1FN

- "1. Eliminar los grupos repetitivos de la tablas individuales.
- 2. Crear una tabla separada por cada grupo de datos relacionados.
- 3. Identificar cada grupo de datos relacionados con una clave primaria."

2FN

- "1.Crear tablas separadas para aquellos grupos de datos que se aplican a varios registros.
- 2. Relacionar estas tablas mediante una clave externa. "

3FN

- "1.Eliminar aquellos campos que no dependan de la clave"

IMPLEMENTACIÓN DE LA BASE DE DATOS

Saber más …

Como es lógico las bases de datos han ido evolucionando desde sus comienzos a principios de la década del 70 a la actualidad.

Las primeras implementaciones llamadas de tipo o modelo Jerárquicas no permitían las relaciones muchos a muchos. A estas le siguieron las bases de datos en Red, que salvaban dicha limitación, pero ambas eran difíciles de implementar ya que su diseño físico era muy complejo y de él dependía que su rendimiento fuese eficiente.

En la actualidad las bases de datos son del tipo relacional, también orientada a objetos. Pero es el primer tipo el más difundido.

Existe mucho software de distintos fabricantes de bases de datos relacionales. Podemos mencionar Informix de IBM, SQL Server de Microsoft y Oracle entre los productos pagos.

MySql es un sistema de distribución libre que se puede correr tanto en computadoras con sistema operativo Windows como Linux. MySql es utilizado en la mayoría de las implementaciones de páginas web dinámicas.

Para materializar la base de datos que hemos diseñado, debemos elegir el tipo de base de datos y el software que utilizaremos.

En el curso implementaremos una base de datos relacional con MySql en computadoras con sistema operativo Windows.

BASES DE DATOS RELACOIONALES

Como hemos visto en la primera unidad, una base de datos está formada por un conjunto de programas que permite su funcionamiento y administración y los archivos que almacenan los datos de los usuarios. Estos tienen una organización física compleja que permite mediante índices la localización de cualquier dato en forma rápida.

Por fortuna el diseñador de la base no tiene que trabajar sobre los archivos físicos sino que lo hace sobre la vista lógica de los datos.

La vista lógica de una base de datos relacional consiste en un conjunto de tablas, a cada una de las cuales se le asigna un nombre exclusivo.

Una tabla está formada por filas y columnas, la primera fila contiene el nombre de la columna.

Saber más …

Cada fila de la tabla representa una relación entre un conjunto de valores. Dado que cada tabla es un conjunto de dichas relaciones, hay una fuerte correspondencia entre el concepto de tabla y el concepto matemático de relación, del que toma su nombre el modelo de datos relacional

Cada entidad definida tendrá una tabla donde se almacenan los datos. Por cada atributo que hemos relevado habrá una columna. Se debe definir qué tipo de valores pueden tener los atributos (números enteros, con decimales, valores alfabéticos, fechas, etc.).

Saber más …

Al conjunto de los valores que puede tomar un atributo se lo llama dominio.

Las filas se denominan tuplas, la relación sólo contendrá un subconjunto del conjunto de todas las tuplas posibles resultantes del producto cartesiano de los dominios de los atributos.

Se exigirá que, para todas las relaciones, los dominios de todos los atributos sean atómicos. Un dominio es atómico si los elementos del dominio se consideran unidades indivisibles.

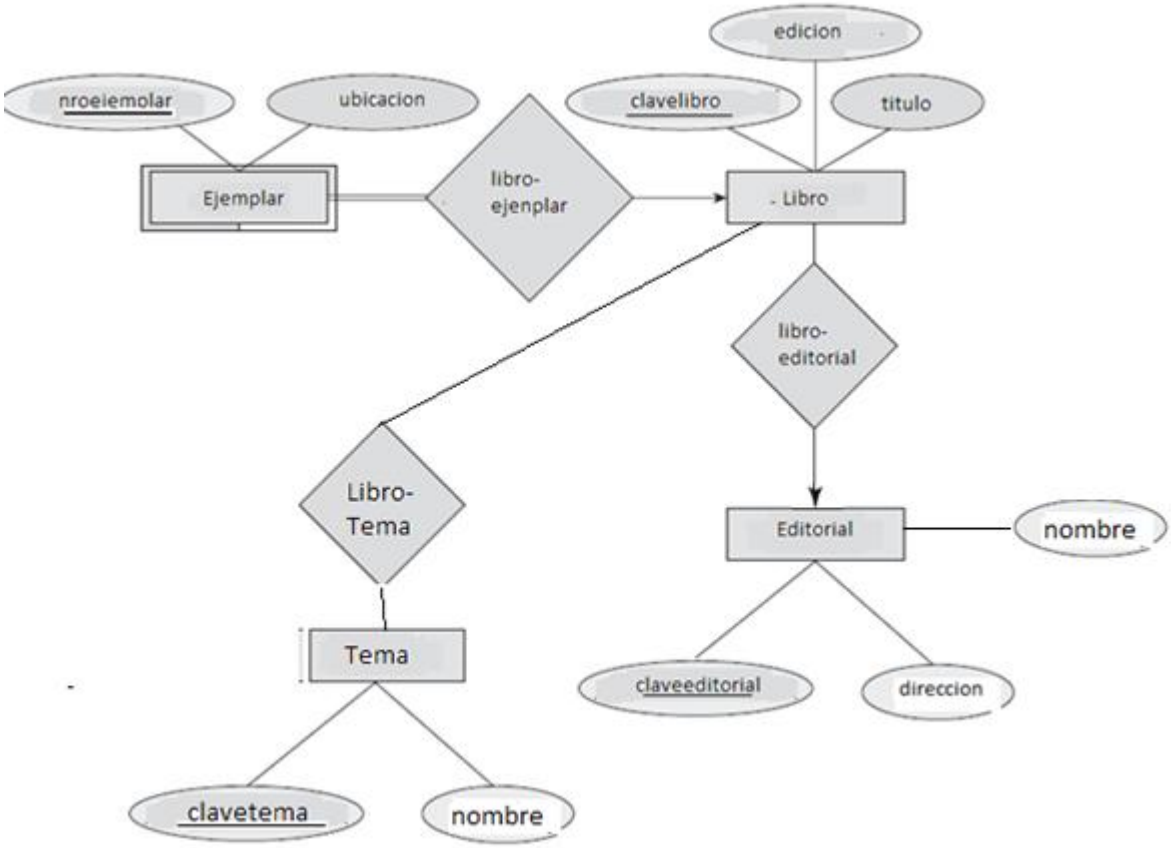
El concepto de relación se corresponde con el conceptode variable de los lenguajes de programación.

El concepto de esquema de la relación se corresponde con el concepto de definición de tipos.

PASAJE DEL MODELO E-R A TABLAS

El pasaje del modelo E-R a tablas es directo en el caso de las entidades, cada entidad dará lugar a una tabla, la clave primaria definida para la entidad será la clave primaria de la tabla.

Tomaremos para ejemplificar el siguiente Diseño E-R:



El mismo corresponde a parte del diseño de una base de datos para una biblioteca.

Tenemos cuatro entidades, *Libro* con los atributos *clave*, *título* y *edición*; *Ejemplar* con *número de orden* y *ubicación*; *Editorial* con *clave*, *nombre* y *dirección*; por último *Tema* con *clave* y *nombre*.

Definimos todas las claves como números enteros y los restantes campos de texto con una longitud de hasta 40 caracteres.

Tabla Libros

Clave-libro	**Título**	**Edición**
100	Programación en C	2003
105	MySQL y Webs dinámicas	2010

Tabla Editorial

Clave-editorial	**Nombre**	**Dirección**
10	El ateneo	Santa Fe 1433, CABA
21	Cúspide	Cerrito 520, CABA
23	Paraninfo	Tucumán 2210

Tabla tema

Clave-tema	**Nombre**
1	Programación en html
2	Sintaxis de SQL
3	Administración de servidores WEB
4	Archivos en C
5	Vectores y matrices

Tabla ejemplar

Numero-orden	**Mueble**	**Estante**
1	A	7
2	A	7
1	A	7
2	A	7
3	A	7

En las tres primeras tablas el campo subrayado será la clave primaria, esos valores no se repiten en toda la tabla.

En *Ejemplares* los números comienzan en 1 para cada ejemplar de un libro. Por ejemplo si del libro "*Martín Fierro*" edición 1990 de editorial *Ateneo* hay tres ejemplares serán los números 1, 2 y 3. Las ubicaciones de los ejemplares son letras y números que hacen referencia al nombre del mueble y al estante donde se guardan.

Vemos que *Ejemplar* es una entidad débil ya que para cada libro los ejemplares se numeran a partir de uno, y la ubicación hace referencia al armario y estante. Por lo que vemos que no hay un conjunto de valores que puedan ser únicos en cada fila para constituir la clave primaria.

Almacenando relaciones uno a muchos

Para representar la relaciones *uno a varios* la *clave primaria* de la entidad *uno* es incluida como atributo en la entidad *varios* y declarada como *clave extranjera* para que se valide la integridad referencial.

Ahora las tablas quedan:

Tabla Libros

Clave-libro	**Título**	**Edición**	***Clave-editorial***
100	Programación en C	2003	21
105	MySQL y Webs dinámicas	2010	10

Tabla ejemplar

Numero-orden	***Clave-libro***	**Mueble**	**Estante**
1	100	A	7
2	100	A	7
1	100	A	7
2	105	A	7
3	105	A	7

De ejemplar a libro hay una relación muchos a uno. Para representar la relación entre libro y ejemplar la clave de libro es incluida en la tabla de ejemplar (en cursiva por ser una clave extranjera).

Ahora el conjunto numero-orden + clave-libro en único y conforma la clave primaria.

La clave primaria se formará con atributos propios más la clave primaria de la entidad a la que está subordinada.

Almacenando relaciones *muchos a muchos*

Para representar las relaciones *muchos a muchos* se crea una tabla que tiene las claves primarias de ambas entidades que serán declaradas como claves extranjeras y serán clave primaria esta tabla.

Tabla tema-libro

Clave-tema	***Clave-libro***
10	100
21	105
21	110

EJERCICIO MODELO

En la unidad anterior presentamos el diseño de una base de datos para una biblioteca. Realizaremos el paso siguiente para la implantación del diseño en una base relacional, es decir definir las tablas que surgen del mismo y sus claves primarias.

Del DER surgen siete entidades: *libros*, *ejemplares*, *temas*, *editoriales*, *autor*, *socio* y *prestamos*.

También seis relaciones:

- *Uno a muchos* de préstamo a socio.
- *Uno a muchos* de préstamo a ejemplar.
- *Uno a muchos* de ejemplar a libro.
- *Uno a muchos* de libro editorial.
- *Muchos a muchos* de libro a autor.
- *Muchos a muchos* de libro a tema.

Por lo tanto el pasaje a tablas implicará la definición de nueve tablas, una por cada entidad y una por cada relación *muchos a muchos*.

Las definiremos con el nombre de la tabla seguido de la lista de atributos entre paréntesis. El o los atributos seleccionados como clave primaria se subrayan y se aclara con la letra PK (Primary Key).

Como complemento se podrá incluir al lado de cada nombre de atributo el tipo de dato que deberá cargarse y si ese dato es obligatorio o no.

Para materializar las relaciones *uno a muchos* la clave de la entidad *uno* es incluida como atributo en la entidad *uno* incluyendo al lado del nombre las letras FK que indican que es la clave de otra entidad.

Estas definiciones sirven de base para escribir el código en lenguaje SQL para la creación de la tabla en la base de datos.

Socio(DNI PK número, *apellido* letras 40, dirección letras 40)

Préstamo(*Número* PK número, DNI FK numero obligatorio, *código_libro* FK número obligatorio, *fecha_prestamo* fecha obligatorio, *fecha_devolución* fecha obligatorio)

Editorial(*Codigo_editorial* PK número, *nombre* letras 30 obligatorio)

Tema(*código_tema* PK número, *nombre* letras 50 obligatorio)

Libro(*código_libro* PK número, *código_editorial* FK número obligatorio, *título* letras 50 obligatorio, *edición* número)

Ejemplar(*Número* PK número, *código_libro* PK FK número, *armario* letras 4, *estante* número)

Autor(*Clave_autor* PK número, *nombre* letras 50 obligatorio)

Tablas para las relaciones muchos a muchos

Escrito_por(*Clave_libro* PK FK número; *Clave_autor* PK FK número)

Trata_sobre(*Clave_libro* PK FK número; *Clave_tema* PK FK número)

NORMALIZACIÓN

Hemos realizado el diseño para una base de datos relacional definiendo las tablas que la conformarán. ¿Es correcto este diseño? ¿Se puede mejorar?

Un diseño correcto evita datos que se repitan, lo que implicaría que puedan estar desactualizados, por ejemplo si el teléfono de una persona está en varias tablas puede estar en nuevo número en unas y el viejo en otras, además esto desperdiciaría espacio en disco.

Los únicos datos que se repiten en varias tablas son las claves primarias necesarias para representar las relaciones. El SGBD asegura que si se modifica en la tabla que es clave primaria automáticamente lo hace en todas en las que es clave extranjera.

El proceso de eliminar las redundancias innecesarias se llama normalización, que significa aplicar la norma es decir que se ajuste a una regla.

El proceso de normalización generalmente implica la reducción de una tabla dada, descomponiéndola en varias diferentes a fin de eliminar redundancias. Este proceso debe ser sin pérdida de información, los datos desagregados deben poder ser reagrupados de ser necesario.

Primera forma normal

Una tabla está en 1FN *si y sólo si*, en tolas las filas se almacena un sólo valor para cada atributo.

Supongamos que en la entidad libro con los atributos *código* PK, *titulo*, también consideramos *tema* como un atributo que puede tener múltiples valores.

código	**titulo**	**tema**
10	Análisis matemático	Límites, derivadas, integrales
20	Historia antigua	Cultura griega, Esparta, Caldeos, Fenicios
30	Castellano	Análisis sintáctico, conjugación de verbos

Esta tabla no está en primera forma normal un intento podría ser hacer una fila para cada valor de *tema* de un libro, la tabla quedaría así:

código	**titulo**	**tema**
10	Análisis matemático	Límites
10	Análisis matemático	Derivadas
10	Análisis matemático	integrales
20	Historia antigua	Cultura griega
20	Historia antigua	Esparta
20	Historia antigua	Caldeos
20	Historia antigua	Fenicios
30	Castellano	Análisis sintáctico
30	Castellano	Conjugación de verbos

Ahora la tabla está en 1FN, sin embargo esta solución presenta redundancias, el nombre del libro se repite y además *código* no es suficiente para formar clave primaría, para formarla tendríamos que usar *código* más *tema*.

Por otro lado que pasaría si al cargar otro libro de historia antigua que toca el tema de la "*cultura griega*" se registra como tema "*la civilización de los griegos*". Sencillamente cuando se busquen libros que traten "*cultura griega*" este no aparecería.

Como veremos ahora la tabla no cumple la tercera forma normal, para eliminar estos problemas *tema* debe ser considerado una entidad y no un atributo.

Segunda forma normal

Una relación está en 2FN si está en 1FN y todo atributo que no sea clave es dependiente de la clave primaria, no puede haber atributos que dependen de otro atributo, a esto se lo llama dependencias transitivas.

En el ejemplo la tabla *libro* con clave primaría *clave-libro* no está en 2FN ya que *nombre-editorial* depende de *clave-editorial*.

código	**titulo**	**Clave-editorial**	**Nombre-editorial**
10	Análisis matemático	100	Mc Graw Hill
20	Historia antigua	215	Planeta
30	Castellano	130	Kapeluz
40	Historia Moderna	215	Planeta

Tercera forma normal

Una tabla está en 3FN si está en 2FN y todo atributo que no sea clave es dependiente irreduciblemente de la totalidad de la clave primaria, no puede haber dependencias parciales. La clave primaría debe ser compuesta.

En la tabla libro que pasamos a 1FN, al tomar una clave primaria compuesta, título sólo depende de una parte de ella de ahí que se forme un grupo repetitivo.

Otro ejemplo: en la tabla *ejemplar* con clave primaría **clave-libro** + *número de orden* no está en 3FN ya que *titulo* sólo depende de *clave-libro*.

clave-libro	nro-orden	titulo	ubicación
10	1	Análisis I	a-1
10	2	Análisis I	a-1
10	3	Análisis I	a-1
50	1	Geometris descriptiva	d-5
11	2	Geometris descriptiva	d-5
11	1	Analisis Matemático	a-3
11	2	Analisis Matemático	a-3
11	3	Analisis Matemático	a-4