



BASES DE DATOS

LIC. EN INGENIERÍA DE SOFTWARE

Normalización

- Dependencias funcionales
- Primera Forma Normal

Normalización

¿Cómo surgieron?

- ❑ En el proceso de normalización, según la propuesta original de Codd (1972a), se somete un esquema de relación a una serie de pruebas para "certificar" si pertenece o no a una cierta forma normal.
- ❑ En un principio, Codd propuso tres formas normales, a las cuales llamó primera (1FN), segunda (2FN) y tercera (3FN) formas normales.

Normalización

- Posteriormente, Boyce y Codd propusieron una definición más estricta de 3FN, a la que se conoce como forma normal de Boyce-Codd. Todas estas formas normales se basan en las dependencias funcionales entre los atributos de una relación.
- Más adelante se propusieron una cuarta forma normal (4FN) y una quinta (5FN), con fundamento en los conceptos de dependencias multivaluadas y dependencias de reunión, respectivamente

¿Qué es la Normalización?

Normalización de una BASE DE DATOS



TABLA

Código	Nombre	Dirección	Actividad
1	Mario	G. Via, 8	Tecnología
2	Jorge	Juan 3	Textil
3	Isabel	Chile, 5	Textil
...



BASE DE DATOS

Código	Artículo	Familia	Precio C.	Precio V.
1	Impresora	PINFOR	45,00 €	49,00 €
...

Código	Nombre	Dirección	Teléfono
1	Corte	Cubo, 8	534215
2	Alcampo	Pereda	567895
...	234456

Código	Nombre	Dirección	Actividad
1	Mario	G. Via, 8	Tecnología
2	Jorge	Juan 3	Textil
3	Isabel	Chile, 5	Textil
...

Normalización

- La normalización es el proceso empleado para **mejorar el esquema lógico** de la base de datos, de modo que satisfaga ciertas restricciones que **eviten la duplicidad** de datos.
- La normalización garantiza que el esquema resultante se encuentra más **próximo al modelo de la organización**.
- Son reglas que se basan en la **clave primaria** y las **dependencias funcionales**.
- Cada regla que se cumple aumenta el grado de normalización.

Normalización

Debido a que las relaciones resultan de la observación del mundo real y de la transformación del modelo entidad-relación al relacional, pueden presentarse algunos problemas; entre estos problemas cabe destacar los siguientes (Miguel, A. et. al., 2000):

- Incapacidad para almacenar ciertos hechos.
- Redundancias y, por tanto, posibilidad de inconsistencias.
- Ambigüedades.
- Pérdida de información

Normalización

- Pérdida de dependencias funcionales, es decir, de ciertas restricciones de integridad que dan lugar a independencias entre los datos.
- Existencia de valores nulos (inaplicables).
- Aparición, en la base de datos, de estados que no son válidos en el mundo real (anomalías de inserción, borrado y modificación).

Normalización

- Ejemplo:

La relación ESTUDIANTE_SOLICITA_BECA almacena datos sobre los estudiantes que solicitan becas, información sobre las propias becas y la fecha en que un estudiante solicita una beca.

Si observamos con atención, vemos que presenta varios de los problemas enumerados anteriormente:

Normalización

ESTUDIANTE_SOLICITA_BECA

Cod_Estud	Nombre_E	Apellido	DNI	Dirección	Cod_Beca	Nombre	Requisito	Fecha
01234	Roberto	Hens	464367	Antonio López 43	A22321	METRICA	Ing. Téc.	10/10/06
76346	Luis	García	345347	Av. Ciudades 29	B56784	ERASMU	Ing. Téc.	12/11/06
76346	Luis	García	345347	Av. Ciudades 29	A22321	METRICA	Ing. Téc.	14/10/05
76346	Luis	García	345347	Av. Ciudades 29	G65434	HIMMPA	Ingenie.	15/09/06
01234	Roberto	Hens	464367	Antonio López 43	G65434	HIMMPA	Ingenie.	17/09/06
98776	Gregorio	Celada	885764	Pl. Pasises 67	G65434	HIMMPA	Ingenie.	21/09/06
01234	Roberto	Hens	464367	Antonio López 43	B56784	ERASMU	Ing. Téc.	11/11/05
98776	Gregorio	Celada	885764	Pl. Pasises 67	B56784	ERASMU	Ing. Téc.	10/10/05
01234	Roberto	Hens	464367	Antonio López 43	A22321	METRICA	Ing. Téc.	12/11/05
23245	Mercedes	García	809234	Río Miño 2	A22321	METRICA	Ing. Téc.	17/09/06

Normalización

- ❖ Gran cantidad de redundancia: ya que el nombre, apellidos, DNI y dirección de los estudiantes se repite por cada beca que hayan solicitado, y algo análogo sucede, cuando una beca la solicita más de un estudiante, con el nombre de la beca, y los requisitos.
- ❖ Anomalías de modificación; ya que podemos, por ejemplo, cambiar la dirección de un estudiante en una tupla y por error no modificarla en el resto de las que corresponden al mismo estudiante, lo que da lugar a inconsistencias.

Normalización

- ❖ Anomalías de inserción: ya que si se quisiera incluir información sobre algún estudiante que no hubiera solicitado ninguna beca, no sería posible, al formar el atributo Cod_Beca parte de la clave primaria de la relación.
- ❖ Ni tampoco podríamos introducir becas no solicitadas (recuerda que no se permiten nulos en ningún atributo que forme parte de la clave primaria).
- ❖ Además, la inserción de una beca que solicita más de un estudiante obligaría a incluir varias tuplas en la base de datos.

Normalización

- ❖ Anomalías de borrado: ya que si quisiéramos dar de baja una beca, también se perderían datos sobre los estudiantes que la solicitan (si éstos hubieran solicitado sólo esa beca) y, viceversa
- ❖ Si borramos un estudiante desaparecerán de nuestra base de datos las becas solicitadas por él (a no ser que hubieran sido solicitadas por otros estudiantes).

Normalización

- Normalmente, al seguir la metodología de diseño que hemos estudiado, realizando un adecuado diseño conceptual en el modelo entidad-relación, seguido con una cuidadosa transformación al modelo relacional, se pueden evitar estos problemas y obtener un esquema relacional exento de errores.
- Sin embargo, ante las posibles dudas respecto a si un determinado esquema relacional es correcto, es preferible aplicar a dicho esquema un **método formal de análisis** que determine lo que pueda estar equivocado y nos permita llegar a otro que nos asegure el cumplimiento de ciertos requisitos; este **método formal** es la teoría de la **normalización**.

Normalización

- En el proceso de normalización se debe ir comprobando que cada relación cumple una serie de reglas que se basan en la clave primaria y las dependencias funcionales

(que es la principal herramienta para medir formalmente la idoneidad de las agrupaciones de atributos en los esquemas de relación).

- Cada regla que se cumple aumenta el grado de normalización. Si una regla no se cumple, la relación se debe descomponer en varias relaciones que sí la cumplan.

Normalización

- La normalización se lleva a cabo en una serie de pasos.
- Cada paso corresponde a una forma normal que tiene unas propiedades.
- Conforme se va avanzando en la normalización, las relaciones tienen un formato más estricto (más fuerte) y, por lo tanto, son menos vulnerables a las anomalías de actualización.



1FN

2FN

3FN

Normalización

- El modelo relacional sólo requiere un conjunto de relaciones en **primera forma normal**.
- Las restantes formas normales son opcionales.
- Sin embargo, para evitar las anomalías de actualización, es recomendable llegar al menos a la tercera forma normal.

Normalización

- Dependencias funcionales

Normalización

■ DEPENDENCIA FUNCIONAL

- ✓ El concepto individual más importante en el diseño de esquemas relacionales es el de dependencia funcional.
- ✓ Una dependencia funcional es una restricción entre dos conjuntos de atributos de la base de datos.
- ✓ Suponiendo que un esquema de base de datos relacional tienen atributos A_1, A_2, \dots, A_n , y que toda la base de datos se describe con un solo esquema de relación universal.

$R = \{A_1, A_2, \dots, A_n\}$. **No quiere decir que así se almacenará la bd**

Normalización

■ DEPENDENCIA FUNCIONAL

- ✓ Una dependencia funcional, denotada por:

$$X \rightarrow Y$$

Entre dos conjuntos de atributos X y Y que son subconjuntos de R especifica una restricción sobre las posibles tuplas que podrían formar un ejemplar de relación r de R.

Normalización

■ DEPENDENCIA FUNCIONAL

$R = \{A1, A2, \dots An\}$.

$X \rightarrow Y$

✓ La restricción dice que, para cualesquier dos tuplas t_1 y t_2 , de r tales que

$t_1[X] = t_2[X]$, debemos tener también

$t_1[Y] = t_2[Y]$.

IMPARTIR		x	y
PROFESOR	CURSO	TEXTO	
t_1	Silva	Estructuras de datos	Bartram
t_2	Silva	Estructuras de datos	Al-Nour
	Heras	Compiladores	Hoffman
	Bravo	Estructuras de datos	Augenthaler

Normalización

■ DEPENDENCIA FUNCIONAL

$R = \{A1, A2, \dots An\}$.

$X \rightarrow Y$

- ✓ Esto significa que los valores del componente Y de una tupla de r dependen de los valores del componente X, o están determinados por ellos; o bien, que los valores del componente X de una tupla determinan de manera única (o funcionalmente) los valores del componente Y.

IMPARTIR

PROFESOR	CURSO	TEXTO
Silva	Estructuras de datos	Bartram
Silva	Estructuras de datos	Al-Nour
Heras	Compiladores	Hoffman
Bravo	Estructuras de datos	Augenthaler

Normalización

■ DEPENDENCIA FUNCIONAL

$R = \{A_1, A_2, \dots, A_n\}$.

$X \rightarrow Y$

- ✓ La restricción dice que, para cualesquier dos tuplas t_1 y t_2 , de r tales que $t_1[X] = t_2[X]$, debemos tener también $t_1[Y] = t_2[Y]$.
- ✓ También se dice que hay una **dependencia funcional de X a Y** o que Y depende funcionalmente de X.
- ✓ Dependencia funcional = **DF**. El conjunto de atributos X se denomina miembro izquierdo de la DF, y Y es el miembro derecho.

Normalización

■ DEPENDENCIA FUNCIONAL

$R = \{A_1, A_2, \dots, A_n\}$.

$X \rightarrow Y$

- ✓ La restricción dice que, para cualesquier dos tuplas t_1 y t_2 , de r tales que $t_1[X] = t_2[X]$, debemos tener también $t_1[Y] = t_2[Y]$.
- ✓ Así pues, X determina funcionalmente a Y en un esquema de relación R si y sólo si, siempre que dos tuplas $r(R)$ coincidan en su valor X , necesariamente deben coincidir en su valor Y .

Normalización

■ **DEPENDENCIA FUNCIONAL**

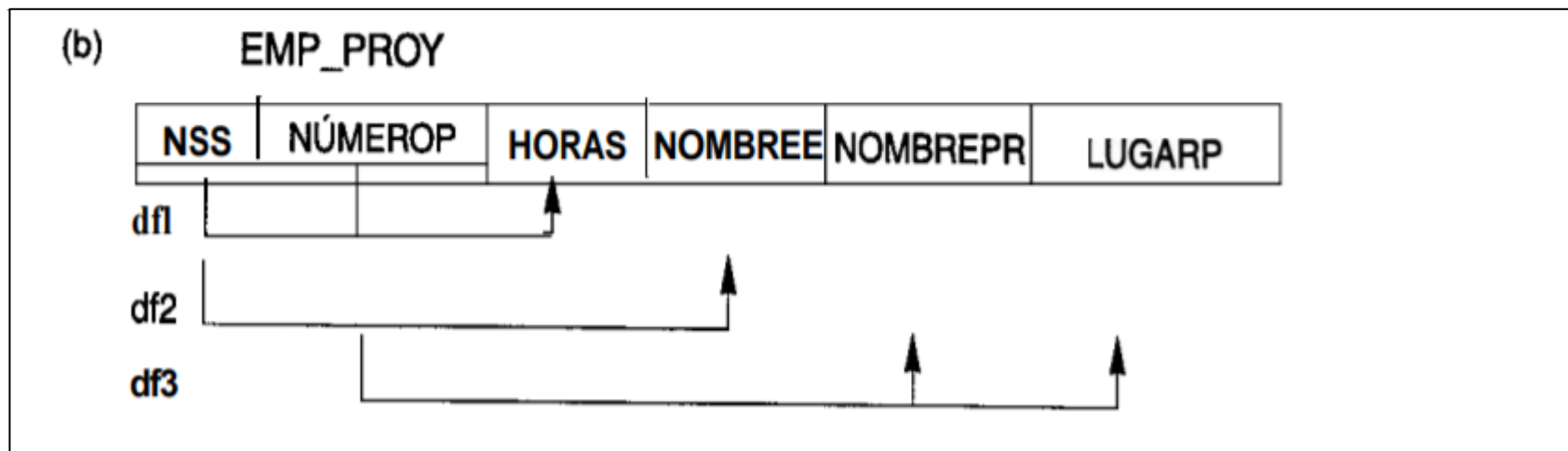
- ✓ La utilidad principal de las dependencias funcionales es describir mejor un esquema de relación R mediante la especificación de restricciones sobre sus atributos que deban cumplirse siempre.

Normalización

■ DEPENDENCIA FUNCIONAL

- ✓ Si una restricción de R dice que no puede haber más de una tupla con un valor X dado en cualquier ejemplar de relación $r(R)$ -esto es, X es una clave candidata de R- esto implica que $X \rightarrow Y$ para cualquier subconjunto de atributos Y de R.
- ✓ Si $X \rightarrow Y$ en R, esto no indica si $Y \rightarrow X$ en R o no.
- ✓ Ejemplo:

Esquema de la relación EMP_PROY y sus dependencias funcionales



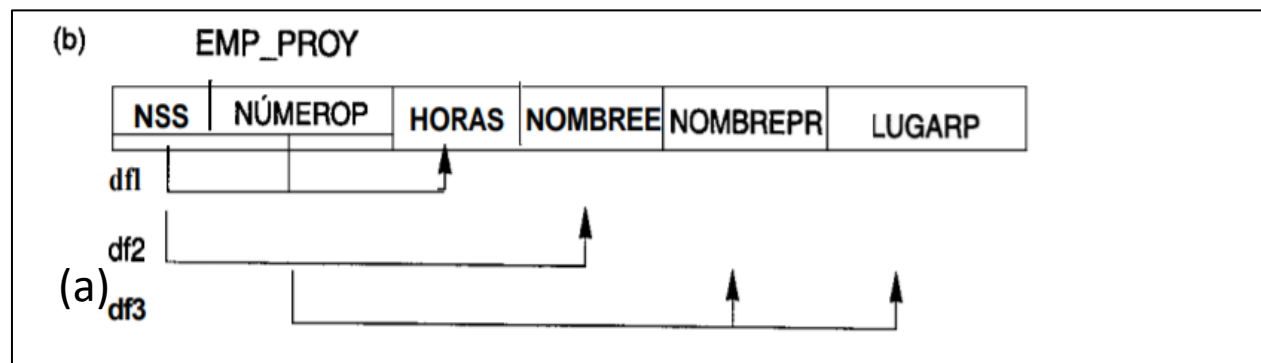
A partir de la semántica de los atributos, sabemos que deben cumplirse las siguientes dependencias funcionales:

(a) $NSS \rightarrow NOMBREE$

(b) $NÚMEROP \rightarrow \{NOMBREPR, LUGARP\}$

(c) $\{NSS, NÚMEROP\} \rightarrow HORAS$

Esquema de la relación EMP_PROY y sus dependencias funcionales



(a) $NSS \rightarrow NOMBREE$

(b) $NÚMEROP \rightarrow \{NOMBREPR, LUGARP\}$

(c) $\{NSS, NÚMEROP\} \rightarrow HORAS$

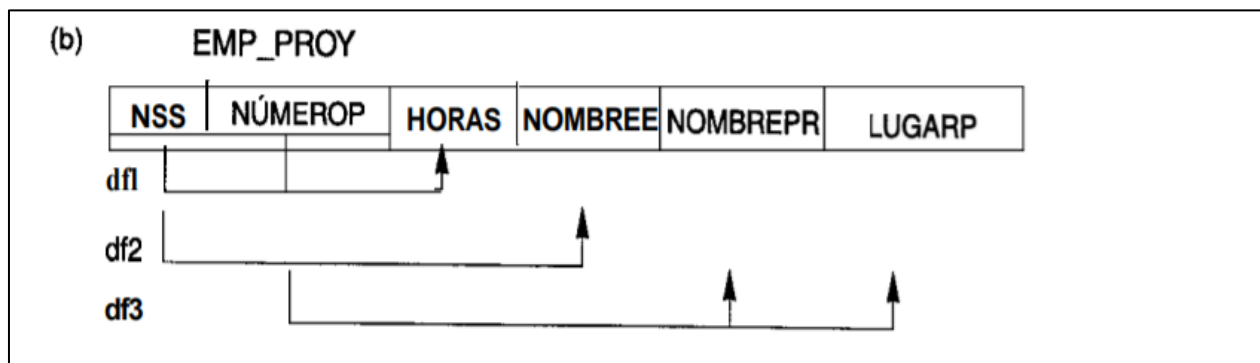
Estas dependencias funcionales especifican que:

(a) El valor del número de seguro social de un empleado (NSS) determina de manera única el nombre de ese empleado (NOMBREE)

(b) El valor del número de un proyecto (NUMEROP) determina de manera única el nombre del proyecto (NOMBREPR) y su lugar (LUGARP)

(c) Una combinación de valores de NSS y NÚMEROP determina de manera única el número de horas que el empleado trabaja en el proyecto cada semana (HORAS).

Esquema de la relación EMP_PROY y sus dependencias funcionales



(a) $NSS \rightarrow NOMBREE$

(b) $NÚMEROP \rightarrow \{NOMBREPR, LUGARP\}$

(c) $\{NSS, NÚMEROP\} \rightarrow HORAS$

De manera alterna, se dice que:

NOMBRE está determinado funcionalmente por (o depende funcionalmente de) NSS, o dado un valor de NSS, conocemos el valor de NOMBREE, y así sucesivamente.

Normalización

■ DEPENDENCIA FUNCIONAL

✓ Ejemplo

Una empresa de alquiler de vehículos desea implementar una base de datos con la información de su negocio. Se tienen **vehículos** identificados por su número de **matrícula**, y de los que se conoce su **marca**, **color**, **modelo** y **año**.

También se tienen **clientes** identificados por su número de **cédula** de identidad, y de los que se conoce su **nombre**, **dirección** y **teléfono**. Un **contrato** de alquiler de vehículo está identificado por un **número de contrato** y se realiza en una **fecha** dada entre un **cliente** y un **vehículo**, registrándose el **periodo** de alquiler en días y el **precio** del servicio.

Se considera que **en una misma fecha no se puede alquilar más de una vez el mismo vehículo al mismo cliente** en la misma fecha.

Una empresa de alquiler de vehículos desea implementar una base de datos con la información de su negocio. Se tienen **vehículos** identificados por su número de **matrícula**, y de los que se conoce su **marca**, **color**, **modelo** y **año**.

También se tienen **clientes** identificados por su número de **cédula** de identidad, y de los que se conoce su **nombre**, **dirección** y **teléfono**. Un **contrato** de alquiler de vehículo está identificado por un **número de contrato** y se realiza en una **fecha** dada entre un **cliente** y un **vehículo**, registrándose el **periodo** de alquiler en días y el **precio** del servicio.

Se considera que **en una misma fecha no se puede alquilar más de una vez el mismo vehículo al mismo cliente** en la misma fecha.

Obtenemos:

$matrícula \rightarrow \{marca, color, modelo, año\}$

$cédula \rightarrow \{nombre, dirección, teléfono\}$

$nroContrato \rightarrow \{fecha, cédula, matrícula, período, precio\}$

$\{fecha, cédula, matrícula\} \rightarrow nroContrato$

Actividad

Determina las dependencias funcionales $X \rightarrow Y$, existentes entre los atributos de cada una de las relaciones del caso “Clínica Veterinaria” que se muestra a continuación:

Actividad

Esquema de las relaciones

Base de datos “ClinicaVeterinaria”

Duenio (idDuenio, nombre, apPaterno, apMaterno, calle, numero, colonia)

Paciente (idPaciente, nombre, día_nac, mes_nac, anio_nac, especie, *idDuenio*)

Cita (numeroCita, día_cita, mes_cita, anio_cita, enfermedad, diagnostico, tratamiento, *idPaciente*, *usuario*)

Veterinario (usuario, primerNombre, apPaterno, apMaterno, nombreUsuario, contrasenia)

Secretaria (usuario, primerNombre, apPaterno, apMaterno, nombreUsuario, contrasenia)

Vacuna_Paciente (idPaciente, nombreVacuna, diaAplicacion, mesAplicacion, anioAplicacion)

Telefono_duenio (idDuenio, teléfono, tipo)

Actividad

Base de datos “ClinicaVeterinaria”. Dependencias Funcionales.

Duenio (idDuenio, nombre, apPaterno, apMaterno, calle, numero, colonia)

idDuenio → {nombre, apPaterno, apMaterno, calle, numero, colonia}

Paciente (idPaciente, nombre, día_nac, mes_nac, año_nac, especie, *idDuenio*)

idPaciente → {idPaciente, nombre, día_nac, mes_nac, año_nac, especie, *idDuenio*}

Cita (numeroCita, día_cita, mes_cita, año_cita, enfermedad, diagnostico, tratamiento, *idPaciente*, *usuario*)

{numeroCita, idPaciente} → {día_cita, mes_cita, año_cita, enfermedad, diagnostico, tratamiento, usuario}

Actividad

Base de datos “ClinicaVeterinaria”. Dependencias Funcionales.

Veterinario (idEmpleado, primerNombre, apPaterno, apMaterno, nombreUsuario, contrasenia)

idEmpleado → {primerNombre, apPaterno, apMaterno, nombreUsuario, contrasenia}

Secretaria (idEmpleado, primerNombre, apPaterno, apMaterno, nombreUsuario, contrasenia)

idEmpleado → {primerNombre, apPaterno, apMaterno, nombreUsuario, contrasenia}

Vacuna_Paciente (idPaciente, nombreVacuna, diaAplicacion, mesAplicacion, anioAplicacion)

No existen dependencias funcionales.

{idPaciente, nombreVacuna, diaAplicacion, mesAplicacion, anioAplicacion}

Actividad

Base de datos “ClinicaVeterinaria”. Dependencias Funcionales.

Telefono_duenio (idDuenio, teléfono, tipo)

{idDuenio, teléfono} → tipo

Normalización

- Primera Forma Normal (1FN)

Normalización

- PRIMERA FORMA NORMAL

"Una relación está en primera forma normal (1FN) si, y sólo si, todos los dominios de la misma contienen valores atómicos, es decir, no hay grupos repetitivos..." (Marqués, 2001).

Un dominio es atómico si se considera que los elementos de ese dominio son unidades indivisibles.

Normalización

■ PRIMERA FORMA NORMAL

- ✓ La primera forma normal se considera ahora parte de la definición formal de relación; históricamente, se definió para prohibir los atributos multivaluados, los atributos compuestos y sus combinaciones.

- ✓ Establece que los dominios de los atributos deben incluir sólo valores , atómicos (simples, indivisibles) y que el valor de cualquier atributo en una tupla debe ser un valor individual proveniente del dominio de ese atributo.

Modelo relacional

Propiedades de las relaciones

- Cada relación tiene un nombre y éste es distinto del nombre de todas las demás.
- **Los valores de los atributos son atómicos: en cada tupla, cada atributo toma un solo valor. Se dice que las relaciones están *normalizadas*.**
- No hay dos atributos que se llamen igual.
- El orden de los atributos no importa: los atributos no están ordenados.
- Cada tupla es distinta de las demás: no hay tuplas duplicadas.
- El orden de las tuplas no importa: las tuplas no están ordenadas.

Modelo relacional

Esquema de una relación

CURSO(Cod_Curso, Nombre, Num_Horas, Materia)

Instancias de una relación

Cod_Curso	Nombre	Num_Horas	Materia
00012	DISEÑO DE BASES DE DATOS	50	Bases de Datos
00034	BASES DE DATOS ORIENTADAS A OBJETOS	30	Sis. Operativos
00167	SISTEMAS OPERATIVOS AVANZADOS	30	Bases de Datos
01521	ALMACENES DE DATOS	25	Bases de Datos
005142	INTRODUCCIÓN A C++	25	Lenguajes
005143	INTRODUCCIÓN A C++	25, 30	Lenguajes

- **Relación:** Película (título, año, duración)

Atributos	Título	Año	Duración
Tuplas	La guerra de las galaxias	1977	123
	El señor de los anillos I	2001	178
	Mar adentro	2004	125

↓
Dominio=textos

↙ ↘
Dominio=enteros

Cardinalidad=3

Grado de la relación=3

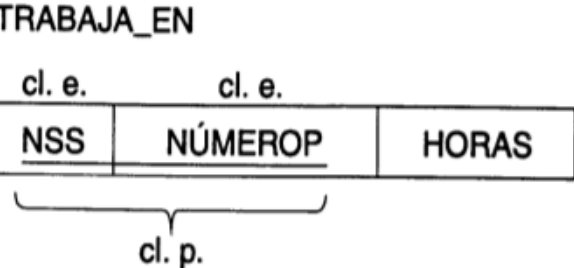
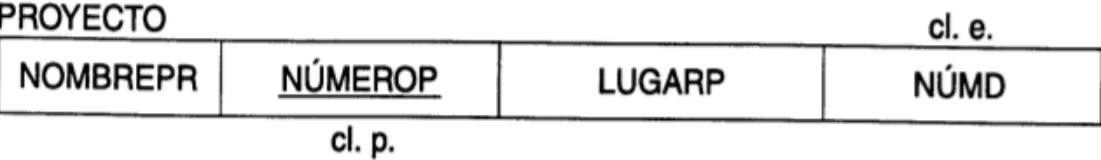
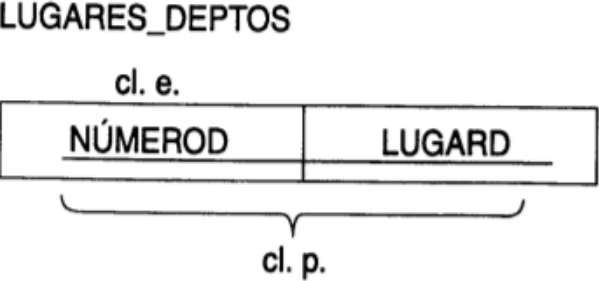
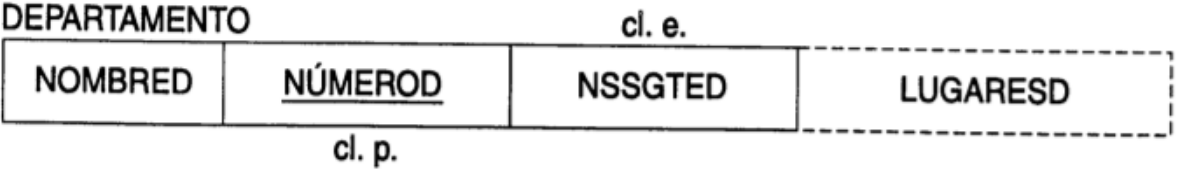
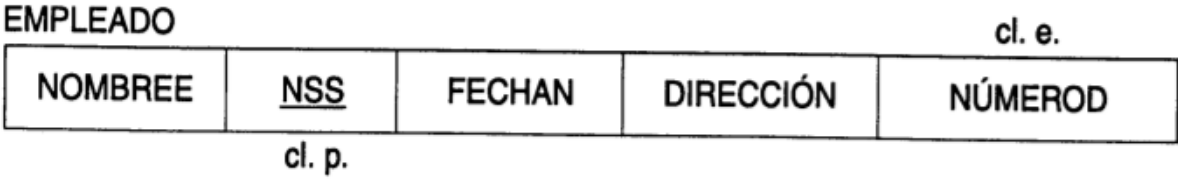
Normalización

- PRIMERA FORMA NORMAL

Se dice que, el esquema de la relación R está en la primera forma normal (1FN) si los dominios de todos los atributos de R son atómicos.

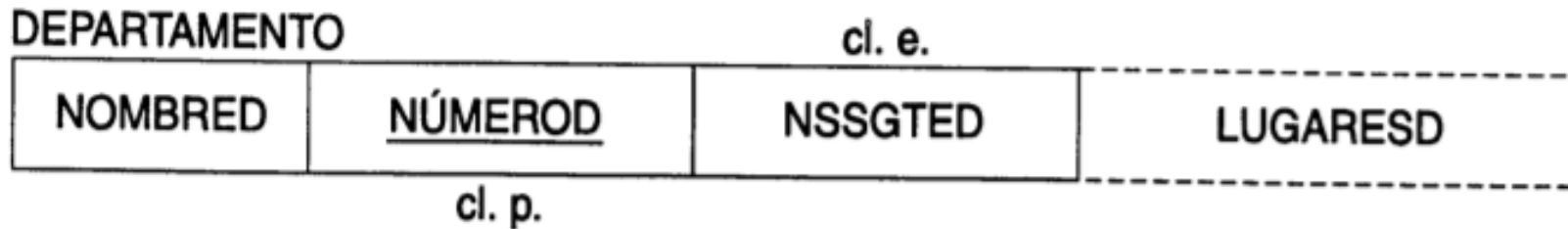
Un dominio es atómico si se considera que los elementos de ese dominio son unidades indivisibles.

Esquema de la base de datos relacional Compañía.



Esquema de la base de datos relacional Compañía.

Esquema de la relación Departamento



Su clave primaria es NUMEROD.

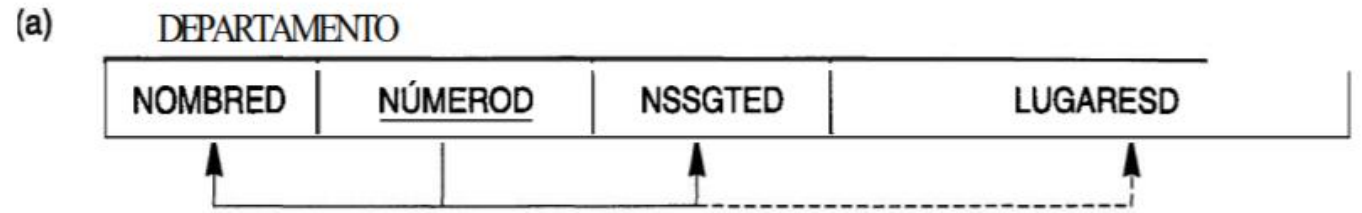
Si agregamos el atributo LUGARESD, suponemos que cada departamento puede tener varios lugares.

Esquema de la base de datos relacional Compañía.

Es evidente que no está en 1FN porque LUGARESD no es un atributo atómico, como puede verse en la primera tupla de la (b)

En la figura se tiene:

- a) Esquema de relación que no está en 1FN.
- b) Ejemplo de ejemplar de relación
- c) Relación 1FN con redundancia



(b)

DEPARTAMENTO			
NOMBRED	<u>NÚMEROD</u>	NSSGTED	LUGARESD
Investigación	5	333445555	(Belén, Sacramento, Higueras)
Administración	4	987654321	(Santiago)
Dirección	1	888665555	(Higueras)

(c)

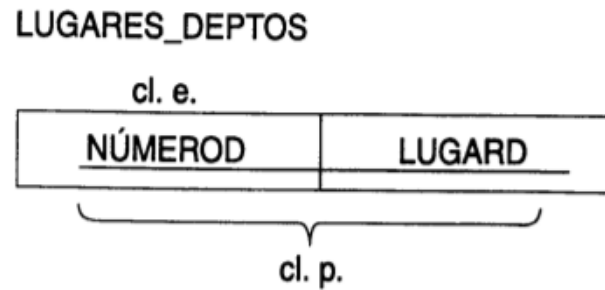
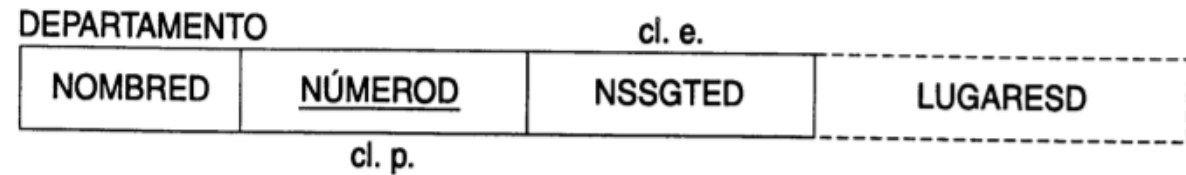
DEPARTAMENTO			
NOMBRED	<u>NÚMEROD</u>	NSSGTED	<u>LUGARESD</u>
Investigación	5	333445555	Belén
Investigación	5	333445555	Sacramento
Investigación	5	333445555	Higueras
Administración	4	987654321	Santiago
Dirección	1	888665555	Higueras

Esquema de la base de datos relacional Compañía.

Para normalizarla a relaciones 1FN, dividimos sus atributos entre dos relaciones: DEPARTAMENTO y LUGARES-DEPTOS

La idea es eliminar el atributo LUGARESD que viola 1FN y colocarlo en una relación aparte LUGARES-DEPTOS junto con la clave primaria NÚMEROD de DEPARTAMENTO.

La clave primaria de esta relación es la combinación {NUMEROD, LUGARD},



Normalización

- Otro ejemplo:

Esquema de la relación:

ALUMNO(id_alumno, nombre_alumno, apellido_alumno, curso)

Ejemplar de la relación:

Id_alumno	nombre_alumno	apellido_alumno	curso
00001	Juanito	Pérez	Matemáticas Español
00002	María	Rosas	Geografía Física Español
00003	Toño	González	Química

Normalización

■ PRIMERA FORMA NORMAL GRUPOS REPETIDOS A TRAVÉS DE COLUMNAS.

Id_alumno	nombre_alumno	apellido_alumno	Curso 1	Curso 2	Curso 3
00001	Juanito	Pérez	Matemáticas	Español	
00002	María	Rosas	Geografía	Física	Español
00003	Toño	González	Química		

Problemas lógicos:

- Dificultad en hacer consultas a la tabla. ¿"Qué alumnos toman el curso Español?" y ¿"Qué pares de alumnos comparten un curso?"
- Al alumno 00003 se le puede dar equivocadamente un valor para el Curso 2 que es exactamente igual que el valor de su Curso 1.
- La restricción de cursos por alumno a tres. Si un alumno tiene cuatro cursos, estamos obligados a guardar solamente tres y dejar el cuarto sin guardar. Esto significa que el diseño de la base de datos está imponiendo restricciones al proceso del negocio, en vez de (como idealmente debe ser el caso) al revés.

Normalización

- PRIMERA FORMA NORMAL REPETICIÓN DE GRUPOS DENTRO DE COLUMNAS.

Id_alumno	nombre_alumno	apellido_alumno	Curso
00001	Juanito	Pérez	Matemáticas, Español
00002	María	Rosas	Geografía, Física, Español
00003	Toño	González	Química

El atributo "Curso" es semánticamente confuso, ya que ahora puede representar, o un curso, o una lista de cursos, o de hecho cualquier cosa. Una consulta como "¿Qué pares de alumnos comparten un curso?" es imposible de formular, dada la necesidad de proveerse de listas de cursos así como cursos individuales. Con este diseño son también imposibles de definir significativas restricciones en cursos.

Id_alumno	nombre_alumno	apellido_alumno	Curso
00001	Juanito	Pérez	Matemáticas, Español
00002	María	Rosas	Geografía, Física, Español
00003	Toño	González	Química

■ PRIMERA FORMA NORMAL

Id_alumno	nombre_alumno	apellido_alumno
00001	Juanito	Pérez
00002	María	Rosas
00003	Toño	González

ALUMNO(id_alumno, nombre_alumno, apellido_alumno)

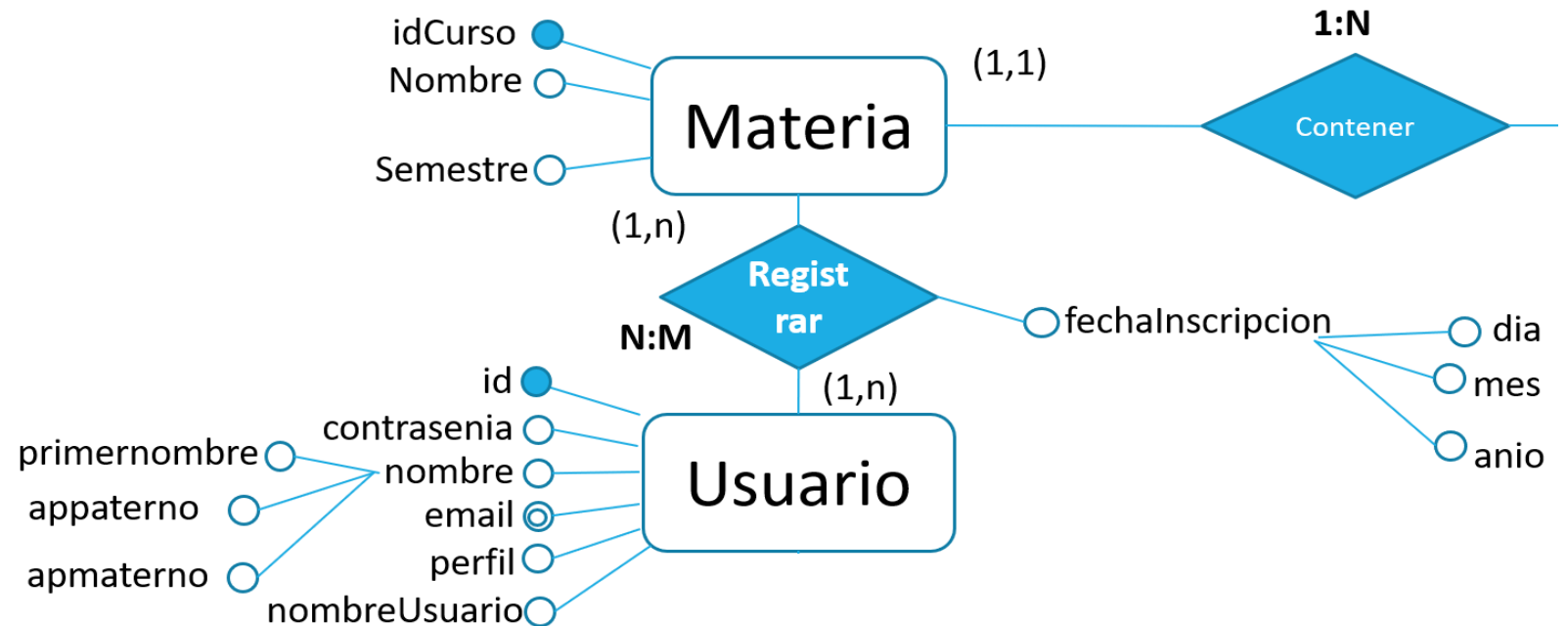
Id_alumno	curso
00001	Matemáticas
00001	Español
00002	Geografía

CURSO_ALUMNO(id_alumno, curso)

Normalización

■ PRIMERA FORMA NORMAL

- ✓ La primera forma normal también prohíbe los atributos compuestos que por sí mismos son multivaluados.



1.2 Diseño lógico de base de datos

En resumen

Primera Forma Normal (1FN):

- a) Cada relación debe tener una clave primaria.
- b) Eliminar grupos repetidos, definiendo atributos llave y no llave apropiadamente.
- c) Atomicidad: Cada atributo debe contener un valor único, no un conjunto de valores.

Normalización

Supongamos que un diseñador principiante desea guardar los nombres y los números telefónicos de los clientes y procede a definir una tabla Cliente como la que sigue:

IDcliente	Nombre	Apellido	Telefono
355	Juan José	Rodríguez	228-8548902,
422	María	Fernández	555-55928713, 555-55339874
833	Fernando	Carreón	229-2485730, 228-2385768, 229-0384759

La 1FN prohíbe a un campo contener más de un valor de su dominio de columna.

Normalización

IDcliente	Nombre	Apellido	Telefono
355	Juan José	Rodríguez	228-8548902
422	María	Fernández	555-55928713 555-55339874
833	Fernando	Carreón	229-2485730 228-2385768 229-0384759

Un diseño que está completamente en 1FN hace uso de dos tablas: una tabla de cliente y una tabla de teléfono del cliente.

CLIENTE

IDcliente	Nombre	Apellido	Telefono
355	Juan José	Rodríguez	228-8548902,
422	María	Fernández	555-55928713, 555-55339874
833	Fernando	Carreón	229-2485730, 228-2385768, 229-0384759

CLIENTE

<u>IDcliente</u>	Nombre	Apellido
355	Juan José	Rodríguez
422	María	Fernández
833	Fernando	Carreón

En este diseño no ocurren grupos repetidos de números telefónicos. En lugar de eso, cada enlace Cliente-a-Teléfono aparece en su propio registro

TELEFONO-CLIENTE

<u>IDcliente</u>	<u>Telefono</u>
355	228-8548902
422	555-55928713
422	555-55339874
833	229-2485730
833	228-2385768
833	229-0384759

Normalización

Otro ejemplo de grupos repetidos:

IDorden	Fecha	IDcliente	Nom_cliente	Estado	Num_prod	Desc_prod	Cant	Precio
2301	10/03/20	101	Pepe	Ver	356	Red	3	200
2301	10/03/20	101	Pepe	Ver	298	Raqueta	6	500
2301	10/03/20	101	Pepe	Ver	900	Paq-2	2	700
2302	15/04/20	107	Martin	Mex	127	Paq-5	1	900
2303	19/05/20	110	Carmen	Gdl	367	PelotasT	8	600
2303	19/05/20	110	Carmen	Gdl	788	Funda	3	1000

En base a los datos de la siguiente tabla, aplicaremos primera forma normal.
Contienen grupos repetido para NUM_PROD, DESC_PROD, CANT y PRECIO

Normalización

Otro ejemplo de grupos repetidos:

La 1FN prohíbe los grupos repetidos, por lo tanto tenemos que convertir a la primera forma normal.

Los pasos a seguir son:

- Eliminar los grupos repetidos.
- Crear una nueva tabla con la PK (Primary Key o Llave Primaria) de la tabla base y el grupo repetido.

Las tuplas deben quedar divididas en dos tablas que llamaremos ÓRDENES y ARTICULOS_ORDENES.

IDorden	Fecha	IDcliente	Nom_cliente	Estado	Num_prod	Desc_prod	Cant	Precio
2301	10/03/20	101	Pepe	Ver	356	Red	3	200
2301	10/03/20	101	Pepe	Ver	298	Raqueta	6	500
2301	10/03/20	101	Pepe	Ver	900	Paq-2	2	700
2302	15/04/20	107	Martin	Mex	127	Paq-5	1	900
2303	19/05/20	110	Carmen	Gdl	367	PelotasT	8	600
2303	19/05/20	110	Carmen	Gdl	788	Funda	3	1000

ÓRDENES

<u>IDorden</u>	Fecha	IDcliente	Nom_cliente	Estado
2301	10/03/20	101	Pepe	Ver
2302	15/04/20	107	Martin	Mex
2303	19/05/20	110	Carmen	Gdl

PRODUCTOS_ORDENES

<u>IDorden</u>	<u>Num_prod</u>	Desc_prod	Cant	Precio
2301	356	Red	3	200
2301	298	Raqueta	6	500
2301	900	Paq-2	2	700
2302	127	Paq-5	1	900
2303	367	PelotasT	8	600
2303	788	Funda	3	1000

Actividad.

Convertir a la primera forma normal la siguiente relación:

PROFESORES

NoPersonal	Nombre	aPaterno	aMaterno	GradoAcad
27654	Erika	Meneses	Rico	Licenciatura en Informática, Maestría en Sistemas
37654	Jesús	Martinez	Vázquez	Licenciatura en Derecho, Maestría en Legislación, Doctorado en Derecho
15287	Verónica	Cortés	Valdés	Licenciatura en Informática, Maestría en Sistemas, Doctorado en Ciencias

Actividad.

Convertir a la primera forma normal la siguiente relación:

ESTUDIANTE_SOLICITA_BECA

Cod_Estud	Nombre_E	Apellido	DNI	Dirección	Cod_Beca	Nombre	Requisito	Fecha
01234	Roberto	Hens	464367	Antonio López 43	A22321	METRICA	Ing. Téc.	10/10/06
76346	Luis	Garcia	345347	Av. Ciudades 29	B56784	ERASMU	Ing. Téc.	12/11/06
76346	Luis	Garcia	345347	Av. Ciudades 29	A22321	METRICA	Ing. Téc.	14/10/05
76346	Luis	García	345347	Av. Ciudades 29	G65434	HIMMPA	Ingenie.	15/09/06
01234	Roberto	Hens	464367	Antonio López 43	G65434	HIMMPA	Ingenie.	17/09/06
98776	Gregorio	Celada	885764	Pl. Pasises 67	G65434	HIMMPA	Ingenie.	21/09/06
01234	Roberto	Hens	464367	Antonio López 43	B56784	ERASMU	Ing. Téc.	11/11/05
98776	Gregorio	Celada	885764	Pl. Pasises 67	B56784	ERASMU	Ing. Téc.	10/10/05
01234	Roberto	Hens	464367	Antonio López 43	A22321	METRICA	Ing. Téc.	12/11/05
23245	Mercedes	García	809234	Río Miño 2	A22321	METRICA	Ing. Téc.	17/09/06

Bibliografía

Miguel, A., Piattini M. y Marcos, E. (2000). *Diseño de bases de datos relacionales*. México: Alfaomega:Ra-Ma.

Silberschatz, A., Korth, H. y Sundarshan, S. (2002). *Fundamentos de Bases de Datos*. (4a ed.). McGraw-Hill: Madrid.

Date, C. J. (2001). *Introducción a los sistemas de bases de datos*. (7a ed.). Tr. Sergio Luis María Ruiz Faudón. México: Pearson Educación.

Elmasri, R. Navathe (2002). *Sistemas de Bases de Datos. Conceptos fundamentales*. (3a ed.). Madrid:Addison-Wesley Iberoamericana:Pearson Educación.

Marqués, M. (2001). *Apuntes de ficheros y bases de datos*. Universitat Jaume I, Campus de Riu Sec. España. consultado el 15 de julio 2006, en: <http://www3.uji.es/~mmarques/f47/apun/apun.html> Si no encuentras el documento, pulsa [aquí](#).