

INGENIERÍA EN SISTEMAS

ASIGNATURA: BASE DE DATOS

CONTENIDISTA: Ing. Silvia Cobialca

UNIDAD 2: DISEÑO LÓGICO DE BASES DE DATOS

INDICE – UNIDAD 2:

DISEÑO LOGICO DE BASES DE DATOS	4
INTRODUCCIÓN.....	4
1. CONCEPTOS DEL MODELO RELACIONAL.....	4
2. MODELO ENTIDAD-RELACIÓN	7
3. ENTIDADES. GRADO. ATRIBUTOS. DOMINIOS.....	9
ACTIVIDAD 1.....	10
4. CLAVES CANDIDATAS, ALTERNATIVAS Y CLAVES PRIMARIAS.....	10
ACTIVIDAD 2.....	12
5. RELACIONES. CARDINALIDAD.	12
ACTIVIDAD 3.....	13
ACTIVIDAD 4.....	15
6. DICCIONARIO DE DATOS.....	16
7. REFINAMIENTO DEL MODELO. DISEÑOS DE CALIDAD.....	21
ACTIVIDAD 5.....	23
ACTIVIDAD 6.....	26
ACTIVIDAD 7.....	27
SÍNTESIS DE LA UNIDAD	28

MAPA DE LA UNIDAD 2: DISEÑO LÓGICO DE BASES DE DATOS

PROPÓSITOS

En esta unidad nos proponemos explicarle qué es un Diagrama Entidad-Relación de una base de datos (o DER por sus siglas en inglés). Veremos las definiciones y utilidad y algunos ejemplos ilustrativos que servirán para aclarar los conceptos.

OBJETIVOS

- ✓ Conocer qué es un DER.
- ✓ Identificar las partes componentes de un DER: entidades, atributos y relaciones.
- ✓ Analizar qué es una clave, cuáles son los tipos de claves y para qué sirven.

CONTENIDOS

Para que alcance los objetivos, los contenidos que abordará son los siguientes:

- 1) Conceptos del modelo relacional.
- 2) Modelo entidad-relación (E-R).
- 3) Entidades. Grado. Atributos Dominios.
- 4) Claves candidatas, alternativas y claves primarias. Integridad de la entidad.
- 5) Relaciones. Cardinalidad.
- 6) Diccionario de datos.
- 7) Refinamiento del modelo. Diseños de calidad. Formas normales.

PALABRAS CLAVES

Entidad, relación, atributo, dominio, clave, DER



BIBLIOGRAFÍA DE CONSULTA

ELMASRI, RAMEZ/NAVATHE, SHAMKANT (2011). *Fundamentos de Sistemas de Base de Datos*. 6ta Ed, EEUU: Pearson / Addison Wesley.

SILBERSCHATZ, ABRAHAM (2006). *Database System Concepts*. 5th Ed, NY: McGraw-Hill Companies Inc.

EVALUACIÓN

A lo largo de cada unidad, encontrará actividades de autoevaluación que le permitirán hacer un seguimiento de su aprendizaje.

DISEÑO LOGICO DE BASES DE DATOS

INTRODUCCIÓN

Para comenzar con esta unidad, ya que en la Unidad 1 vimos lo que es un modelo conceptual, nos vamos a apoyar en ese conocimiento, y veremos cómo se realiza el diseño conceptual de una base de datos y cuáles son las partes del mismo. Veremos además algunos ejemplos de aplicación que servirán para aclarar las ideas vertidas.



ABORDEMOS EL LOGRO DEL PRIMER OBJETIVO DE LA UNIDAD:

- ✓ Conocer qué es un DER

Comencemos...

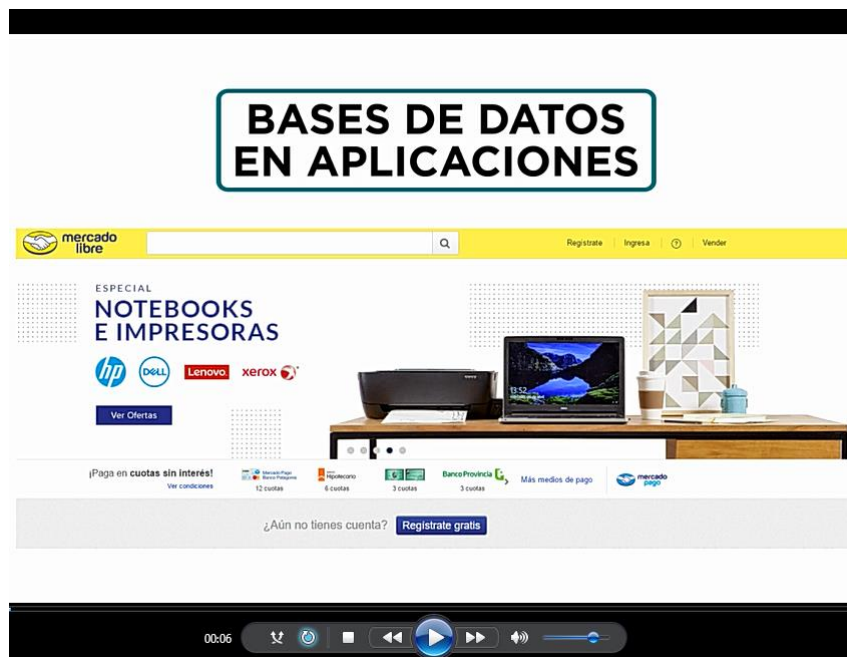


1. CONCEPTOS DEL MODELO RELACIONAL

Como vimos en la Unidad 1, el diseñador arma un esquema conceptual que refleja el sistema o negocio de interés, lo que hemos llamado “el mini-mundo” a modelar. Es decir, antes que nada, debe tener una idea de cuál es ese pequeño entorno que va a modelar, o sea cuál es el alcance del mismo, dónde comienza y dónde termina.

Para saber esto, generalmente los diseñadores se reúnen con el cliente y éste les “cuenta”, con sus palabras, lo que necesita. Es un trabajo arduo que implica mucha capacidad de interpretación por parte de los diseñadores ya que, a menudo, el cliente asume muchas funcionalidades por su amplio conocimiento del negocio y está en la habilidad del diseñador el hacer las preguntas correctas para terminar de entender lo que se desea modelar.

Veamos el siguiente video que nos aclara los conceptos vistos:



Esto es bastante parecido a lo que muestra la presentación de la Unidad 1. Le sugerimos volver a leerla para aclarar dudas de este proceso.

El cliente suele hablar de los datos que necesita para realizar su operación, atándolos a ciertas acciones que tienen que ver con el negocio.

Por ejemplo, podría decir:

“Tengo una serie de clientes a quienes vendo mis productos. Mis vendedores los visitan personalmente en sus locales, realizan la venta y luego tienen que mandarles un mail con la lista de los productos que compraron, la cantidad y el monto total de la factura, para que puedan verificar los valores y realizar el pago”.



Se preguntará en este momento, ¿Cómo pasamos de este simple relato a construir el modelo para una base de datos?

Bien, para “atar” el relato o “relevamiento del negocio” al modelo relacional, se deben encontrar aquellos **sustantivos importantes** para el cliente dentro del texto. Usted podrá notar, además,

la existencia de **otros sustantivos que en general están asociados a los sustantivos “importantes”**.

Por ejemplo, cliente y vendedor son sustantivos importantes. Ahora, mail también es un sustantivo, pero en el relato dice claramente que hay que enviarle un mail al cliente, es decir nos va a interesar conocer cuál es la dirección de correo electrónico del cliente.





ABORDEMOS EL LOGRO DEL SEGUNDO OBJETIVO DE LA UNIDAD:

- ✓ Identificar las partes componentes de un der: entidades, atributos y relaciones



2. MODELO ENTIDAD-RELACIÓN

Continuando desde el apartado anterior y ya conociendo cómo vamos a descubrir los diferentes elementos del modelo para un cierto problema, veremos ahora los pasos a seguir para el armado de dicho modelo.

Los sustantivos importantes son las ENTIDADES del modelo. Los sustantivos menos importantes que corresponden a las entidades (en nuestro ejemplo habíamos visto que el mail era uno de ellos) son los ATRIBUTOS.

Las entidades interactúan entre sí por medio de acciones, por ejemplo, los VENEDORES “visitan” LOCALES que “pertenecen a” los CLIENTES para “venderles” los PRODUCTOS. Las acciones son las “relaciones” del modelo.

Las entidades luego contendrán los datos relevantes que hemos indicado son sus atributos. Así en nuestro ejemplo, sabemos que del cliente necesitaremos saber su mail, seguramente también su nombre y dirección o número de teléfono. Tendríamos así los diferentes clientes:

		Atributos			
Entidad	Cliente	Nombre	Dirección	Email	Teléfono
	1	J Perez	Araoz 1340	jperez@g...	33029823
	2	A López	Frías 239	alopez@y...	44729929
	3	M Arias	Palpa 2237	marias@h...	48839376

Ahora que sabemos esto, estamos en condiciones de definir qué es el modelo relacional:

El modelo relacional representa a la base de datos como una colección de “entidades” relacionadas entre sí, donde cada entidad se puede ver como una “tabla”.

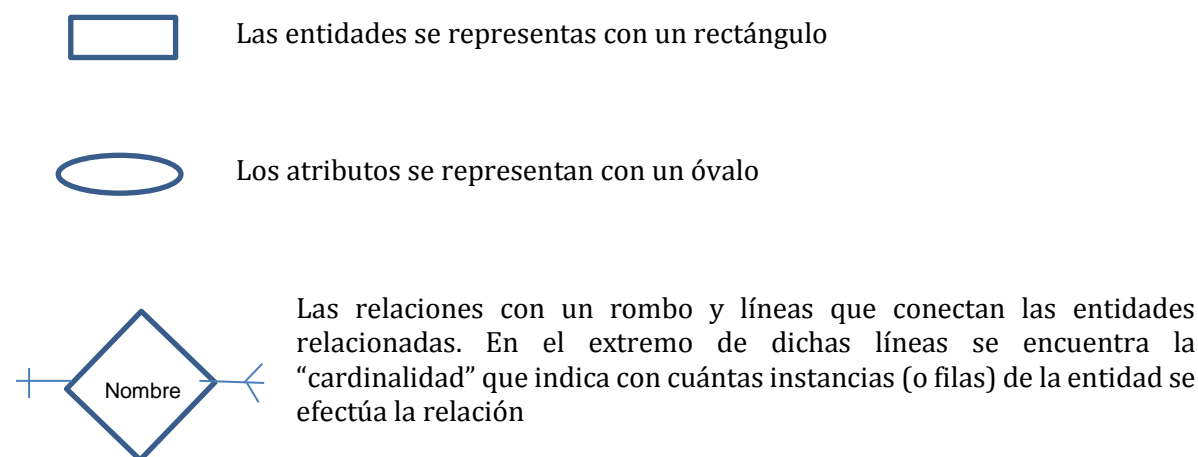
Podría pensarse en una tabla como si fuera una hoja de cálculo, con sus filas y columnas. Las filas simbolizan cada una de las instancias de la tabla, por ejemplo un cliente, un vendedor o una venta. Las columnas sirven para interpretar el significado de los valores de la fila.

En el modelo relacional la fila se denomina “T-upla”.

Las columnas de la tabla son los “atributos”.

La tabla, como dijimos anteriormente, se denomina “entidad”.

Además, como tenemos que construir un diagrama vamos a tener que definir cómo representar cada una de sus partes. Entonces podemos usar la siguiente convención:



Los diferentes valores que pueden aparecer en una columna o atributo llevan el nombre de “dominio” de ese atributo, y está muy relacionado con el tipo de dato del atributo, es decir si es un número o un texto.

Veamos un ejemplo de cómo sería el modelo del relato de los vendedores:

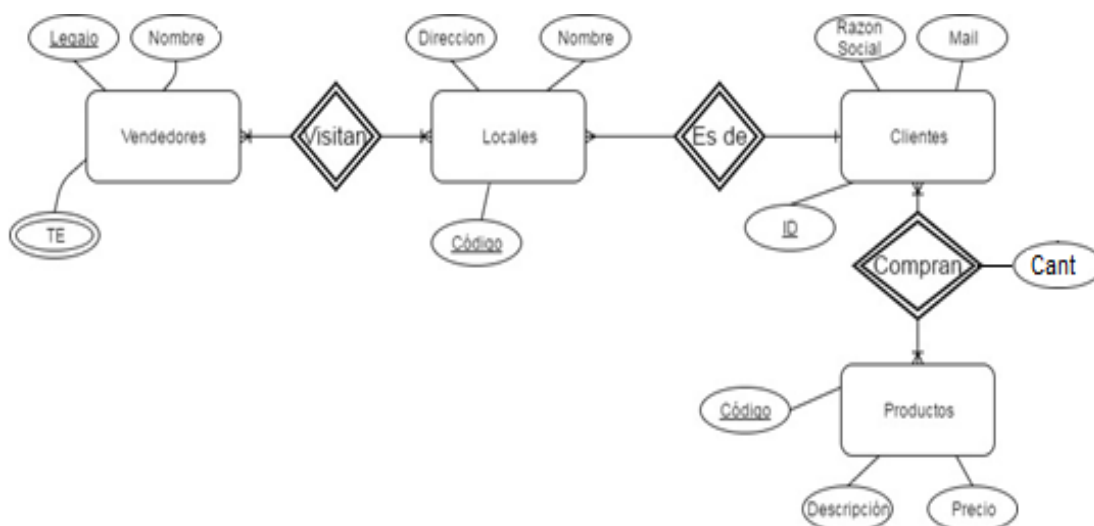


Figura 1 – Diagrama Entidad-Relación del ejemplo

Usted notará que entre clientes y productos hay una relación “compran” que simboliza el hecho de que los clientes van a solicitar ciertos productos a los vendedores. Se puede ver que hay un

atributo en la relación. Es decir el atributo no pertenece a las entidades sino que es propio de la relación. Este punto lo aclararemos en el apartado siguiente cuando veamos más ejemplos.



¿Puede a partir del diagrama, “leer” el relato que había hecho el cliente al diseñador?





3. ENTIDADES. GRADO. ATRIBUTOS. DOMINIOS.

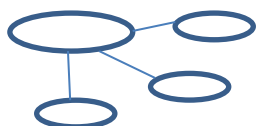
Hemos visto entonces que las entidades contienen atributos o mejor dicho, que los atributos describen a las entidades. La cantidad de atributos que tiene una entidad se denomina “grado” de la entidad.

Dijimos que los atributos se simbolizaban en el modelo E-R con un óvalo.


En el diagrama de la figura 1 se puede ver que hay diferentes tipos de óvalos, que simbolizan diferentes tipos de atributos. Estos son:

 **Atributo común:** corresponde a un atributo que tiene un valor único para cada tupla de la entidad, es indivisible y no derivable a partir de ningún otro atributo. Por ejemplo, el nombre de una persona, su apellido, un color (en el caso de un producto).

 **Atributo multivaluado:** es un atributo que puede tomar más de un valor. Un ejemplo es el del teléfono, ya que una persona puede tener más de un teléfono de contacto. Otro podría ser el email.



Atributo compuesto: es un atributo que puede formarse por la unión de varios atributos, por ejemplo la dirección es un caso de atributo compuesto, ya que se forma con la calle, el número, el código postal y la localidad.

 **Atributo calculado o derivado:** se puede calcular a partir del valor de otro atributo. Un ejemplo típico es la edad que puede calcularse a partir de la fecha de nacimiento.



¿Puede pensar otro ejemplo de cada tipo de atributo?

Seguramente usted habrá pensado ejemplos parecidos a los siguientes:

Compuestos: Fecha: {día, mes, año}; NombreApellido: {Nombre, Apellido}

Derivado: Monto = precio * cantidad * 1,21 (para calcular el monto a pagar con IVA incluido)

Multivaluado: teléfono, email, podría ser también una dirección de entrega

Comunes: Color, Altura, Descripción, Talle, Departamento, etc.

Cada atributo puede tomar un conjunto de valores posibles distintos. A ese conjunto de valores lo llamamos “**dominio**” del atributo. Existe la posibilidad de que diferentes atributos compartan el mismo dominio, por ejemplo en el atributo compuesto dirección, están las calles. Estas calles pueden ser las mismas para las direcciones de los clientes, o las de los empleados;

lo mismo puede pasar con las localidades. Se trata de atributos de diferentes entidades y sin embargo sus dominios coinciden.



ACTIVIDAD 1

Le pedimos que a continuación realice los ejercicios 1, 2 y 3 de la práctica 1 del campus, dichos ejercicios serán revisados durante el encuentro virtual de la semana. Si lo desea, puede utilizar el foro de la unidad para intercambiar ideas con sus compañeros durante la resolución y de esa manera ir afianzando conceptos vistos hasta aquí.



ABORDEMOS EL LOGRO DEL TERCER Y ÚLTIMO OBJETIVO DE LA UNIDAD:

- ✓ Analizar qué es una clave, cuáles son los tipos de claves y para qué sirven



4. CLAVES CANDIDATAS, ALTERNATIVAS Y CLAVES PRIMARIAS.

En una entidad o tabla, hemos visto que existen muchas t-uplas, tantas como datos tenga la tabla. Podría suceder que en el ejemplo de los vendedores, por accidente, dos vendedores diferentes visiten el mismo local e ingresen datos del cliente, generando de esta manera dos t-uplas para el mismo cliente. Para evitar esto es que existe un atributo específico de cada tabla o entidad que se utiliza para diferenciar una t-upla de la otra. Generalmente ese atributo es un tipo de dato de tamaño pequeño y fácil de identificar y diferenciar para los usuarios del sistema.

Por lo tanto:

La clave candidata es aquel atributo de una entidad que sirve para poder identificar unívocamente un set de valores de esa entidad (cada t-upla o instancia de la entidad).

Si tomáramos ese atributo conjuntamente con otro, por ejemplo, el código, que es nuestro atributo clave, y el nombre del socio de la biblioteca, ambos conformarían una super-clave, ya que entre ambos también podremos identificar unívocamente un dato de otro de dicha entidad pero, sabemos que con solo el código nos alcanzaría. Veamos el ejemplo específico:

Socios:

Nombre	Apellido	CUIT	Código
Juan	Pérez	27-112329826-0	1234
María	Gonzalez	20-156678901-2	2312
Juan	Pérez	20-214549742-1	2133
Jorge	Peres	30-178629743-0	2213
J	Perez	20-146652445-7	2132

En el ejemplo vemos varias t-uplas que corresponden a cierto socio Juan Pérez (o Jorge Peres? O J Perez?).



¿Cómo sabemos que no se trata del mismo socio y que alguien al tipear sus datos no cometió un error?

En principio, como el código es diferente, debemos suponer que se trata de un socio distinto. Seguramente al ingresarlo se le han tomado los datos suficientes para asegurarse que no se trata de la misma persona. Podés observar que el CUIT es diferente en todos los casos.



¿Por qué no se utilizó el CUIT como clave entonces?

Tranquilamente se podría haber utilizado el CUIT como clave. Es decir, en efecto, el CUIT era una clave candidata, tal como lo era el código, ya que ambos podían utilizarse para diferenciar una t-upla de la otra en la entidad Socios. Pero se eligió finalmente al código porque era un número más pequeño (o sea por comodidad).

El hecho que ambos atributos podían usarse indistintamente como claves hace que sean **claves candidatas o alternativas**.

Se podrían usar un par de atributos juntos como claves, por ejemplo nombre y apellido, pero vemos que tenemos dos socios con el mismo nombre y apellido, lo que invalida la posibilidad de utilizar este par como clave.

Ahora bien, ¿Podríamos utilizar CUIT y apellido como clave?

La respuesta es NO, porque en realidad el par {CUIT, Apellido} son una **SUPER clave** porque si solo usamos el CUIT nos alcanza para identificar una t-upla de otra, es decir nos sobraría el apellido (de ahí que es SUPER, más grande que la clave)

De toda esta discusión se desprende entonces que:

La clave primaria es aquella clave candidata que ha sido elegida para identificar unívocamente un set de valores de esa entidad.

O sea, de todas las posibles claves candidatas, hemos elegido a la clave primaria para identificar un set de valores o t-upla, respecto de otra en una cierta entidad.



ACTIVIDAD 2

Le pedimos que a continuación realice la actividad de Autoevaluación que le proponemos en el campus

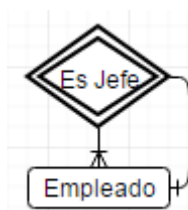


5. RELACIONES. CARDINALIDAD.

Ya vamos profundizando en los conceptos que conciernen a las diferentes partes del modelo relacional, y esta vez, nos centraremos en **las relaciones**.

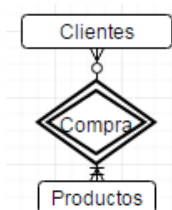
Vimos que la relación se simboliza con un rombo y líneas que unen las entidades relacionadas.

Según cuántas entidades estamos relacionando, tendremos relaciones de diferentes tipos, a saber:

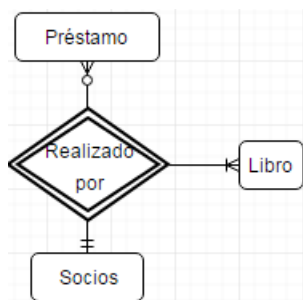


Unarias: la entidad se relaciona consigo misma, es decir solo una entidad participa de la relación.

El ejemplo típico es el de los empleados y sus jefes. Un jefe es también un empleado por lo tanto tendremos que el empleado jefe supervisa a muchos empleados (sus subordinados)



Binarias: dos entidades participan de la relación. Es el tipo más común.



Ternarias: en la relación participan tres entidades.



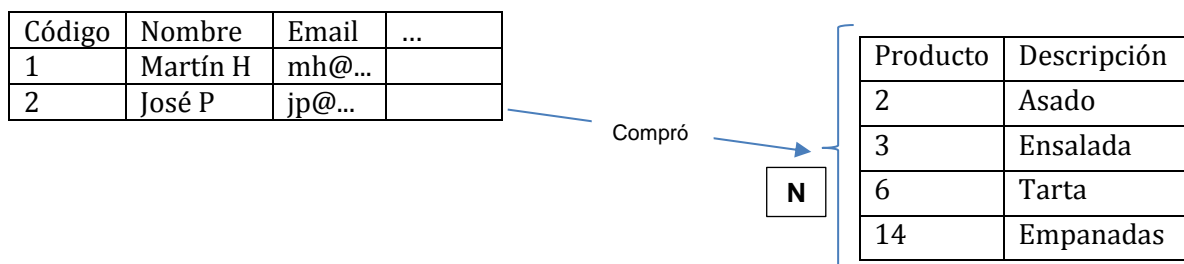
ACTIVIDAD 3

Le pedimos que a continuación realice los ejercicios 4 al 7 de la práctica 1 del campus, dichos ejercicios serán revisados durante el encuentro virtual de la semana. Si lo desea, puede utilizar el foro de la unidad para intercambiar ideas con sus compañeros durante la resolución y de esa manera ir afianzando conceptos vistos hasta aquí.

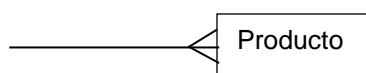
Continuemos....



Veamos en el caso de la relación binaria, “los clientes compran productos”. José que es un cliente, ¿cuántos productos de nuestra compañía podría comprar? Obviamente le permitiremos comprar muchos, y si es un cliente, es porque nos ha comprado alguna vez, y deseamos que siga comprando. Eso significa que en la relación entre ambas entidades, en el lado de los productos vamos a poner un símbolo que signifique muchos (o N).



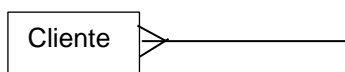
Esto lo hacemos con una patita de gallo como hemos mostrado previamente en el diagrama:



Continuemos ahora del otro lado de la relación, y lo pensamos así:

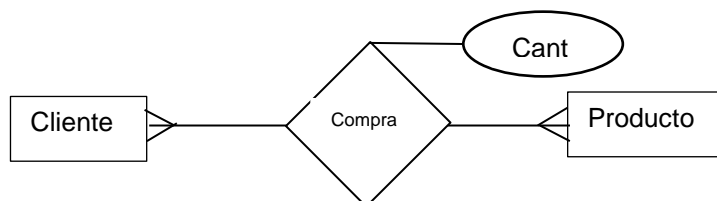
Al producto XXX (por ejemplo el producto de código=3, ensalada) ¿cuántos clientes me lo podrán comprar?

Seguramente si quiero que mi negocio funcione bien, querré vender mucho cada uno de mis productos, por lo tanto desearé que muchos clientes me compren el producto XXX. O sea pondremos una patita de gallo también del lado de los clientes.

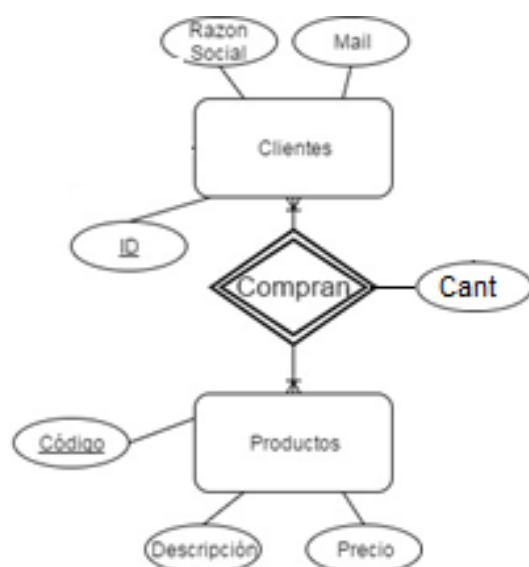


Entonces vemos que las relaciones tienen otro parámetro importante que es el que nos va a indicar cuántas t-uplas de cada entidad están involucradas en la relación. Esta es precisamente la "Cardinalidad".

Como vimos en el ejemplo, nuestra relación Clientes-Productos resulta entonces de cardinalidad N-M (muchos a muchos), **la cantidad comprada es un atributo de la relación y no de las entidades.**



Ahora volvamos al diagrama:



Ya vimos que en el ejemplo de los vendedores y los locales de principios de la unidad, la cantidad era un atributo de la relación N-M entre clientes y productos.

Ahora que sabemos que ambas tablas tienen una clave, podemos entender que la cantidad está modelando cuántas unidades de cada producto compró un cierto cliente.



¿Qué pasaría si pusiéramos la cantidad como atributo de Clientes?

Como sabemos que cada cliente solo aparece una vez en esa tabla, entonces la cantidad que figura para su t-upla indicaría que está destinado a comprar siempre la misma cantidad sin importar qué producto adquiriera.

De manera similar, si ponemos cantidad en la tabla de productos, indicaría que de un cierto producto solo se admitiría que los clientes compren siempre esa cantidad.

La cantidad, modelada dentro de la relación y no en las entidades que la conforman, indicará que es la cantidad de cada producto que ha comprado un cliente específico.

Pasemos ahora a otro ejemplo para ilustrar otro tipo de Cardinalidad: la de los empleados y sus jefes.

Empleados |— Pensemos en Juan, uno de nuestros empleados, ¿cuántos jefes directos tiene?

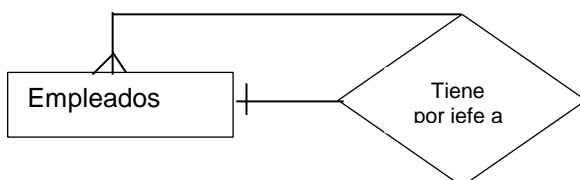
Seguramente uno solo, Federico. Por lo tanto del lado de los empleados jefes, la Cardinalidad es 1 (recordá que los jefes son también empleados).

Ahora veamos la relación del lado de Federico.

Legajo	Nombre	Jefe
2	Juan	4
1	Maria	4
4	Federico	9
7	Paula	4

Es un empleado que es jefe y pensemos cuantos empleados tiene a cargo. Lo más seguro es que sea más de uno, no importa el número exacto, solo si es uno o muchos. O sea del lado de los empleados la Cardinalidad es N.

Resulta entonces que la relación “Empleado tiene por jefe a” es N-1. O sea:



ACTIVIDAD 4

Le pedimos que a continuación repase los ejercicios realizados en las actividades anteriores y decida si las claves elegidas son correctas. ¿Qué cambiaría? Discuta con sus compañeros en el foro de la unidad, los ejercicios serán revisados con el docente en el encuentro virtual de la semana.



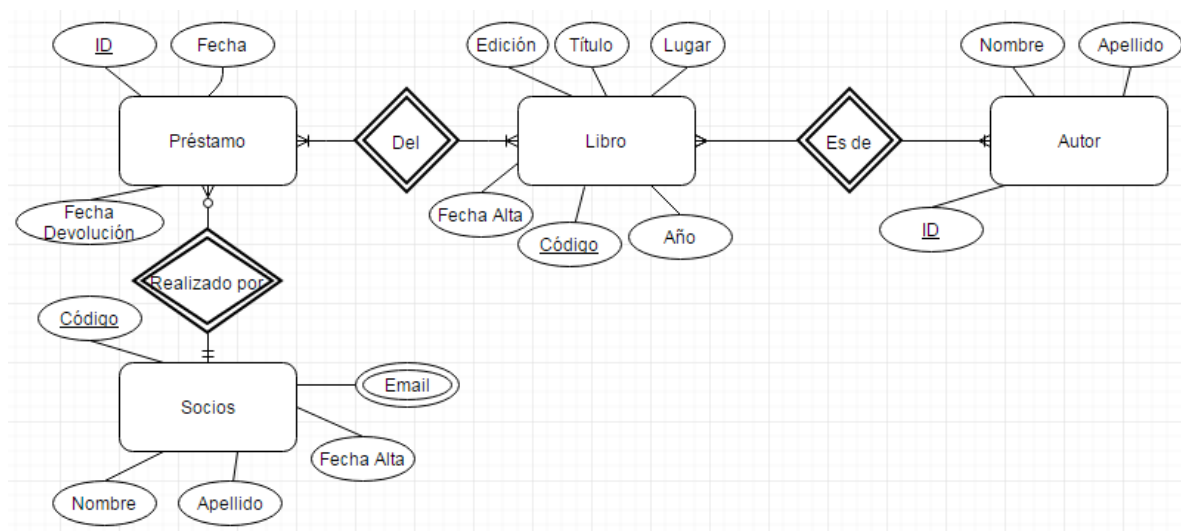
6. DICCIONARIO DE DATOS.

Ya hemos visto en los apartados anteriores cómo debemos hacer para armar el modelo conceptual o modelo E-R, primero identificando las entidades involucradas y sus atributos, y luego relacionándolas entre sí.

Pero en la Unidad 1 también habíamos expresado que el diseño conceptual luego derivaba (o se mapeaba) al esquema interno y al de vistas o externo.

Para que esto suceda, los desarrolladores y los diseñadores deben trabajar juntos y definir el diccionario de datos pasando el esquema conceptual “a tablas”.

Para mostrar cómo es este proceso, vamos a volver al esquema de los préstamos de libros de principios de la unidad.



Primero tomaremos cada entidad y armaremos la tabla correspondiente.

Nos quedará de esta manera:

Libro: {Código, Título, Lugar, Año, Fecha Alta, Edición}

Autor: {ID, Nombre, Apellido}

Socios: {Código, Fecha Alta, Nombre, Apellido}

Préstamo: {ID, Fecha, Fecha Devolución} (tentativo y a revisar)



Se preguntará ¿Qué pasó con el email de los socios que no aparece entre los atributos de la tabla socios, que corresponde a la entidad socios del modelo conceptual?

Veamos este caso con un ejemplo: tomemos uno de los socios María González cuyo código (o clave de la entidad Socios) es 14.

Ella, al momento de asociarse a nuestra biblioteca, nos dio dos mails, el de su trabajo `magonzalez@usu.com.ar` y su mail personal `magonza@gmail.com`.

Como recordará, cuando definimos el atributo clave, dijimos que era un atributo que identificaba unívocamente cada t-upla o fila de la tabla. Esto significa que no puede haber dos filas que tengan el código 14. Por lo tanto sería imposible poner ambos emails en la tabla de socios.

Entonces volvemos a la pregunta:



¿Cómo hacemos para modelar el email?

Seguramente usted estará respondiendo que vamos a necesitar una tabla para el Email, de lo contrario no podríamos tener guardado en nuestro modelo más de un email por socio.

Vamos a ver a continuación cómo se debe modelar esta nueva tabla.

Hay que recordar que debemos mantener los diferentes mails para cada socio y, como hemos establecido, es muy importante tener un atributo clave que nos servirá para poder identificar las t-uplas entre sí.

No podremos usar el mail como atributo clave porque, como ya mencionamos, siempre conviene que este atributo sea un dato de tipo numérico, por lo tanto debemos utilizar otro atributo como clave.

Podemos poner un número que se vaya incrementando en 1 cada vez que creamos una t-upla nueva (los DBMS tienen esta funcionalidad). Lo denominaremos ID (o código).

¿Y qué nos falta? ¿Cómo lo hacemos?



¡Nos falta una manera de “atar” los números de teléfono a María, por supuesto!

Vimos que una de las reglas del mapeo del esquema conceptual al esquema interno, es que cada atributo multivaluado se pasa como una tabla nueva.

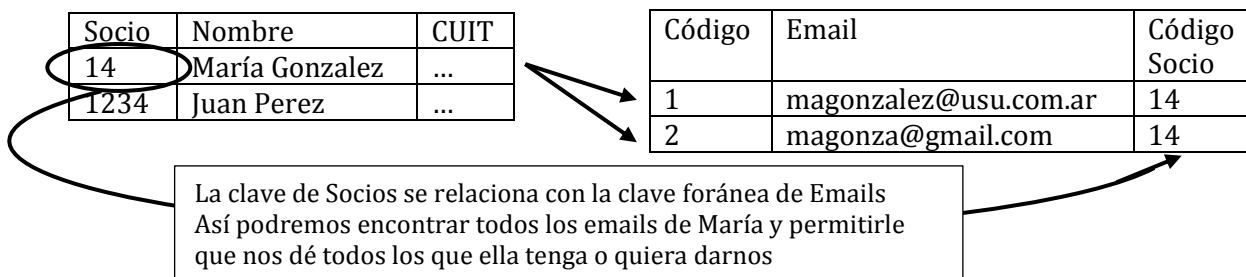
Pero... es necesario que en esta nueva tabla haya además un atributo que se utilizará para relacionar las t-uplas de la tabla con su “padre”, en este caso, el socio.



Seguramente se estará preguntando ¿cómo?

Agregando en la nueva tabla de Emails, el código del socio (o sea que para estos dos números nos quedará asociado el código 14)

Este nuevo atributo se llama **clave foránea**, ya que es exactamente el mismo valor de la clave de dicha tabla. Mostramos como nos quedarían las dos tablas con sus datos de ejemplo:



Queda entonces la nueva tabla:

Email: {Código, Dirección de Mail, Código de Socio (FK Socio)}

Le agregamos FK ya que queremos indicar que es un atributo que proviene de la tabla Socios (la clave foránea por sus siglas en inglés, **foreign key**).

Como este atributo de la tabla préstamos, es a su vez clave foránea de la tabla socios, quiere decir que está apuntando a aquella t-upla de socios cuya clave tiene el mismo valor.

En el ejemplo de la socia María, con código 14, vemos que la clave foránea posee, obviamente, el mismo valor y ambas t-uplas contienen los dos mails que María ha dado como contacto al momento de asociarse.

Es importante entender en qué orden se crean los datos en ambas tablas para darse cuenta que es imposible pensar que las dos t-uplas de la tabla Emails con los datos que le corresponden a María (con código de socio = 14) no pueden existir si primero no existe la t-upla con los datos de María en la tabla Socios.

Por lo tanto, el DBMS tiene una operación que “chequea” que exista el socio con código=14 cuando se está ingresando el mail que le corresponde. Si no existe ese socio, el DBMS no permite la creación de la t-upla en la tabla Emails.

Es decir, no podríamos tener un Código de Socio en esta tabla de Emails, que no existiera previamente en la tabla Socios. Esta característica de las claves foráneas de las bases de datos se denomina **Integridad Referencial**.

Un detalle muy importante que nos queda por corregir en el pasaje a tablas es que si observás la tabla Préstamos, no tenemos manera de saber qué socio realizó el préstamo. La relación entre Préstamos y Socios es 1-N, un socio puede realizar muchos pedidos de préstamos pero el préstamo es únicamente de él.

Para modelar correctamente esta relación, y en adelante todas las relaciones de cardinalidad 1-N, debemos tener una clave foránea del socio en la entidad del lado “Muchos”, o sea en Préstamos.



Por ejemplo, si la misma María Gonzalez en algún momento pidió los libros “cien años de Soledad” y “Sobre héroes y tumbas” tendríamos en la tabla Préstamos:

ID	Fecha	Fecha Devolución	Código de Socio
2	2/11/2014	15/11/2014	14
27	13/04/2017	20/04/2017	18
22	14/10/2016	28/10/2016	14

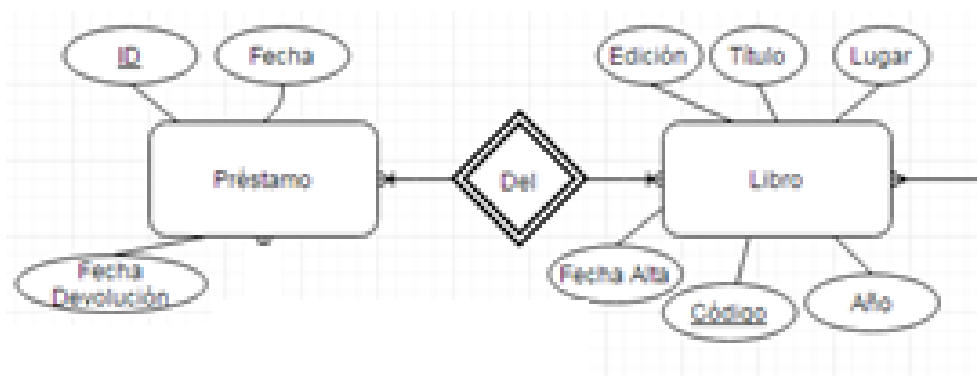
Que sería entonces así:

Préstamo: {ID, Fecha, Fecha Devolución, Código de Socio (FK Socios)}

Además tendríamos las t-uplas correspondientes en la tabla Libros:

Libro	Título	Edición	...
10	Cien Años de Soledad	2012	...
19	Sobre Héroes y Tumbas	2010	...

Y para terminar, veremos cómo se pasa a tablas una relación N-M, tomando por ejemplo la relación Préstamos-Libros, donde vemos que cada libro puede ser prestado muchas veces y en un mismo préstamo se puede prestar más de un libro.



Vimos que María había pedido en el pasado dos libros, uno de Sábato y otro de García Márquez.

Si Jorge (código de socio = 18) solicitó los mismos libros esta semana, y se los otorgaron con el préstamo código= 27 hasta la semana que viene, vamos a ver cómo quedarían las tablas correspondientes con los datos, para que puedas darte cuenta de cómo modelar estos casos.

Necesitamos poder prever todos los casos en que se han prestado los libros mencionados pero debemos tener además especial cuidado con la clave que nos debe permitir poder identificar unívocamente cada caso:

CodPréstamo	CodLibro
2	10
22	19
27	10
27	19

Vemos que tenemos en estas t-uplas las combinaciones de los préstamos de María (el 2 y el 22) y Jorge (el 27) de ambos libros (el 10 y el 19).

Lo importante para poder diferenciar las t-uplas, es que no haya pares repetidos, es decir no podríamos tener otro préstamo 27 con otro libro de código 19, ya que ese par ya existe.

La clave de esta tabla está formada por el par CódLibro-CodPréstamo

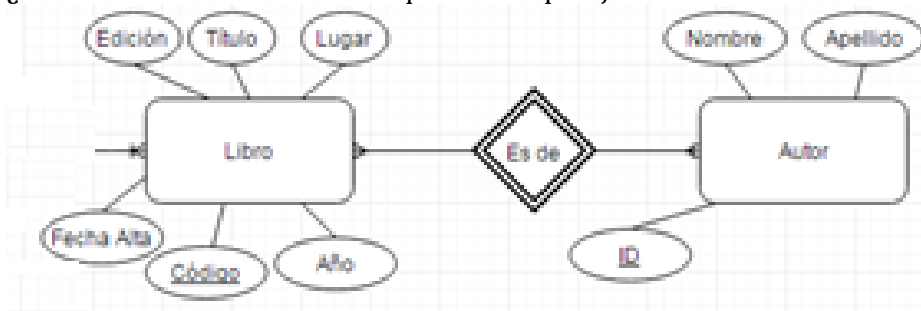
Entonces para resumir:

Cuando tengamos una relación N-M, necesitamos una nueva tabla a la que llamaremos, por convención, con la combinación de los nombres de ambas tablas separados por un guión, y la misma tendrá los siguientes atributos: la clave de dicha tabla será la combinación de las claves de las tablas origen o entidades participantes en la relación, que serán a su vez claves foráneas de las mismas. Es decir, para nuestro ejemplo:

Préstamo_Libro. {CodLibro (FK Libros), CodPréstamo(FK Préstamos)}



¿Se anima ahora a armar cómo quedaría el pasaje a tablas de la relación Autor-Libro?



Usted se habrá imaginado que, de la misma manera que hicimos con Préstamo-Libro debemos agregar una tabla denominada Autor-Libro, de acuerdo a la convención antes mencionada. La tabla tendrá las claves foráneas a sus respectivas tablas padre:

Autor_Libro. {CodLibro (FK Libros), Cod Autor(FK Autor)}

Como regla general entonces tendremos que al finalizar el pasaje a tablas habrá:

- 1 tabla por cada entidad
- 1 tabla por cada atributo multivaluado
- 1 tabla por cada relación N-M

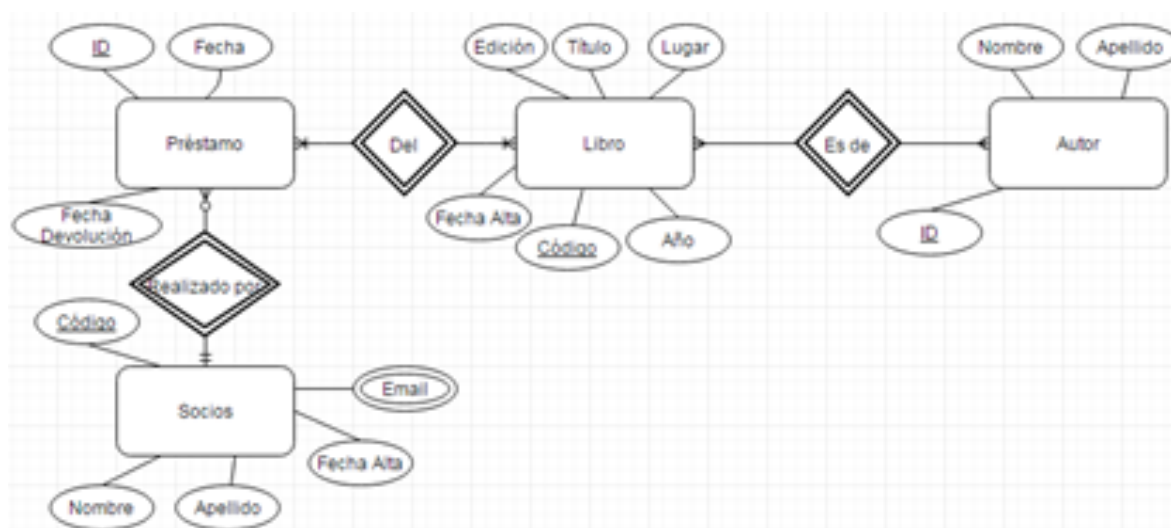
Si tenemos esta regla presente, podremos controlar que el pasaje a tablas sea coherente con el esquema conceptual para luego poder ir armando la base de datos.

El diccionario de datos es entonces el resultado del pasaje a tablas del esquema conceptual donde cada entidad y sus atributos quedan plasmados ya en la base de datos.



¿Puede calcular cuántas tablas tendremos entonces para nuestro diagrama E-R de la biblioteca?

Le mostramos el diagrama:



Seguramente habrá calculado: 4 entidades => 4 tablas.

1 atributo multivaluado (Email)=> 1 tabla

2 relaciones N-M=> 2* (1 tabla)

Total= 4 + 1 + 2 * 1 = 7 Tablas



7. REFINAMIENTO DEL MODELO. DISEÑOS DE CALIDAD.

Vamos a abordar ahora algunos detalles de diseño que se pueden tener en cuenta cuando se arman los modelos de entidad relación. Estos detalles van a contribuir a una mejor calidad en el diseño de la base de datos.

El primero de estos refinamientos es el uso de la **Especialización**.

Utilizaremos un ejemplo para ilustrar un caso donde se puede utilizar este refinamiento:

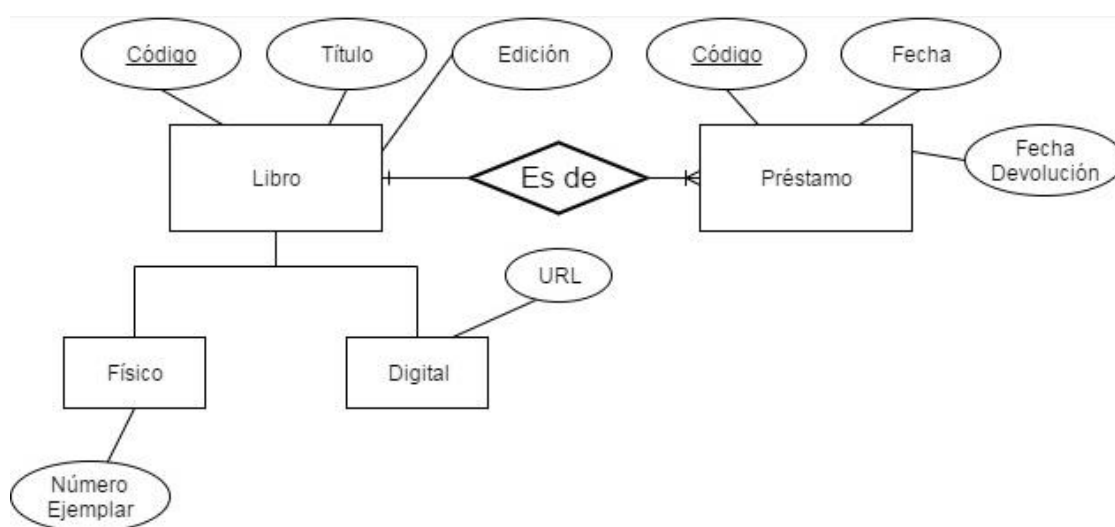
Cuando vimos el ejemplo de los préstamos de libros, no profundizamos en los diferentes tipos de libros que podríamos tener en la biblioteca.

Imaginemos por un momento que los libros fueran digitales o reales (los libros físicos).

Podría pasar que los digitales se puedan leer desde una página de internet con la utilización de un código de préstamo que permitiría su uso durante el tiempo que dure el préstamo, al cabo del cual sería inaccesible.

Un ejemplo así nos permitiría tener un trato distintivo a ambos tipos de libros: los préstamos de los libros digitales necesitan un código de acceso y los libros físicos no.

Veamos cómo se modela esto:



Cuando hay una especialización vamos a ver que siempre hay una entidad, en este caso Libro, que es la superclase de la cual se derivan las subclases especializadas, en este caso Físico y Digital. No tendría sentido que un libro físico tenga una URL o que un libro digital tenga número de ejemplar, por lo que se utiliza la especialización para agregar estos atributos solamente a quien corresponda.



Se preguntará ¿Qué pasa con los atributos que son comunes a todos los libros, como el autor o el título?

¿Y con las claves?

Bueno, estos atributos solo se colocan en la entidad padre o superclase y son “heredados” por las entidades especializadas.

Cuando pasamos a tablas la especialización veremos que tenemos por supuesto 1 tabla para cada entidad. La superclase tendrá la clave y todos los atributos que son comunes a todas las entidades.

Las entidades subclase van a heredar la clave del atributo de la superclase como clave foránea, quedará entonces de esta manera el pasaje a tablas:

Libro: {Código, Título, Edición}

Digital: {ID (FK Libro), URL}

Físico: {ID (FK Libro), Número Ejemplar}

Préstamo: {Código, Fecha, Fecha Devolución, CódLibro (FK Libro)}



ACTIVIDAD 5.

Le pedimos que a continuación realice los ejercicios 8 al 11 de la práctica 1 del campus, dichos ejercicios serán revisados durante el encuentro virtual de la semana. Si lo desea, puede utilizar el foro de la unidad para intercambiar ideas con sus compañeros durante la resolución y de esa manera ir afianzando conceptos vistos hasta aquí.

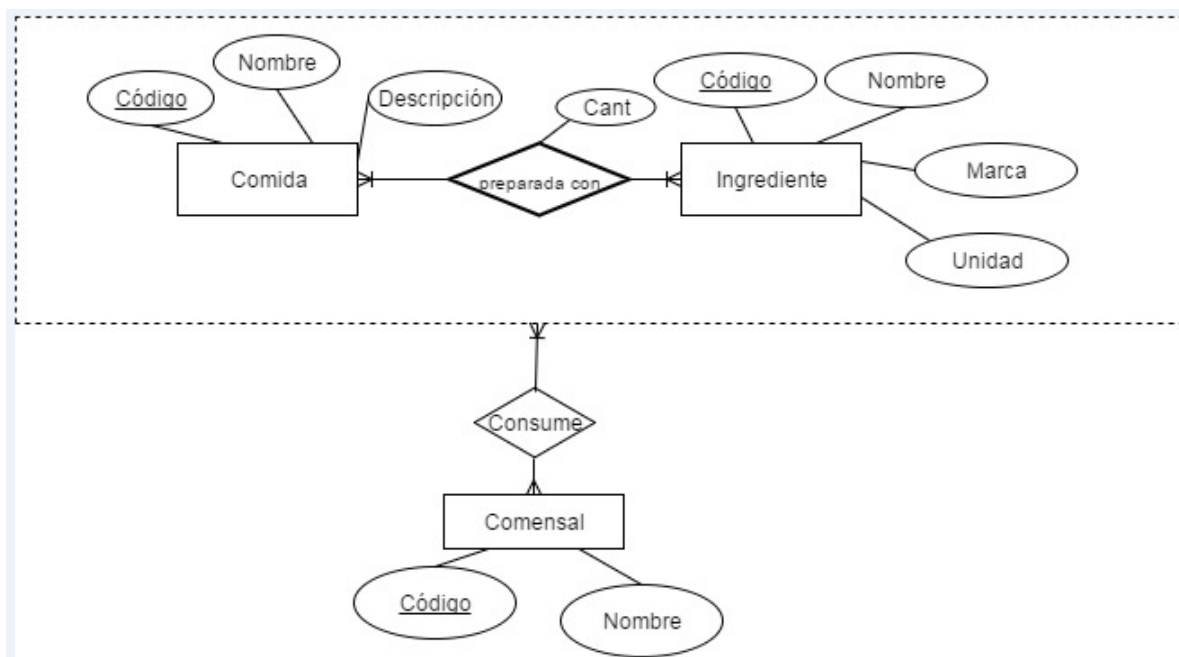
Continuemos...

Un último caso de refinamiento de los modelos sería el que se utiliza para modelar una especie de “menú”.

Por ejemplo, pensemos en el menú de un restaurante: donde tenemos ciertas comidas que están formadas por una cantidad de ingredientes. Sería impensable que una entrada reuniera dulce de leche, aceite y lechuga. ¡Ciertamente no estaría dentro de un menú!

Pero dentro de las entradas podremos tener todas las verduras crudas que se utilizan normalmente en las ensaladas, tales como: lechuga, tomate, zanahorias, cebolla; combinadas con los aderezos: aceite, vinagre, limón, mostaza, mayonesa. Lo mismo para los platos principales y los postres.

Veamos cómo se vería el diseño del modelo del ejemplo de las comidas del menú del restaurante:



Un menú es una combinación de casos apareados entre las t-uplas provenientes de dos entidades o tablas, en los que no son elegibles todas las combinaciones posibles. Este menú se arma primero y termina siendo una tabla.

La existencia del menú es únicamente posible si va a ser utilizado posteriormente, sino, no tiene sentido conformar la tabla subyacente.

Vamos a revisar ahora este diseño para estar seguros de que quedó coherente con todo lo que hablamos:

Vemos que una comida está preparada con ingredientes en diversas cantidades. Este sería el menú.

Una porción individual de la comida tortilla de papas se prepara con 200 grs de papas, 100 cc de aceite, ½ cebolla picada (100 grs) y 2 huevos.

Cierto día entra un cliente al restaurante, que podría ser nuestra socia de la biblioteca, María Gonzalez. Ella se sienta, el mozo se acerca y le entrega el menú para que elija lo que desea almorzar. Ella elige una porción individual de tortilla de papas y de postre el flan con crema y dulce de leche.

El mozo entrega el pedido a cocina y el personal que trabaja allí sabrá perfectamente cómo prepararla y con qué proporción de ingredientes, porque las mismas se encuentran en la tabla que corresponde al menú:

La tabla Ingrediente:

Código	Descripción	Unidad	Marca
1	Huevo	1	GoldenFarm
2	Papas	kg	La Visión
3	Aceite	Litro	Patito

4	Leche	Litro	La Serenísima
5	Cebollas	Kg	La Visión
6	Dulce de Leche	Kg	La Serenísima
7	Crema de Leche	Litro	La Serenísima
8	Azúcar	Kg	Ledesma

La tabla Comida:

Código	Nombre	Descripción
12	Tortilla de Papas	Tortilla de tamaño individual (no es porción)
24	Flan	Individual
33	Flan c/Dulce de Leche	Adicional por porción
29	Flan c/Crema Chantilli	Adicional por porción
39	Flan c/DdL y C Chantilli	Flan con crema chantillí y dulce de leche

Y finalmente la tabla que corresponde al menú del cual estuvimos hablando hasta ahora, nuestra “agregación” que lleva el nombre de ambas tablas por convención o sea Comida-Ingrediente con lo que eligió María:

CodComida	CodIngrediente	Cant
12	1	2
12	2	0,200
12	3	0,150
12	5	0,100
39	1	1
39	4	0,150
39	8	0,075
39	6	0,060
39	7	0,060

En el diccionario de datos, esta tabla por supuesto corresponderá a:
Comida-Ingrediente: {CodComida (FK Comida), CodIngrediente (FK Ingrediente), Cant}

Veamos ahora lo más importante, que es lo que hace que la agregación exista: el hecho de que María haya podido elegir esas combinaciones de comida-Ingrediente pre-existentes (¡ya que no existía una combinación de tortilla con el adicional de dulce de leche, obviamente hubiera sido incomible!). Suponiendo que María es el cliente número 37 del restaurante:

Comensal	Comida	Ingrediente
37	12	1
37	12	2
37	12	3
37	12	5
37	39	1
37	39	4
37	39	8
37	39	6
37	39	7

Esta tabla, que corresponde también a la relación N-M entre el menú (la agregación) y la entidad Comensal, se arma con las tres tablas provenientes de las entidades involucradas, PERO lo más importante es la manera en que se combinan las claves foráneas en esta tabla:

Comensal-Comida-Ingrediente: {CodComensal (FK Comensal), CodComida (FK Comida-Ingrediente), CodIngrediente (FK Comida-Ingrediente)}

Cabe aclarar la importancia de entender que las claves foráneas de Ingrediente y Comida solo corresponden a las combinaciones existentes en la tabla de la agregación. Es decir, NO SON VÁLIDAS todas las combinaciones posibles de las claves foráneas provenientes de las tablas Comida e Ingrediente. ¡Si así fuera, hubiera sido posible para María pedir la tortilla con el adicional de dulce de leche!



ACTIVIDAD 6

Revise los ejercicios de la práctica ya realizados y piense si en alguno de ellos no sería mejor aplicar estos conceptos más avanzados. Comente con sus compañeros en el foro de la unidad. Los ejercicios y las dudas serán revisados con el docente durante el encuentro virtual de la semana.

Formas Normales

Las formas normales brindan una serie de reglas para dar contexto al diseño de bases de datos de manera que aseguren que éste sea lo más correcto posible. Estas reglas ayudan a los diseñadores a realizar su trabajo ya que las deben aplicar cuando arman los DER. Cabe aclarar que las formas normales son de estructura jerárquica, es decir, si un diseño no está en 1ra forma normal entonces no podrá estar tampoco en 2da o 3ra forma normal. Solamente veremos estas tres formas normales ya que son las formas básicas.

Primera Forma Normal

Esta primera forma es muy simple y en realidad lo que indica es que todos los atributos que se manejen y utilicen en una base de datos deben ser simples o atómicos. Es decir, no deben ser divisibles en atributos más pequeños. Ejemplos típicos podrían ser el nombre de un contacto, que debería estar separado en nombre y apellido, dos atributos en vez de uno. Otro ejemplo sería la dirección, que debería en realidad ser calle, número, piso, departamento, localidad, provincia y país, o si utilizamos bases de datos con atributos geográficos, se podría también agregar la latitud y la longitud. De todas maneras, obviamente no es solo un atributo "dirección" con todos esos datos incluídos en el mismo, sino varios atributos. Si la base de datos tiene atributos de este tipo, NO está ni siquiera en primera forma normal.

Segunda Forma Normal

En una tabla, todos los atributos de la misma deben depender DIRECTAMENTE de la clave primaria. Esto quiere decir que no puede haber un atributo totalmente desconectado del resto

de los atributos de la tabla, especialmente de su clave. Un ejemplo sería si en la tabla empleados, cuya clave es el legajo, tuviéramos el departamento donde trabaja (por ejemplo “Compras”) ya sea el nombre o el código del departamento. Esto claramente violaría la 2da forma normal ya que el departamento no depende del legajo. Para solucionar esto, se debe separar el departamento junto con el *legajo* y armar una tabla nueva que se llame por ejemplo Empleados_Departamentos con ambos atributos como clave.

Tercera Forma Normal

Esta forma normal establece que todas las claves no primarias de una relación no deben depender de otros atributos que tampoco son claves primarias. Es decir que no puede haber dependencias transitivas entre los atributos de una tabla (o relación, como se la llama en su forma más general). Por ejemplo, la tabla *Empleado* no podría contener como atributos al *Código del departamento* y al *Nombre del departamento* donde trabaja el empleado ya que la clave de la tabla empleado es *legajo*, el nombre del departamento depende del código del departamento (que no es clave en la tabla empleado). Para solucionar esto, obviamente se debe separar en una tabla aparte, la tabla Departamento, ambos atributos colocando como clave al *Código* y como atributo dependiente de la clave al *Nombre*. Quedan entonces en la tabla Empleado, todos atributos que dependen de la clave directamente y no queda ninguna dependencia transitiva. Con lo cual está en 1ra, 2da y 3ra forma normal.



ACTIVIDAD 7

Le pedimos que a continuación realice la Actividad que le proponemos y la envíe al tutor para su corrección.



Si usted realizó con éxito todas las actividades, habrá logrado comprender las definiciones y conceptos necesarios para trabajar con diagramas E-R de las bases de datos.

¡FELICITACIONES!

Ha logrado alcanzar los objetivos propuestos para esta Unidad y puede pasar al siguiente donde estudiaremos cómo interactuar con los datos



SÍNTESIS DE LA UNIDAD

Hagamos un resumen de todo lo visto en esta Unidad.

Hemos visto los conceptos básicos que se deben conocer antes de comenzar a trabajar con modelado de datos y bases de datos. Usted podrá utilizar estos conocimientos en la Unidad 3 cuando comencemos a usar las sentencias de los lenguajes DML y DDL.

El ESQUEMA CONCEPTUAL de una base de datos está representado por el DIAGRAMA E-R. Este diagrama está compuesto por ENTIDADES, RELACIONES y ATRIBUTOS. Las ENTIDADES representan TABLAS y las TABLAS están formadas por T-uplas.

Las ENTIDADES a su vez están compuestas por los ATRIBUTOS. Los ATRIBUTOS pueden ser de tipo COMPUESTOS, MULTIVALUADOS, COMUNES o DERIVADOS.

La Cardinalidad especifica la cantidad de RELACIONES que puede ser 1-N, N-M o 1-1. Además, las RELACIONES pueden ser UNARIAS, BINARIAS y TERNARIAS,

Veamos el siguiente mapa conceptual que relaciona los conceptos planteados.



GUÍA DE AUTO-EVALUACIÓN

Aquí le ofrecemos la posibilidad de reflexionar sobre su aprendizaje y sobre la forma como está organizando y llevando a cabo su trabajo. Es una herramienta muy importante. Úsela y si tiene comentarios para compartir, póngalos en el Foro de Interacción con los Tutores o si no son consultas y sólo quiere compartir, en el Bar.

Aspecto a evaluar	MB	B	R	M
1. Mi distribución del tiempo de estudio y trabajo.				
2. Mi entrenamiento en técnicas de estudio en esta unidad. (¿Lo hice, aprendí, usé lo que aprendí?)				
3. Nuevos aprendizajes.				
4. Mi participación en los foros.				
5. Mi participación en el Glosario.				
6. Mi manejo del campus y medios técnicos.				
7. Mi contacto con los compañeros. (¿Intercambio ideas, socializo en el bar?)				
8. Mi contacto con los tutores (¿Pregunto, comento, pido aclaraciones?)				

Considere: ¿Qué debe mejorar?