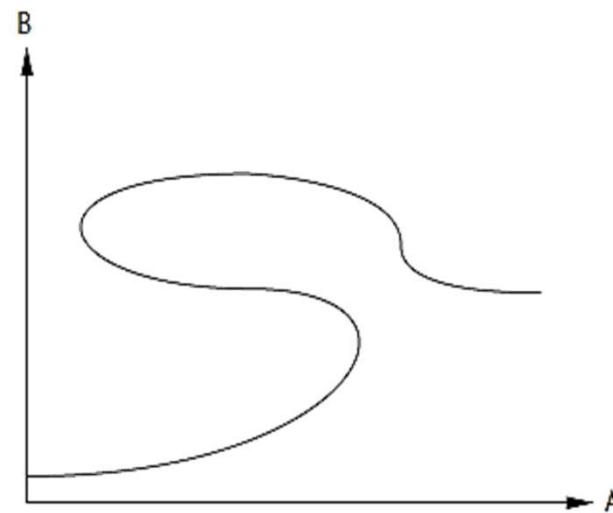


# Dependencias funcionales

- Las **dependencias funcionales** ayudan a especificar formalmente cuándo un diseño **es correcto**.
- Se trata de una relación unidireccional entre 2 atributos de tal forma que en un momento dado, para cada valor único de A, sólo un valor de B se asocia con él a través de la relación.



A functionally determines B. Each value of A corresponds to only one value of B.



A does not functionally determine B. Some values of A correspond to more than one value of B.

- Una **DF** que denotaremos por  $X \rightarrow Y$ , sucede entre dos conjuntos de atributos X e Y que son subconjuntos de R.

# ...Dependencias funcionales

Por ejemplo, vamos a suponer que tenemos la relación:

**Pedidos** (*idCliente, nombre, paterno, materno, calle, ciudad, estado, CP, teléfono, noPedido fechaOrden, noArticulo, producto, precio, ¿enviado?*)

Se espera, por ejemplo que:

**idCliente** → nombre, paterno, materno, calle, ciudad, estado, CP, teléfono

**noPedido** → idCliente, fechaOrden

**noArticulo** → producto, precio

**noPedido + noArticulo** → ¿enviado?

*Aunque los valores de los atributos pueden cambiar, en cualquier momento, sólo hay uno.*

Las DFs se utilizan para:

- **Especificar restricciones sobre el conjunto de relaciones.**
- **Examinar las relaciones y determinar si son legales bajo un conjunto de dependencias funcionales dado.**

# ...Dependencias funcionales

- Si **X** es una **llave** de **R** esto implica que  $X \rightarrow Y$  para cualquier subconjunto de atributos **Y** de **R**.
- Si  $X \rightarrow Y$  en **R**, esto no implica que  $Y \rightarrow X$
- Una **DF** es una **propiedad semántica** de los atributos, la cual se debe cumplir para la **extensión** en una relación.
- Las *extensiones* que satisfacen las restricciones de DFs se denominan **extensiones legales** o **estados legales** debido a que obedecen las restricciones de la DF.

A	B	C	D
a1	b1	c1	d1
a1	b2	c1	d2
a2	b2	c2	d2
a2	b3	c2	d3
a3	b3	c2	d4

¿ $A \rightarrow B$ ?

¿ $A \rightarrow C$ ?

¿ $AB \rightarrow CD$ ?



Una **llave** puede definirse como un conjunto de atributos  $\{A_1, A_2, \dots, A_n\}$  tales que:

- Determinan funcionalmente cualquier otro atributo de la relación. Es decir, es imposible para dos *tuplas* distintas de R coincidir en todos los atributos  $\{A_1, A_2, \dots, A_n\}$ .
- Ningún subconjunto propio de  $\{A_1, A_2, \dots, A_n\}$  determina funcionalmente a los otros atributos de R, es decir, debe ser mínimo.
- Una **superllave** es un conjunto de atributos que contiene una **llave**.



# Reglas de inferencia de Armstrong

---

- Fueron desarrolladas por **William W. Armstrong** en **1974**: ***Dependency Structures of Data Base Relationships***.
- Se trata de un conjunto de reglas que permiten deducir **todas** las dependencias funcionales que tienen lugar en un conjunto de atributos dados, como **consecuencia** de aquellas que se asumen como **ciertas**, a partir del conocimiento del problema.
- Este resultado permite establecer un conjunto de algoritmos sencillos para:
  1. **Encontrar el conjunto cerrado de un conjunto de dependencias funcionales.**
  2. **Encontrar la equivalencia lógica de esquemas.**
  3. **Deducir dependencias funcionales.**
  4. **Calcular las llaves de un esquema.**

# ...Reglas de inferencia de Armstrong

1. Regla de la **reflexividad**: Si  $Y \subseteq X$ , entonces  $X \rightarrow Y$
2. Regla del **aumento**:  $\{X \rightarrow Y\} \Rightarrow XZ \rightarrow YZ$
3. Regla de la **transitividad**:  $\{X \rightarrow Y, Y \rightarrow Z\} \Rightarrow X \rightarrow Z$
4. Regla de la **descomposición**:  $\{X \rightarrow YZ\} \Rightarrow X \rightarrow Y$  y  $X \rightarrow Z$
5. Regla de la **unión**:  $\{X \rightarrow Y, X \rightarrow Z\} \Rightarrow X \rightarrow YZ$
6. Regla de la **pseudo-transitividad**:  $\{X \rightarrow Y, WY \rightarrow Z\} \Rightarrow WX \rightarrow Z$

## Ejemplo

Sean

$R = (A, B, C, G, H, I)$  y  $F = \{A \rightarrow B, A \rightarrow C, CG \rightarrow H, CG \rightarrow I, B \rightarrow H\}$

Algunos miembros de  $F^+$  serían:

- Unión:  $A \rightarrow BC, CG \rightarrow HI$
- Pseudo-transitividad:  $AG \rightarrow I, AG \rightarrow H$
- Aumento:  $AG \rightarrow CG$
- Transitividad:  $A \rightarrow H$

# Cerradura del conjunto de atributos

- Para crear las dependencias funcionales de un esquema, requerimos:
  - ❑ **Especificar a partir de la semántica de los atributos de R, el conjunto F de DFs.**
  - ❑ **Usando las reglas de inferencia de Armstrong obtener otras DFs.**
- Para obtener todas las dependencias funcionales de manera sistemática:
  - ❑ **Determinar el conjunto de atributos X del lado izquierdo de alguna DF en F.**
  - ❑ **Determinar el conjunto  $X^+$  de todos los atributos que son dependientes de X.**
- Dado  $\{A_1, A_2, \dots, A_n\}$  un conjunto de atributos y F un conjunto de DFs, la **cerradura del conjunto de atributos**  $(\{A_1, A_2, \dots, A_n\}^+)$  bajo las dependencias en F es el conjunto de atributos B tales que cada relación que satisface todas las dependencias en F también satisface:

$$A_1, A_2, \dots, A_n \rightarrow B$$

# Algoritmo para calcular $X^+$ bajo $F$

**Repetir**

**anterior $X^+$  =  $X^+$ ;**

**Para cada  $Y \rightarrow Z$  en  $F$  hacer**

**Sí  $Y \subseteq X^+$  entonces  $X^+ = X^+ \cup Z$ ;**

**hasta que anterior $X^+$  =  $X^+$ ;**

Ejemplos:

1. Sean  $R = \{A, B, C, G, H, I\}$  y  $F = \{A \rightarrow B, A \rightarrow C, CG \rightarrow H, CG \rightarrow I, B \rightarrow H\}$ , ¿cuál sería la cerradura **AG**?

**$\{AG\}^+ = \{AGBCIH\}$**

2. Sea  $R = \{A, B, C, D, E, F\}$  y  $F = \{AB \rightarrow C, BC \rightarrow D, D \rightarrow E, CF \rightarrow B\}$ , ¿cuál sería la cerradura **AB**?

**$\{AB\}^+ = \{ABCDE\}$**

Esto implica que:  **$AB \rightarrow CDE$**

Si queremos probar si una dependencia funcional  $A_1, A_2, \dots, A_n \rightarrow B$ , se deduce de un conjunto de dependencias  $F$ , debemos calcular  $\{A_1, A_2, \dots, A_n\}^+$ , si  $B$  está ahí, entonces la DF se deduce del conjunto  $F$ , en caso contrario no es deducida de  $F$ .

Por ejemplo, sea  $R = \{A, B, C, D, E, F\}$  y  $F = \{AB \rightarrow C, BC \rightarrow D, D \rightarrow E, CF \rightarrow B\}$

- Probar que  $AB \rightarrow D$

Se empieza por calcular  $\{A, B\}^+ = \{A, B, C, D, E\}$ , como  $D \in \{A, B\}^+$  entonces ésta se deduce

- Probar que  $D \rightarrow A$

Se tiene que  $D^+ = \{D, E\}$ , como  $A$  no está en  $D^+$ , entonces la DF no se deduce de  $F$ .

Y ahora...  
¿Cómo normalizo una  
Base de Datos?



No estés muy orgulloso de haber comprendido estas notas.

La habilidad para manejar la Normalización de BD es insignificante comparado con el poder de la Fuerza.

