

# BASE DE DATOS

16/04/2024 y 20/04/2024

UPN.EDU.PE

# **Semana 04**



## **Normalización de bases de datos**

# Presentación de la sesión



## Logro de sesión

Al término de la sesión los estudiantes podrán identificar los conceptos necesarios para crear un modelo de datos entidad-relación extendido y determinar la lógica necesaria para aplicar Normalización, de manera clara y precisa de acuerdo a los esquemas y procedimientos presentados en clase. De la misma forma, resolverán problemas de diagramas entidad-relación extendido y de normalización.

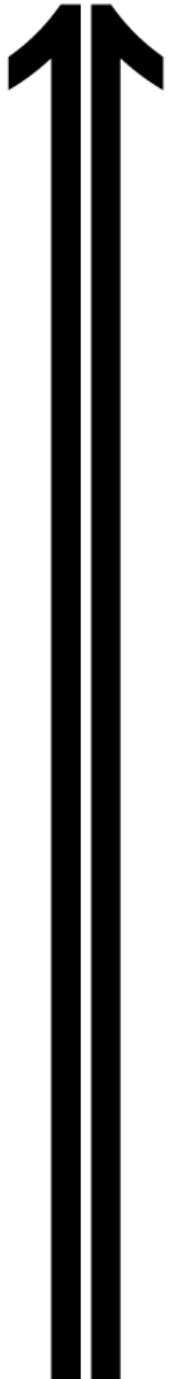
## Temario

- Normalización de una base de datos
- Redundancia de datos
- Formas normales



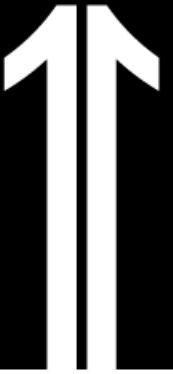
¿Qué entiende por normalización de datos?

¿Qué entiende por redundancia de datos?



# **Anomalías en las operaciones de datos**

# ANOMALÍAS EN LAS OPERACIONES DE DATOS



## Anomalías en las operaciones de datos

- Efecto secundario inesperado de una operación de fila
  - consecuencia no intencional (a veces bueno, a veces malo)
- Anomalía de modificación:
  - No se pueden modificar sólo los datos deseados
  - Debe modificar más de los datos deseados
- Causa:
  - Redundancia: hechos almacenados varias veces
  - Solución: Eliminar las redundancias (no deseados) para eliminar anomalías



## TABLA DE BASE DE DATOS DE UNA UNIVERSIDAD

<b>StdNo</b>	<b>StdClas</b>	<b>ProgNo</b>	<b>ProgAnio</b>	<b>ProgNota</b>	<b>CursoNo</b>	<b>CursoDesc</b>
S1	JUN	O1	2013	3.5	C1	DB
S1	JUN	O2	2013	3.3	C2	VB
S2	JUN	O3	2014	3.1	C3	OO
S2	JUN	O2	2013	3.4	C2	VB

- Toda la base de datos en una tabla (error de principiante)
- ¿encuentra algún error de inserción, eliminación o actualización en la tabla?



## TABLA DE BASE DE DATOS DE UNA UNIVERSIDAD

StdNo	StdClass	ProgNo	ProgAnio	ProgNota	CursoNo	CursoDesc
S1	JUN	O1	2013	3.5	C1	DB
S1	JUN	O2	2013	3.3	C2	VB
S2	JUN	O3	2014	3.1	C3	OO
S2	JUN	O2	2013	3.4	C2	VB

- ¿Qué pasa si queremos insertar un curso?
- No se puede insertar un curso nuevo sin un estudiante sin inscribirlo en una programado (ProgNo parte de PK)
- La creación del curso debe ser independiente





## TABLA DE BASE DE DATOS DE UNA UNIVERSIDAD

<u>StdNo</u>	StdClass	<u>ProgNo</u>	ProgAnio	ProgNota	CursoNo	CursoDesc
S1	JUN	O1	2013	3.5	C1	DB
S1	JUN	O2	2013	3.3	C2	VB
S2	JUN	O3	2014	3.1	C3	OO
S2	JUN	O2	2013	3.4	C2	VB

- ¿Qué pasa si queremos actualizar la descripción de un curso?
- Cambiar la descripción del curso; debemos cambiar la descripción en cada inscripción del curso
- La actualización se debe realizar en un solo lugar



## TABLA DE BASE DE DATOS DE UNA UNIVERSIDAD

<b>StdNo</b>	<b>StdClas</b>	<b>ProgNo</b>	<b>ProgAnio</b>	<b>ProgNota</b>	<b>CursoNo</b>	<b>CursoDesc</b>
S1	JUN	O1	2013	3.5	C1	DB
S1	JUN	O2	2013	3.3	C2	VB
S2	JUN	O3	2014	3.1	C3	OO
S2	JUN	O2	2013	3.4	C2	VB

- ¿Qué pasa si queremos eliminar la inscripción del estudiante S1 en el curso C1 que esta en la primera fila?
- Al eliminar la primera fila; perderemos información sobre el curso C1 y la programación O1
- Debe ser independiente la operacion



# VENTAJAS Y DESVENTAJAS EN LAS ANOMALÍAS DE DATOS

- Más fácil de consultar: sin uniones (joins)
- Más difícil de cambiar: actualizaciones, eliminaciones, insert con problemas que acabamos de analizar.



# **DEPENDENCIAS FUNCIONALES**



## DEPENDENCIAS FUNCIONALES

- Afirmaciones o Restricciones sobre los datos (reglas de negocio)
- Una dependencia funcional es una restricción de datos en una tabla
  - Clave Primaria, las claves foráneas
  - Una restricción implica una comparación de columnas
- Las restricciones deben plasmar las limitaciones que existen en el modelo de negocio
- Es necesario conocer las reglas del negocio
- Las restricciones son normativas
- No se puede automatizar, es diferente en cada proceso, en cada organización
- La dependencia funcional es una restricción entre una o más columnas de una tabla

## ↑↑ NOTACIÓN

- Clasificación:  $X \rightarrow Y$
- $X$  (funcionalmente) determina  $Y$
- **Para cada valor  $X$ , hay como máximo un valor  $Y$**
- $\text{StdNo} \rightarrow \text{StdCiudad}$  si cada valor de  $\text{stdNo}$  tiene como máximo un valor  $\text{StdCiudad}$
- $X$ : determinante

## EJEMPLO

<b>StdNo</b>	StdClas	<b>ProgNo</b>	ProgAnio	ProgNota	CursoNo	CursoDesc
S1	JUN	O1	2013	3.5	C1	DB
S1	JUN	O2	2013	3.3	C2	VB
S2	JUN	O3	2014	3.1	C3	OO
S2	JUN	O2	2013	3.4	C2	VB

- Ejemplo ProgNo -> ProgAnio
  - Hay como máximo un año para cada programación
  - Colocar ProgNo y ProgAnio en la misma tabla: ProgNo es único
- Ejemplo ProgNo, StdNo -> ProgNota
  - Hay como máximo una nota para cada combinación de ProgNo y StdNo
  - Coloque ProgNo y StdNo en la misma tabla con ProgNota: <ProgNo, StdNo> es único

## EJEMPLO: ENCONTRANDO DEPENDENCIAS FUNCIONALES

<b>StdNo</b>	StdClas	<b>ProgNo</b>	ProgAnio	ProgNota	CursoNo	CursoDesc
S1	JUN	O1	2013	3.5	C1	DB
S1	JUN	O2	2013	3.3	C2	VB
S2	JUN	O3	2014	3.1	C3	OO
S2	JUN	O2	2013	3.4	C2	VB

- Primero analicemos las columnas que forman la clave primaria (individualmente) (analizamos contra todas las columnas)
- **StdNo** → StdClas :SI      **StdNo** → ProgAnio :NO      **StdNo** → ProgNota :NO
- **StdNo** → CursoNo :NO      **StdNo** → CursoDesc :NO
- Solo queda como dependencia funcional **SrtNo→StdClas** por lo que ya no analizaremos esa columna (esos dos deben ir juntos en la misma tabla)



## EJEMPLO: ENCONTRANDO DEPENDENCIAS FUNCIONALES

<u>StdNo</u>	StdClas	<u>ProgNo</u>	ProgAnio	ProgNota	CursoNo	CursoDesc
S1	JUN	O1	2013	3.5	C1	DB
S1	JUN	O2	2013	3.3	C2	VB
S2	JUN	O3	2014	3.1	C3	OO
S2	JUN	O2	2013	3.4	C2	VB

- Analizamos la siguiente columna de la clave primaria
- **ProgNo** → ProgAnio :SI **ProgNo** → ProgNota :NO, **ProgNo** → CursoNo :SI, **ProgNo** → CursoDesc :SI
- Queda como dependencia funcional lo siguiente:
- **ProgNo** → **ProgAnio, CursoNo, CursoDesc** (deben ir juntos en una tabla)

## EJEMPLO: ENCONTRANDO DEPENDENCIAS FUNCIONALES

<u>StdNo</u>	StdClas	<u>ProgNo</u>	ProgAnio	ProgNota	CursoNo	CursoDesc
S1	JUN	O1	2013	3.5	C1	DB
S1	JUN	O2	2013	3.3	C2	VB
S2	JUN	O3	2014	3.1	C3	OO
S2	JUN	O2	2013	3.4	C2	VB

- Como ya acabamos las columnas individuales ahora analicemos la combinación de ambas contra las columnas que aun no tienen asignada una dependencia funcional
- **StdNo,ProgNo** → ProgNota :SI
- Queda como dependencia funcional lo siguiente:
- **StdNo,ProgNo** → **ProgNota** (deben ir juntos en una tabla)

## EJEMPLO: ENCONTRANDO DEPENDENCIAS FUNCIONALES

<b>StdNo</b>	StdClas	<b>ProgNo</b>	ProgAnio	ProgNota	CursoNo	CursoDesc
S1	JUN	O1	2013	3.5	C1	DB
S1	JUN	O2	2013	3.3	C2	VB
S2	JUN	O3	2014	3.1	C3	OO
S2	JUN	O2	2013	3.4	C2	VB

- Las dependencias funcionales identificadas hasta el momento (las que deben ir en la misma tabla) son:
- **SrtNo → StdClas**
- **ProgNo → ProgAnio, CursoNo, CursoDesc**
- **StdNo, ProgNo → ProgNota**

## EJEMPLO: ENCONTRANDO DEPENDENCIAS FUNCIONALES

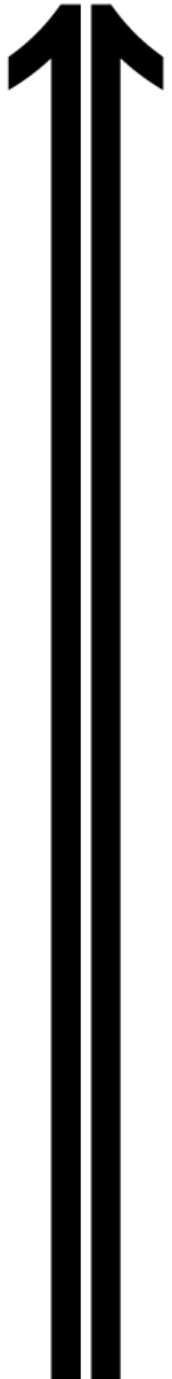
<b>StdNo</b>	StdClas	<b>ProgNo</b>	ProgAnio	ProgNota	CursoNo	CursoDesc
S1	JUN	O1	2013	3.5	C1	DB
S1	JUN	O2	2013	3.3	C2	VB
S2	JUN	O3	2014	3.1	C3	OO
S2	JUN	O2	2013	3.4	C2	VB

- **ProgNo → ProgAnio, CursoNo, CursoDesc**
- ¿tendrá dependencias funcionales entre sus atributos (dependencias transitivas)?
- ProgAnio→CursoNo :NO, ProgAnio→ CursoDesc :NO
- CursoNo→CursoDesc :SI Deben ir en la misma tabla (por lo que se saca un atributo de la dependencia funcional)
- **ProgNo → ProgAnio, CursoNo**
- **CursoNo → CursoDesc**

## EJEMPLO: ENCONTRANDO DEPENDENCIAS FUNCIONALES

StdNo	StdClas	ProgNo	ProgAnio	ProgNota	CursoNo	CursoDesc
S1	JUN	O1	2013	3.5	C1	DB
S1	JUN	O2	2013	3.3	C2	VB
S2	JUN	O3	2014	3.1	C3	OO
S2	JUN	O2	2013	3.4	C2	VB

- Al final las tablas resultantes (dependencias funcionales y transitivas) :
- **SrtNo  $\rightarrow$  StdClas**
- **ProgNo  $\rightarrow$  ProgAnio, CursoNo**
- **CursoNo  $\rightarrow$  CursoDesc**
- **StdNo, ProgNo  $\rightarrow$  ProgNota**



# **Normalización de una base de datos**

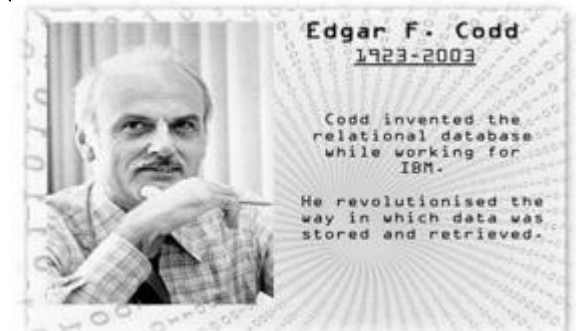
# ¿QUÉ ES UNA NORMALIZACIÓN?

Es una regla para obtener estructuras de datos eficientes que eviten las **anomalías** de actualización.

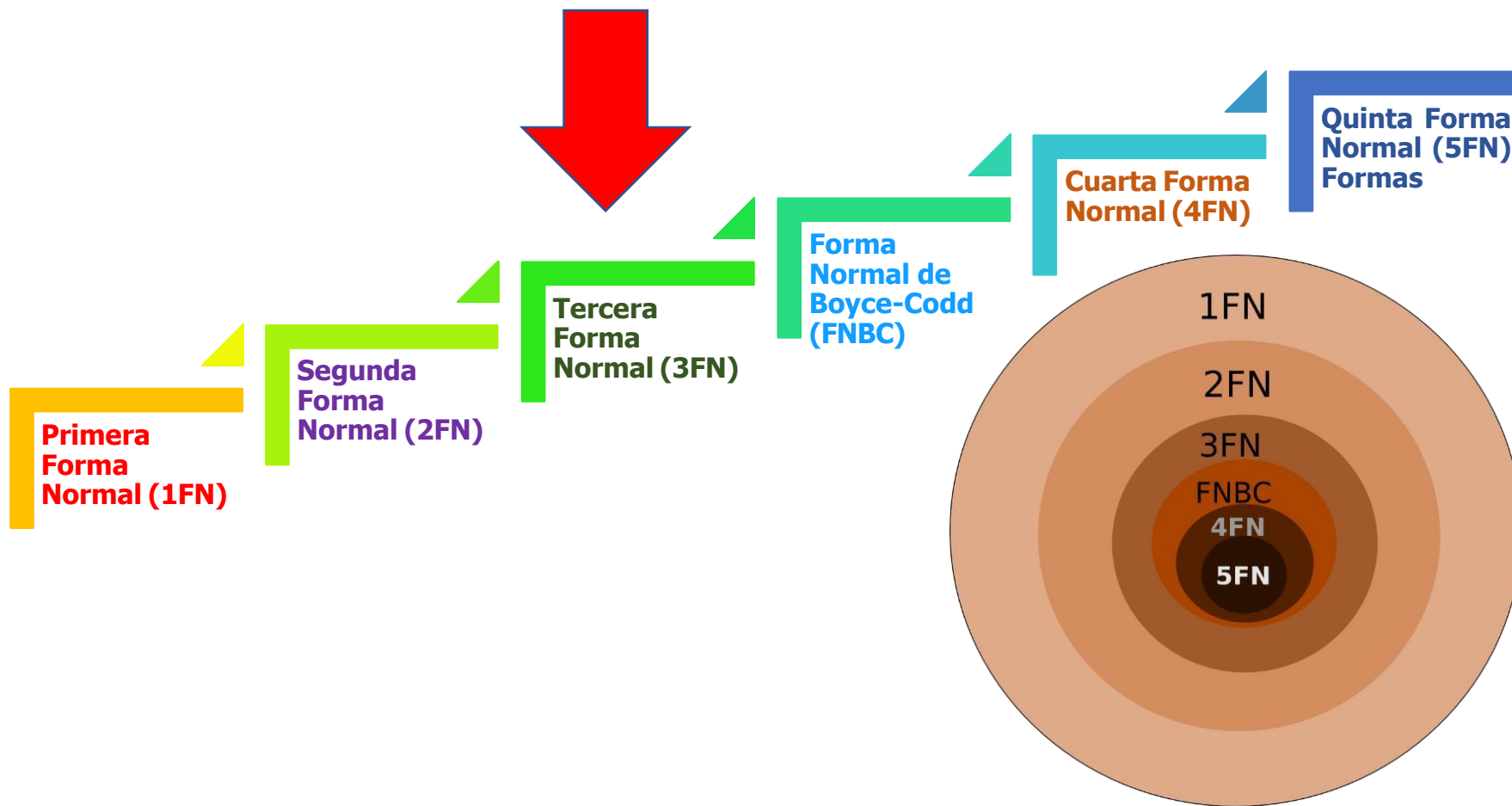
Es el proceso que consiste en aplicar una serie de reglas a las relaciones obtenidas tras el paso del modelo E-R (entidad-relación) al modelo relacional.

Es el concepto que ha introducido Edgar Frank Codd pensando, para aplicarse a sistemas relacionales.

Es la expresión formal del modo de realizar un buen diseño

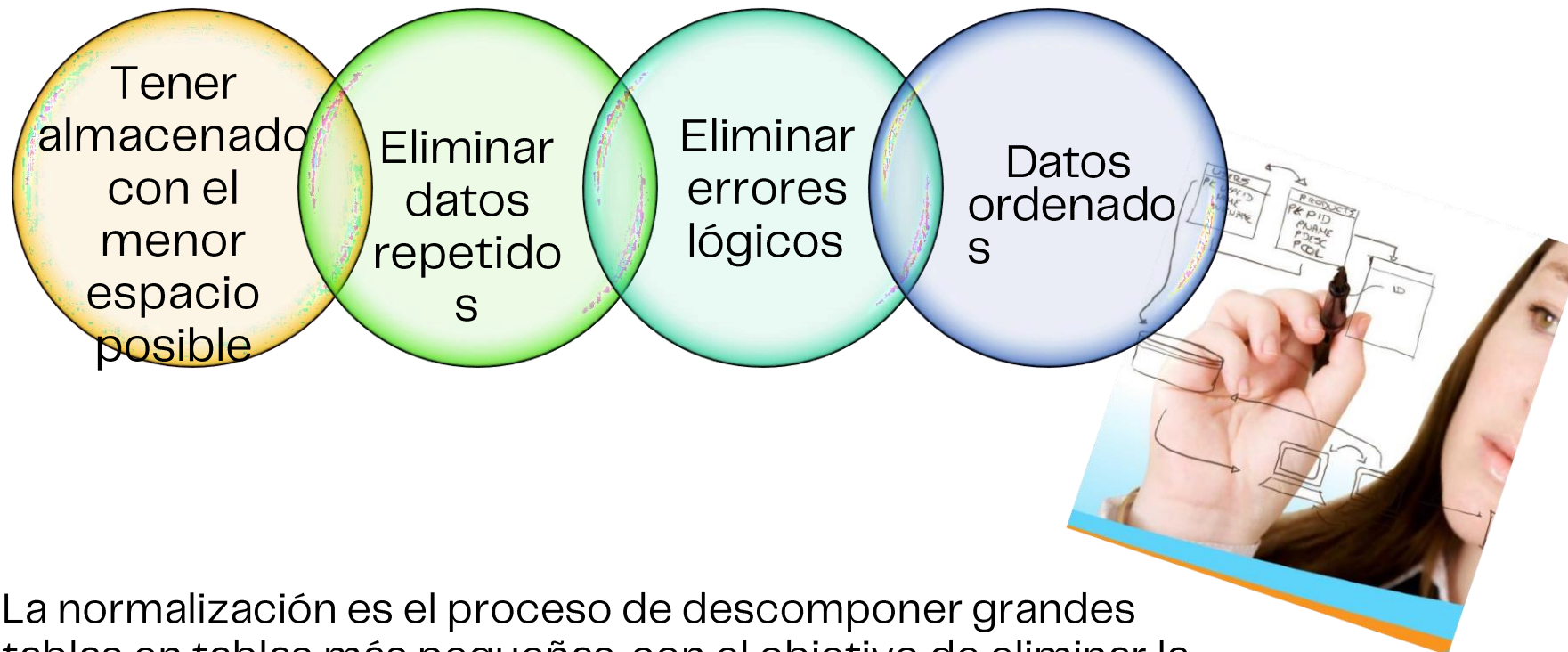


# FASES





# OBJETIVOS DE LA NORMALIZACIÓN



- La normalización es el proceso de descomponer grandes tablas en tablas más pequeñas, con el objetivo de eliminar la redundancia de datos.
- Para ello se usan las **formas normales**.

# PRIMERA FORMA NORMAL

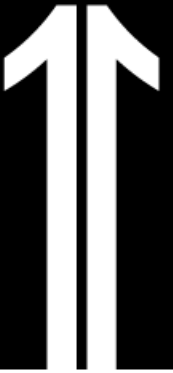


- No hay grupos en los datos (valores únicos)
- Una tabla de empleados con sus hijos y sus fechas de cumpleaños

NroEmp	Nombre	Apellido	Hijos	Cumpleaños
1001	Jane	Doe	Mary, Sam	01/09/02, 15/05/14
1002	Jhon	Doe	Lisa, David	01/09/00, 15/04/01
1003	Jane	Smith	Jhon, Pat, Lee, Mary	10/05/04, 10/12/00, 06/06/06, 21/08/04
1004	Jhon	Smith	Michael	07/04/06
1005	Jane	Jones	Edward, Martha	21/10/05, 15/10/99

- Primer problema: no hay una forma exacta de saber que fecha de cumpleaños corresponde a cada hijo
- Para buscar que empleados tuvieron hijos después del año 2005 es muy complicado o no?

# PRIMERA FORMA NORMAL

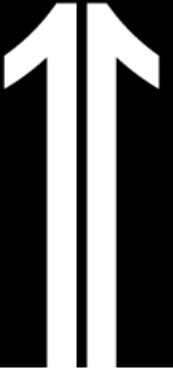


- Solución:

NroEmp	Nombre	Apellido	Hijos1	Cumpleaños1	Hijos2	Cumpleaños2	Hijos3	Cumpleaños3
1001	Jane	Doe	Mary	01/09/02	Sam	15/05/14		
1002	Jhon	Doe	Lisa	01/09/00	David	15/04/01		
1003	Jane	Smith	Jhon	10/05/04	Pat	10/12/00	Lee	06/06/06
1004	Jhon	Smith	Michael	07/04/06				
1005	Jane	Jones	Edward	21/10/05	Martha	15/10/99		

- Se limitan 3 hijos
- El cuarto hijo de Jane Smith, no tiene lugar
- Empleados no tienen 3 hijos tienen menos perdida de espacio
- Búsqueda, que empleados no tienen hijos, buscar en todos los campos

# PRIMERA FORMA NORMAL



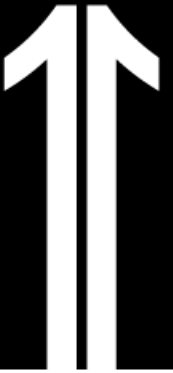
- Solución:

NroEmp	Nombre	Apellido
1001	Jane	Doe
1002	Jhon	Doe
1003	Jane	Smith
1004	Jhon	Smith
1005	Jane	Jones

NroEmp	Nombre	fechaNacimiento
1001	Mary	01/01/02
1001	Sam	15/04/04
1002	Lisa	01/01/00
1002	David	10/05/04
1002	Jhon	10/12/05

- No limite de numero de hijos
- No perdida de espacio
- Buscar un hijo especifico es mas fácil

# PRIMERA FORMA NORMAL

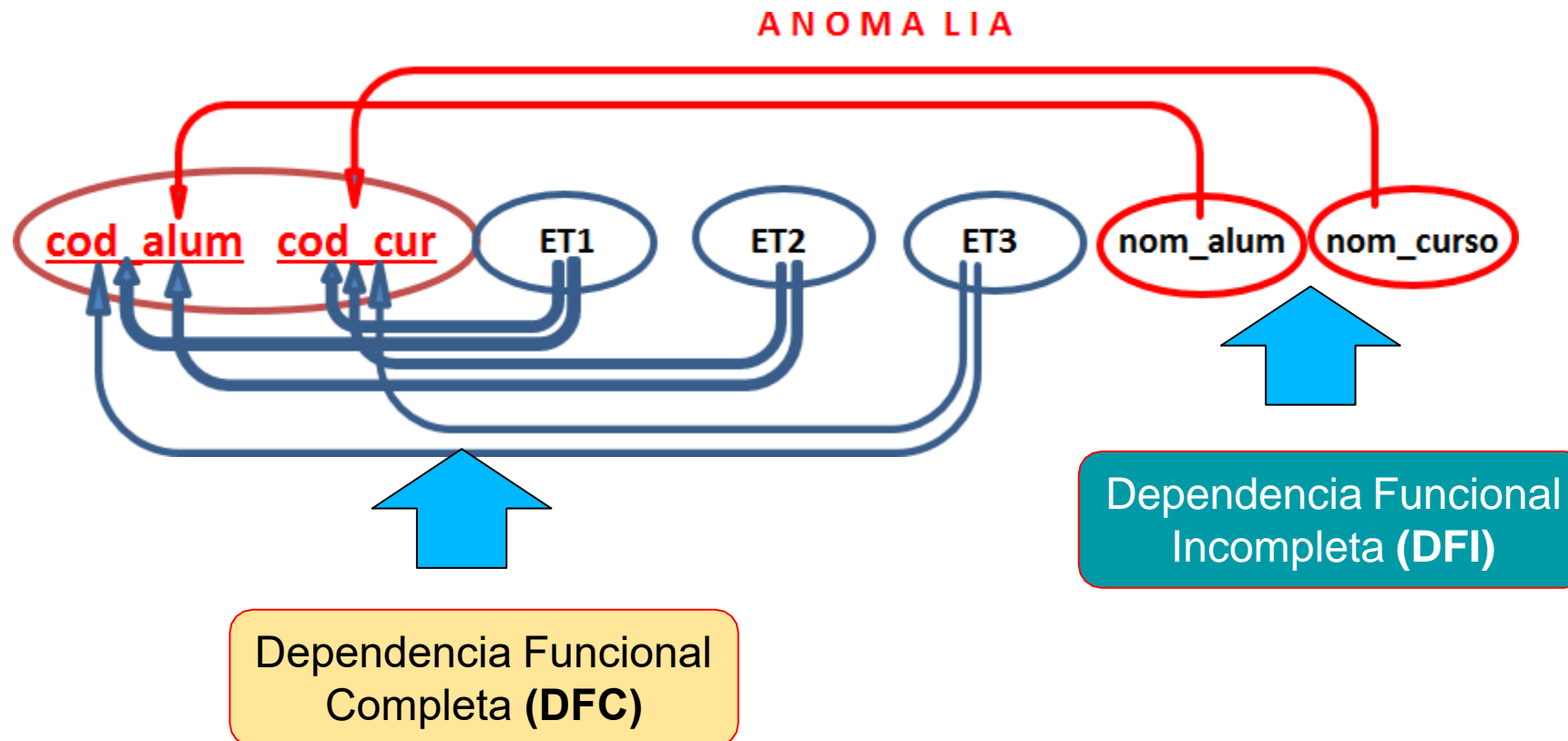


- Esta en primera forma normal?

<u>StdNo</u>	StdClas	<u>ProgNo</u>	ProgAnio	ProgNota	CursoNo	CursoDesc
S1	JUN	O1	2013	3.5	C1	DB
S1	JUN	O2	2013	3.3	C2	VB
S2	JUN	O3	2014	3.1	C3	OO
S2	JUN	O2	2013	3.4	C2	VB

- Tiene problemas...?
- Inserción, modificación, eliminación
- SI

# SEGUNDA FORMA NORMAL



# SEGUNDA FORMA NORMAL



- Debe estar en primera forma normal
- Los atributos de la tabla deben ser funcionalmente dependientes de su clave primaria.

<u>DNI</u>	<u>CURSO</u>	NOMBRE	APELLIDO	NOTA
12121219A	34	Pedro	Valiente	9
12121219A	25	Pedro	Valiente	8
3457775G	34	Ana	Fernandez	6
5674378J	25	Sara	Crespo	7
5674378J	34	Sara	Crespo	3

- **DNI→Nombre :SI** **DNI→Apellido :SI** DNI→nota :NO
- CURSO→Nota :NO
- **DNI,CURSO→Nota :SI**

# SEGUNDA FORMA NORMAL



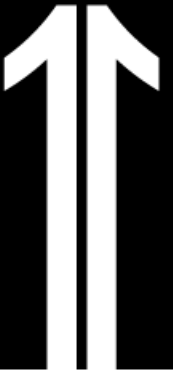
- Las tablas resultantes serian:

<u>DNI</u>	NOMBRE	APELLIDO
12121219A	Pedro	Valiente
12121219A	Pedro	Valiente
3457775G	Ana	Fernandez
5674378J	Sara	Crespo
5674378J	Sara	Crespo

<u>DNI</u>	<u>CURSO</u>	NOTA
12121219A	34	9
12121219A	25	8
3457775G	34	6
5674378J	25	7
5674378J	34	3



# SEGUNDA FORMA NORMAL

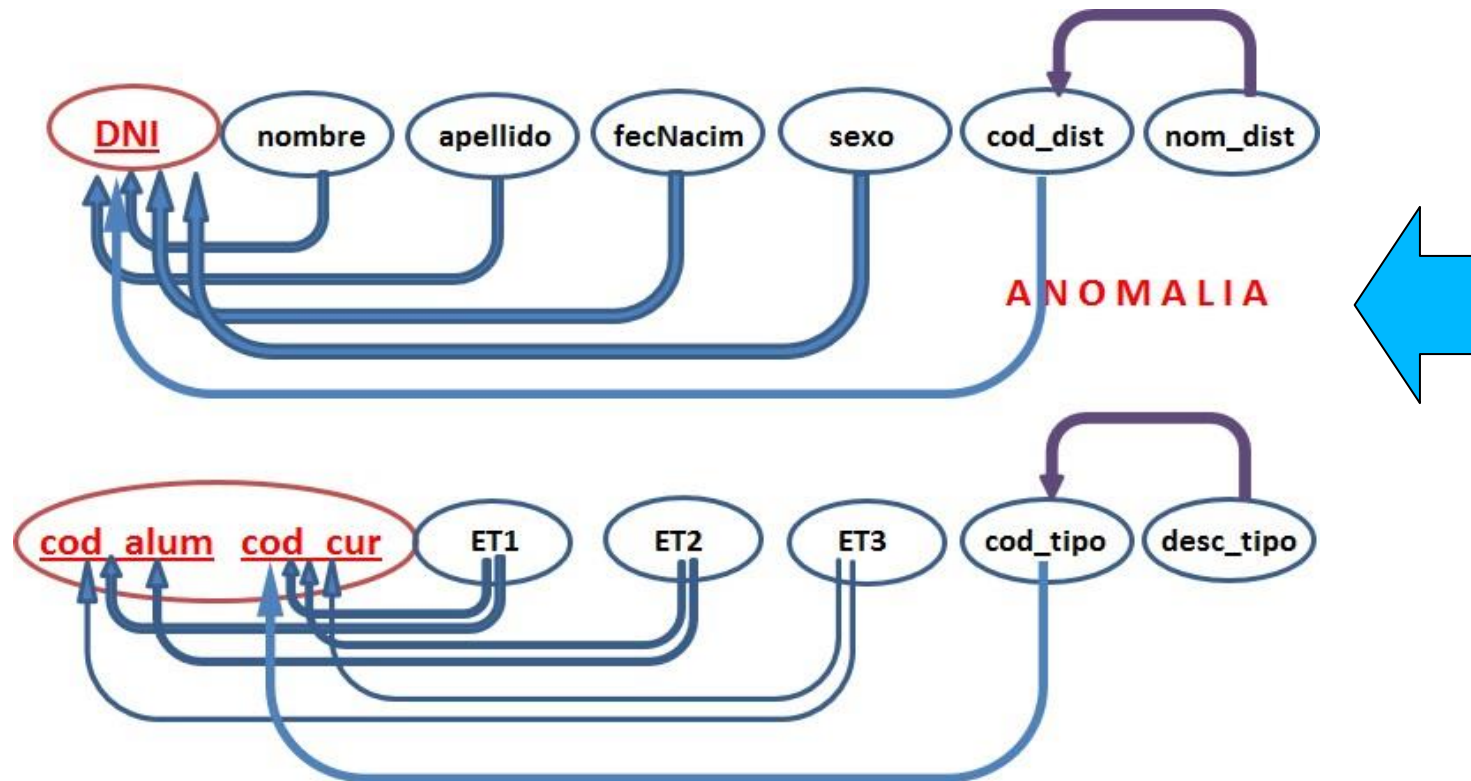


- Esta en primera segunda normal?

<u>StdNo</u>	StdClas	<u>ProgNo</u>	ProgAnio	ProgNota	CursoNo	CursoDesc
S1	JUN	O1	2013	3.5	C1	DB
S1	JUN	O2	2013	3.3	C2	VB
S2	JUN	O3	2014	3.1	C3	OO
S2	JUN	O2	2013	3.4	C2	VB

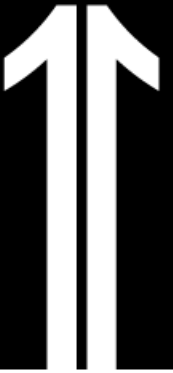
- NO
- El resultado seria: (ya lo hicimos en la parte de dependencias funcionales)
- **StdNo → StdClas**
- **ProgNo → ProgAnio, CursoNo, CursoDesc**
- **StdNo, ProgNo → ProgNota**

# TERCERA FORMA NORMAL



Dependencia Transitiva (DT)

# TERCERA FORMA NORMAL

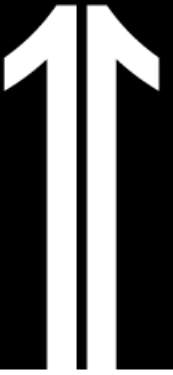


- Debe estar en Segunda forma normal
- No debe tener dependencias transitivas

<u>ProgNo</u>	ProgAnio	CursoNro	CursoDesc
O1	2013	C1	DB
O2	2013	C2	VB
O3	2014	C3	OO
O4	2013	C2	VB

- SrtNo → StdClas
- **ProgNo → ProgAnio, CursoNo, CursoDesc**
- StdNo, ProgNo → ProgNota

# TERCERA FORMA NORMAL



- Ya lo hicimos en dependencias funcionales

<u>ProgNo</u>	ProgAnio	CursoNro	CursoDesc
O1	2013	C1	DB
O2	2013	C2	VB
O3	2014	C3	OO
O4	2013	C2	VB

- El resultado será:
- **SrtNo → StdClas**
- **ProgNo → ProgAnio, CursoNo**
- **CursoNo → CursoDesc**
- **StdNo, ProgNo → ProgNota**

*CursoNro	DescCurso
C1	DB
C2	VB
C3	OO



**NORMALIZACIÓN**

# 1. LA PRIMERA FORMA NORMAL (1FN)

## • Enunciado:

**“Eliminar los grupos de datos repetitivos”**

## • Qué hacer:

- Crear una tabla separada por cada grupo de datos repetitivos
- Identificar la nueva tabla con una clave primaria la misma que deberá estar relacionada con la tabla base.

## • Ejemplo

**Registrar factura Nro=90 con 3 productos adquiridos**

nro	Fecha	IdCli	RazonSocial	Direcc	Obs	Tventa	IdPro	Descrip	u-med	cantid	precio	stotal	lgv	total
90	01/01/07	c1	LOS COCOS	Av Amer	Xx	Cr	p1	cemento	bsa	10	20	200	90	590
90	01/01/07	c1	LOS COCOS	Av Amer	Xx	Cr	p2	yeso	bsa	20	10	200	90	590
90	01/01/07	c1	LOS COCOS	Av Amer	Xx	Cr	p3	fierro	unid	5	10	100	90	590

**Qué atributos se repiten en la factura nro90 ?**

# 1. LA PRIMERA FORMA NORMAL (1FN)

## ● Enunciado:

Qué información se repite en la factura nro90

nro	Fecha	IdCli	RazonSocial	Direcc	Obs	Tventa	IdPro	Descrip	u-med	cantid	precio	stotal	lgv	total
90	01/01/07	c1	LOS COCOS	Av Amer	Xx	Cr	p1	cemento	bsa	10	20	200	90	590
90	01/01/07	c1	LOS COCOS	Av Amer	Xx	Cr	p2	yeso	bsa	20	10	200	90	590
90	01/01/07	c1	LOS COCOS	Av Amer	Xx	Cr	p3	fierro	unid	5	10	100	90	590

## ● Qué hacer:

- Crear una tabla separada por cada grupo de datos repetitivos: DETALLE
- Identificar DETALLE con IDENTIFICADOR y relacionarla con la tabla base.

nro	IdProd	Descrip	u-med	cantid	precio	Stotal
90	p1	Ceme	bsa	10	20	200
90	p2	Yeso	bsa	20	10	200
90	p3	fierro	unid	5	10	100

Nueva Tabla:  
DETALLE

Tabla Base  
VENTA

nro	Fecha	IdCliente	RazonSocial	Direcc	Obs	Tventa	lgv	total
90	01/01/07	c1	LOS COCOS	Av Amer	Xx	cr	98	598

## 2. SEGUNDA FORMA NORMAL (2FN)

- **Enunciado.**

Eliminar los atributos que no dependan del identificador o clave primaria

- **Qué hacer:**

- Crear una tabla separada con los atributos que no dependan de la tabla base
- Identificar la nueva tabla con un IDENTIFICADOR la misma que deberá estar relacionada con la tabla base.

- **Ejemplo:**

nro	Fecha	IdCLiente	RazonSocial	Direcc	Obs	Tventa	igv	total
90	01/01/20	c1	LOS COCOS	Av America 20	Xx	cr	90	590
100	02/01/20	c1	LOS COCOS	Av America 20	Yy	co	18	118
200	02/02/20	C2	LAS LIRAS	Av Volkan 500	TT	co	36	136

Qué datos no Dependen del Nro de Factura ?



## 2. SEGUNDA FORMA NORMAL (2FN)

### • Enunciado:

Qué datos no dependen del nro factura

nro	Fecha	IdCLiente	RazonSocial	Direcc	Obs	Tventa	igv	total
90	01/01/20	c1	LOS COCOS	Av America 20	Xx	cr	90	590
100	02/01/20	c1	LOS COCOS	Av America 20	Yy	co	18	118
200	02/02/20	C2	LAS LIRAS	Av Volkan 500	TT	co	36	136

### • Qué hacer:

- Crear una tabla con atributos que no dependan de la tabla base: CLIENTE
- Poner un identificado a CLIENTE y relacionarla con la tabla base.

IdCLiente	RazonSocial	Direcc
c1	LOS COCOS	Av America 20
C2	LAS LIRAS	Av Volkan 500

Nueva Tabla:  
CLIENTE

Tabla Base  
VENTA

nro	Fecha	Obs	Tventa	igv	total	IdCLiente
90	01/02/20	xx	cr	98	598	c1
100	02/01/20	Yy	co	18	118	C1
200	02/02/20	TT	co	36	136	

### 3. TERCERA FORMA NORMAL (3FN)

- **Enunciado.**

**Eliminar los atributos que dependen de otro identificador o clave alterna**

- **Qué hacer:**

- **Crear una tabla separada con los atributos que no dependan de la tabla base**
- **Identificar la nueva tabla con identificador, que deberá estar relacionada con la tabla base.**

- **Ejemplo:**

Nro	IdProd	Descripcion	u-med	cantid	Precio	Stotal
90	p1	Cemento	bsa	10	20	200
90	p2	Yeso	bsa	20	10	200
90	p3	Fierro	unid	5	10	100
100	P1	Cemento	Bs	20	20	400
100	P3	Fierro	Univ	10	10	100

**Qué atributos dependen de otro identificador?**

### 3. TERCERA FORMA NORMAL (3FN)

- **Enunciado:**

**Qué atributos dependen de otro identificador?**

Nro	IdProd	Descripcion	u-med	cantid	Precio	Stotal
90	p1	Cemento	bsa	10	20	200
90	p2	Yeso	bsa	20	10	200
90	p3	Fierro	unid	5	10	100
100	P1	Cemento	Bs	20	20	400
100	P3	Fierro	Univ	10	10	100

- **Qué hacer:**

- Crear una tabla separada con los atributos que no dependan de la tabla base: **PRODUCTO**
- Identificar la nueva tabla con un atributo, y relacionar con la tabla base.

IdProd	Descrip	u-med	Precio
p1	Cemento	bsa	20
p2	Yeso	bsa	10
p3	Fierro	unid	10

**Nueva Tabla:  
PRODUCTO**

**Tabla Base  
DETALLE**

Nro	IdProd	Cantid	Precio
90	P1	10	20
90	P2	20	10
90	P3	5	10

# ↑↑ CUÁNDO “DESNORMALIZAR”

- Note que al atributo Precio Unitario (precio) se ha mantenido en la tabla base. Esto es justificable por que el precio con el que se vendió DEBE MANTENERSE EN EL TIEMPO.
- Ejemplo: si el precio del producto: p1, cambia de 20 a 22, los 20 deben de mantenerse en la entidad DETALLE

**Detalle, sin Precio.  
Y si varía el Precio en Producto?  
Afectaría el precio del detalle**



**Detalle, CON Precio.  
Y si varía el Precio en  
Producto?,  
Afectaría el precio del detalle**

## EJERCICIO N.º 01

Se tiene el siguiente cuadro de Ordenes y se desea normalizar con la finalidad de tener centralizado y ordenada la información de la empresa

### Ordenes

Id_orden	Fecha	Id_cliente	Nom_cliente	Ciudad	Id_art	Nom_art	Cant	Precio
2301	23/02/11	101	Martin	Riobamba	3786	Red	3	35,00
2301	23/02/11	101	Martin	Riobamba	4011	Raqueta	6	65,00
2301	23/02/11	101	Martin	Riobamba	9132	Paq-3	8	4,75
2302	25/02/11	107	Herman	Ambato	5794	Paq-6	4	5,00
2303	27/02/11	110	Pedro	Quito	4011	Raqueta	2	65,00
2303	27/02/11	110	Pedro	Quito	3141	Funda	2	10,00



## EJERCICIO N.º 02


La empresa **EcuadorSYSTEMS** lo ha contratado para sistematizar la facturación, usted debe analizar toda la información disponible de la factura y aplicar el proceso de normalización

EcuadorSYSTEMS					
Accesorios y Partes del Computador					
Factura de Compra			No. 5656801		
			Fecha: Octubre 23/2014		
Nombre Cliente		JOSE PEREZ PRITO			
Direccion		CRA 7 NO. 50-50			
Cedula/NIT		80005000			
Ciudad		Santafe de Bogota, DC			
Telefono		35555555			
LISTA DE PRODUCTOS					
CATEGORIA	CODIGO	DESCRIPCION	VR.UNIT	CANTIDAD	PRECIO
Memorias	51238	Dim PC133 256 MB	\$ 115.000	2	\$ 230.000
Monitores	68989	Samsung Viamaster 550	\$ 380.000	1	\$ 380.000
Motherboards	589898	MSI KM266 ATA 133	\$ 220.000	1	\$ 220.000
Unidades	18888	CDROM LG 52X	\$ 70.000	1	\$ 70.000
Unidades	198999	Quemador LG 52x32x52x	\$ 150.000	1	\$ 150.000
Perifericos	898666	Teclado Genius PS2	\$ 15.000	1	\$ 30.000
Perifericos	638888	Mouse Genius Netscroll+	\$ 13.000	2	\$ 26.000

## ACERCA DE LAS FORMAS NORMALES

- No es necesario aplicar siempre las 3FN
- Puede aplicarlas en cualquier orden.





¿Preguntas o comentarios?





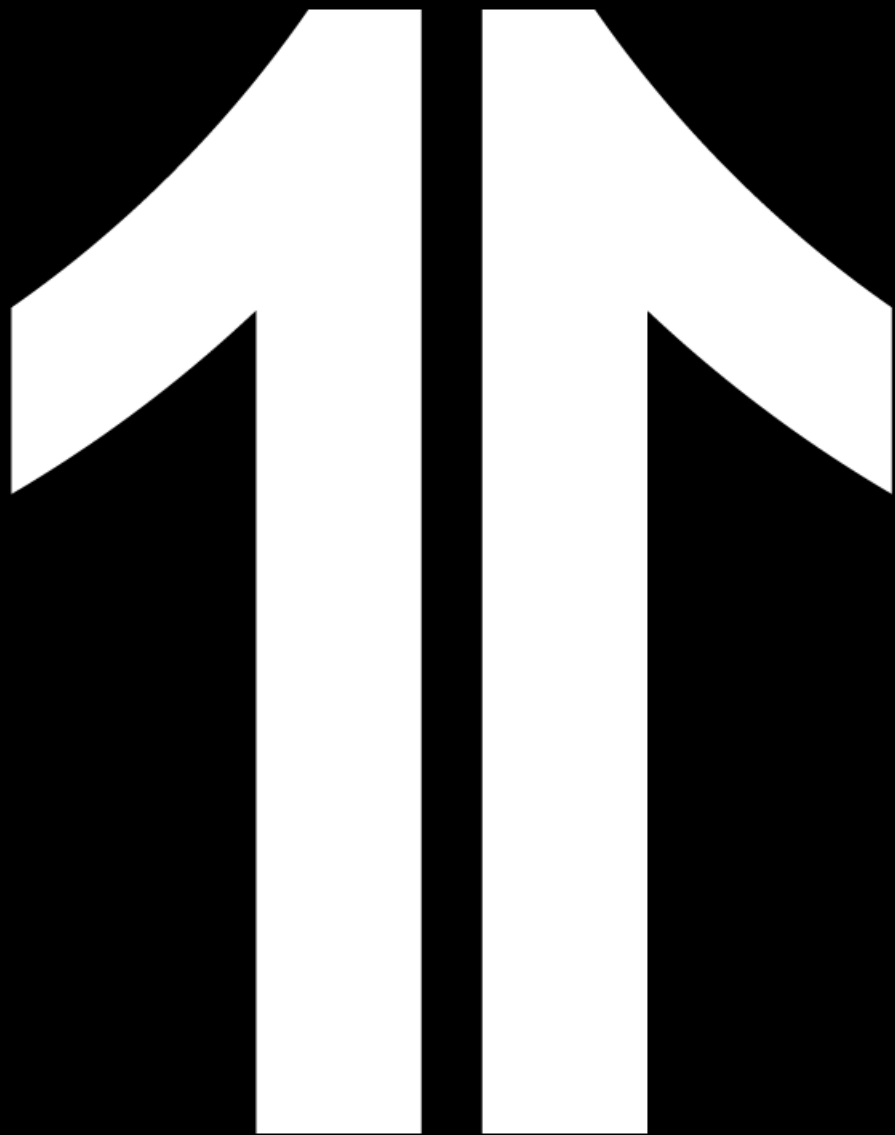
# CONCLUSIONES

- La normalización es importante para obtener registros de calidad que permitan la adecuada recuperación y transferencia de la información.
- La normalización es una técnica utilizada para diseñar tablas en las que las redundancias de datos se reducen al mínimo. Las primeras tres formas normales (1FN, 2FN y 3FN) son las más utilizadas. Desde un punto de vista estructural, las formas de mayor nivel son mejores que las de menor nivel, porque aquellas producen relativamente pocas redundancias de datos en la base de datos. En otras palabras, 3FN es mejor que 2FN y ésta, a su vez, es mejor que 1FN. Casi todos los diseños de negocios utilizan la 3FN como forma ideal.



## REFERENCIAS

Coronel, Carlos; Morris Steven y Rob, Peter. Bases de Datos: Diseño, Implementación y Administración. 2011



**GRACIAS**

