B.D 03 - CONTENIDOS.

MODELO RELACIONAL.

1. Caso práctico.

En **SI Andalucía** continúan trabajando en la aplicación para la compañía de seguros. Ya elaboraron el diagrama E-R, pero el proceso no termina ahí. No es más que una primera aproximación. **María** le cuenta a **Víctor** que a partir del diagrama E-R habrá que aplicar una serie de normas de transformación para obtener el diseño del **Modelo Relacional.** Ese diseño es un nuevo diagrama, al que llamamos esquema relacional, y que representa el modelo lógico de la base de datos. **Víctor** le pregunta qué es lo que muestra ese esquema, y **Carmen**, que también participa en la formación de **Víctor**, le cuenta que el modelo relacional nos indicará de forma clara y esquemática qué tablas tendrá finalmente la base de datos, qué campos o columnas tendrá cada tabla, cómo se relacionan unas tablas con otras, y cuales serán las claves primarias de cada tabla. De esta forma, el diseño de la base de datos estará completo, y pasar esas tablas a un SGBD concreto será una tarea rutinaria.

Víctor recuerda que el modelo Entidad-Relación estaba basado en los conceptos de entidad y relación, que no eran más que aplicaciones de los conceptos matemáticos de teoría de conjuntos, y le pregunta a **María** si el modelo Relacional también está basado en conceptos matemáticos. Ésta le responde que sí, que el modelo relacional se basa en el concepto matemático de relación, que se representa gráficamente mediante una tabla, lo que lo hace muy intuitivo. Es en definitiva un modelo matemático, ideado por E.F. Codd, pero que afortunadamente la mayoría de las operaciones de transformación que permiten pasar del modelo entidad-relación al modelo relacional son bastante "automáticas", y pueden aplicarse incluso sin dominar esos conceptos matemáticos, aunque conocer en qué se basan esas reglas ayuda a aprender a usarlas correctamente con más rapidez.

Carmen le propone a **Víctor** que la invite a cenar, y de esta forma podrá explicarle con más detalle cómo se realiza ese proceso de transformación, y en qué reglas se basa. Naturalmente **Víctor** acepta, porque realmente tiene ganas de aprender y de aplicar lo aprendido al diseño de la base de datos de la compañía de seguros. Y porque cenar en compañía de **Carmen** también le resulta por sí sólo suficientemente estimulante.

2. Introducción.

La **arquitectura relacional** se puede expresar en términos de tres niveles de abstracción:

- Nivel interno,
- Nivel conceptual y
- Nivel de visión.

Veamos cómo se relacionan los distintos componentes de la arquitectura relacional con los distintos niveles de abstracción:

1. **Modelo relacional de datos:**

En el nivel conceptual, el modelo relacional de datos está representado por una colección de relaciones (tablas) almacenadas.

2. **Submodelo de datos:**

En el nivel de visión, los esquemas externos de un sistema relacional se llaman submodelos relacionales de datos. Cada uno consta de una o más **vistas** para describir los datos requeridos por una aplicación dada.

Una vista puede incluir datos de una o más tablas de datos. Cada programa de aplicación está provisto de un buffer ("Área de trabajo de usuario") donde el SGBD

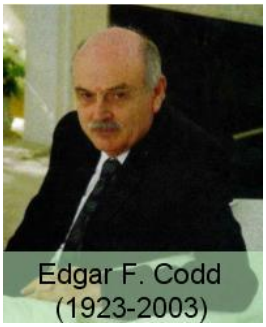
puede depositar los datos recuperados de la base para su procesamiento, o puede guardar temporalmente sus salidas antes de que el SGBD las escriba en la base de datos.

3. **Esquema de almacenamiento:**

   En el nivel interno, cada tabla base se implanta como un archivo almacenado. Para las recuperaciones sobre las claves principal o secundaria se pueden establecer uno o más índices para indexar un archivo almacenado.

¿De qué nos vamos a ocupar en esta unidad?

Pues de conocer el modelo relacional de datos, que es el modelo lógico en el que se basan la mayoría de los Sistemas Gestores de Bases de Datos (SGBD) usados en la actualidad, tales como ORACLE, Access, dBaseIV. MySQL, MS SQL, ...



A finales de los años sesenta **Edgar F. Codd** introdujo la teoría de las relaciones en el campo de las bases de datos. De esta manera los datos se estructuran lógicamente en forma de relaciones.

De manera simple, una relación representa una **tabla**, en que cada fila incluye una colección de valores que describen una entidad del mundo real. Cada fila se denomina **tupla**.

PARA SABER MÁS

Si quieres un poco acerca de la biografía de Edgar Frank Codd, consulta el siguiente enlace:

Breve biografía de Edgar Frank Codd
http://es.wikipedia.org/wiki/E._F._Codd

Los objetivos que buscaba Codd con el modelo relacional van encaminados a obtener:

- **Independencia física**.- Almacenamiento/manipulación. Un cambio físico en la base de datos no afecta a los programas.
- **Independencia lógica**.- Añadir, eliminar o modificar elementos en la BD no debe repercutir en los programas y/o usuarios que acceden a ellos.
- **Flexibilidad**.- Ofrecer al usuario los datos en la forma más adecuada a cada aplicación.
- **Uniformidad**.- Las estructuras lógicas de los datos son tablas. Facilita la concepción y utilización de la BD por parte de los usuarios.
- **Sencillez**.- Por las características anteriores y por los sencillos lenguajes de usuario que utiliza, el modelo relacional es fácil de comprender y utilizar por parte del usuario final.

Es notoria la importancia que Codd concede al tema de la **independencia de la representación lógica de los datos respecto a su almacenamiento interno** (independencia de **ordenación,** independencia de **indexación** e independencia de la **forma de acceso**), tal como expresa Codd desde su primer artículo dedicado al modelo relacional, en cuyo resumen se puede leer:

"... se propone un modelo relacional de datos como una base para proteger a los usuarios de sistemas de datos formateados de los cambios que potencialmente pueden alterar la representación de los datos, causados por el crecimiento del banco de datos y por los cambios en los caminos de acceso".

El modelo relacional representa la segunda generación de los SGBD. En él, todos los datos están estructurados a nivel lógico como tablas formadas por filas y columnas, aunque a nivel físico pueden tener una estructura completamente distinta. Un punto importante del modelo relacional es **la sencillez de su estructura lógica**. Pero detrás de esa simple estructura hay un fundamento teórico matemático importante que hace que el modelo sea seguro y robusto.

AUTOEVALUACIÓN:

La arquitectura relacional se puede expresar en términos de tres niveles de abstracción:

a) Nivel interno, que se corresponde con el esquema de almacenamiento.
b) Nivel conceptual, que se corresponde con el modelo relacional de datos.
c) Nivel de visión, que se corresponde con el submodelo de datos.
d) Todas las respuestas son correctas. (CORRECTA)

PARA SABER MÁS

Si quieres conocer un buen razonamiento de la importancia de la teoría del modelo relacional, consulta el siguiente enlace:

Introducción al modelo relacional
http://www3.uji.es/~mmarques/f47/apun/node44.html

3. El modelo relacional: Estructura.

CASO.

Cuando **Carmen** ha llegado a casa de **Víctor**, se ha encontrado con que la cena estaba en el horno, (lomo horneado con queso y mostaza) y que todavía le faltaba un buen rato para estar lista. En consecuencia, mientras tomaban un refrigerio, **Carmen** ha aprovechado para empezar a estudiar y a explicarle a **Víctor** los conceptos del modelo relacional, en lo relativo a su estructura (relaciones, conjunto de operadores del Álgebra relacional y teoría de dependencias funcionales y Normalización. Sólo es una primera aproximación, pero **Víctor** intuye que detrás de esos conceptos hay mucha más miga… En lo relativo a las Relaciones, la cosa parece más fácil a primera vista:
Constan de una cabecera o intensión y un cuerpo o extensión.

- La cabecera incluye el nombre de la tabla, el nombre de los atributos o columnas de la tabla, y el dominio de cada uno de esos atributos.
- El cuerpo incluye los valores concretos, de forma que cada fila (también llamada tupla) contiene los valores correspondientes a un mismo ente del mundo real.

Víctor piensa que no es nada distinto a cualquier tabla que hayamos podido usar en nuestra vida corriente, en la que no es raro trabajar con tablas. Incluso pensando en términos de ficheros, cada tabla o relación sería un fichero que contiene registros con la misma estructura, cada fila o tupla sería un registro concreto, y cada columna o atributo sería un campo de ese registro.

Después de darle un repaso a algunas de las propiedades de esas tablas, y a la analogía entre la representación física, la representación intuitiva, y el modelo matemático, y a algunos tipos de relaciones, deciden dar por terminada la clase de hoy, y dar buena cuenta de la cena, que vendrá seguida de una larga velada de charla sobre otros temas menos técnicos, pero igual de importantes para ellos.

¿En qué consiste el Modelo Relacional?

El modelo relacional se basa en el concepto matemático de relación, que se representa gráficamente mediante una tabla.

Codd, que era un experto matemático, utilizó terminología matemática para definir el modelo relacional, en concreto la de la teoría de conjuntos y de la lógica de predicados.

La **estructura del Modelo Relacional** está formada por tres partes claramente diferenciadas:

- **La parte estructural**: Utiliza una estructura de datos muy sencilla, **la relación**.
- **La parte manipulativa**: Compuesta por **un conjunto de operadores** que permiten manejar la estructura anterior, **el Álgebra Relacional**. Los estudiaremos en apartados posteriores.
- **La Teoría de las Dependencias Funcionales y de la Normalización:** Compuesta por un conjunto de definiciones que permite estudiar las dependencias entre los atributos de las relaciones y proporciona métodos para un correcto diseño de las bases de datos relacionales. Esta teoría se verá con detenimiento en la siguiente unidad.

4. El modelo relacional: relaciones.

Veamos ahora en qué se traduce exactamente una **relación** dentro de una Base de Datos:

La relación es el elemento básico del modelo relacional y está compuesta por dos partes:

- **Cabecera** (intensión). La cabecera está formada por un conjunto fijo de atributos. Es la parte invariante de la relación y se le denomina también **esquema de la relación**. Está constituida por:
  - El nombre del conjunto (tabla).
  - El nombre de los atributos (columnas de la tabla).
  - Los dominios de los que toman valores.

- **Cuerpo** (extensión). El cuerpo está formado por un conjunto de tuplas. Como consecuencia de su parecido con un elemento matemático, podemos nombrar una relación con el nombre de **tabla**, la cual está compuesta por filas y columnas, donde cada fila (**tupla**) representa un conjunto de valores relacionados entre sí (hechos del mundo real), y las columnas (**atributos**) tienen la función de ayudar a interpretar el significado de los valores que están en cada fila de la tabla.

El número de columnas que tiene una relación recibe el nombre de **grado de la relación**, y el número de filas recibe el nombre de **cardinalidad de la relación**.

Como ejemplo, la figura siguiente representa la relación `PERSONA`.

Cabecera o Intensión

NOMBRE	APELLIDO1	APELLIDO2	DIRECCION	EDAD	TELEFONO_FIJO
Alfonso	Bonillo	Sierra	C/ Isla de Arosa, 7	38	950 363366
Sebastián	López	Ojeda	C/ Islas Vírgenes, 21	37	950 192021
Narciso	Jáimez	Toro	C/ Isla de Java, 11	38	950 111213
Gonzalo	Fernández	Hernández	C/ Ibiza, 10	29	950 353535
Diego	Rodríguez	Gracia	C/ Menorca, 17	36	950 151617
Silvia	Thomas	Barrós	C/ Islas del Caribe, 19	34	950 190405
Miguel Ángel	Pérez	Martínez	C/ Isla Cristina, 12	31	950 343434
Alberto	Domínguez	Vega	C/ Formentera, 9	35	950 123454
Manuel	Rubia	Mateos	C/ Mallorca, 6	36	950 100908

Cuerpo o Extensión

En este ejemplo el esquema de la relación o intensión es:

```
PERSONA (NOMBRE, APELLIDO1, APELLIDO2, DIRECCION, EDAD, TELEFONO_FIJO)
```

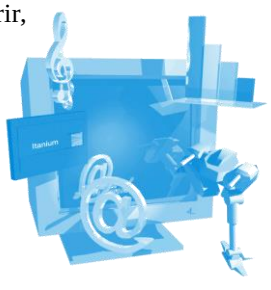
La extensión es el conjunto de 9 tuplas con los datos concretos de personas, el grado de la relación es 6 y la cardinalidad 9.

¿Cómo podemos diferenciar de una manera lógica el término Relación y el término Tabla? De la siguiente manera:

Una relación es una especie abstracta de objeto y una tabla es una representación concreta de ese objeto abstracto.

Estas tablas poseen ciertas propiedades, todas ellas consecuencia inmediata de la relación:

- **No existen tuplas repetidas**: Esta propiedad es consecuencia del hecho de que el cuerpo de la relación es un conjunto matemático (es decir, un conjunto de tuplas) y en matemáticas por definición los conjuntos no incluyen elementos repetidos. Esto se traduce en que dos registros de una misma relación deben diferir, al menos, en el valor de un campo.
- **Las tuplas no están ordenadas**: Esta propiedad sirve para ilustrar la diferencia entre una relación y una tabla, porque las filas de una tabla tienen un orden obvio de arriba hacia abajo, en tanto que las tuplas de una relación carecen de tal orden.



- **Los atributos no están ordenados**: Esta propiedad desprende el hecho de que la cabecera de una relación se define también como conjunto. Las columnas de una tabla tienen un orden evidente de izquierda a derecha, pero los atributos de una relación carecen de tal orden.
- **Todos los valores de los atributos son atómicos**. Es decir, un atributo sólo puede tomar un valor en cada tupla. Otra forma de expresar esta propiedad es que **toda tabla es plana**, es decir, en el cruce de un atributo y una tupla sólo puede haber un valor.

En la siguiente tabla:

PERSONA

NOMBRE	APELLIDO1	APELLIDO2	DIRECCION	EDAD	TELEFONO_FIJO
Alfonso	Bonillo	Sierra	C/ Isla de Arosa, 7	38	950 363366 677586952
Sebastián	López	Ojeda	C/ Islas Vírgenes, 21	37	950 192021
Narciso	Jáimez	Toro	C/ Isla de Java, 11	38	950 111213
Gonzalo	Fernández	Hernández	C/ Ibiza, 10 C/ Calasparra, s/n	29	950 353535
Silvia	Thomas	Barrós	C/ Islas del Caribe, 19	34	950 190405

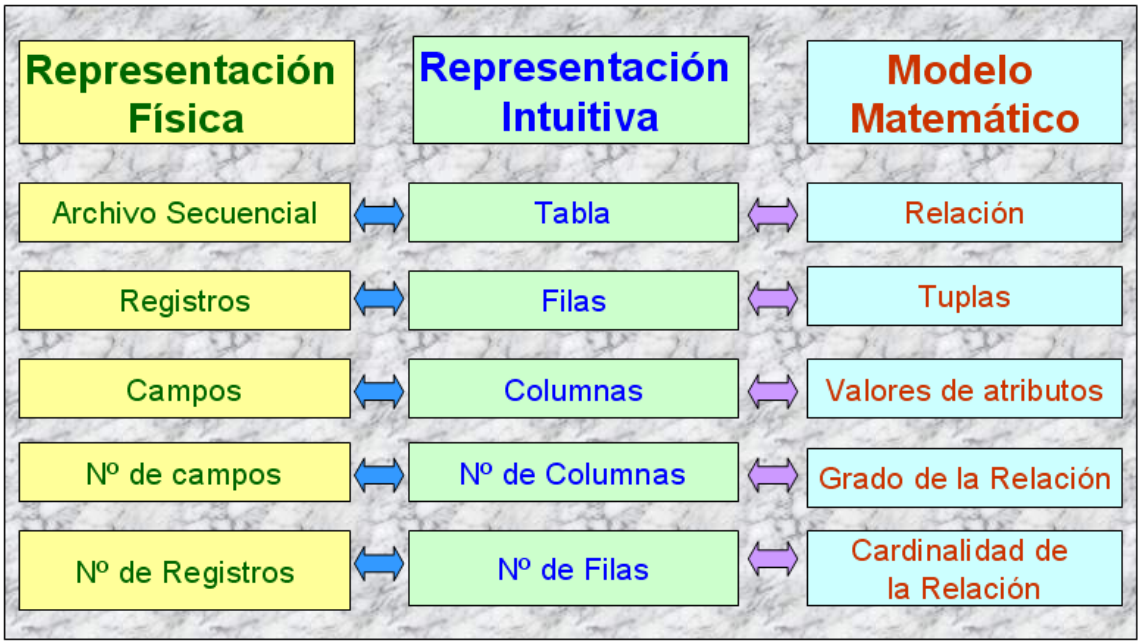
Las tuplas correspondientes a Alfonso Bonillo Sierra y Gonzalo Fernández Hernández no serían válidas por tener dos números de teléfono el primero y dos direcciones el segundo. Eso suele indicarse diciendo que **la tabla no es plana.**

AUTOEVALUACIÓN:

Las relaciones del modelo relacional son la traducción…

a) Únicamente de las entidades del diagrama E/R.
b) Únicamente de las interrelaciones del diagrama E/R.
c) Tanto de las entidades como de las interrelaciones del diagrama E/R.(CORRECTA)
d) Todas las respuestas anteriores son incorrectas

Veamos con una imagen la correspondencia de los distintos modelos que representan una misma realidad:



Intuitivamente una Base de datos no es más que un **conjunto de tablas entrelazadas entre sí a través de los campos con información común.**

En un SGBD relacional pueden existir varios **tipos de relaciones**, aunque no todos manejan todos los tipos.

- **Relaciones Base**. Son relaciones reales que tienen nombre y forman parte directa de la base de datos almacenada (son autónomas).

- **Vistas**. También denominadas relaciones virtuales, son relaciones con nombre y relaciones derivadas. Se representan mediante su definición en términos de otras relaciones con nombre y no poseen datos propios almacenados.
- **Relaciones Instantánea***s*. Son relaciones con nombre y derivadas, pero a diferencia de las vistas, son reales, no virtuales. Están representadas no sólo por su definición en términos de otras relaciones con nombre, sino también por sus propios datos almacenados. Son relaciones sólo de lectura y se refrescan periódicamente.

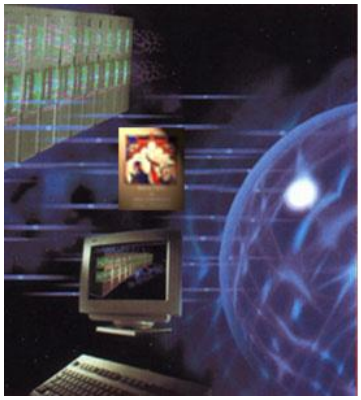
- **Resultados de consultas**. Son las relaciones resultantes de alguna consulta especificada. Pueden o no tener nombre y no persisten en la base de datos.
- **Resultados intermedios**. Son las relaciones que contienen los resultados de las subconsultas. Normalmente no tienen nombre y tampoco persisten en la base de datos.
- **Resultados temporales**. Son relaciones con nombre, similares a las relaciones base o a las instantáneas, pero la diferencia es que se destruyen automáticamente en algún momento apropiado.

AUTOEVALUACIÓN:

Hemos visto que existe una clara correspondencia entre los distintos elementos de los modelos que representan una misma realidad. Elige de entre las que se proponen a continuación la correspondencia que en tu opinión sería correcta.

a) Número de registros → Número de filas → Cardinalidad.
b) Registros → Filas → Tuplas.
c) Campos → Columnas → Valores atribuidos.
d) Todas las respuestas anteriores son correctas. (CORRECTA)
e) Ninguna de las respuestas anteriores es correcta.

5. Tablas, tuplas y columnas.



¿Tenemos clara la diferencia entre tabla y relación?

Sí, sabemos que una tabla no es más que la materialización de un objeto abstracto llamado relación. Por lo tanto, es importante no confundir estas relaciones con las que vimos en el Modelo E/R, y hay que tener cuidado ya que **de ahora en adelante llamaremos indistintamente tablas o relaciones a las relaciones que representan la información que queremos guardar**.

Veamos con más detenimiento la composición de una tabla. Ya sabemos que se compone de una cabecera y un cuerpo, pero ¿de qué está compuesta la cabecera? ¿y el cuerpo?

Si has leído con detenimiento el apartado anterior, no te costará trabajo responder brevemente a las cuestiones anteriores, pero detrás de las tuplas y las columnas hay conceptos que aún no hemos definido.

Las columnas son los campos o atributos que definen la tabla. Cada **campo** está **definido por**:

- **Nombre:** que describe los datos almacenados en él.
- **Dominio:** que indica el tipo de valores que contendrá dicho campo. Es un conjunto finito de valores homogéneos (son todos del mismo tipo) y atómicos (son indivisibles en cuanto a lo que representan se refiere).

Cada **atributo** de una base de datos relacional se define sobre un **dominio**, pudiendo haber varios atributos definidos sobre el mismo dominio. Es evidente que dos atributos de la misma relación no pueden tener el mismo nombre. En unidades posteriores veremos los tipos de valores que se pueden definir para un atributo, aunque como ya sabes los tipos de datos básicos son comunes para la mayoría de los lenguajes de programación, aunque depende del SGBD que utilicemos la definición exacta de los mismos.

La siguiente tabla nos muestra los dominios de los atributos de la relación `OFICINA`.

OFICINA

ATRIBUTO	NOMBRE DEL DOMINIO	DESCRIPCIÓN	DEFINICIÓN
Numero	**NUM_OFICINA**	Posibles valores de número de oficina	3 caracteres; rango 01-099
Calle	**NOM_CALLE**	Nombres de calles de España	25 caracteres
Área	**NOM_AREA**	Nombres de áreas de las poblaciones de España	20 caracteres
Población	**NOM_POBLACION**	Nombres de las poblaciones de España	15 caracteres
teléfono	**NUM_TEL**	Números de teléfono de España	9 caracteres
Fax	**NUM_FAX**	Números de fax de España	9 caracteres

De esta manera la **cabecera de una relación está constituida por un conjunto de "m" campos**, y es expresada de la siguiente manera:

```
relacion (nombre_campo_1:dominio_1, nombre_campo_2:dominio_2,…,
          nombre_campo_m: dominio_m)
```

A esta expresión se le conoce también como **intensión de la relación**.
En el caso de la relación OFICINA su esquema sería:

```
OFICINA (NUM_OFICINA:cadena(3), NOM_CALLE:cadena(25), NOM_AREA:cadena(20),
         NOM_POBLACION:cadena (15), NUM_TEL:cadena(9), NUM_FAX:cadena(9))
```

Y su cuerpo, o extensión sería:

OFICINA

NUM_OFICINA	NOM_CALLE	NOM_AREA	NOM_POBLACION	NUM_TEL	NUM_FAX
005	Islas Vírgenes, 19	Sur	Aguadulce	950 191919	950 363366
002	Islas del Caribe, 21	Norte	Aguadulce	950 212019	950 192021
006	Menorca, 15	Sur	Almería	950 242563	950 111213
007	Isla Cristina, 12	Centro	Murcia	968 256985	968 353535
004	Rejoneador, 16		Ronda	95 2658965	95 2151617

Si consideramos la relación siguiente:

PERSONA

NOMBRE	APELLIDO1	APELLIDO2	DIRECCION	EDAD	TELEFONO_FIJO
Alfonso	Bonillo	Sierra	C/ Isla de Arosa, 7. Almería	38	950 363366
Sebastián	López	Ojeda	C/ Islas Vírgenes, 21. Córdoba.	37	950 192021
Narciso	Jáimez	Toro	C/ Isla de Java, 11. Málaga	38	950 111213
Gonzalo	Fernández	Hernández	C/ Ibiza, 10. Murcia	29	950 353535
Diego	Rodríguez	Gracia	C/ Menorca, 17. Valencia	36	950 151617
Silvia	Thomas	Barrós	C/ Islas del Caribe, 19. Granada	34	950 190405
Miguel Ángel	Pérez	Martínez	C/ Isla Cristina, 12. Murcia	31	950 343434
Alberto	Domínguez	Vega	C/ Formentera, 9. Sevilla	35	950 123454
Manuel	Rubia	Mateos	C/ Mallorca, 6. Jaén	36	950 100908

La cabecera de la relación vendría dada por:

```
PERSONA (NOMBRE: cadena(30), APELLIDO1: cadena(30), APELLIDO2: cadena(30),
DIRECCION: cadena(100), EDAD: número, TELEFONO_FIJO: cadena
(9))
```

Como podemos observar el dominio para el campo `NOMBRE` es una cadena de caracteres de a lo sumo 30 caracteres. Es el mismo dominio que para los campos `APELLIDO1` y `APELLIDO2`. Para el campo `DIRECCION` tenemos un dominio de una cadena de caracteres de a lo sumo 100 caracteres. Para el campo `EDAD` el dominio es un número y por último, para el campo `TELEFONO_FIJO`, el dominio es una cadena de caracteres de longitud 9 caracteres.

De esta manera estamos diciendo que el valor que consideremos para un campo concreto ha de ser siempre del tipo de dato que hemos definido el dominio. Esto es lo que significa que los **valores** de un campo son **homogéneos**.

Lo vemos con un ejemplo:

Para el campo NOMBRE no sería nunca válido el valor "2006", sin embargo sí serían válidas todas las cadenas de caracteres que tuvieran como máximo 30 caracteres.

También hemos definido los valores de los campos como **atómicos**, es decir, indivisibles… y te podrías preguntar ¿las direcciones de la relación PERSONA son atómicas? En principio puede parecer que no, ya que podríamos dividirlas en dirección y población…pero ahí está la cuestión, un valor se considera atómico o no según las restricciones del sistema de información que tengamos. De tal manera que en unos casos nos interesa tener la dirección lo más descompuesta posible para poder tener más posibilidades de combinación de los campos y en otros casos nos interesa más tener toda la dirección en un único campo.

Las tuplas constituyen los registros de la tabla. Un registro es un conjunto de **m** valores, correspondientes a los m campos de la relación. En nuestro ejemplo una de las tuplas de la relación PERSONA es:

(Alfonso, Bonillo, Sierra, C/ Isla de Arosa, 7. Almería, 38, 950 363366)

Es decir, tenemos 9 tuplas o registros en nuestra relación (tabla), como ya hemos visto con anterioridad esta relación tiene cardinalidad 9.

Cabe destacar que **todos los registros de una relación deben tener el mismo número de campos**, aunque alguno esté vacío (se admite el valor NULO).

AUTOEVALUACIÓN:

Las tablas están formadas por…

a) Una cabecera y un cuerpo. (CORRECTA)
b) Una extensión y una intensión. (CORRECTA)
c) Un conjunto de registros con distinto número de campos cada registro.
d) Todas las respuestas anteriores son falsas.

6. Clave primaria y claves ajenas.

CASO.

Al día siguiente en la empresa continúan trabajando sobre la elaboración del modelo relacional a partir del diagrama E-R. A **Víctor** le llama la atención ver que en el modelo Relacional también hay que tener en cuenta los conceptos de llave primaria y llave externa o ajena, tal y como se hacía en el modelo E-R. De hecho, son casi lo mismo, por lo que un rápido repaso le basta para refrescarlos. **María** le llama la atención sobre las propiedades que deben cumplir tanto claves primarias como llaves ajenas, ya que son estas claves las que permiten relacionar la información de unas tablas con la de otras, y permitir consultar toda la información que realmente contiene la base de datos.

Hemos visto que en una relación no hay registros repetidos, lo que quiere decir que se pueden distinguir unos de otros de manera única. La forma de identificarlos es a través de los valores que toman sus atributos.

¿Recuerdas los conceptos de clave, clave primaria y clave ajena de la unidad anterior? Pues en el caso del Modelo Relacional también existen estos conceptos, y se parecen en gran medida a los vistos en el Modelo Entidad Relación.

Veamos los distintos tipos de claves que existen en el Modelo Relacional:

- **Clave Candidata**: Es un conjunto mínimo y no vacío de atributos que identifica unívocamente cada registro de una relación.
- **Clave Primaria**: Es la clave candidata que elige el usuario para identificar los registros de una relación. Se dice que una **clave primaria es compuesta cuando está formada por más de un atributo**. Se

representa en el Modelo Relacional marcándola en negrita y con subrayado continuo.

- **Clave Alternativa**: Es cualquiera de las claves candidatas que no han sido elegidas como clave primaria.
- **Clave Ajena**: Es un conjunto no vacío de atributos de una relación R1 cuyos valores han de coincidir con los valores de la clave primaria de otra relación R2. R1 y R2 no tienen que ser necesariamente distintas (es el caso de las relaciones reflexivas). Se representa en el Modelo Relacional marcándola en negrita y subrayado discontinuo.

Pero seguro que todo esto te queda más claro con algunos ejemplos.

Si consideramos la relación `OFICINA`:

OFICINA

NUM_OFICINA	NOM_CALLE	NOM_AREA	NOM_POBLACION	NUM_TEL	NUM_FAX
005	Islas Vírgenes, 19	Sur	Aguadulce	950 191919	950 363366
002	Islas del Caribe, 21	Norte	Aguadulce	950 212019	950 192021
006	menoría, 15	Sur	Almería	950 242563	950 111213
007	Isla Cristina, 12	Centro	Murcia	968 256985	968 353535
004	Rejoneador, 16		Ronda	95 2658965	95 2151617

Es evidente que el atributo `NUM_OFICINA` es una clave candidata ya que identifica de manera única cada registro de la tabla, y en este caso además es la clave principal por no existir ninguna otra clave candidata. Pero si consideramos la tabla `EMPLEADO`:

EMPLEADO

NUM_EMPLEADO	DNI	OFICINA	TELEF_FIJO	TELEF_MOVIL	NUM_FAX
1234	12345678	Sur	950 235689	655 191919	950 363366
1235	23456789	Norte	950 235698	655 212019	950 192021
1236	13467925	Sur	950 457812	655 242563	950 111213
1237	25836941	Centro	950 456985	655 256985	950 353535
1238	42589637	Centro	950 191921	655 658965	950 151617

Es evidente que tenemos dos claves candidatas, el atributo `NUM_EMPLEADO` y el atributo `DNI`, dado que ambos identifican de manera única a cada registro de la tabla, y por tanto cualquiera de ellos puede ser clave primaria de la relación. En este caso vamos a considerar el atributo `DNI` como clave primaria, con lo que el atributo `NUM_EMPLEADO` pasa a ser una clave alternativa de la tabla.

Si la relación `OFICINA` tuviera la siguiente estructura:

OFICINA

NUM_OFICINA	NOM_CALLE	NOM_AREA	NOM_POBLACION	NUM_TEL	DIRECTOR
005	Islas Vírgenes, 19	Sur	Aguadulce	950 191919	12345678
002	Islas del Caribe, 2	Norte	Aguadulce	950 212019	23456789
006	menoría, 15	Sur	Almería	950 242563	25836941
007	Isla Cristina, 12	Centro	Murcia	968 256985	23456789
004	Rejoneador, 16	Norte	Ronda	95 2658965	25836941

Podemos ver con claridad que el atributo `DIRECTOR` hace referencia al atributo `DNI` de la tabla `EMPLEADO`, que además es la clave primaria de dicha tabla, por lo que `DIRECTOR` es la clave ajena de la tabla `OFICINA`.

Las **claves primarias y ajenas** cumplen una serie de propiedades:

- Las claves ajenas son esenciales en el Modelo Relacional ya que permiten enlazar distintas tablas de la base de datos.
- Una clave ajena y la clave primaria de la tabla referenciada asociada han de estar definidas sobre los mismos dominios.
- Una tabla puede poseer más de una clave ajena, de hecho tendrá una clave ajena por cada tabla referenciada de la cual dependa.
- Una tabla puede no tener ninguna clave ajena.
- Una clave ajena puede relacionar una tabla consigo misma (relaciones reflexivas).

AUTOEVALUACIÓN:

Las claves en el Modelo Relacional…

a) No tienen ninguna relación con las claves del Modelo Entidad-Relación.
b) Pueden ser Primarias y Ajenas entre otras. (CORRECTA)
c) Aparecen en todas las tablas del esquema relacional. (CORRECTA)
d) Todas las tablas deben tener al menos una clave ajena.

7. La integridad en el modelo relacional.

CASO.

Víctor creía que obtener el modelo relacional era más simple, pero está convencido de que todo lo que está aprendiendo le va a facilitar la aplicación práctica. Por ejemplo, el concepto de integridad en el modelo relacional es sumamente importante para conseguir que la información que se almacena en la base de datos sea en todo momento correcta y consistente. Aprende que las reglas de integridad de entidad y de integridad referencial son sumamente importantes para evitar que al manipular los datos la información de la base de datos se deteriore.

María cree conveniente ponerle un **ejemplo**: en el caso de la compañía de seguros, no es posible que el atributo NIF, que es la clave primaria de la tabla CLIENTE, contenga valores nulos. Por tanto el NIF debe marcarse como un campo obligatorio, ya que si introdujéramos un cliente sin NIF, no sería posible distinguir entre sí a dos clientes que se llamaran igual, o más claramente, si no introducimos la matrícula para el vehículo asegurado, será imposible saber de qué vehículo concreto se trata en aquellos casos en que coincida el modelo, el color, y el resto de atributos del vehículo. **Eso es lo que establece la integridad de entidad.**

Por otra parte, si damos de baja un cliente, lo más razonable es que cancelemos todas las pólizas que tuviera contratadas con nosotros, o mirándolo desde otro punto de vista, no debería permitirse

dar de baja a un cliente hasta que no cancele todas las pólizas que tenía contratadas con la compañía. En cualquier caso, lo que no se debe permitir es que tengamos pólizas a nombre de clientes que no figuran en nuestra base de datos. Es absolutamente necesario para la lógica del negocio que todas las referencias a un NIF del cliente dentro de la tabla PÓLIZA se refieran a un NIF que realmente existe en la tabla CLIENTE. **Eso es lo que establece la integridad referencial.**

A **Víctor** le surge la duda de quién tiene que comprobar que se cumplen las reglas de integridad, si el usuario, el programador o el SGBD. Pero **Carmen** le hace ver rápidamente lo fácil que es responder a su pregunta, haciéndole que piense lo siguiente:

- **Carmen:** "¿Te imaginas que cada vez que se hiciera una nueva póliza en la compañía de seguros tuvieran que revisar todos los registros de la base de datos para comprobar si el NIF y la matrícula introducidos existen realmente en las tablas CLIENTE y VEHÍCULO? Y el programador ni siquiera está presente en ese momento"
- **Víctor:** "Por tanto deberá ser el SGBD" .
- **Carmen:** "Lógicamente…"

Ahora que ya conocemos la estructura de datos del Modelo Relacional, ¿cómo podemos saber que nuestra representación es correcta y se corresponde con el universo del discurso?

Existen una serie de reglas, llamadas **reglas de integridad, que al cumplirse nos garantizan que los datos almacenados en nuestra base de datos son correctos, es decir, son una serie de normas que mantienen la corrección semántica de la base de datos.**

En los siguientes subapartados vamos a ir viendo cuáles son esas reglas y las implicaciones que tienen sobre el diseño y el uso de la base de datos.

7.1 Restricciones inherentes.

Como hemos visto con anterioridad, detrás del concepto de relación hay una fuerte base matemática, lo que implica una serie de **restricciones** derivadas de su definición matemática. Éstas eran:

- No hay dos tuplas iguales.
- El orden de las tuplas no es significativo.
- El orden de los atributos (columnas) no es significativo.

- Cada atributo sólo puede tomar un único valor del dominio, es decir, los atributos son atómicos y homogéneos.

Además tenemos que añadir dos reglas que completan el conjunto de restricciones inherentes:

- Regla de integridad de entidad.
- Regla de integridad referencial.

Pero antes tenemos que definir el concepto de **NULO** en el Modelo Relacional.

Decimos que un **atributo es NULO (null) cuando dicho atributo es desconocido**. Un atributo nulo no se representa por el valor cero, ni por la cadena vacía, estos valores tienen significado. Se utiliza el valor null en un atributo cuando dicho atributo no tiene asociado ningún valor o que "su valor es desconocido".

Por ejemplo, se usa el valor null cuando "no se sabe la dirección de una persona", "se desconoce su fecha de nacimiento", etc.

Null puede asignarse a atributos de cualquier tipo, pero no se pueden comparar valores null de atributos de distinto tipo, lo que sí puede considerarse es si un atributo es null o no.

Veamos algunos ejemplos en los que se hace uso del valor null en distintos tipos de atributo en las tablas definidas con anterioridad:

`EMPLEADO (1234, 12345678, "Sur", null, 655191919, 950363366)` en este caso no conocemos el valor del campo "`TELEF_FIJO`".

`OFICINA (05, "Islas Vírgenes, 19", null, null, 950191919, 950363366)` en este caso no conocemos ni el campo "`NOM_AREA`", ni el "`NOM_POBLACION`".

7.1.1 Integridad de entidad.

La regla de integridad de entidad es el mecanismo que garantiza la identificación y unicidad de las tuplas en una relación.

¿Qué dice esta regla?

Ningún atributo que forme parte de la clave primaria de una relación puede tomar un valor nulo.

De esta manera nos aseguramos que no se pueden repetir dos tuplas en una relación. Veamos que esto es así tanto si la clave primaria es un único atributo, o si está compuesta por más de un atributo.

- **La clave primaria** es un único atributo: Consideramos la siguiente relación `ALUMNO`

ALUMNO

DNI	NOMBRE	DIRECCION	CURSO
23456987	S. López Ojeda	Islas Vírgenes, 19	Primero
22335566	S. Thomas Barrós	Islas del Caribe, 2	Primero
34562358	D. Rodríguez Gracia	Menorca, 15	Segundo
23569856	N. Jáimez Toro	Isla Cristina, 12	Tercero
Null	A. Bonilla Sierra	Rejoneador, 16	Tercero
Null	A. Bonillo Sierra	Rejoneador, 16	Tercero

Este registro corresponde a Alfonso Bonillo Sierra

Este registro corresponde a Andrés Bonillo Sierra

En esta relación la clave primaria es el atributo `DNI`, si dicho atributo fuese null y quisiéramos introducir una tupla nueva con los datos de Andrés Bonillo Sierra, hermano de Alfonso Bonillo Sierra, y considerando que los nombres se introducen con el formato: Inicial del nombre y dos apellidos… ¿cómo podríamos distinguir a los dos hermanos Bonillo Sierra? Es a través de la condición de que la clave primaria no puede ser null que podemos distinguir los distintos registros de nuestras tablas.

- La clave primaria está formada por un conjunto de atributos: Consideramos la relación `PEDIDO`

PEDIDO

DNI_CLIENTE	COD_TIENDA	FECHA_PEDIDO	CUANTIA
23456987	005	19/04/05	1904€
22335566	007	08/02/06	806€
34562358	019	21/05/06	2525€
23569856	012	20/06/05	235€

Es evidente que para que los registros de esta relación sean únicos, necesitamos que la clave principal esté formada por los atributos `DNI_CLIENTE`, `COD_TIENDA` y `FECHA_PEDIDO`. Es por esta misma razón, que si alguno de los atributos que componen la clave primaria fuera null y aún así pudiéramos identificar claramente unos registros de otros, es porque ese atributo no debe formar parte de la clave principal. Es más esa no sería la clave principal de la relación porque no cumpliría la condición de minimalidad.

AUTOEVALUACIÓN:

La integridad en el Modelo Relacional…

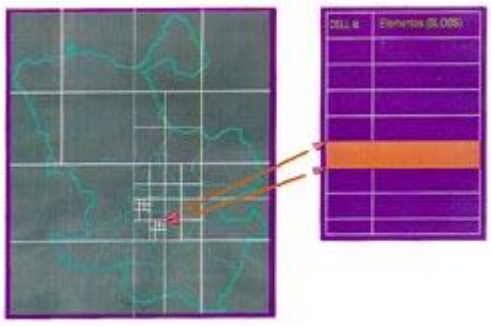
a) Es un concepto de poca importancia en el conjunto del Modelo Relacional.
b) Nos asegura que la representación que hemos obtenido de la realidad que pretendemos modelizar es correcta. (CORRECTA)
c) Se asegura en parte, gracias a las reglas de integridad de entidad y de integridad referencial.(CORRECTA)
d) Nos asegura que una clave primaria puede ser nula.

7.1.2 Integridad referencial.

- ¿Qué ocurriría si introdujésemos en un pedido el código de un proveedor que no existe en la base de datos?
- ¿Qué ocurrirá con los pedidos que contienen un código de proveedor cuando ese proveedor fue dado de baja en la base de datos porque ya no nos suministra?
- ¿Deberíamos borrar todos los pedidos de ese proveedor?
- ¿Deberíamos permitir pedidos con un proveedor que ya no existe?

De estas cuestiones se ocupa la integridad referencial.

La regla de integridad referencial controla los casos de representación de interrelaciones (no tablas) en el universo del discurso, de modo que permite representar vínculos existentes entre tablas, evitando referencias no permitidas. Pero ¿qué dice esta regla?



Los valores de una clave ajena, o bien coinciden con los de la clave primaria a la que referencian o bien son nulos.

Dicho de otra manera**, si una relación R2 (relación que referencia) tiene un atributo que es la clave primaria de otra relación R1 (relación referenciada), todo valor de dicho atributo debe coincidir con un valor de la clave primaria de R1 o ser nulo**. El atributo en cuestión es, por tanto, una clave ajena de la relación R2.

Si consideramos las siguientes relaciones `OFICINA` y `EMPLEADO`

OFICINA

NUM_OFICINA	NOM_CALLE	NOM_AREA	NOM_POBLACION	NUM_TEL	DIRECTOR
005	Islas Vírgenes, 19	Sur	Aguadulce	950 191919	12345678
002	Islas del Caribe, 2	Norte	Aguadulce	950 212019	23456789
006	menoría, 15	Sur	Almería	950 242563	25836941
007	Isla Cristina, 12	Centro	Murcia	968 256985	23456789
004	Rejoneador, 16	Norte	Ronda	95 2658965	25836941

EMPLEADO

NUM_EMPLEADO	DNI	NUM_OFICINA	TELEF_FIJO	NUM_FAX
1234	12345678	005	950 235689	950 363366
1235	23456789	005	950 235698	950 192021
1236	13467925	002	950 457812	950 111213
1237	25836941	007	950 456985	950 353535
1238	42589637	null	950 191921	950 151617

Es evidente que el atributo NUM_OFICINA en la tabla EMPLEADO es clave ajena, que hace referencia a la clave primaria NUM_OFICINA de la tabla OFICINA.

- **Si la clave ajena referencia una tupla**, por ejemplo, NUM_OFICINA de la relación EMPLEADO, si toma un valor no nulo, este valor necesariamente ha de existir en una tupla de la relación OFICINA. No se admite ningún valor que no esté correctamente referenciado, de lo contrario estaríamos asignando un empleado a una oficina que no existe y eso sería una incoherencia.
- **Si la clave ajena no referencia una tupla**, por ejemplo, NUM_OFICINA de un empleado es null, significaría que ese empleado no tiene asignada ninguna oficina en ese momento.

AUTOEVALUACIÓN:

La integridad referencial…

a) Evita que una clave primaria haga referencia a una clave ajena inexistente.
b) Evita que una clave ajena haga referencia a una clave primaria inexistente. (CORRECTA)
c) Permite representar vínculos existentes entre tablas, evitando referencias no permitidas. (CORRECTA)
d) Evita introducir claves ajenas nulas.

7.2 Restricciones semánticas.

Son restricciones que dependen de la **semántica del problema** y sirven para reflejar de la forma más fiel posible el mundo real que se modela. Algunas de estas restricciones se pueden obtener claramente de las definiciones de clave primaria y clave ajena. Vemos algunas de estas restricciones:

- **Clave primaria**. Permite declarar un atributo o un conjunto de atributos como clave primaria de una relación, por lo que sus valores no se podrán repetir ni se admitirán valores nulos.
- **Unicidad.** Mediante la cual se indica que los valores de un conjunto de atributos (uno o más) no pueden repetirse en una relación. Esta restricción permite la definición de claves candidatas.
- **Obligatoriedad (NOT NULL)** de uno o más atributos, indicando de esta manera que ese conjunto de atributos no admite valores nulos.
- **Verificación**, comprueba, en toda operación de actualización de la tabla, si el valor a introducir es correcto y en caso de que no lo sea, rechaza la operación. Una restricción de verificación se define sobre un único atributo y puede o no ser nombrada. Cada relación puede tener tantas restricciones de verificación como atributos tenga.
- **Disparador**, restricción en la que el usuario puede especificar libremente la respuesta del SGBD ante una determinada condición. Los disparadores son reglas de integridad procedimentales, siendo necesario que el usuario escriba el procedimiento que ha de aplicarse en caso de que se cumpla la condición.

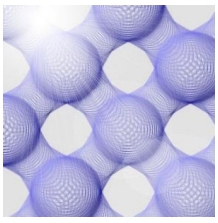
Éstas son algunas de las principales restricciones semánticas que contempla el Modelo Relacional, en unidades posteriores se verán algunas más y se analizarán éstas con más detalle.

AUTOEVALUACIÓN:

Las restricciones semánticas…

a) Permiten repetir valores en la clave primaria.
b) No tienen en cuenta la semántica concreta del problema.
c) Sirven para reflejar de la forma más fiel posible el mundo real que se modela. (CORRECTA)
d) Pueden ser especificadas por el usuario según el problema considerado. (CORRECTA)

7.3 Mantenimiento de la integridad.



Hemos visto la importancia de la integridad en el Modelo Relacional, y es por tanto necesario mantener en todo momento esa integridad. ¿Cómo mantenemos dicha integridad?

Gracias a un principio básico, **la integridad ha de ser mantenida en todo momento por el sistema**.

Veamos cómo se logra tal objetivo considerando las dos reglas de integridad más importantes:

- **Integridad de entidad**: Se debe comprobar que los atributos que forman parte de una clave primaria son no nulos y que el valor de dichos atributos no se repite en los procesos de **inserción** y **actualización**.
- **Integridad referencial**:
  - **En inserción**, comprobar que el valor de la clave ajena es nula o coincide con un valor existente de la clave primaria de la tabla que referencia.
  - **En actualización**, si se actualiza la clave ajena, comprobar las condiciones que definen la clave ajena (ver si el nuevo valor existe como clave primaria en la relación referenciada), y si se actualiza la clave primaria, actualizar en cadena la clave ajena.
  - **En borrado**, si se borra la clave primaria, borrar en cascada o poner a null la clave ajena que hace referencia a dicha clave primaria, es decir, si eliminamos un registro de la tabla R1 referenciada, se eliminan en cascada todos los registros de la tabla R2 que hacen referencia al registro de R1, o se ponen a null las claves ajenas de R2 que referencia a dicha clave primaria.

  Decimos que se produce un borrado en cascada porque al borrar los registros de la tabla R2 puede hacerse posible borrar más registros de una tabla R3 que referenciaran a esos registros de R2, y así sucesivamente a lo largo de toda la cadena de referencias que existieran.

8. Transformación de diagramas E-R al Modelo Relacional.

CASO.

Una vez conocidos los conceptos básicos del modelo Relacional, ha llegado el momento de ver cuáles son la reglas prácticas que se aplican para obtener el esquema relacional a partir del modelo entidad relación.

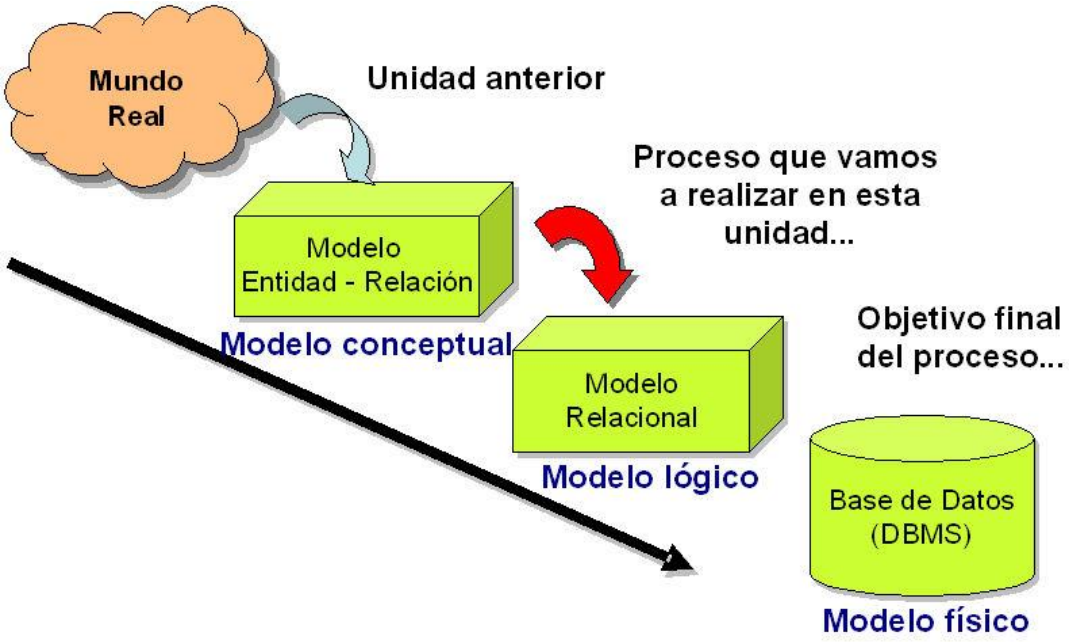
La verdad es que **Víctor** esperaba que fueran extremadamente complicadas, pero sin embargo se ha sorprendido de lo fácil y lógico que es aplicar esas reglas de trasformación del diagrama E-R al modelo relacional. Como **Carmen** le dice, ésa es una de sus grandes ventajas, que piensa siempre de forma lógica, y por eso entiende casi siempre todo a la primera.

Una vez vistas estas reglas, no queda más que aplicarlas, y **Víctor** se pone a elaborar su primer modelo relacional: el de la base de datos de la compañía de seguros.

María le llama la atención sobre algunos detalles que debe tener en cuenta, y al final tuvo que corregirle algún pequeño fallo, pero en general puede decirse que para ser el primer ejercicio de este tipo que hace, **Víctor** lo ha resuelto brillantemente, y en un tiempo bastante reducido.

Si recuerdas de la unidad anterior, el **Modelo Entidad Relación** se basa en una percepción del mundo real basada en **objetos** (entidades) y **relaciones** entre dichos objetos. Se desarrolló para facilitar el proceso de diseño de bases de datos y utiliza diagramas para representar la estructura lógica general de las mismas. Constituye el modelo conceptual del sistema de información que queremos modelizar.

En esta unidad hemos dado un paso más en el proceso, y hemos estudiado el modelo lógico, esto es, el Modelo Relacional.



Por tanto, en esta unidad debemos ver los pasos a seguir para transformar un diagrama E/R (del modelo Entidad-Relación) al esquema relacional (del modelo relacional).

8.1 Principios de transformación.

La transformación de un diagrama E/R al Modelo Relacional está basado en los siguientes principios:

- **Toda entidad se convierte en una relación (tabla).**
- **Toda interrelación (relación) N:M se transforma en una relación (tabla)**
- **Toda interrelación (relación) 1:N se traduce en el fenómeno de "propagación de clave" (se crea una clave ajena)**

Después de analizar los principios anteriores podemos pensar que en el paso del diseño conceptual (diagrama ER) al diseño lógico (esquema relacional) **se pierde información semántica**, ya que tanto las entidades como las relaciones se transforman en tablas, sin que haya una diferencia entre las tablas que provienen de entidades y las que provienen de relaciones. Pero la realidad es **diferente**, ya que gracias a la definición de las restricciones de integridad necesarias, nos aseguramos la conservación de la integridad de la base de datos, es decir que toda la semántica del universo del discurso quede reflejada en el esquema relacional.

8.2 Transformación de las entidades y sus atributos.

Veamos cómo aplicar estos principios generales a cada elemento concreto:

- **Transformación de las Entidades.**

Cada entidad que aparezca en el diagrama E/R se convierte en una tabla.

- **Transformación de Entidades Débiles.**

Las entidades débiles se transforman en una tabla, propagando la clave de la entidad fuerte, que pasa a formar parte de la clave primaria de la entidad débil.

- **Transformación de los Atributos de las entidades**

Cada atributo de una entidad se transforma en una columna en la relación a la que ha dado lugar la entidad. Veamos cómo se definen cada uno de los tipos de atributos:

  - El, o los atributos principales de una entidad (es decir, la clave primaria de la entidad) pasan a ser la clave primaria de la relación. Se debe especificar que no son nulos.
  - El resto de atributos pasan a ser columnas de la tabla, pudiendo tomar valores nulos, a no ser que se indique lo contrario por restricciones de nuestro sistema de información.

- **Transformación de Atributos Compuestos.**

El modelo relacional no permite representar atributos compuestos, por lo que se busca una alternativa, como:

  - Considerar el atributo compuesto como un atributo simple.
  - Eliminar el atributo compuesto y considerar cada uno de sus atributos como atributo simple de la entidad.

AUTOEVALUACIÓN:

Las tres reglas principales para la transformación del diagrama E/R al modelo relacional nos permiten…

a) Convertir cada entidad del diagrama en una tabla. (CORRECTA)
b) Convertir sólo los atributos de la clave principal en campos de una tabla.
c) Eliminar las entidades débiles ya que con la entidad fuerte mantenemos toda la información necesaria.
d) Que los atributos que no forman parte de la clave primaria puedan ser nulos. (CORRECTA)

8.3 Transformación de las relaciones y sus atributos.

- **Transformación de las relaciones (interrelaciones)**

Dependiendo del tipo de relación y de la cardinalidad que tenga, existen diversas maneras de transformarlas:

  - **Relaciones N:M**. Se crea una nueva tabla que incluye los atributos de la propia relación (si los tuviera) y las claves primarias de las dos entidades, que forman la clave primaria de la nueva relación.

  - **Relaciones 1:N**. Estas interrelaciones se pueden transformar de dos maneras diferentes:

    a) **Propagar la clave principal** de la entidad que tiene cardinalidad máxima 1 a la que tiene N, y hacer desaparecer la tabla de la relación como tal.

    b) **Transformarla en una nueva tabla** como si fuese de una relación de tipo N:M, es decir, incluyendo los atributos de la relación y las claves primarias de las dos entidades. Esta acción es recomendable sólo:
      - cuando es posible que aparezcan muchos nulos porque existen pocos elementos relacionados,
      - o cuando se prevé que dicha relación pasará en un futuro a ser de tipo N:M,
      - o cuando la relación tiene atributos propios.

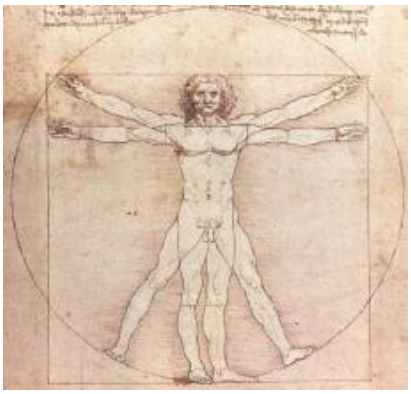
    Cuando la cardinalidad mínima de la entidad uno es igual a uno, se suele utilizar la primera opción. En cambio si la cardinalidad mínima fuera cero, se suele utilizar la segunda opción.

  - **Relaciones 1:1.** Este es un caso particular de cualquiera de los dos casos anteriores, por lo que se podrían aplicar las reglas anteriores. De todas formas es recomendable tener en cuenta las siguientes recomendaciones:

    - Si la **relación es entre entidades con cardinalidades (0,1) y (0,1),** es mejor crear una relación para evitar tener muchos nulos como propagación de alguna de las claves a la otra.

- Si la **relación es entre entidades con cardinalidades (0,1) y (1,1),** es mejor propagar la clave de la entidad (1,1) a la (0,1).

- Si la **relación es entre entidades con cardinalidades (1,1) y (1,1),** la propagación es indiferente, y se hará atendiendo a los criterios de frecuencia de acceso (consulta, modificación, inserción, etc.) a cada una de las tablas en cuestión.



- **Transformación de los atributos de relaciones.**

Si la relación se transforma en una tabla, todos sus atributos pasan a ser columnas de la tabla. En el caso en que alguno de los atributos de la relación sea clave primaria, deberá ser incluido como parte de la clave primaria en dicha tabla.

- **Transformación de relaciones exclusivas.**

Para soportar relaciones exclusivas debemos definir las restricciones pertinentes en cada caso. Por **ejemplo**, en el caso en que exista una exclusividad en la edición de un libro por parte de una editorial o de una universidad, estas dos relaciones se resuelven mediante el mecanismo de propagación de la clave, llevando las claves primarias de editorial y universidad a libro. Ya veremos en unidades posteriores, que tendremos que utilizar entonces la cláusula `CHECK` de SQL para introducir las restricciones pertinentes.

AUTOEVALUACIÓN:

La transformación de las relaciones al modelo relacional…

a) Se realizan sin tener en cuenta la cardinalidad de las relaciones.
b) No siempre da lugar a una nueva tabla, depende del tipo de relación que se trate. (CORRECTA)
c) Sólo en el caso de que la cardinalidad de la relación sea N:M se genera una tabla para dicha relación. (CORRECTA)
d) No tiene en cuenta los atributos de las relaciones.

8.4 Transformación en casos especiales.

- **Transformación de relaciones ternarias (grado 3).**

  - **Relaciones muchos a muchos a muchos:**

Este tipo de relación se transforma en una tabla cuya clave primaria es la concatenación de las claves primarias de las tablas surgidas al transformar las entidades que forman parte de la relación. Junto a estos atributos se incluyen los atributos propios de la relación. Cada uno de los atributos que forman la clave primaria de esta tabla son a la vez claves ajenas respecto a cada una de las tablas donde dicho atributo es clave primaria.

  - **Relaciones muchos a muchos a uno:**

Este tipo de relación se transforma en una tabla cuya clave primaria es la concatenación de las claves primarias de las tablas que corresponden a la cardinalidad **M** surgidas al transformar las entidades que forman parte de la relación. Junto a estos atributos se incluyen los atributos propios de la relación más la clave primaria de la tabla que corresponde a la cardinalidad 1. Cada uno de los atributos que forman la clave primaria de esta tabla (y los atributos añadidos de la relación de cardinalidad 1) son claves ajenas respecto a cada una de las tablas donde dicho atributo es clave primaria.

- **Transformación de la generalización (tipos y subtipos).**

Los tipos y subtipos no son objetos que se puedan representar en el modelo relacional estándar. Existen pues, varias posibilidades para su transformación:

  - Englobar todos los atributos de una entidad y sus subtipos en **una sola tabla**, añadiendo el atributo que permite distinguir los subtipos. También habrá que especificar las restricciones semánticas adecuadas.

- Crear una **tabla para el supertipo**, y tantas tablas como subtipos existan. Ésta es la mejor opción desde el punto de vista semántico, pero es menos eficiente que la opción anterior.

- Crear sólo tablas para los subtipos, añadiendo en cada una de ellas los atributos pertenecientes al supertipo. Habitualmente ésta es la opción más utilizada.

- **Transformación de la agregación.**

  Se trata de transformar primero el nivel más alto de la agregación (aplicando las reglas adecuadas) y trataremos la relación resultante como si fuera "una nueva entidad a relacionar" con el nivel más bajo (aplicando las normas anteriores).

AUTOEVALUACIÓN

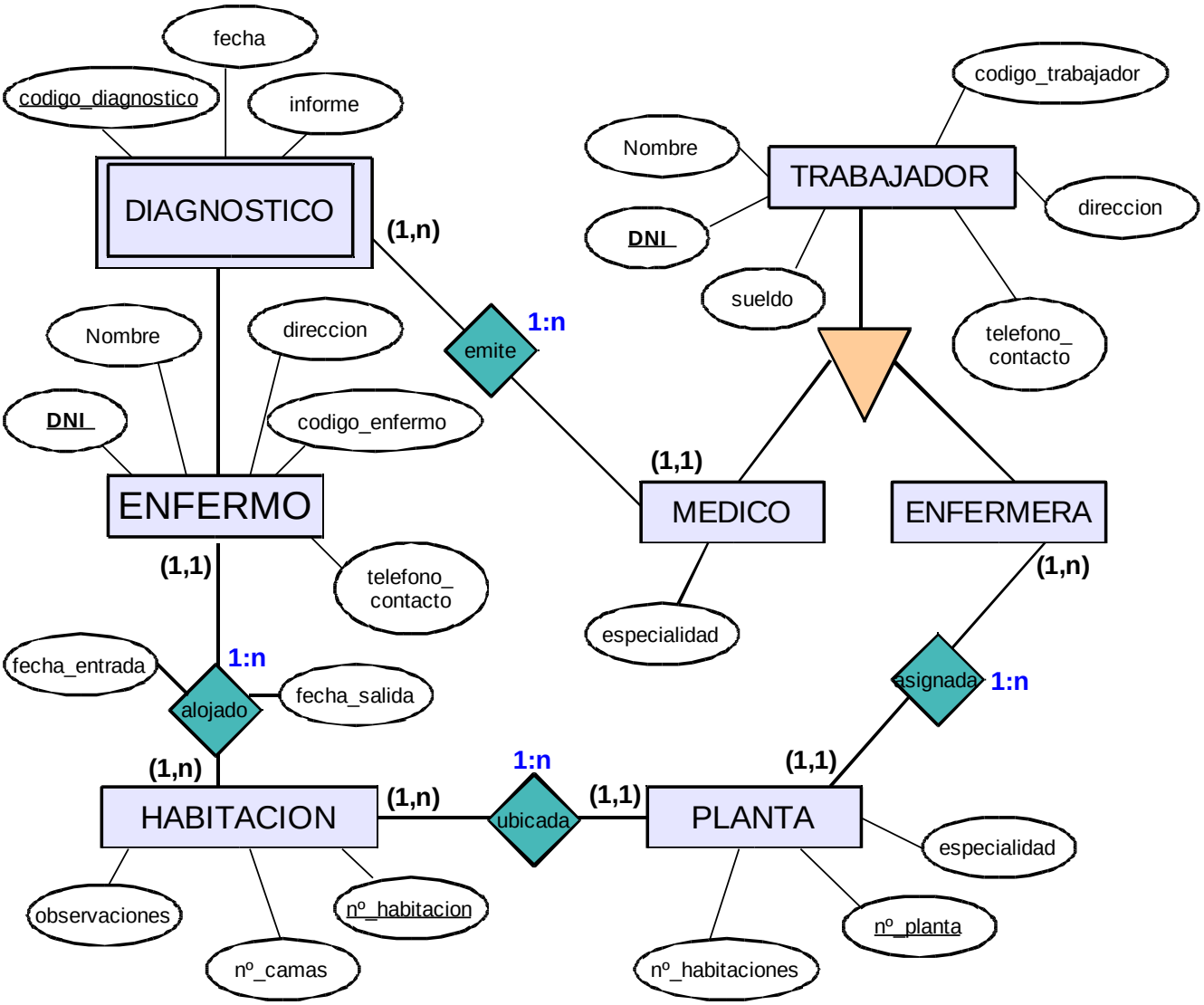
Señala las opciones correctas

a) Al transformar los elementos de una generalización, siempre hay que transformar todas las entidades que participan en la generalización.
b) Al transformar los elementos de una generalización, existen varias posibilidades para su transformación. (CORRECTA)
c) Al transformar una relación ternaria, tenemos que tener en cuenta sus cardinalidades para transformarla. (CORRECTA)
d) Ninguna de las respuestas anteriores es correcta.

9. Ejemplo 1 de transformación: Hospital.

Veamos todo lo anterior con **un ejemplo concreto**.

Para ello vamos a utilizar el ejemplo del **hospital** que hicimos en la unidad anterior. A continuación volvemos a reproducir el diagrama E/R que obtuvimos, y que será el punto de partida sobre el que comenzaremos la tarea de obtener el esquema relacional usando las reglas de transformación que hemos ido viendo en los apartados anteriores:



9.1 Transformando las entidadades.

Empezamos nuestra transformación al Modelo Relacional:

- **Entidades que nos encontramos**: `HABITACION`, `PLANTA` y `ENFERMO` como entidades fuertes, y `DIAGNOSTICO` como entidad débil.

  Las tablas serían:

  ```
  HABITACION( )
  PLANTA( )
  ENFERMO( )
  DIAGNOSTICO( )
  ```

- **Consideramos los atributos para cada entidad**: Debemos recordar que las entidades débiles heredan los atributos de la clave primaria de la entidad fuerte de la que dependen, por lo que la entidad `DIAGNOSTICO` hereda el campo `DNI` de la entidad `ENFERMO` de la que depende.

- **Es habitual modificar el nombre del campo heredado si es necesario para evitar posibles ambigüedades o confusiones en la identificación de los atributos heredados**. Siguiendo esta recomendación notamos el `DNI` de `ENFERMO` como `DNI_enfermo` en la tabla `DIAGNOSTICO` para tener presente en todo momento de dónde procede dicho atributo.

  ```
  HABITACION(n°_habitacion, n°_camas, observaciones)
  PLANTA(n°_planta, n°_habitaciones, especialidad)
  ```

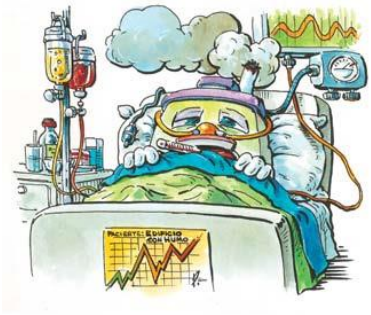
```
ENFERMO(DNI, codigo_enfermo, Nombre, direccion, telefono_contacto)
DIAGNOSTICO(codigo_diagnostico, DNI_enfermo, fecha, informe)
```

- **Seleccionamos las claves primarias**: Sabemos que coinciden con las claves primarias definidas en el diagrama E/R.

  El único caso que merece especial atención es el de la entidad débil DIAGNOSTICO, que al ser débil en existencia y no en identificación, hereda la clave primaria de la entidad fuerte de la que depende, es decir, de ENFERMO, pero dicho atributo no forma parte de la clave primaria de DIAGNOSTICO.

```
HABITACION (n° habitacion, n°_camas, observaciones )
PLANTA ( n° planta, n°_habitaciones, especialidad)
ENFERMO ( DNI, codigo_enfermo, Nombre, direccion, telefono_contacto )
DIAGNOSTICO ( codigo diagnostico, DNI_enfermo, fecha, informe)
```

9.2 Transformamos las relaciones.



- **Ahora pasamos las relaciones a tablas**: Sabemos que, en principio, salvo propagación de claves, cada relación pasa a ser una tabla en el Modelo Relacional, de esta manera tenemos las siguientes tablas:

```
alojado( )
emite( )
ubicada( )
asignada( )
```

- **Añadimos atributos y seleccionamos las claves primarias**: Podemos observar que todas las relaciones son del tipo uno a muchos, con cardinalidad mínima uno de la entidad que participa a uno, por lo que todas las relaciones son candidatas a propagar la clave primaria de la relación que participa con cardinalidad 1 a la relación que participa con cardinalidad N, salvo que la relación tenga atributos propios, como es el caso de "alojado".

```
alojado ( n° habitacion, DNI, fecha_entrada, fecha_salida )
emite (codigo diagnostico,DNI_medico )
ubicada ( n° habitacion, n°_planta)
asignada ( n°_planta, DNI_enfermera)
```

9.3 Transformamos los casos especiales y fusionamos.

Por último tenemos que ocuparnos de la transformación de aquellas entidades que presenten generalización (o especialización, según se mire). En este caso, tanto ENFERMERA como MEDICO pueden considerarse claramente como casos de especialización sobre la entidad TRABAJADOR. (O bien TRABAJADOR como una generalización de MEDICO y ENFERMERA)

- **Pasamos la especialización a tablas**: De entre las distintas posibilidades que tenemos para pasar a tablas una especialización, nos quedamos con la opción de convertir las entidades especializadas en tablas y no la generalización, es decir, pasamos a tablas las entidades MEDICO y ENFERMERA y no TRABAJADOR debido a que existen atributos específicos de una de las entidades y además cada una de ellas establece una relación concreta con otras entidades, lo que le da peso específico para existir de manera independiente.

  Debemos recordar que las entidades que componen la especialización heredan los atributos de la entidad más general y se le añaden los atributos propios si los tuviera.

  Con estos recordatorios podemos considerar:

```
MEDICO(DNI medico, codigo_trabajador, Nombre, sueldo, direccion,
telefono_contacto, especialidad)
ENFERMERA(DNI enfermera, codigo_trabajador, Nombre, sueldo, direccion,
telefono_contacto)
```

- **Fusionamos las tablas siguiendo el criterio de propagación de claves visto con anterioridad, es decir, las claves primarias de las entidades que participan con uno en la relación uno a muchos se propagan a la entidad que participa con muchos.**

HABITACION(**n°_habitacion**, n°_camas, observaciones)
PLANTA(**n°_planta**, n°_habitaciones, especialidad)
ENFERMO(**DNI**, codigo_enfermo, Nombre, direccion, telefono_contacto)
DIAGNOSTICO(**codigo_diagnostico**, DNI_enfermo, fecha, informe, DNI_medico)
MEDICO(**DNI_medico**, codigo_trabajador, Nombre, sueldo, direccion, telefono_contacto, especialidad)
ENFERMERA(**DNI_enfermera**, codigo_trabajador, Nombre, sueldo, direccion, telefono_contacto)
alojado(**n°_habitacion**, DNI, fecha_entrada, fecha_salida)
~~**emite**(**codigo_diagnostico**, DNI_medico)~~
~~**ubicada**(**n°_habitacion**, n°_planta)~~
~~**asignada**(n°_planta, **DNI_enfermera**)~~

Como puedes observar, las tres últimas tablas las hemos **tachado**. Eso se debe a que desaparecen como consecuencia de la propagación de las claves primarias.

9.4 Resultado final.

Hasta este momento no hemos tendido en cuenta las claves ajenas del Modelo Relacional que estamos construyendo. Ahora es el momento de especificar estas claves, y lo hacemos de dos maneras distintas, subrayando con línea discontinua el atributo, y vinculándolo con la clave primaria que referencia. De esta manera nos queda el siguiente esquema relacional:

HABITACION (**nº_habitacion**, nº_camas, observaciones,**nº_planta, DNI_enfermo**)

PLANTA (**nº_planta**, nº_habitaciones, especialidad)

ENFERMO (**DNI**, codigo_enfermo, Nombre, direccion, telefono_contacto)

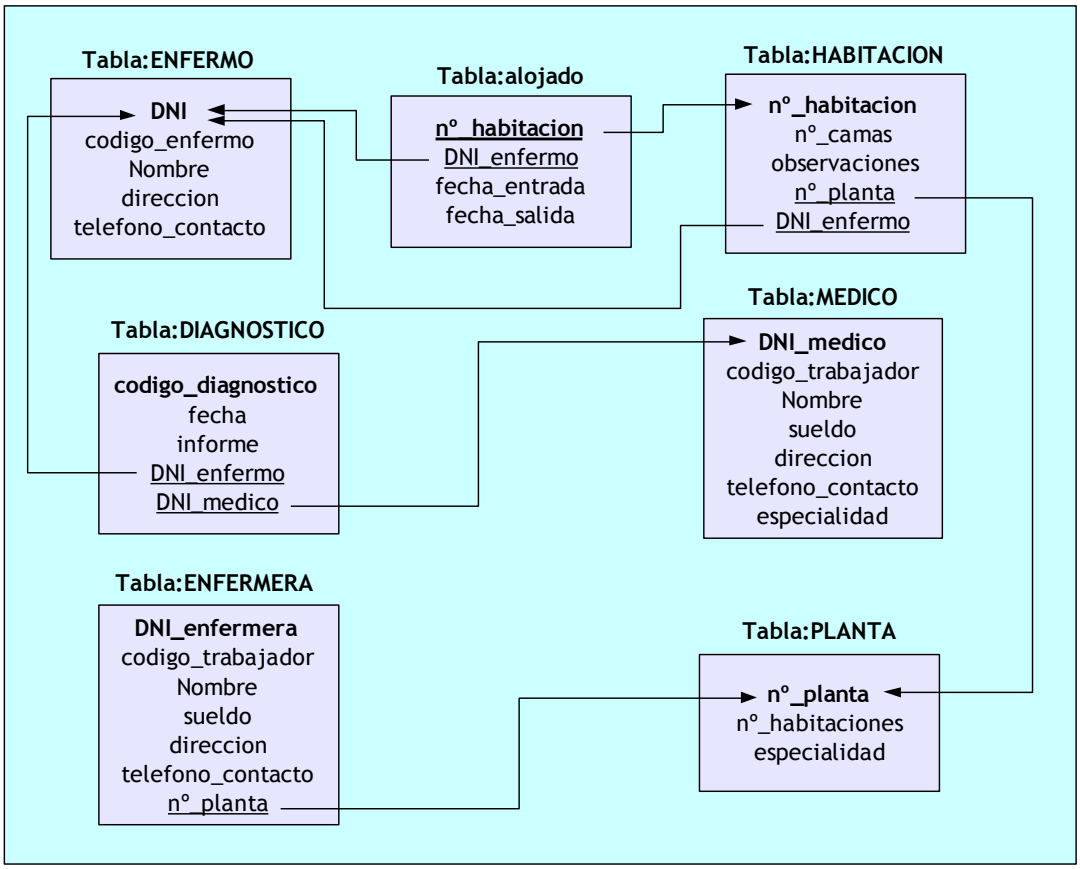
DIAGNOSTICO (**codigo_diagnostico**, DNI_enfermo, fecha, informe,**DNI_medico**)

MEDICO (**DNI_medico**, codigo_trabajador, Nombre, sueldo, direccion, telefono_contacto, especialidad)

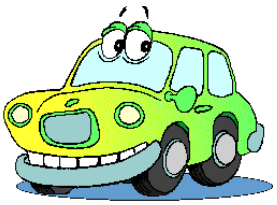
ENFERMERA (**DNI_enfermera**, codigo_trabajador, Nombre, sueldo, direccion, telefono_contacto,nº_planta)

alojado (**nº_habitacion**, **DNI_enfermo**, fecha_entrada, fecha_salida)

Otra manera de expresar el esquema relacional es la siguiente. Como podrás comprobar es más visual e intuitiva que la anterior:



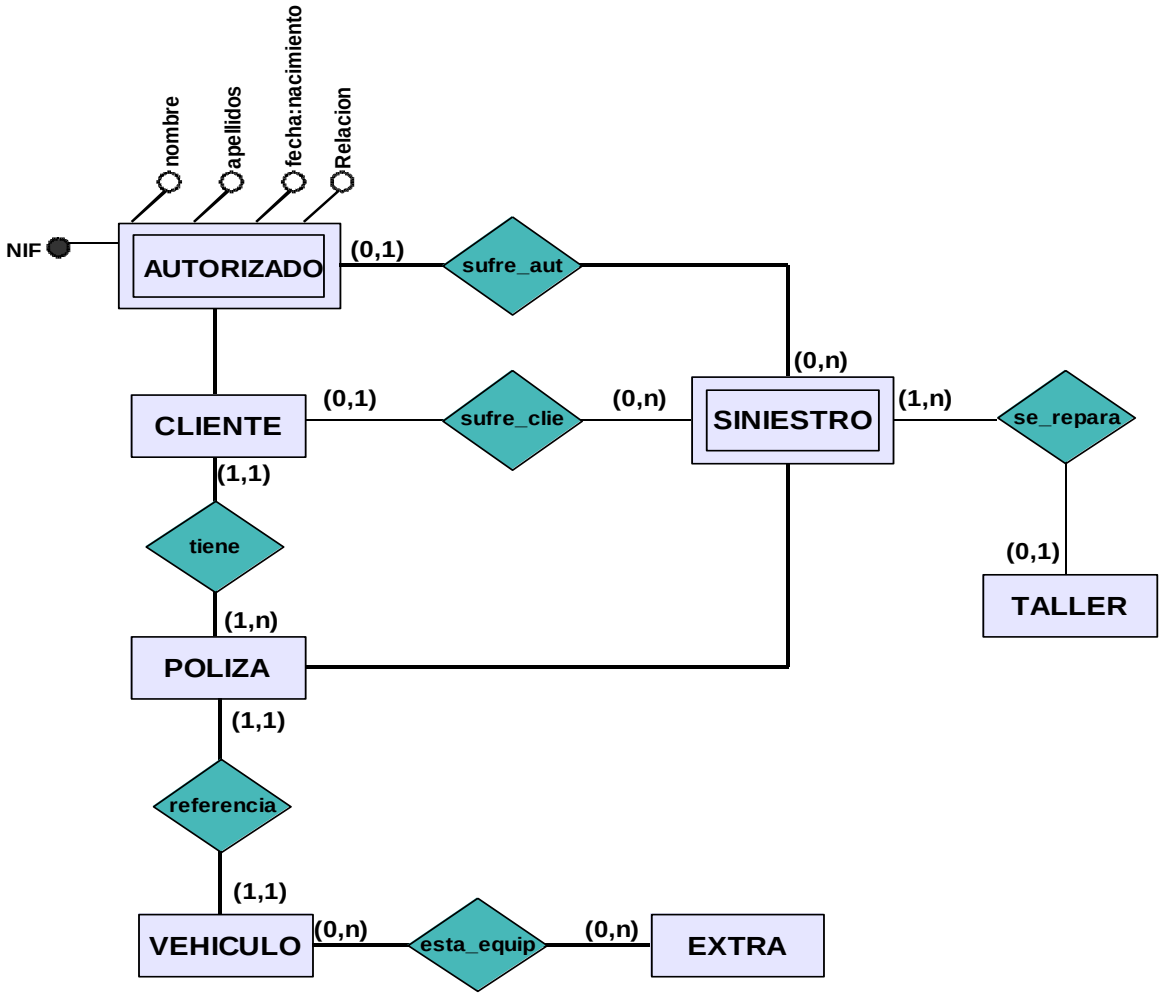
10. Ejemplo 2 de transformación: Seguros.



Igual que para aprender a hacer diagramas E/R la mejor manera era practicar y hacer varios de ellos, para aprender a transformar los diagramas E/R en esquemas relacionales, también hay que practicar y hacer varios ejemplos de transformaciones antes de saber hacerlas bien. Por eso es conveniente que te proporcionemos algún **ejemplo** más resuelto.

En esta ocasión también vamos a utilizar uno de los ejemplos que vimos en la unidad anterior, concretamente el segundo.

Si recuerdas, en aquella ocasión el diagrama E/R lo hicimos sin especificar los atributos de las entidades y relaciones para que el diagrama no resultara demasiado complicado, pero ahora sí que **vamos a tener en cuenta todos los atributos** para crear el esquema relacional, así que tendrás que volver a mirar el enunciado del problema para poder obtener todos los atributos.



10.1 Transformamos las entidades.

Empezamos nuestra transformación al Modelo Relacional con este segundo ejemplo:

Entidades, atributos y claves primarias: Como entidades fuertes tenemos `CLIENTE`, `POLIZA`, `VEHICULO`, `EXTRA` y `TALLER`, mientras que como entidades débiles tememos `AUTORIZADO` y `SINIESTRO`.

Volvemos a recordar que las **entidades débiles heredan los atributos de la clave primaria de la entidad fuerte de la que dependen** y que hay que tener en cuenta si la dependencia es en existencia o en identificación para decidir cual es la clave principal de la tabla que representa la entidad débil.

Te recordamos también que es habitual modificar el nombre del campo heredado si es necesario para evitar posibles ambigüedades o confusiones en la identificación de los atributos heredados.

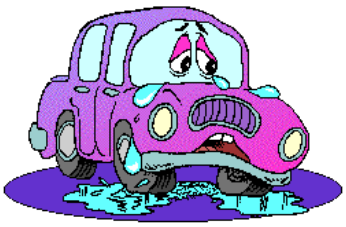
Teniendo en cuenta todas estas consideraciones, las tablas de este ejemplo son las siguientes:

```
CLIENTE(NIF,nombre,apellidos,direccion,telefono,fecha_nacimiento,
        fecha_permiso)
POLIZA(num_poliza,tipo,cobertura,estatus,matricula)
VEHICULO(matricula,num_chasis,marca,modelo,potencia,anno_fabricacion,color)
EXTRA(cod_extra,descripcion)
TALLER(cod_taller,nombre)
AUTORIZADO(NIF,nombre,apellidos,telefono,fecha_nacimiento,relacion,
           NIF_cliente)
SINIESTRO(num_siniestro,num_poliza,fecha_siniestro)
```

Los únicos casos que merecen especial atención en cuanto a la elección de la clave primaria son los de las **entidades débiles** AUTORIZADO y SINIESTRO. La entidad débil SINIESTRO es débil en existencia, por lo que hereda la clave primaria de POLIZA, pero no para formar parte de su clave primaria, sin embargo, la entidad débil AUTORIZADO es débil en identificación, ya que un mismo autorizado, lo puede ser para varios clientes, con lo que además de heredar la clave primaria de CLIENTE, ésta forma parte de su propia clave primaria.

10.2 Transformamos las relaciones.

Pasamos ahora las relaciones a tablas, con los atributos y las claves primarias: Sabemos que, en principio, salvo propagación de claves, cada relación pasa a ser una tabla en el Modelo Relacional, de esta manera tenemos las siguientes tablas:



sufre_aut (**num_siniestro, NIF_cliente, NIF_autorizado**)
sufre_cliente (**num_siniestro, NIF_cliente**)
tiene (NIF_cliente, **num_poliza**)
referencia (num_poliza, **matricula**)
esta_equip (**cod_extra, matricula**)
se_repara (**num_siniestro, cod_taller,** fecha, importe)

La relación **referencia**, al ser de cardinalidad uno a uno, puede tomar como clave primaria cualquiera de los dos atributos.
Para decidir cuáles son las claves primarias de estas tablas hemos observado las cardinalidades de cada relación. Sabemos que las relaciones sufre_aut, sufre_cliente y se_repara son del tipo uno a muchos, con cardinalidad mínima cero de la entidad que participa a uno, por lo que todas las relaciones se tratan como si fueran del tipo muchos a muchos, y por consiguiente heredan las claves primarias de las entidades que relacionan y además dichos atributos constituyen la clave primaria de la relación.

Como podemos ver en el diagrama E/R, la relación esta_equip es de muchos a muchos, por lo que tiene como clave primaria la unión de las claves primarias de las entidades que relaciona.

10.3 Fusionamos las tablas.

Fusionamos las tablas siguiendo el **criterio de propagación de claves** visto con anterioridad, es decir:

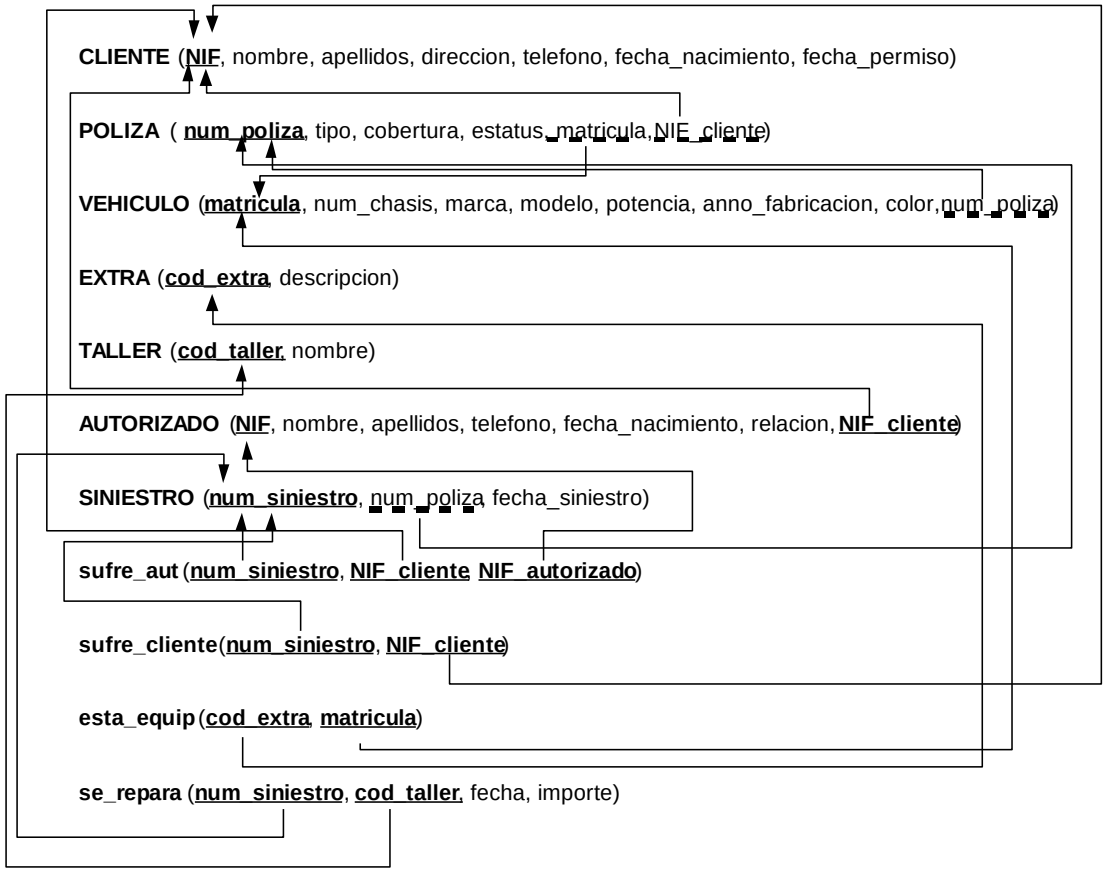
- **Las claves primarias de las entidades que participan con uno en la relación uno a muchos y tienen cardinalidad mínima uno, se propagan a la entidad que participa con muchos.**
- **Además también tenemos en cuenta que las relaciones uno a uno propagan también la clave primaria de una de las entidades que relaciona a la otra.**

CLIENTE(**NIF**,nombre,apellidos,direccion,telefono,
fecha_nacimiento,fecha_permiso)
POLIZA(**num_poliza**,tipo,cobertura,estatus,matricula,
NIF_cliente)
VEHICULO(**matricula**,num_chasis,marca,modelo,potencia,
anno_fabricacion,color,num_poliza)
EXTRA(**cod_extra**,descripcion)
TALLER(**cod_taller**,nombre)
AUTORIZADO(**NIF**,nombre,apellidos,telefono,fecha_nacimiento,relacion,
NIF_cliente)
SINIESTRO(**num_siniestro**,num_poliza,fecha_siniestro)
sufre_aut(**num_siniestro**,**NIF_cliente**,**NIF_autorizado**)
sufre_cliente(**num_siniestro**,**NIF_cliente**)
~~**tiene**(NIF_cliente,**num_poliza**)~~
~~**referencia**(num_poliza,**matricula**)~~
esta_equip(**cod_extra**,**matricula**)
se_repara(**num_siniestro**,**cod_taller**,fecha,importe)

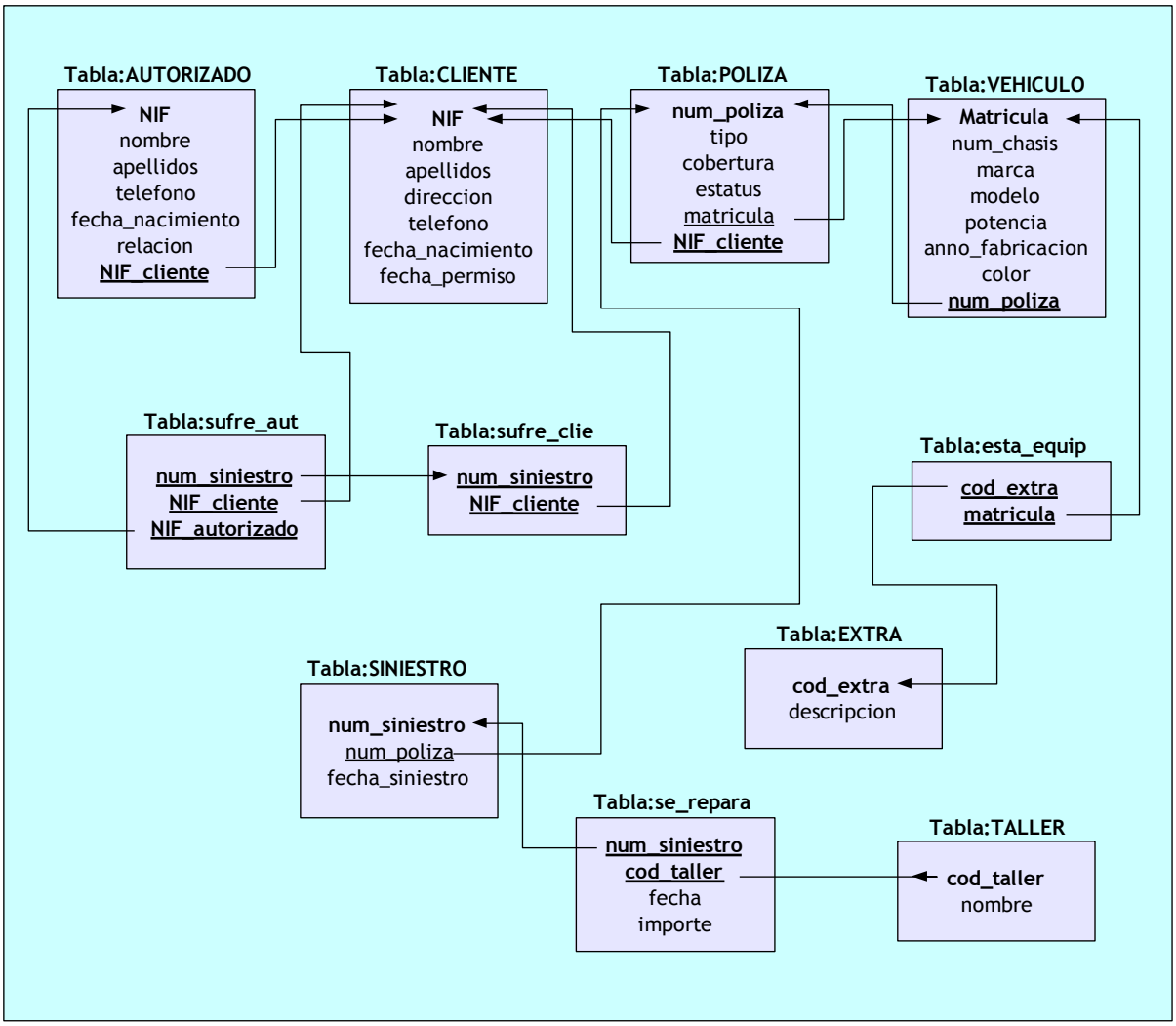
De nuevo, las dos tablas que aparecen **tachadas** porque desaparecen como consecuencia de la propagación de las claves primarias.

10.4 Resultado final.

De nuevo no hemos tenido en cuenta las claves ajenas del Modelo Relacional que estamos construyendo. Ahora es el momento de **especificar** estas claves, y lo hacemos de dos maneras distintas, subrayando con línea discontinua el atributo, y vinculándolo con la clave primaria que referencia, como hemos hecho en el ejemplo anterior. De esta manera nos queda el siguiente esquema relacional:

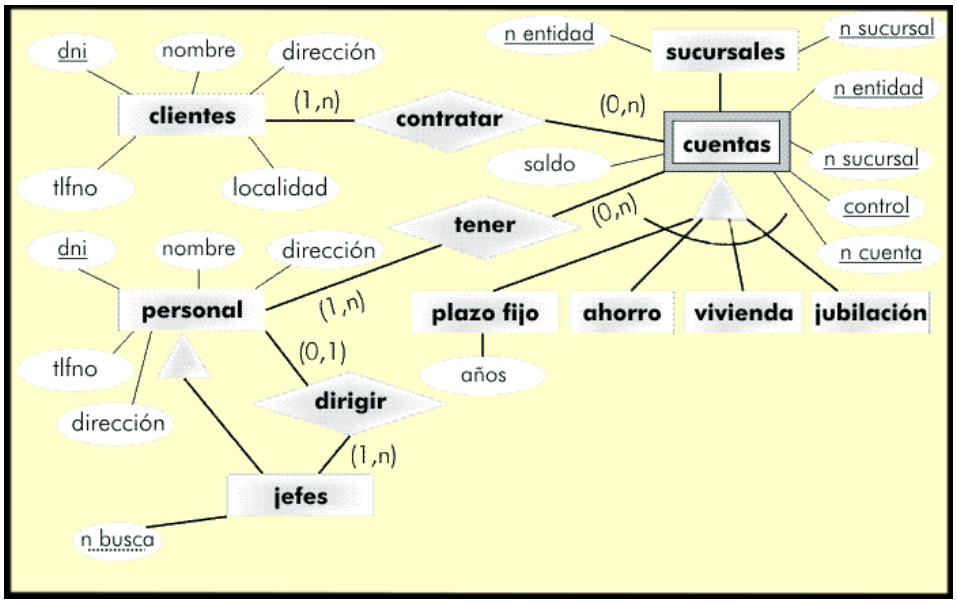


Y ahora vemos la otra manera de representar el Modelo Relacional:



11. Otros ejemplos: un banco.

Te presentamos el diagrama E/R correspondiente a un banco. Intenta hacer la transformación para obtener el esquema relacional.

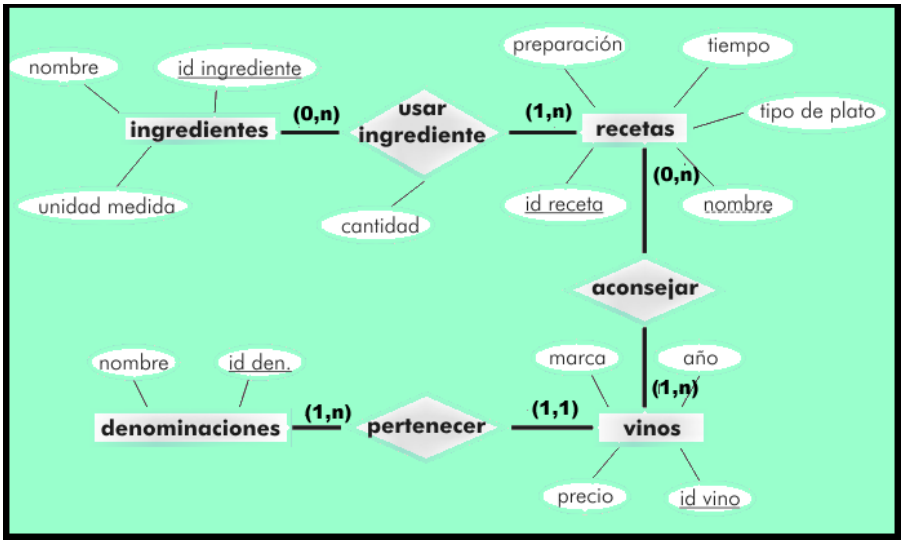


Después de haberlo completado por ti mismo, puedes comprobar la solución en el siguiente enlace.

Insertar aquí 4GL03_Recurso01.gif

12. Otros ejemplos: recetas.

En esta ocasión te proporcionamos el diagrama E/R para un sistema de recetas de comida. ¿Sabrías transformarlo al modelo relacional?

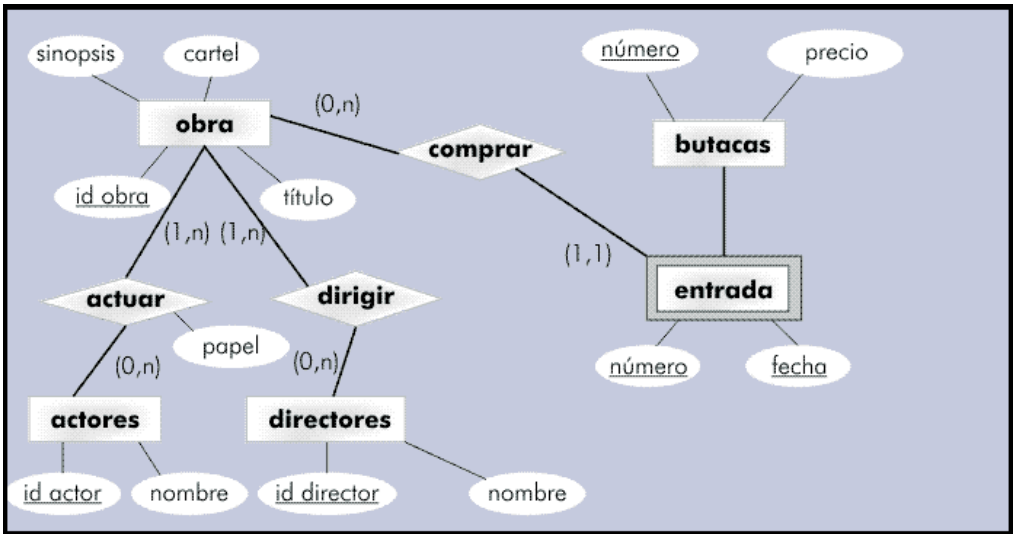


Una vez más, te recomendamos que intentes resolverlo por ti mismo, y sólo entonces, que compruebes la solución en el siguiente enlace:

Insertar aquí 4GL03_Recurso02.gif

13. Otros ejemplos: teatro.

Y por último te mostramos el diagrama E/R que gestiona un teatro. ¿Te atreves a un último ejercicio para obtener el esquema relacional?



Y la solución para contrastarla con la que tú has encontrado en el siguiente enlace:

Insertar aquí 4GL03_Recurso03.gif

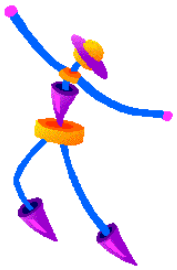
PARA SABER MÁS

En la siguiente página web encontrarás algunos ejemplos más en los que te proporciona un breve enunciado, el diagrama Entidad-Relación y su transformación en el correspondiente esquema relacional. Ya sabes que este tipo de materia se asimila mejor ejercitando los conocimientos adquiridos, y las destrezas se adquieren entrenándose. No hay mejor manera de

convertirse en un experto. Por eso te aconsejamos que en todos ellos primero intentes resolver el problema por ti mismo, y sólo entonces consultes la solución propuesta.

Ejercicios Modelo Relacional resueltos
http://www.jorgesanchez.net/bd/ejercicios.html

14. Las 12 reglas de Codd.



Hasta ahora hemos visto toda la teoría necesaria para poder obtener el esquema relacional, desde un punto de vista práctico, pero detrás de todo esto hay una teoría aún más extensa. En este apartado vamos a dar una visión general de esta teoría. Por cierto, ¿te imaginas quién puede ser el responsable de toda esta teoría?

En un artículo de 1985 publicado en Computerworld, el Dr. Codd presentó doce reglas que una base de datos debe obedecer para que sea considerada relacional. **Las doce reglas de Codd se han convertido en la definición teórica de una base de datos relacional**. Éstas se derivan del trabajo teórico de Codd sobre el modelo relacional y representan realmente más **un objetivo ideal** que una definición de una base de datos relacional. Dichas doce reglas son:

1. **Regla de información.** Toda la información de una base de datos relacional está representada explícitamente a nivel lógico y exactamente de un modo: Mediante valores en tablas.

2. **Regla de acceso garantizado.** Todos y cada uno de los datos de una base de datos relacional se garantiza que sean lógicamente accesibles recurriendo a una combinación de nombre de tabla, valor de clave primaria y nombre de columna, como verás más adelante con SQL**.**

3. **Tratamiento sistemático de valores nulo.** Los valores nulos (distinto de la cadena de caracteres vacía o de una cadena de caracteres en blanco y distinta del cero o de cualquier otro número) se soportan en los SGBD completamente relacionales para representar la falta de información y la información inaplicable de un modo sistemático e independiente del tipo de datos.

4. **Catálogo en línea dinámico basado en el modelo relacional.** La descripción de la base de datos se representa a nivel lógico del mismo modo que los datos ordinarios, de modo que los usuarios autorizados puedan aplicar a su consulta el mismo lenguaje relacional que aplican a los datos regulares.

5. **Regla de sublenguaje completo de datos.** Un sistema relacional puede soportar varios lenguajes y varios modos de uso terminal (por ejemplo, el modo de rellenar con espacios en blanco). Sin embargo, debe haber al menos un lenguaje cuyas sentencias sean expresables mediante alguna sintaxis bien definida, como cadenas de caracteres, y que sea completa en cuanto al soporte de todos los puntos siguientes:
   - Definición de datos.
   - Definición de vista.
   - Manipulación de datos (interactiva y por programa).
   - Restricciones de integridad.
   - Autorización.
   - Fronteras de transacciones (comienzo, cumplimiento y vuelta atrás).

6. **Regla de actualización de vista.** Todas las vistas que sean teóricamente actualizables son también actualizables por el sistema.

7. **Inserción, actualización y supresión de alto nivel.** La capacidad de manejar una relación de base de datos o una relación derivada como un único operando se aplica, no solamente a la recuperación de datos, sino también a la inserción, actualización y supresión de los datos.

8. **Independencia física de los datos.** Los programas de aplicación y las actividades terminales permanecen lógicamente inalterados cualquiera que sean los cambios efectuados, ya sea a las representaciones de almacenamiento o a los métodos de acceso.

9. **Independencia lógica de los datos.** Los programas de aplicación y las actividades terminales permanecen lógicamente inalterados cuando se efectúen, sobre las tablas de

base, cambios preservadores de la información de cualquier tipo que teóricamente permita alteraciones.

10. **Independencia de integridad.** Las restricciones de integridad específicas para una base de datos relacional particular deben ser definibles en el sublenguaje de datos relacional y almacenables en el catálogo, no en los programas de aplicación.

11. **Independencia de distribución**. Un SGBD relacional tiene independencia de distribución.

12. **Regla de no subversión**. Si un sistema relacional tiene un lenguaje de bajo nivel (un solo registro a la vez), ese bajo nivel no puede ser utilizado para subvertir o suprimir las reglas de integridad y las restricciones expresadas en el lenguaje relacional de nivel superior (múltiples registros a la vez).

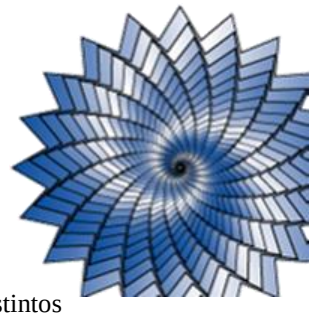
Ningún SGBD relacional actualmente disponible satisface totalmente las doce reglas de Codd.

PARA SABER MÁS

Si quieres profundizar en el significado de las 12 reglas de Codd, entra en el siguiente enlace:

Las doce reglas de Codd
http://es.wikipedia.org/wiki/RDBMS

15. Álgebra relacional.



El Álgebra Relacional se define como la parte dinámica del Modelo Relacional. Se basa en una serie de operadores matemáticos que Codd utilizó para poder realizar diferentes consultas a las tablas definidas en el modelo. Estos operadores establecen la forma de obtener la información que nos interesa de entre toda la información disponible en la base de datos. En este apartado vamos a ver una breve introducción a esta teoría.

Los principales operadores utilizados en el Álgebra Relacional se dividen en Operadores Primitivos y Operadores Derivados.

Vamos a considerar las siguientes tablas para ver con ejemplos cómo funcionan los distintos operadores:

PELICULA

ID_Pelicula	Nombre	Año
1	La guerra de las galaxias	1977
2	El señor de los anillos 1	2001
3	Mar Adentro	2004
4	El viaje de Chihiro	2001

ACTOR

ID_Actor	Nombre	Apellido
1	Mark	Hamill
2	Christopher	Lee
3	Javier	Bardem
4	Hugo	Weaving

DIRECTOR

ID_Director	Nombre	Apellido
1	George	Lucas
2	Meter	Jackson
3	Javier	Bardem
4	Hayao	Miyazaki
5	Alejandro	Amenábar

ESTUDIO

ID_Estudio	Nombre
1	Ghibli
2	New Line Cinema

3	Lucasfilms
4	Sogecine

15.1 Operadores primitivos.

Los operadores primitivos son los operadores tomados de las matemáticas para desarrollar todo el Álgebra Relacional. Existe una clasificación de estos operadores en función del número de tablas implicadas en la operación:

- **OPERADORES UNARIOS.** Son los operadores que sólo involucran una tabla.
    - **PROYECCIÓN. (π)** La proyección de una relación sobre un conjunto de sus atributos es otra relación definida sobre ellos, eliminando las tuplas duplicadas que hubieran podido resultar. Devuelve columnas completas

$\pi_{Año}(Película) =$
{<1977>,<2001>,<2004>,<2001>}

$\pi_{ID\_Película,Año}(Película) =$
{<1,1977>,<2,2001>,<3,2004>,<4,2001>}

$\pi_{Nombre}(Actor) =$
{<Mark>,<Cristopher>,<Javier>,<Hugo>}

    - **SELECCIÓN. (σ)** La selección de una relación mediante una expresión lógica (predicado de selección) da como resultado una relación formada por el conjunto de tuplas que satisfacen dicha expresión. Devuelve filas completas.

$\sigma_{Apellido=Lee}(Actor) =$
{<2,Cristopher,Lee>}

$\sigma_{Año>2000}(Película) =$
{<2,La comunidad del anillo,2001>,
<4,El viaje de Chihiro,2001>}

- **OPERADORES BINARIOS.** Son los operadores que involucran dos tablas.

    - **UNIÓN. (U)** La unión de dos tablas R1 y R2, compatibles con su esquema, es otra relación definida sobre el mismo esquema de relación, cuya extensión estará constituida por el conjunto de tuplas que pertenezcan a R1, a R2 o a ambas (sin duplicar).

ACTOR U DIRECTOR=
{
<1, George, Lucas>,
<2, Meter, Jackson>,
<3, Alejandro, Amenábar>,
<4, Hayao, Miyazaki>,
<1, Mark, Hamill>,
<2, Christopher, Lee>,
<3, Javier, Bardem>,
<4, Hugo, Weaving>
}

- **DIFERENCIA. (−)** La diferencia de dos relaciones R1 y R2, compatibles en su esquema, es otra relación definida sobre el mismo esquema de relación, cuya extensión estará constituida por el conjunto de tuplas que pertenecen a R1 y no pertenecen a R2.

ACTOR – DIRECTOR =
{
<1, Mark, Hamill>,
<2, Christopher, Lee>,
<4, Hugo, Weaving>
}

- **PRODUCTO CARTESIANO. (X)** El producto cartesiano de dos relaciones R1 y R2 de cardinalidades m1 y m2 respectivamente, es una relación definida sobre la unión de los atributos de ambas relaciones y cuya extensión estará constituida por las **m1 x m2** tuplas formadas concatenando cada tupla de la primera relación con cada una de las tuplas de la segunda relación.

***Película*×*Estudio* =**
{
<1,La guerra de las galaxias,1977,3,1,Ghibli>,
<1,La guerra de las galaxias,1977,3,2,New Line Cinema>,
<1,La guerra de las galaxias,1977,3,3,Lucasfilms>,
<1,La guerra de las galaxias,1977,3,4,Sogecine>,
<2,La comunidad del anillo,2001,2,1,Ghibli>,
<2,La comunidad del anillo,2001,2,2,New Line Cinema>,
<2,La comunidad del anillo,2001,2,3,Lucasfilms>,
<2,La comunidad del anillo,2001,2,4,Sogecine>,
<3,Mar adentro,2004,4,1,Ghibli>,
<3,Mar adentro,2004,4,2,New Line Cinema>,
... }

AUTOEVALUACIÓN:

Sabemos que el Modelo Relacional tiene una fuerte base matemática, que se fundamenta en la teoría de conjuntos. Los principales operadores utilizados en este modelo se pueden clasificar en:

a) Operadores unarios y derivados.
b) Operadores unarios y binarios dentro de los operadores primitivos. (CORRECTA)

c) Operadores primitivos y derivados. (CORRECTA)
d) Ninguna de las respuestas anteriores es correcta.

- **OPERADORES DERIVADOS**

Estos operadores se obtienen como combinación de los operadores primitivos. Principalmente se consideran los tres siguientes, aunque existen algunos más:

  - la combinación,
  - la intersección y
  - la división.

PARA SABER MÁS

Si quieres profundizar más acerca de esta teoría de conjuntos, pulsa en los siguientes enlaces en los que encontrarás explicaciones de los distintos operadores y algunos ejemplos para comprender mejor su funcionamiento.

Álgebra relacional
http://ict.udlap.mx/people/carlos/is341/bases04.html

Álgebra relacional: Definiciones básicas.
http://www-etsi2.ugr.es/depar/ccia/bd1/bd15.ppt