

Base de Datos Relacionales

Clase 2



Introducción a las Bases de Datos

- Auditoría
- Control de concurrencia y simultaneidad
- Diseño de Bases de Datos

Capacidad de auditoria

Objetivos



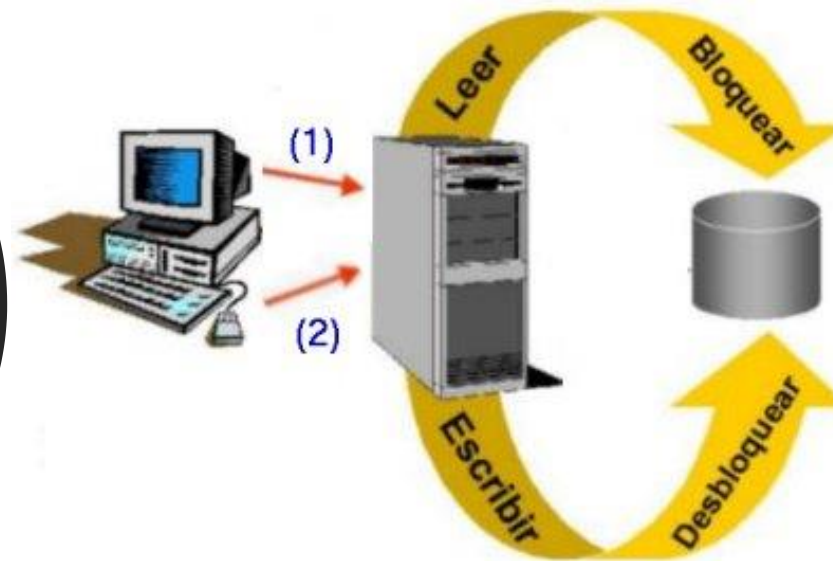
Vigilar el entorno de la base de datos para que los datos estén protegidos. A través de métodos:

- Procedimientos de Seguridad.
- Respaldo y Recuperación.
- Concurrencia adecuados

Poder determinar quién interactuó con la base, no en el sentido de modificar los datos, sino de quién recuperó unos datos y cuales fueron esos datos. Se puede emplear el mismo archivo diario. Puede plantear problemas, como:

- Tener un diario muy grande.
- Aumentar el tiempo necesario para acceder a los datos al tener que anotar hasta las lecturas realizadas.

Concurrencia



Gracias al SGBD existe la posibilidad de que **varios usuarios** tengan acceso de forma rápida y eficiente a los datos de la base. Al centralizar los datos en una base de datos, aumentan las probabilidades de que se dé este caso. Si el SGBD permite esto, seguramente el trabajo realizado por los usuarios se vería dañado, por eso el SGBD de proteger los datos de la actualización simultánea por otro usuario; para ello utiliza mecanismos sofisticados de bloqueo.



ORACLE

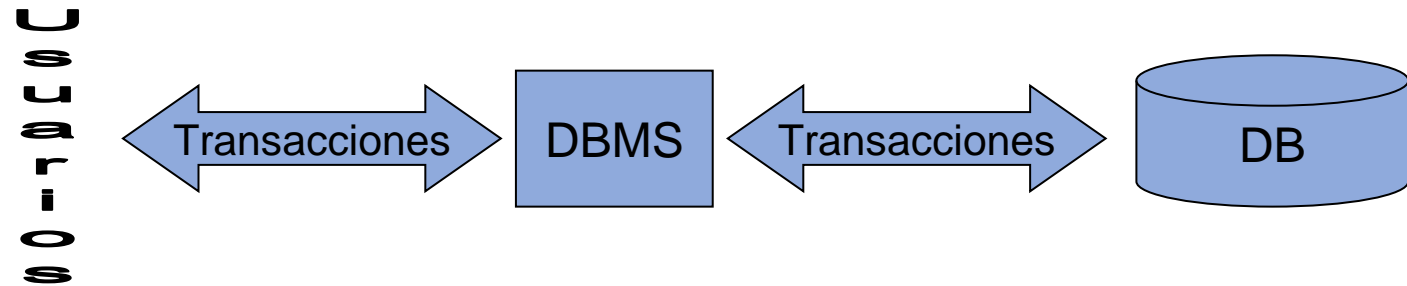


Microsoft SQL Server

Multiprogramación:

Consiste en la ejecución concurrente de dos o más programas diferentes e independientes en la misma computadora, con el fin de incrementar las prestaciones del procesador y realizar un número mayor de transacciones en un intervalo de tiempo dado.

Control de
concurrency y
simultaneidad



Ejemplo

En una aerolínea quedan 25 plazas para un determinado vuelo, y los empleados A y B casi de forma simultánea solicitan reservas de ese vuelo para la misma fecha, y quieren reservar 4 y 2 plazas respectivamente.

Ejemplo

Tiempo	Empleado A	Empleado B
T1	Lee registro <i>(quedan 25 plazas)</i>	
T2		Lee registro <i>(quedan 25 plazas)</i>
T3	Resta 4 lugares y escribe el registro actualizado. <i>(21 plazas)</i>	
T4		Resta 2 lugares y escribe el registro actualizado. <i>(23 plazas)</i>
	<i>Deberían quedar 19 plazas</i>	

Solución al problema de la concurrencia

Consiste en aplicar una técnica conocida como **cerrado**.

El DBMS está diseñado para que, una vez iniciada una actualización con un determinado elemento de datos, cualquier intento de iniciar otra actualización sobre el mismo dato no pueda efectuarse, por encontrarse el acceso bloqueado o cerrado hasta que la primera transacción haya concluido; es decir, se trata de secuenciar actualizaciones.



Diseño de bases de datos

Diseño

En una base de datos bien diseñada, cada tabla almacena datos sobre un tema en **concreto, como empleados o productos**. Una tabla tiene **registros (filas) y campos (columnas)**. Los campos tienen diferentes tipos de datos, como texto, números, fechas e hipervínculos.

Customers				
	ID	Company	First Name	Last Name
+	1	Company A	Anna	Bedecs
+	2	Company B	Antonio	Gratacos Solsona
+	3	Company C	Thomas	Axen

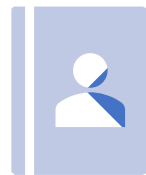
Registro y propiedades

Un registro: contiene datos específicos, como información acerca de un determinado empleado o un producto.

Un campo: contiene datos sobre un aspecto del asunto de la tabla, como el nombre o la dirección de correo electrónico.


Un valor de campo: cada registro tiene un valor de campo. Por ejemplo, Contoso, Ltd. o alguien@ejemplo.com.


Propiedades de tabla y campo





Las tablas y los campos también tienen propiedades que se puede establecer para controlar sus características o su comportamiento.


Propiedades


 Básico


 Opciones


 Índices (1)

 Llaves foráneas (0)

 Comprobar restricciones (0)

 Particiones

 Código CREATE




Nombre:


legajos


Comentario:


Ejemplo de tabla con datos únicos del personal


Columnas:

 Agregar

 Borrar

 Subir

 Bajar

#	Nombre	Tipo de datos	Longitud/Co...	Sin signo	Permitir NULL	Rellenar con cer...	Predeterminado
 1	legajo	INT	4	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Sin valor predeterminado
2	dni	INT	8	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Sin valor predeterminado
3	apellido	VARCHAR	30	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Sin valor predeterminado
4	nombre	VARCHAR	40	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Sin valor predeterminado

Registros

Host: 127.0.0.1 Base de datos: clase1 Tabla: legajos

clase1.legajos: 3 filas en total (aproximadamente)

legajo	dni	apellido	nombre
100	32.001.002	Perez	Juan
101	32.001.003	Baez	Juan
102	33.554.555	Gómez	Esteban

Indicador de clave primaria

Nombre de Columna/
campo/
atributos.
Especifica el
nombre del
dato a
almacenar

Valor del atributo
/ campo

Clave Primaria


Una **clave primaria** es una **columna especial o conjunto de columnas** en una tabla de base de datos que identifica de manera única cada registro en la tabla. En el diseño de bases de datos relacionales, se llama clave primaria a un campo o a una combinación de campos que identifica de forma única a cada fila de una tabla

Host: 127.0.0.1

Base de datos: clase1

Tabla: legajos

clase1.legajos: 3 filas en total (aproximadamente)

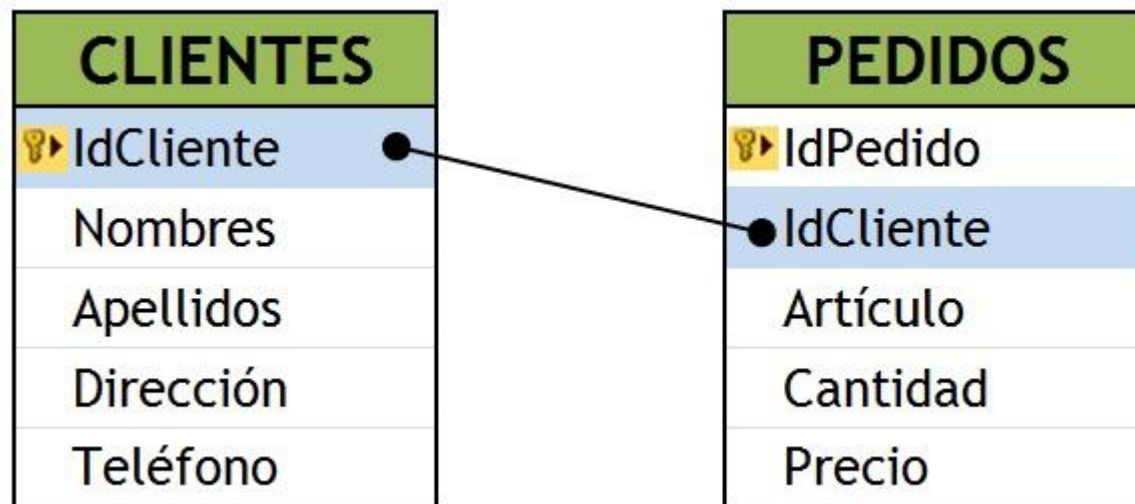
legajo 	dni	apellido	nombre
100	32.001.002	Perez	Juan
101	32.001.003	Baez	Juan
102			

Legajo es PK
100 es el valor
clave

Identifica un registro único

Relaciones entre tablas

Una relación es una **conexión entre dos tablas que contienen datos**: una columna en cada tabla es la base de la relación. Para ver por qué son útiles las relaciones, imagine que realiza el seguimiento de los datos de los pedidos de los clientes de su negocio.



Diseño de Bases de Datos

Objetivos del Diseño.

Entre los distintos objetivos en el diseño de una base de datos se pueden considerar:

- La base de datos resultante tiene que ser capaz de almacenar toda la información necesaria. El primer paso será determinar los atributos que van a formar parte de la base de datos y reunirlos en una relación Universal. Hasta que se hayan concretado los campos necesarios, no podrá el diseñador establecer las relaciones con ellos.
- Eliminar la información redundante siempre que se a posible.
- Mantener el nro de relaciones al mínimo entre los componentes de la base con el fin de facilitar la programación o uso.
- Las relaciones obtenidas deben estar normalizadas con el fin de minimizar los problemas de actualización.

Normalización

Se entiende por normalización, la descomposición o subdivisión de una relación en dos o más relaciones para evitar la redundancia. En definitiva, que “Cada hecho esté en su lugar”.

La relación Universal.

Supongamos que se desea implementar en una base de datos las ventas de una determinada empresa a sus clientes, por la relación:

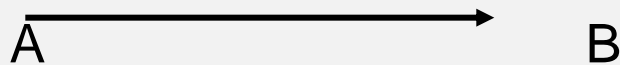
ORDENES-VENTA(*NCLI, NOMBRE, LOCALIDAD, CT, NAR, ARTÍCULO, CANT, PVP, FECHA*).

Dependencia Funcional: DF

- La normalización se basa en la dependencia funcional.
- El concepto de dependencia funcional se tomó de las matemáticas $Y=f(x)$, Y es función de X si el valor de Y está siempre determinado por el valor X.

La dependencia funcional (DF),

Se define como: Dados dos atributos **A** y **B** de una relación **R**, se dice que **B** es funcionalmente dependiente del atributo **A** si para cada valor de **A** existe un valor de **B**, y solo uno asociado con él. En otros términos si en cualquier instante, conocido el valor de **A**, podemos conocer el valor de **B**. Se simboliza:



- Tanto A como B pueden ser un conjunto de atributos en lugar de atributos simples.
- La dependencia funcional no permite establecer condiciones entre atributos de diferentes relaciones.
- Las DF se determinan al estudiar las propiedades de todos los atributos de la relación y deducir cómo están relacionados los atributos entre sí.
- La DF está íntimamente ligado con el concepto de **clave**.

Claves

Clave Candidata: Conjunto de uno o mas atributos que podría ser utilizado como clave principal de una relación.

Super Clave: Conjunto de uno o mas atributos que, juntos, permiten identificar de forma única a una entidad dentro de una relación.

Clave Principal: *Es la clave candidata en la que ningún componente puede tomar valor nulo.*

Para encontrar la clave candidata, se debe estudiar las DF y, a partir de ellas, obtener el mínimo conjunto posible de atributos tales que, una vez conocidos sus valores en la tupla, los demás queden definidos

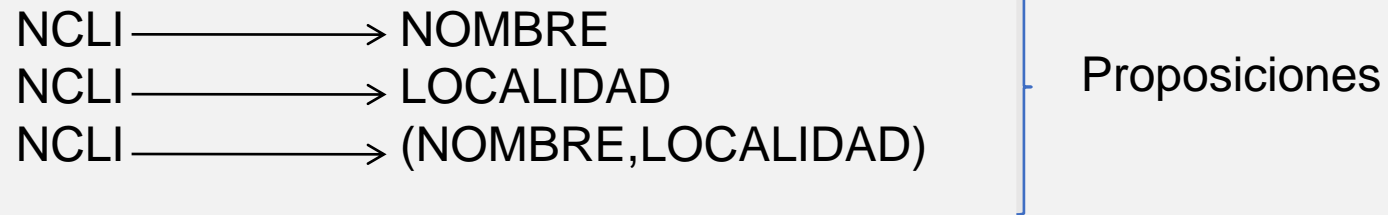
Relación universal

Relación Ordenes - Venta

NCLI	NOMBRE	LOCALIDAD	CT	NART	ARTICULO	CANT	PVP	FECHA
11	Luis	Maipú	0,8	a1	Papel	100	5	03-May
11	Luis	Maipú	0,8	a3	Cinta	50	500	05-May
11	Luis	Maipú	0,8	a9	Disco	25	200	07-May
44	Ana	Godoy Cruz	1,1	a1	Papel	100	5	10-May
55	José	Luján	1,4	a4	Grampas	30	50	03-May

Información deseada para las ORDENES-VENTA

D.F. y Determinantes



Determinante: Si $A \rightarrow B$ es una DF y B *no* es funcionalmente dependiente de A, se dice que A es el determinante de B.
Un *determinante* son todos los atributos situados en el lado *izquierdo* de una DF

Diagrama de Dependencia Funcional

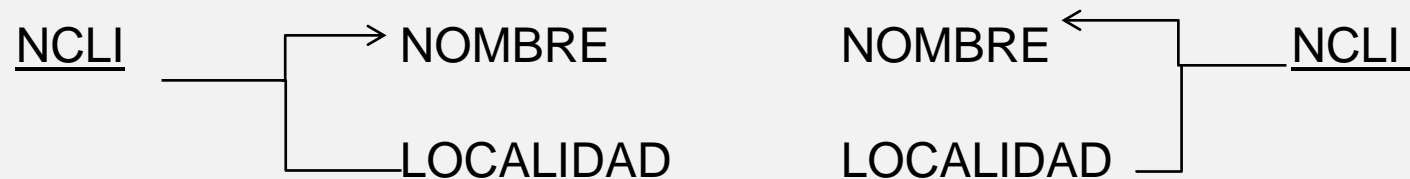
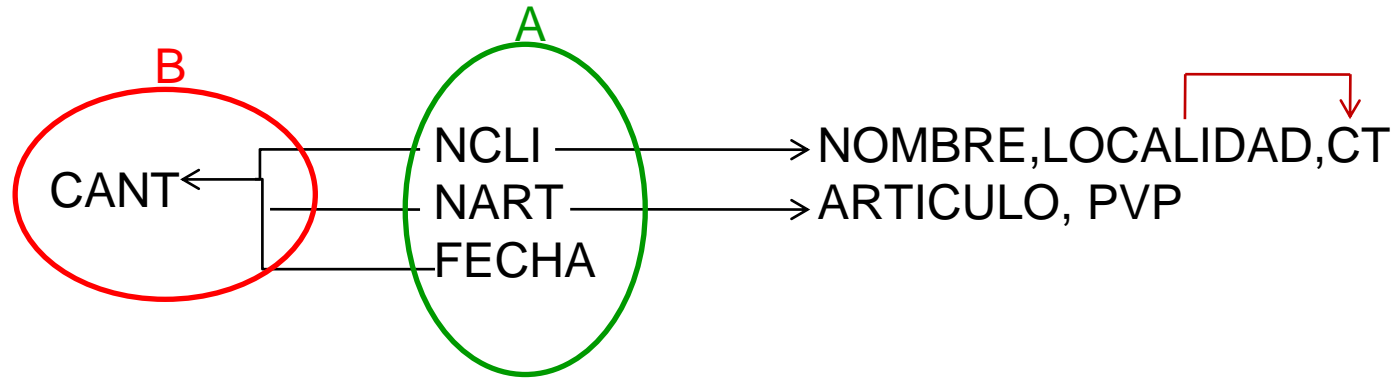


Diagrama funcional para la relación Ordenes de Venta



El atributo CANT es totalmente dependiente de los atributos NCLI, NART y FECHA.

Esto da lugar al concepto de **DF Total**.

En una relación **R**, un atributo o colección de atributos **B** tiene una DFT de otra colección de atributos **A** de la relación **R**, si **B** es funcionalmente dependiente de todos los atributos de **A**, pero no de un subconjunto de **A**.

Primera forma normal

Una relación está en 1FN si todo atributo contiene un valor indivisible, atómico.

Relación Ordenes - Venta

NCLI	NOMBRE	LOCALIDAD	CT	NART	ARTICULO	CANT	PVP	FECHA
11	Luis	Maipú	0,8	a1	Papel	100	5	03-May
11	Luis	Maipú	0,8	a3	Cinta	50	500	05-May
11	Luis	Maipú	0,8	a9	Disco	25	200	07-May
44	Ana	Godoy Cruz	1,1	a1	Papel	100	5	10-May
55	José	Luján	1,4	a4	Grampas	30	50	03-May

Se puede normalizar con la creación de un registro nuevo por cada uno de los distintos valores de un campo, tal que permita expresar la relación como una tabla.

Normalización de la 1 FN

- a) Dividiendo la RU en nuevas relaciones
- b) Cada R tiene la propiedad de que su clave, en su totalidad, es necesaria para definir cada uno de los campos no clave.

Descomposición de la RU

Descomposición sin pérdida: es la descomposición de una relación R en R1, R2,...RN, tal que:

$$R=R1*R2*.....*RN$$

Cuando se actualiza la base de datos, el sistema debe poder comprobar que la actualización no va a generar relaciones ilegales, o una que no satisfaga todas la DF establecidas.

Para llevar a cabo el proceso de Normalización es aconsejable dar los siguientes pasos:

1. Elegir una clave primaria que puede representar de forma única a cada registro de la relación.
2. Construir un diagrama de dependencia en función a esas claves.
3. Construir las nuevas relaciones basándose en dichas claves.

En el ejemplo,

Para el paso 1. En la relación ORDENES-VENTAS, los atributos que forman la clave NCLI,NART,FECHA.

Relación CLIENTES NCLI → NOMBRE,LOCALIDAD,CT

Relación ARTICULOS NART → ARTICULO,PVP

Relación VENTAS NCLI
NART
FECHA → CANT

Diagramas de dependencias funcionales

Relación CLIENTES

NCLI	NOMBRE	LOCALIDAD	CT
11	Luis	Maipú	0,8
44	Ana	Godoy Cruz	1,1
55	José	Luján	1,4

Relación ARTICULOS

NART	ARTICULO	PVP
a1	Papel	5
a3	Cinta	500
a9	Disco	200
a4	Grampas	50

Relación Ventas

NCLI	NART	FECHA	CANT
11	A1	03-May	100
11	a3	05-May	50
11	a9	07-May	25
44	a1	10-May	100
55	a4	03-May	30

Dependencia Transitiva.

$R(A, B, C)$

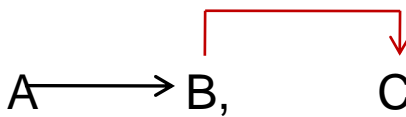
Si $A \longrightarrow B$, $B \longrightarrow C$
Y $B \not\longrightarrow A$

Entonces

C **depende transitivamente de** A

En un diagrama de dependencia funcional

Relación R $A \longrightarrow B$, C



Se puede descomponer en dos relaciones por la proyección del último eslabón de la forma:

Relación R1 $A \longrightarrow B$
Relación R2 $B \longrightarrow C$

Normalización de la relación 2FN

Las anomalías de almacenamiento, originadas por la dependencia transitiva en una relación en 2FN, se puede normalizar mediante los siguientes pasos:

1. En una relación, determinar el atributo que es funcionalmente dependiente de un atributo no clave y dibujar el diagrama de dependencia funcional.
2. Crear una nueva relación para almacenar el atributo no clave y su determinante.

Relación CLIENTES **NCLI** —————> NOMBRE,LOCALIDAD,CT

Relación TRANSPORTE **LOCALIDAD** —>CT

Relación ARTICULOS **NART** —————>ARTICULO,PVP

Relación VENTAS **NLCI** 
 NART —————>CANT
 FECHA

Diagramas de dependencias funcionales

Relación CLIENTES

NCLI	NOMBRE	LOCALIDAD
11	Luis	Maipú
44	Ana	Godoy Cruz
55	José	Luján

Relación ARTICULOS

NART	ARTICULO	PVP
a1	Papel	5
a3	Cinta	500
a9	Disco	200
a4	Grampas	50

Relación Ventas

NCLI	NART	FECHA	CANT
11	A1	03-May	100
11	a3	05-May	50
11	a9	07-May	25
44	a1	10-May	100
55	a4	03-May	30

Relación Localidades

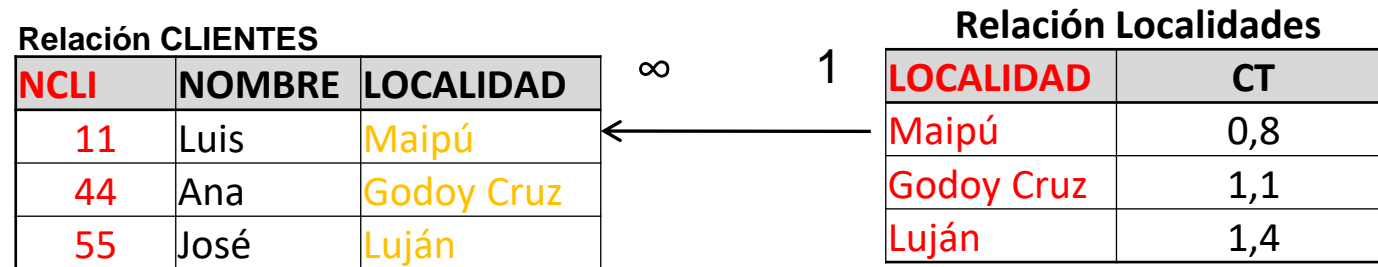
LOCALIDAD	CT
Maipú	0,8
Godoy Cruz	1,1
Luján	1,4

TERCERA FORMA NORMAL 3FN

Una relación está en 3FN si, y solo si:

1. Está en 2FN
2. Todo atributo que no pertenezca a la clave no depende de un atributo no clave.

La 3FN elimina las redundancias ocasionadas por las dependencias transitivas.



Forma Normal de Boyce Codd: FNBC

Una relación R está en forma normal de FNBC, si, y sólo si, cada determinante de la Relación **es una clave candidata**.

En la mayoría de los casos una relación en 3FN NO es FNBC , sin embargo toda relación que esté en FNBC también está 3FN.

Descomposición de la 3FN

R(A,B,C,D,E.....) que no está en FNBC
Y sea C----->D una de las DF causantes.
Siendo C un determinante pero una clave candidata se forman dos nuevas relaciones.

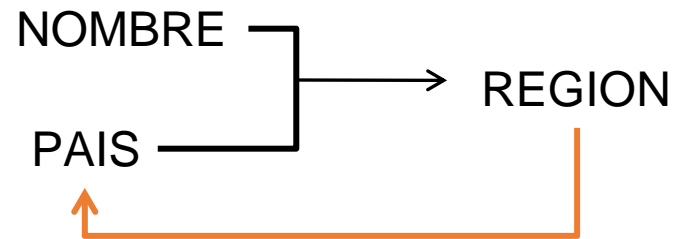
R(A,B,E) y R(C,D)

Ejemplo:

QUESOS(NOMBRE, PAIS, REGION)

REGION → PAIS

NOMBRE, PAIS → REGION



QUESOS(NOMBRE, REGION)

NOMBRE → REGION

REGIONES(REGION, PAIS)

REGION → PAIS

DEPENDENCIAS DE VALORES MULTIPLES: 4FN

Un valor o atributo A determina un conjunto de valores múltiples B

A \longrightarrow B

ESTUDIANTE(NUM, CURSO, DEPORTE)

num	curso	deporte
10	Base de datos	Basquet
10	Base de datos	Natación
10	Base de datos	Tenis
20	Física	Basquet
20	Física	Futbol

NUM \longrightarrow CURSO

NUM \longrightarrow DEPORTE

Normalización de la 4FN

1. Es FNBC
2. No contiene dependencias Multivaluadas

R(A, B, C)



NUM	CURSO
10	Base de datos
20	Física

NUM	DEPORTES
10	Basquet
10	Natación
10	Tenis
20	Basquet
20	Futbol

Ejercicio Normalización Nro 1

COD-LIBRO	TITULO	CAUTOR	AUTOR	CEDITORIAL	EDITORIAL	COD-LECTOR	LECTOR	DEVOLUCION
1001	Variable Compleja	100	Murray Spiegel	E980	McGraw Hill	A401	Pérez Gómez, Juan	15/04/2017
1004	C#	120	E. Petronic	E120	Anaya	A345	Ríos Terán, Ana	17/04/2017
1005	Estadística	100	Murray Spiegel	E980	McGraw Hill	A320	Roca, René	16/04/2017
1001	Variable Compleja	121	Stewart James	E980	McGraw Hill	A500	Moreno, Beatriz	19/04/2017
1006	Oracle University	205	Greenberg Nancy y Nathan Priya	E581	Oracle Corp.	A420	García, Roque	20/04/2017
1007	Java	300	Ramalho	E980	McGraw Hill	A401	Pérez Gómez, Juan	18/04/2017

Realizar el proceso de normalización para la administración de una biblioteca, donde se pretende Conocer en todo momento:

1. Qué lector tiene un libro
2. En que fecha se deben producir las devoluciones
3. Qué libros posee cada autor
4. Qué libros pertenecen a cada editorial



Gracias...