



BASES DE DATOS

Temario completo

DESCRIPCIÓN:

Resumen del temario completo tomado del libro: "Bases de datos, desde Chen hasta Codd" junto con apuntes de clase.

Eloy G.

CONTENIDO

Tema 1: Introducción a las bases de datos	Pág. 1
1.1 Características de las bases de datos	Pág. 1
1.2 Bases de datos y sistemas de gestión de bases de datos	Pág. 2
Tema 2: Representación de los problemas del mundo real	Pág. 3
2.1 Los problemas del mundo real	Pág. 3
2.2 Los modelos de datos	Pág. 4
2.3 El modelo Entidad-Interrelación	Pág. 4
Tema 3: El modelo de datos relacional	Pág. 11
3.2 El modelo de datos relacional	Pág. 11
3.3 Normalización de relaciones	Pág. 12



Tema 1: introducción a las bases de datos

1.1 Características de las bases de datos

El término bases de datos no aparece hasta mediados de los años 60, cuando empiezan a usarse un conjunto de ficheros, generalmente planos. Existía gran redundancia y había poca garantía en la integridad de la información.

Una base de datos garantiza la independencia de los datos con respecto a los procedimientos a dos niveles:

- **Lógica:** La modificación de los procedimientos lógicos no afecta a los programas que la manipulan.
- **Física:** La distribución de los datos en las unidades de almacenamiento es independiente de los procedimientos que manejan la misma.

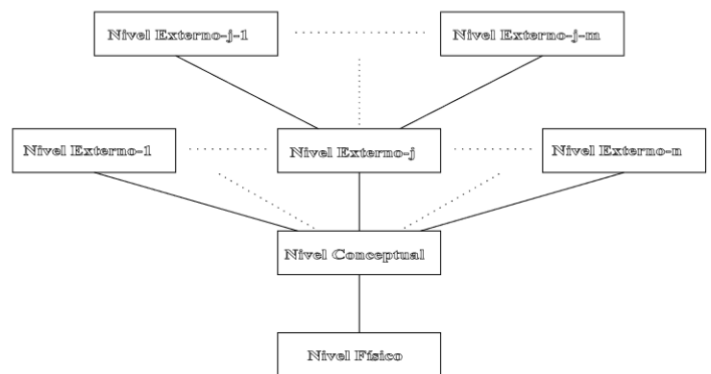
Las bases de datos siguen una serie de 9 características:

- **Versatilidad para la representación de la información:** Aunque la información que forma parte del problema es única, pueden existir diferentes visiones de esa información:
- **Desempeño:** Deben asegurar un tiempo de respuesta adecuado entre el hombre y la máquina, permitiendo el acceso simultáneo a los ítems de datos por el mismo o distinto procedimiento.
- **Capacidad de acceso:** Continuamente se reclama información a la base de datos.
- **Mínima redundancia:** Evitar la alta redundancia de las estructuras planas.
- **Simplicidad:** Deben de estar basadas en representaciones lógicas simples que permitan la verificación en la representación del problema que representan.
- **Integridad:** Hace referencia a la veracidad de los datos almacenados con respecto a la información existente en el dominio del problema que trata la misma.
- **Seguridad y privacidad:** Proteger los datos contra su pérdida total o parcial por fallos del sistema o por accesos a los mismos.
- **Afinación:** Organización física de la información de la base de datos, la cual

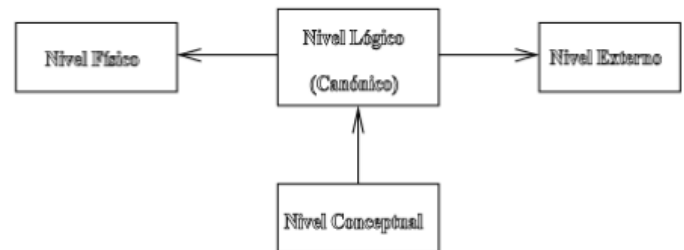
determina directamente el tiempo de respuesta de los procedimientos que operan sobre la misma.

- **Interfaz con el pasado y el futuro:** El dominio de un problema va evolucionando con el tiempo.

Existen tres niveles de abstracción de los datos en una base de datos: **visión externa** (es la visión que tienen los usuarios finales, **visión conceptual** (representación del problema y **visión física** (representación del almacenamiento de la información).



En cuanto a la independencia:



Siendo el **nivel conceptual** el más importante, sólo depende del problema del mundo real que se está representando, si éste no cambia, aunque cambie el mecanismo lógico, no cambiará la descripción conceptual. Conceptos:

Granularidad: Es el nivel de detalle en que pueden ser descritas las representaciones externas derivadas, es decir, a mayor granularidad (menor información a considerar) mayor será la independencia, sin embargo, trae mayor complejidad en el software.

Ligadura: La integridad de una base de datos necesita que los procedimientos que la manejan a un determinado nivel de abstracción tengan en cuenta cómo se representa la información en otros niveles. Existen dos tipos de ligadura:



- **Lógica:** vinculación que se produce entre las representaciones externas y la lógica.
- **Física:** vinculación entre la representación lógica y la física.

1.2 Bases de datos y sistemas de gestión de bases de datos

Definición: Una base de datos es una colección de archivos relacionados que almacenan tanto una representación abstracta del dominio de un problema del mundo real como los datos correspondientes a la información acerca del mismo. Sus restricciones son las innatas al problema y las que garantizan la integridad.

Definición: Un Sistema de Gestión de Bases de Datos es una colección de programas de aplicación que proporcionan al usuario de la base de datos los medios necesarios para realizar las siguientes tareas:

- **Definición de los datos** a los distintos niveles de abstracción.
- **Manipulación de los datos**, es decir, inserción, modificación, borrado y acceso o consulta de los mismos.
- **Mantenimiento de la integridad**, en cuanto a los datos en sí, sus valores y las relaciones entre ellos

Componentes de los SGBD:

1. **El lenguaje de definición de datos (DDL):**
Es un lenguaje artificial basado en un determinado modelo de datos que permite la representación lógica de éstos, esta representación es almacenada en el Diccionario de Datos.
2. **Lenguaje de definición de almacenamiento de datos.**
3. **Lenguaje de manipulación de datos (DML):** Es un lenguaje artificial mediante el cual se realizan dos funciones bien diferentes en la gestión de los datos: la definición de los datos y la manipulación de éstos. Dependiendo del modelo de datos y del SGBD existen dos tipos de DML: **procedimentales** (requieren que en las sentencias se especifique qué operaciones deben realizar) y **no procedimentales** (donde el DML se encarga de determinar los procedimientos más efectivos)

4. **El diccionario de datos:** Se trata de una base de datos que contiene información sobre otra base de datos. Así, en el diccionario de datos se encuentra almacenado:
 - El esquema lógico de la bd.
 - El esquema físico de la bd.
 - Los subesquemas de la bd.
 - Las restricciones de privacidad y acceso a los datos.
 - Información que garantiza la integridad de los datos.
5. **Monitor de la base de datos:** Es el responsable de garantizar la privacidad, seguridad, integridad y acceso de los datos y de interactuar con el sistema operativo mediante el *procesador de consultas*.
6. **El administrador de la base de datos:** Se encarga de la definición de diccionario de datos.
7. **Los usuarios de la base de datos:** Pueden ser:
 - **Usuarios terminales:** Aquellos usuarios que interactúan con la base de datos
 - **Usuarios técnicos:** Aquellos que desarrollan los programas de aplicación usados por los usuarios terminales
 - **Usuarios especializados:** Aquellos que utilizan el SGBD como una herramienta en el desarrollo de otros sistemas.
 - **Usuarios críticos:** Son aquellos usuarios pertenecientes al *staff* de las empresas donde se ha instalado la base de datos.



Tema 2: Representación de los problemas del mundo real

2.1 Los problemas del mundo real

Para representar los problemas del mundo real, nos basamos en la capacidad de abstracción para simplificar la naturaleza del problema.

Al conjunto de las propiedades que caracterizan un fenómeno se le denomina **datos**, y al conjunto de valores, relaciones o dependencias se les denomina **información**.

Un **modelo** es una representación abstracta y holística de un determinado proceso, donde los datos que proporcionan la información deben representar valores medibles de las propiedades que caracterizan el fenómeno. Para esta medición se definen los **tipos de datos (dominio)** (entero, real, booleano, etc.)

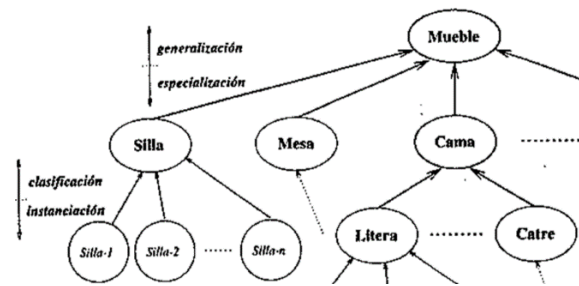
Sin embargo, un fenómeno puede ser descrito mediante un conjunto de datos o propiedades que pueden cambiar a lo largo del tiempo y que necesitan una representación para tener significado (Kg, Litro, metro, etc.) Para ello usamos los **modelos de datos**, un conjunto de reglas de acuerdo a las cuales puede ser descrito un fenómeno, aunque no proporciona una interpretación completa, sino únicamente a qué operaciones puede ser sometida la información.

2.1 Los problemas del mundo real

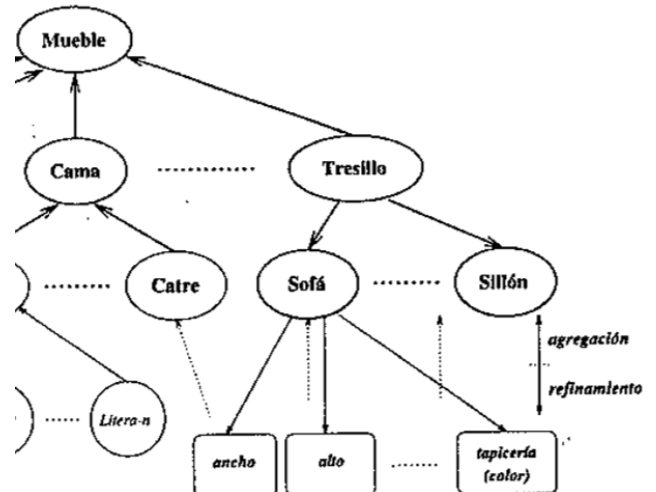
El primer paso en la representación de un problema es la determinación de sus límites, es decir, qué datos intervienen en el sistema y cómo pueden ser medidos basándonos en la abstracción.

2.1.1 Abstracción

La abstracción es la capacidad mediante la cual una serie de objetos con propiedades comunes se categorizan en una nueva **clase** o **tipo de objeto**. En la definición de datos, la abstracción es utilizada de dos formas: generalización y agregación: la **generalización** es la abstracción por la cual un conjunto de clases de objetos puede ser visto como una nueva clase más general, la generalización de objetos simples en una clase es denominada **clasificación**, a los procesos inversos se les denomina **especialización** e **instanciación**. Por ejemplo:



La **agregación** es la capacidad de considerar un objeto basándose en los elementos que lo constituyen, por ejemplo:

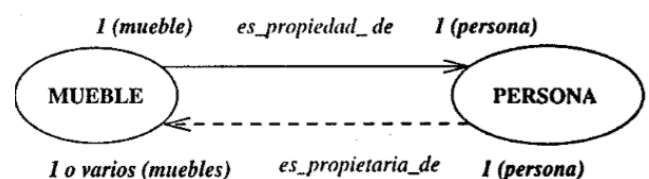


La clase Tresillo en la figura es vista como la agregación de otras dos clases, la clase Sofá y la clase Sillón. El proceso inverso a la agregación se denomina **refinamiento**, mediante el cual se puede representar a aquellos objetos simples o propiedades que caracterizan a una clase de objetos.

La generalización puede asociarse con el concepto **ES_UN**, mientras que la agregación se asociaría con el concepto **PARTE_DE**.

2.1.2 Representación de los problemas del mundo real

Además de las relaciones **ES_UN** y **PARTE_DE** consideradas, en la mayor parte de problemas existen otro tipo de relaciones entre los elementos. También se deben representar las interdependencias:





2.1.3 Análisis de los problemas

El análisis del problema consiste en determinar las entidades del sistema, sus dependencias, su comportamiento y las interdependencias con otros sistemas.

De forma general, la representación de un problema requiere el seguimiento de los siguientes pasos:

1. **La definición del problema:** Mediante una descripción simple y de cuál es la función que el sistema intenta alcanzar a grandes rasgos.
2. **Definición de la arquitectura del problema:** Mediante una descripción de las partes importantes del sistema: los subsistemas y su interdependencia.

En estos dos pasos se están definiendo los límites del sistema y de cada uno de los subsistemas que lo componen, diferenciando el problema de otros conjuntos.

3. **Definición de la estructura del problema:** En esta fase se determinarán qué objetos, entidades, datos o variables son las que forman el problema. Para cada uno de estos objetos se determina:
 - **Definición del objeto:** Función que desempeña el objeto dentro del problema.
 - **Medida del objeto:** Valores que pueden ser medidos o puede tomar el objeto para el problema. A cada uno de los objetos de le asignará un *tipo de datos* (int, bool...). Este paso se realiza mediante el uso de la *generalización*, de forma que los objetos son generalizados mediante una relación *ES_UN* a una clase básica.
 - **Relaciones entre objetos:** Descripción de las interdependencias entre los objetos que intervienen. Se agregan objetos particulares en objetos más generales.
 - **Descripción de las restricciones** inherentes a los objetos.

4. **Definición de la dinámica del problema:** Es decir, la descripción de la evolución que el problema va a tener con el tiempo, proponiendo modelos mediante los cuales pueden preverse futuros estados del sistema

5. Estudio del comportamiento del modelo propuesto.

2.2 Los modelos de datos

Mediante un modelo de datos el sistema es descrito como una clase de objeto que se relaciona con otras clases de objetos (sistemas), este modelo tiene diferentes niveles de abstracción a la hora de representar un problema:

NIVEL DE DESCRIPCIÓN	ESTRUCTURA	COMPORTAMIENTO
Modelo Conceptual	Descripción de los objetos del mundo real, de sus atributos o propiedades y de las relaciones entre los objetos.	Las acciones y procesos que estos objetos realizan sobre otros objetos.
Modelo Lógico	Descripción de los objetos, así como las relaciones existentes entre los objetos lógicos, identificando los atributos por los cuales pueden ser identificados	Descripción de las tareas que se deben realizar para representar el comportamiento de los objetos. Estas tareas se agruparán en fases y procedimientos
Modelo Físico	La estructura y relaciones de los objetos son representadas de forma adecuada para su posterior almacenamiento, recuperación y tratamiento	Descripción de las acciones elementales que se deben realizar para representar el comportamiento de los objetos.

2.2.1 Modelos de datos y sistemas de gestión de bases de datos

Cada SGBD está basado en el uso de un modelo de datos y en el uso de su teoría par la descripción y manipulación de los datos. Se tienen:

SGBD	MODELOS DE DATOS
Jerárquicos	Jerárquico
En red	Plex
Relacionales	Relacional
Orientados a objeto	Orientados a objeto

2.3 El modelo Entidad-Interrelación

El modelo E-R fue propuesto por Peter Chen a mediados de los años 70 para la representación conceptual de los problemas. Este modelo está soportado en la representación de los datos haciendo uso de grafos y tablas, mediante un conjunto de símbolos y haciendo uso de un conjunto de reglas para representar los elementos que forman parte del sistema y las relaciones entre ellos.



Conceptos básicos para el modelo E-R:

Conjunto: Agregación de una serie de objetos elementales mediante una función de pertenencia. La función de pertenencia caracteriza a los elementos del conjunto, no siendo importante ni el orden ni la duplicación. Por ejemplo $\{1, 2, 3\}$ es igual a $\{2, 2, 1, 3, 1\}$.

Relación: Conjunto que representa la **correspondencia entre dos o más conjuntos**, es, por tanto, un nuevo conjunto en el que cada elemento está formado por la agregación de los elementos de los conjuntos que intervienen en la relación. El orden de la relación es importante, por lo que el par $\langle c1, c2 \rangle$ no tiene por qué ser igual al par $\langle c2, c1 \rangle$.

Además, las relaciones pueden ser binarias, ternarias o n-arias y pueden ser definidas como el producto cartesiano de los conjuntos que intervienen en la relación ($R \subseteq C1 \times C2 \times \dots \times Cn$)

Intención y extensión: La intención representa la clasificación de una serie de elementos individuales en un tipo o clase de objeto al que se ha denominado conjunto o relación. **La extensión representa la instanciación de un tipo o clase de objeto.**

Así, **la intención es la descripción del tipo** o clase de objeto y la extensión es la descripción de los elementos del conjunto. Por ejemplo, consideremos los conjuntos C_1 y C_2 , donde $C_1 = \{c = \text{letra}\}$ y $C_2 = \{c = 0, 1\}$ y la correspondencia entre los mismos definida por la relación $R \subseteq C_1 \times C_2$. Así el conjunto $C_3 = \{\langle a, 1 \rangle, \langle b, 0 \rangle, \langle c, 1 \rangle\}$ representa la extensión de la relación R .

Mientras la intención es invariable con el tiempo, pues describe la estructura del conjunto, la extensión sí varía con el tiempo, pues depende de los elementos del conjunto en un momento dado.

Dominio: Se denomina dominio a los **conjuntos cuyos elementos son homogéneos**. En términos de abstracción, es una especialización de un conjunto. Por ejemplo, un conjunto de los números entre el 10 y el 99 es un dominio del conjunto infinito de los números enteros.

Atributo: Se denomina atributo de un dominio a la intención de ese dominio, y el valor del atributo será la extensión del dominio. **Un atributo identifica el significado de un dato.** Por ejemplo, el atributo *edad* de una serie de objetos *Persona*,

puede ser definido sobre la base de los números enteros.

Entidad: Es un conjunto definido en base a la agregación de una serie de atributos, corresponde a la caracterización de objetos del mundo real. Las entidades tienen, como los conjuntos, intenciones y extensiones. La intención de una entidad se denomina Tipo de entidad, que representa la clasificación de las entidades individuales. La extensión es denominada *Conjunto de entidades* y se corresponde con todos los valores que en un momento dado son asociados con cada atributo que define el tipo de entidad.

En definitiva, un **tipo de entidad** es una estructura genérica que describe un conjunto de entidades aplicando la abstracción a la clasificación y por tanto, una **entidad** es un ejemplar de ese tipo de entidad, resultado de la clasificación de ese tipo de entidad.

Interrelación: Representa la relación existente entre entidades, denominándose **tipo de interrelación** a la intención de la relación existente entre dos tipos de entidad. La extensión de un tipo de interrelaciones es denominada **Conjunto de interrelaciones** y representa a cada una de las posibles correspondencias entre los conjuntos de entidades que intervienen en el tipo de interrelación.

En definitiva, un tipo de interrelación es una estructura genérica que describe un conjunto de interrelaciones, y cada interrelación es un ejemplar concreto de ese conjunto, entendiendo por interrelación una asociación, vinculación o correspondencia entre entidades.

2.3.1 Entidades e interrelaciones en el modelo E-R

El modelo E-R considera que una entidad es un objeto real o abstracto que forma parte del problema y que tiene **existencia propia** (la entidad existe como un elemento que interviene en el comportamiento global del sistema), es **distinguible** del resto de entidades, y las entidades del mismo tipo están definidas en base a un mismo conjunto de atributos.

Un tipo de interrelación es definido como una relación matemática entre n tipos de entidades:

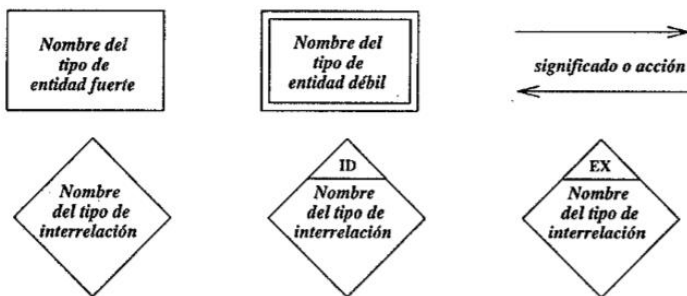
$$R = \{[e_1, e_2, \dots, e_n] \mid e_1 \in E_1, e_2 \in E_2, \dots, e_n \in E_n\}$$

Donde E_1, E_2, \dots, E_n son tipos de entidad.



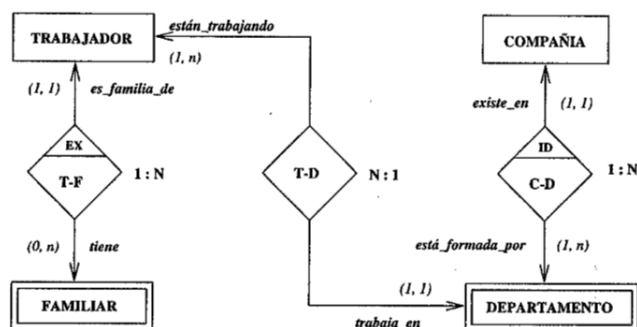
En el modelo E-R se consideran dos tipos de entidad:

- **Tipo de entidades fuertes:** Cuya existencia no depende de la existencia de ningún otro tipo de entidad en la consideración del problema.
- **Tipo de entidades débiles:** Cuya existencia depende de la existencia de un tipo de entidad fuerte. Estas entidades débiles pueden ser de dos tipos:
 - **Debilidad por identificación:** No puede ser identificada (reconocida y diferenciada del resto de entidades) a no ser que se identifique una entidad fuerte por cuya existencia está presente la debilidad.
 - **Debilidad por existencia:** Puede ser identificada sin necesidad de identificar la entidad fuerte por la cual existe.



Una debilidad de identificación implica una debilidad de existencia, pero no al contrario.

Ejemplo:



La debilidad del tipo de entidad *Familiar* es por existencia, ya que no existirá si no existe *Trabajador*. Por otro lado, la debilidad del tipo de entidad *Departamento* es tanto por existencia como por identificación, pues *Departamento* no existirá si no existe una entidad *Compañía* asociada con la cual esté relacionada (C-D), y, además, una entidad *Departamento* no puede ser identificada por sí sola, sino que es necesario identificar a la entidad *Compañía* con la cual

está relacionada para poder diferenciarla del resto de entidades *Departamento*.

En un diagrama E-R es necesario representar, para cada tipo de interrelación, las **cardinalidades** con las que cada tipo de entidad interviene en el tipo de interrelación. Si un tipo de interrelación representa una correspondencia entre tipos de entidad, es necesario explicitar el número de entidades que intervienen. Se representan mediante una pareja de datos:

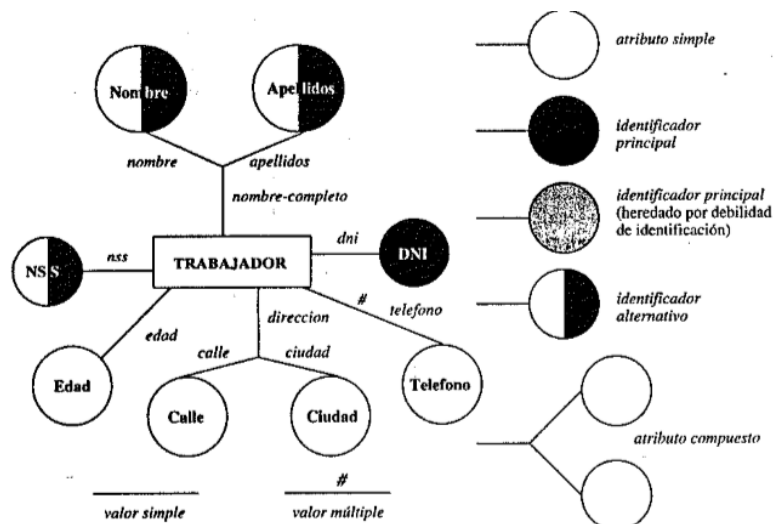
(cardinalidad mínima, cardinalidad máxima)

En el ejemplo anterior, para el tipo de interrelación (T-F), el tipo de entidad *Trabajador* y *Familiar*, el tipo de entidad *Trabajador* participa con la cardinalidad (1, 1), la cual informa que una entidad de tipo *Familiar* está relacionada con 1 de tipo *Trabajador*. Y una entidad del tipo *Trabajador* puede estar relacionada, a través del tipo de interrelación (T-F), con 0 como mínimo y *n* entidades como máximo del tipo *Familiar*.

Además, los tipos de interrelación se deben acompañar con las cardinalidades máximas con las que intervienen los tipos de entidad de la interrelación, por ejemplo, en (T-F) aparece como máximo 1 *trabajador* y como máximo *n* *Familiares*: 1:N.

2.3.2 Descripción de los tipos de entidad e interrelación

Los tipos de entidad son caracterizados en base a un conjunto de propiedades denominados **atributos**. Un atributo es una interpretación de un dominio para un determinado tipo de entidad, aportando un significado en el mundo del problema en el cual ese tipo de entidad es considerado.





Generalmente los atributos pueden tomar un único valor para cada una de las entidades con las que están relacionados, sin embargo, algunos pueden tomar un conjunto de valores, se les denomina *atributos múltiples* y son representados mediante el signo # acompañando al nombre del atributo, o bien indicando el número de valores que pueden tomar (2, 3, ..., n).

Los **atributos identificadores** de un tipo de entidad son un conjunto de atributos (o uno solo) que no toma el mismo valor para dos entidades diferentes del mismo tipo, estos conjuntos de atributos deben cumplir la condición de ser mínimos; es decir, el conjunto de atributos identificadores será tal que, si se eliminase algún atributo de este conjunto, dejaría de cumplir la propiedad de *identificador*.

Para un tipo de entidad puede haber más de un conjunto de atributos que satisfagan esta condición, siendo *candidatos* para desempeñar el papel de identificación. En estos casos, se asigna a uno de los grupos el papel de **identificador principal** y al resto el de **identificador alternativo**.

Siempre que no se trate de una entidad débil, debe haber un atributo identificador.

Ejemplos:

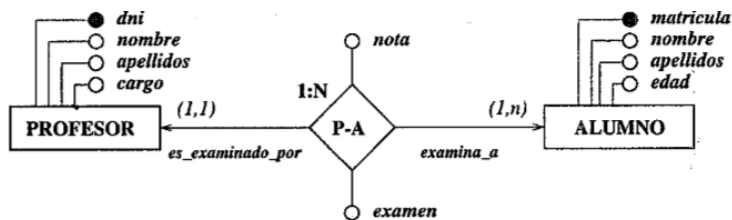


Figura 2.7 Diagrama E-R de la relación entre profesores y alumnos

En la figura 2.7 se muestran dos tipos de entidad, *Profesor* y *Alumno* relacionados mediante un tipo de interrelación que representa el hecho de que un *Profesor* imparte docencia a uno o varios alumnos mientras que a un *Alumno* sólo le imparte docencia sólo un *Profesor*. Sin embargo, aunque la relación (P-A) tiene asociados dos atributos *examen* y *nota*, debería representar que, aunque a un alumno sólo le imparte docencia un profesor, existe una serie de calificaciones que el profesor imputa a cada alumno para los exámenes que realiza.

En muchas ocasiones, un tipo de interrelación en la que estén presentes atributos puede descomponerse en un tipo de entidad que

mantiene una serie de interrelaciones con los tipos de entidad que participaban en el tipo de interrelación, dependiendo del problema será necesaria una nueva interrelación *n-aria* o *n* interrelaciones binarias.

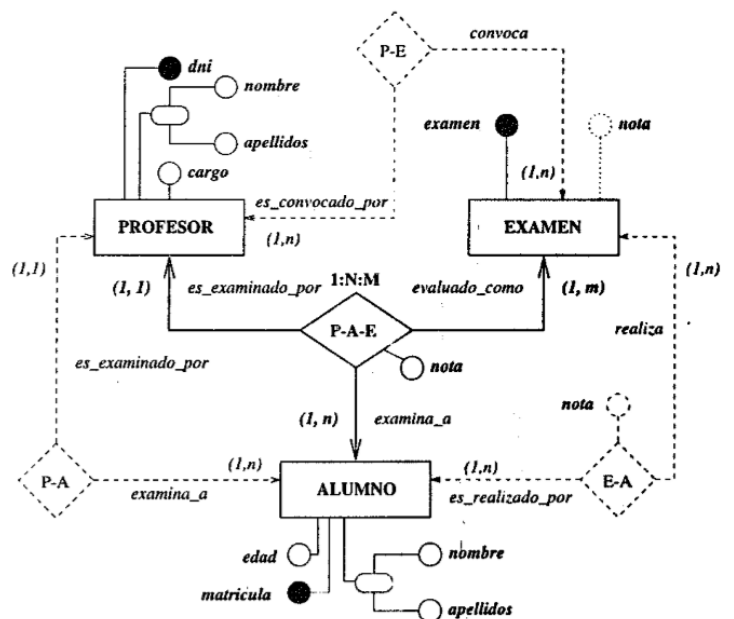


Figura 2.8 Refinamiento del diagrama E-R de los profesores y alumnos

En la figura 2.8 se muestra esa descomposición, sin embargo, si se observa el diagrama, se puede apreciar que para que *nota* sea un atributo del tipo de entidad *Examen*, se debe cumplir que *Examen* sea una entidad débil por identificación con respecto a *Alumno* y, por tanto, el identificador del tipo de entidad *Examen* sea, por ejemplo, una concatenación de los atributos *examen* y *matrícula*. En caso contrario, todos los alumnos que realizaran un mismo examen tendrían el mismo valor del atributo *nota*, pues éste sólo es dependiente del valor del atributo *examen* y no del alumno que lo realiza, dado que varios alumnos pueden realizar el mismo examen.

La relación entre los tipos de entidad *Examen* y *Alumno* sería $1 \rightarrow$ muchos, dado que un examen correspondería a un único alumno, pues vendrá identificado por el par (examen, matrícula).

Como esta situación no es la que se desea representar, es necesario que el atributo *nota* se considere como un atributo del tipo de interrelación entre *Examen* y *Alumno*, como así se muestra en la figura.

Nota: El diagrama correcto, por tanto, es el que viene definido en línea de puntos, eliminando la interrelación (P-A-E) y el atributo *nota* de la entidad *Examen*.

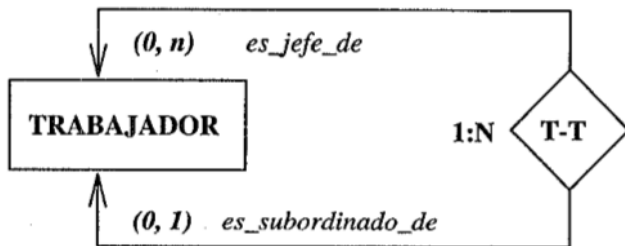


2.3.3 Los tipos de interrelación en el modelo E-R

Las últimas actualizaciones del modelo E-R han dado lugar a lo que se denomina *Modelo Entidad-Interrelación-Extendido (EE-R)*, permiten la representación de cualquier tipo de relaciones existentes entre clases de objetos.

Interrelaciones reflexivas

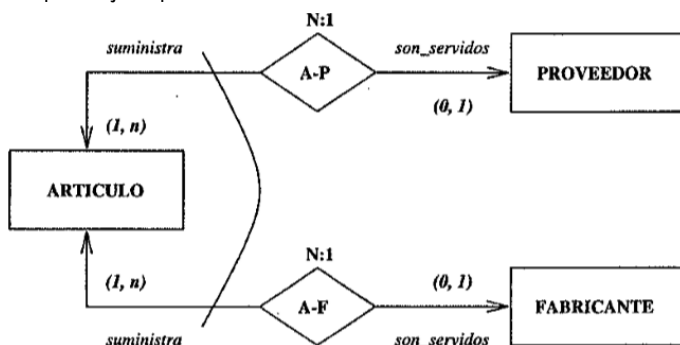
Son relaciones unarias y, por tanto, consideran que en el tipo de interrelación se ve involucrado un único tipo de entidad, por ejemplo:



Se trata de un tipo de interrelación reflexiva en la que interviene un único tipo de entidad que desempeña dos papeles diferentes en el mismo tipo de interrelación (Un trabajador es jefe de 0 o varios trabajadores, mientras que un trabajador sólo es dirigido por 0 (el jefe de todos) o 1 trabajador)

Interrelaciones exclusivas

Un tipo de entidad puede mantener relaciones con un conjunto de otros tipos de entidad, pero no siempre estas relaciones son independientes, por ejemplo:



En la figura se presentan tres tipos de entidad *Artículo*, *Proveedor* y *Fabricante*, el problema indica que los artículos son suministrados por los proveedores o por los fabricantes, pero un artículo nunca puede ser suministrado por un proveedor que no fabrica el artículo, de forma que si el fabricante puede suministrarlo, en ningún momento será solicitado a un proveedor.

2.3.4 Generalización y herencia en el modelo EE-R

El modelo EE-R permite representar las relaciones jerárquicas existentes entre los tipos de entidad de los problemas del mundo real. Este tipo de relaciones entre tipos de entidad implica la consideración de tipos de entidad (o supertipos) y subtipos de entidad (clases, superclases y subclases de objetos)

Un **subtipo de entidad** es un tipo de entidad que mantiene un tipo de interrelación jerárquica con otro tipo de entidad, y que:

- Representa a un conjunto de entidades cuyas propiedades y comportamiento general es considerado por la entidad supertipo
- La relación jerárquica puede ser n-aria entre un tipo de entidad y un conjunto de subtipos
- Las propiedades y el comportamiento de los subtipos son heredados del supertipo
- Las propiedades y/o el comportamiento de un subtipo deben cambiar con respecto a otros subtipos que intervengan en la misma relación. Estos deben distinguirse sin ambigüedad.
- Para cada subtipo de entidad pueden ser redefinidas tanto las propiedades como el comportamiento del supertipo
- Un tipo de entidad puede ser un subtipo para más de un tipo de entidad con las que puede mantener diferentes relaciones jerárquicas.

Un tipo de interrelación jerárquica representa una especialización de un tipo de entidad en otros tipos de entidad. Esta especialización puede ser debida a:

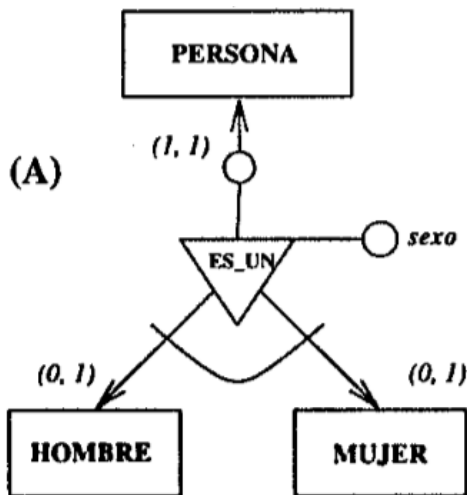
1. Diferencia en cuanto al número de propiedades que definen los subtipos de entidad
2. Diferentes valores que pueden ser medidos para una propiedad del conjunto de subtipos
3. Que se cumplan ambas condiciones

Esta especialización puede ser **exclusiva**, representa el hecho de que una instancia del supertipo sólo pueda pertenecer o estar asociada a una instancia de los subtipos o **inclusiva**, que representa el hecho de que una instancia del tipo de entidad más general puede tener asociadas instancias de cualquier subtipo.



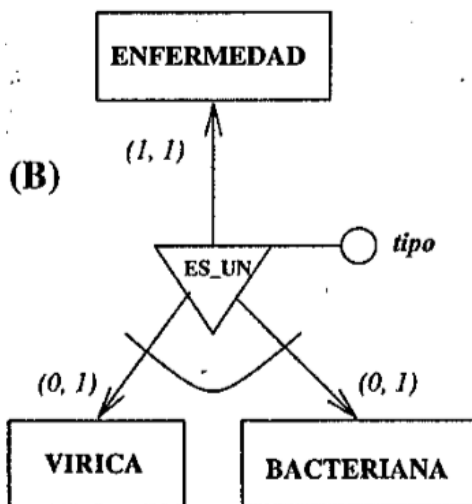
Por otro lado, la especialización puede ser **total**, que representa el hecho de que las entidades son de alguno de los subtipos especializados, no existiendo entidades que no pertenezcan a alguno de los subtipos. Una especialización **parcial** representa el hecho de que pueden existir entidades que pertenezcan al tipo de entidad y no a ninguno de los subtipos especializados. Por lo tanto, tenemos:

Total exclusiva



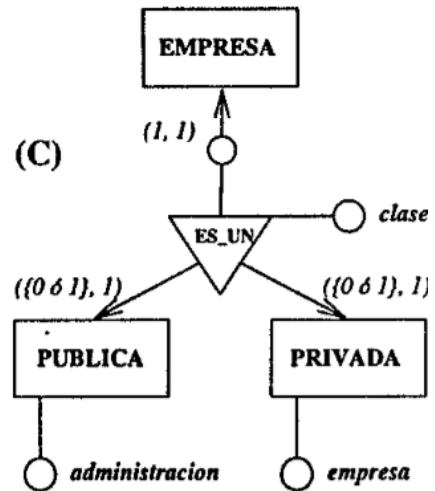
No existirá una entidad *Persona* que no sea de alguno de los dos subtipos y además de forma exclusiva, por lo que una entidad pertenecerá a uno y sólo a uno de los subtipos.

Parcial exclusiva



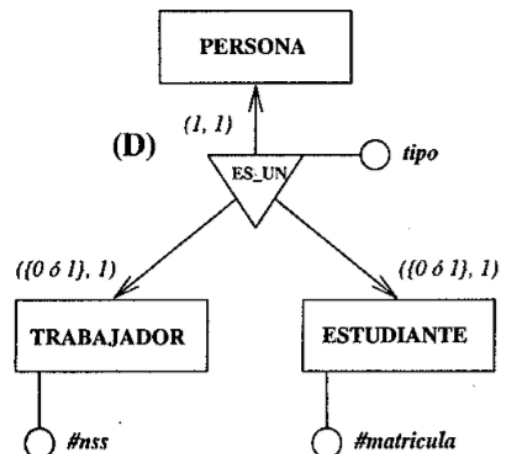
Se consideran un conjunto de entidades *Enfermedad* las cuales pertenecerán bien a alguno de los subtipos considerados *Virica* o *Bacteriana*, pero que además existirán entidades *Enfermedad* las cuales no puedan ser clasificadas en ninguno de estos subtipos debido, posiblemente, al desconocimiento del valor del atributo *tipo* utilizado como discriminador.

Total inclusiva



Se está representando el hecho de que podrán existir en el dominio del problema entidades que puedan ser consideradas tanto del tipo *Pública* como *Privada*, o bien ambos tipos al mismo tiempo y, además el hecho de que no podrán existir entidades que no puedan ser especializadas en alguno de estos dos subtipos.

Parcial inclusiva



Este ejemplo representa que una entidad *Persona* puede ser del tipo *Trabajador* y/o del tipo *Estudiante* y que además pueden existir entidades *Persona* que no puedan clasificarse en ninguno de estos dos subtipos.

En estos dos últimos ejemplos, los subtipos de entidad incorporan nuevos atributos mediante los cuales pueden diferenciarse entidades pertenecientes a los distintos subtipos. Igualmente podrían existir atributos pertenecientes al tipo de interrelación jerárquica fuera esta diferenciación de las entidades pertenecientes a los subtipos.

**Cardinalidades en la jerarquía:**

Dependiendo del tipo de interrelación jerárquica que se represente, los tipos de entidad participarán con un número determinado de ocurrencias:

- El tipo de entidad más general que es especializado participa con la cardinalidad mínima 1 y máxima 1.
- La entidad de los subtipos tiene una cardinalidad máxima de 1, ya que representa una especialización.
- Si es total o parcial sin solapamiento, los subtipos tendrán cardinalidad mínima 0, puesto que una entidad pertenece a un subtipo únicamente.
- Si la interrelación es total o parcial con solapamiento, los subtipos tendrán cardinalidad mínima 0 ó 1, ya que se puede especializar en varios subtipos.



Tema 3: El modelo de datos relacional

3.2 El modelo de datos relacional

Este modelo lógico, desarrollado por E.F Codd, está basado en conceptos muy sencillos teniendo asociada la *teoría de normalización de relaciones* para eliminar los comportamientos anómalos de las relaciones a la hora de manejar la información. Propone una representación de la información que:

- Origine esquemas que representen fielmente la información, los objetos y las relaciones entre ellos existentes en el dominio del problema.
- Pueda ser entendida fácilmente por los usuarios no especializados.
- Haga posible ampliar el esquema de la base de datos sin modificar la estructura lógica existente (programa/aplicación).
- Permita la máxima flexibilidad en la formulación de los interrogantes previstos y no previstos sobre la información mantenida en la base de datos.

3.2.1 Terminología del modelo relacional

La forma de representar la información es sobre la base tabular plana. Una tabla bidimensional mediante la cual se representan tanto los objetos como las relaciones entre ellos.

Una **tabla** es una matriz rectangular que puede ser descrita de forma simple matemáticamente y que posee las siguientes propiedades:

1. Cada entrada de la tabla (cada elemento) representa un ítem de datos elemental
2. Una tabla es homogénea por columnas, es decir, todos los ítems de datos elementales de una columna son de la misma clase, están definidos en el mismo dominio y representan una misma propiedad del problema.
3. Cada columna tiene asignado un nombre único en el conjunto de columnas de esa tabla.
4. Para una tabla, todas las filas son diferentes
5. Tanto las filas como las columnas pueden ser consideradas en cualquier secuencia sin afectar al contenido de la información ni a la representación semántica de la misma.

Terminología: A una tabla o matriz rectangular se la denomina **relación**, a las filas de la misma se les denomina **tuplas** y al conjunto de sus

columnas **dominio** de la relación. Así, una base de datos relacional estará formada por un conjunto de relaciones.

3.2.2 Relación

Dada una serie de conjuntos $D_1, D_2, D_3, \dots, D_n$, no necesariamente distintos, se dice que R es una relación entre estos n conjuntos si es un dominio de n tuplas no ordenadas $(d_1, d_2, d_3, \dots, d_n)$ tales que $d_1 \in D_1, d_2 \in D_2, d_3 \in D_3, \dots, d_n \in D_n$. A los conjuntos $D_1, D_2, D_3, \dots, D_n$ se les denomina **dominios** de R , y el valor de n es el grado de la relación R . Ejemplo:

ALUMNO	matricula#	nombre	apellidos	curso	nota
	3456	José	Pérez de la Lastra	1	5.25
	0101	María	Antúnez Sastre	2	7.80
	8743	Lourdes	Sánchez Argote	1	4.50
	1234	Antonio	Soria Madrid	3	6.35
	5674	Luis	González Silos	1	3.20
	0678	Pilar	Alcántara Badajoz	2	5.50
	0345	Dolores	Almiz Márquez	3	7.30
	2985	Manuel	Rives Fuentes	3	3.50

Al número de tuplas de una relación en un instante dado se le denomina **cardinalidad** de la relación (ocho en este caso), al número de columnas de una relación se le denomina **grado** de la relación (cinco en este caso).

Mientras la cardinalidad de una relación depende del momento en que ésta sea considerada, el grado es independiente del tiempo, pues hace referencia al número de dominios que define la relación. Dependiendo del grado de la relación éstas se denominan: unarias, binarias, ternarias, etc.

3.2.3 Dominios y atributos

Un **atributo** representa el uso de un dominio para una determinada relación, es decir, un atributo aporta significado semántico a un dominio. Mientras que un **dominio** es un conjunto homogéneo definido mediante el uso de la abstracción en base a otro conjunto (*Tema 2). Ejemplo:

Define Dominio	expediente	entero (4)	Fin definición;
Define Dominio	primer-nombre	carácter (15)	Fin definición;
Define Dominio	final-nombre	carácter (40)	Fin definición;
Define Dominio	estudios	entero (2)	Fin definición;
Define Dominio	nota	real (4)	Fin definición;
Define Relación	Alumno		
	(matricula#:	Dominio	expediente,
	nombre:	Dominio	primer-nombre,
	apellidos:	Dominio	final-nombre,
	curso:	Dominio	estudios,
	nota:	Dominio	nota);

Como se puede observar, se han definido primero cinco dominios, y en base a ellos, se han definido cinco atributos de la relación.



3.2.4 Intención y extensión de relaciones

La **intención** va a ser siempre *invariable con el tiempo y se trata de la definición de las propiedades* o atributos del objeto del mundo real representado por la relación, de las cuales cada una está definida en su correspondiente dominio de datos. Define dos aspectos:

1. Una estructura de datos nominada en la que tanto la estructura (el objeto) como los datos que la componen (atributos/propiedades) tienen asignado un nombre único y están definidos en un determinado dominio
2. Un conjunto de restricciones de integridad.

Por otro lado, la **extensión** de una relación representa a cada uno de los objetos (tuplas), pertenecientes a un mismo tipo (relación), existentes en el dominio del problema en un momento dado.

3.2.5 Consistencia de la representación lógica relacional

La representación lógica, independientemente del modelo utilizado, debe satisfacer la representación conceptual de ese problema y, por tanto, las especificaciones, restricciones y requisitos definidos en el proceso de análisis del problema del mundo real que se esté tratando, por lo que existen una serie de **reglas de integridad** que permiten representar de forma consistente la semántica del problema. Aunque antes se definirán una serie de términos:

Claves de las relaciones

Es usual que en el conjunto de atributos que forman parte de la intención de una relación específica, un conjunto de ellos tenga la propiedad de tomar valores únicos en el dominio del problema, identificando sin ambigüedad y de forma única a una y sólo una, de las tuplas de esa relación. Por ejemplo, en la relación *Alumno* utilizada en las secciones anteriores se puede considerar:

- Que el atributo *matrícula#* tiene la propiedad de identificación única.
- Que sobre la base de la extensión de la relación *Alumno* presentada en la anterior tabla, la agregación de los atributos *apellidos* y *nombre* también tiene esta propiedad.

De esta forma, tanto los valores de los atributos *matrícula#* y la agregación de los atributos

(*apellidos* + *nombre*) tienen la propiedad de identificar sin ambigüedad a cada una de las tuplas de la relación *Alumno*.

A los atributos que satisfacen la propiedad de identificación única de las tuplas de una relación se les denomina **claves candidatas**.

Toda relación debe tener alguna clave candidata, además, de entre todas las claves candidatas, en la definición del esquema se debe de especificar cuál de ellas se considera como **clave principal**. El número de atributos de una relación que forman parte de cada clave candidata debe ser el mínimo que satisface la propiedad de identificación única de las tuplas.

Integridad de los esquemas relacionales

La intención de un esquema relacional debe satisfacer las siguientes reglas de integridad para garantizar la integridad de la base de datos:

- **Integridad de la clave:** Ningún atributo que forme parte de la clave candidata de una relación podrá tomar valores nulos.
- **Integridad de referencia:** Sea D un dominio primario, y sea $R_{1,a}$ definido sobre el dominio D , entonces, en cualquier instante dado, cada valor de $R_{1,a}$ en R_1 debe ser nulo o bien igual a algún valor V , el cual existe en ese instante para un atributo $R_{2,b}$ definido en el mismo dominio D sobre la relación R_2 y en la cual está definido como clave primaria. A aquellos atributos $R_{1,a}$ que satisfacen esta regla se les denomina **claves foráneas**.
- **Los valores permitidos** para los atributos que forman parte de las relaciones existentes en el esquema. Por ejemplo, valor máximo, mínimo, lista de valores, etc.
- **Condiciones** que determinan el valor que pueden tomar los atributos

3.3 Normalización de relaciones

Si bien una relación representa un conjunto, no todos los conjuntos pueden ser considerados en un esquema relacional para satisfacer los siguientes objetivos en una base de datos:

1. No-existencia de redundancias superfluas, reduciendo posibles problemas de integridad en la información de la base de datos.



2. Aumentar el desempeño de las operaciones de actualización de la base de datos.
3. Representar de forma coherente los objetos y relaciones existentes en el dominio del problema.
4. Aumentar el desempeño y garantizar la fiabilidad de las interrogaciones sobre la información mantenida en la base de datos.

Para satisfacer estos objetivos, las relaciones que forman parte de un esquema relacional deben satisfacer una serie de reglas que restringen el universo de relaciones/conjuntos que pueden ser considerados en el esquema, a este conjunto de reglas se les denomina *Reglas de normalización de relaciones* y a la teoría en la que se basan se le denomina *Teoría de normalización de relaciones*.

3.3.1 Dependencias funcionales

La normalización de relaciones está basada en otra teoría, la *teoría de las dependencias*, la cual se centra en el estudio de las dependencias que presenta cada atributo de una relación con respecto al resto de atributos de la misma relación.

Dada una relación R, se dice que el atributo $R.y \in R$ es funcionalmente dependiente de otro atributo $R.x \in R$, y se expresa de la forma $R.x \rightarrow R.y$ si, y sólo si, siempre que dos tuplas de R coincidan en sus valores de $R.x$, también, para esas tuplas, existirá una coincidencia en los valores del atributo $R.y$.

En la definición de los esquemas relacionales es de vital importancia el estudio de las dependencias funcionales existentes en las relaciones, y sobre la base de este estudio se deben aplicar las reglas de normalización de las relaciones que forman parte del esquema.

Es conveniente su representación textual y gráfica, por ejemplo, para la relación *Alumno*:

Representación textual de las dependencias funcionales

Formato general

$$R.x \rightarrow (R.a, R.b, \dots, R.n)$$

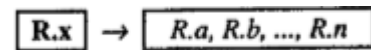
Denotando que los atributos $R.a, R.b, \dots, R.n$ son funcionalmente dependientes del atributo $R.x$

Relación Alumno

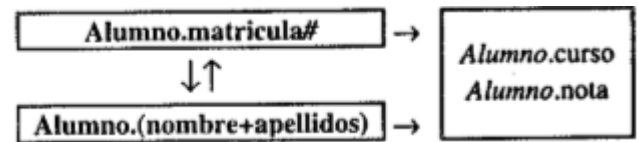
$Alumno.matricula\# \rightarrow (Alumno.nota, Alumno.curso)$
 $Alumno.matricula\# \rightarrow (Alumno.nombre, Alumno.apellidos)$
 $Alumno.matricula\# \leftrightarrow (Alumno.(nombre + apellidos))$

Representación gráfica de las dependencias funcionales

Formato general



Relación Alumno



Es conveniente introducir en estos momentos el concepto de **Dependencia funcional completa**:

Se dice que el atributo $R.y \in R$ es funcionalmente dependiente y de forma completa de otro atributo $R.x \in R$ si, y sólo si, depende funcionalmente de $R.x$ y no de ningún subconjunto de los atributos que formen parte del atributo $R.x$.

Con esta definición se puede apreciar:

- Si el atributo $R.x$ es un agregado formado por la concatenación de varios atributos pertenecientes a la relación, entonces la dependencia funcional $R.x \rightarrow R.y$, puede no ser completa, ya que puede existir una dependencia funcional de la forma $R.z \rightarrow R.y$, para $R.z \subset R.x$.
- Que el atributo $R.y$ sea simple o compuesto no tiene relevancia para que la dependencia funcional $R.x \rightarrow R.y$ sea completa o no.

En la relación *Alumno* se puede considerar que existe una dependencia entre el atributo *nota* y el agregado $(matricula\#+curso)$, $(Alumno.(matricula\#+curso) \rightarrow (matricula\#+curso))$ pero esta dependencia funcional no es completa puesto que el atributo *nota* depende también funcionalmente del atributo *matricula#*

Propiedades de las dependencias funcionales

Las dependencias funcionales satisfacen una serie de 6 propiedades (Axiomas de Armstrong) que es conveniente conocer para el análisis de las relaciones del esquema relacional:

Reflexiva: Dados los atributos $a, b \in R$, para los que se cumple $R.b \subseteq R.a$, entonces, en la relación R está presente una dependencia funcional de la forma $R.a \rightarrow R.b$. En definitiva:

$$R.b \subseteq R.a \Rightarrow R.a \rightarrow R.b$$



Aumento: Dados los atributos $a, b, c \in R$, donde existe una dependencia funcional $R.a \rightarrow R.b$, entonces también estará presente la dependencia funcional $R.(a+c) \rightarrow R.(b+c)$. En definitiva:

$$R.a \rightarrow R.b \Rightarrow R.(a+c) \rightarrow R.(b+c)$$

Transitiva: Dados los atributos $a, b, c \in R$ en la que están presentes las dependencias funcionales $R.a \rightarrow R.b$ y $R.b \rightarrow R.c$. Entonces también estará presente la dependencia funcional $R.a \rightarrow R.c$. En definitiva:

$$R.a \rightarrow R.b \text{ y } R.b \rightarrow R.c \Rightarrow R.a \rightarrow R.c$$

Unión: Dados los atributos $a, b, c \in R$ en la que están presentes las dependencias funcionales $R.a \rightarrow R.b$ y $R.a \rightarrow R.c$, entonces también estará presente la dependencia funcional $R.a \rightarrow R.(b+c)$. En definitiva:

$$R.a \rightarrow R.b \text{ y } R.a \rightarrow R.c \Rightarrow R.a \rightarrow R.(b+c)$$

Pseudo-Transitiva: Dados los atributos $a, b, c, d \in R$ en la que están presentes las dependencias funcionales $R.a \rightarrow R.b$ y $R.(b+c) \rightarrow R.d$, entonces también estará presente la dependencia funcional $R.(a+c) \rightarrow R.d$. En definitiva:

$$R.a \rightarrow R.b \text{ y } R.(b+c) \rightarrow R.d \Rightarrow R.(a+c) \rightarrow R.d$$

Descomposición: Dados los atributos $a, b, c \in R$, donde existe una dependencia funcional $R.a \rightarrow R.b$, y se cumple que $R.c \subseteq R.b$, entonces también estará presente la dependencia funcional $R.a \rightarrow R.c$. En definitiva:

$$R.a \rightarrow R.b \text{ y } R.c \subseteq R.b \Rightarrow R.a \rightarrow R.c$$

3.3.2 Reglas de normalización

Se dice que una relación está en una determinada forma normal si satisface un cierto conjunto específico de restricciones impuestas por la regla de normalización correspondiente.

- La aplicación de una regla de normalización es una operación que toma una relación como argumento de entrada y da como resultado dos o más relaciones.
- La relación objeto de la aplicación de la regla es desestimada en el nuevo esquema relacional considerado.

- No se introducen nuevos atributos en el esquema relacional resultante de la normalización.
- Los atributos de la relación objeto de la normalización pasan a formar parte de la intención de una o más relaciones resultantes.
- En la aplicación de la regla de normalización se ha debido eliminar, al menos, una dependencia existente entre los atributos de la relación objeto de la normalización.

De esta forma, la sucesiva aplicación de las reglas de normalización va a dar lugar a la generación de un número mayor de relaciones y, desde un punto de vista no sólo lógico, una redundancia de los atributos considerados en el esquema.

La aplicación sucesiva de las reglas de normalización restringe, por tanto, el número de relaciones que las satisfacen. Por regla general, se dice que un esquema relacional es consistente si las relaciones satisfacen al menos la forma normal de Boyce-Codd.

PRIMERA FORMA NORMAL. FN1

Una relación R satisface la primera forma normal si, y sólo si, todos los dominios subyacentes de la relación R contienen valores atómicos, es decir, sus atributos sólo pueden tener un único valor a la vez, esta regla se realiza con el análisis del dominio del problema.

No es condición suficiente, aunque sí necesaria, para garantizar la consistencia del esquema relacional.

SEGUNDA FORMA NORMAL. FN2

Una relación R satisface la segunda forma normal si, y sólo si, satisface la primera forma normal y cada atributo de la relación depende funcionalmente de forma completa de la clave primaria de esa relación. Y además no exista una dependencia funcional no completa entre atributos de la relación que forman parte de la clave y la clave de la relación.

Si la clave principal es simple, ya está en forma normal, sólo nos preocupamos cuando son agregados de atributos, en este caso vemos la dependencia funcional con los demás atributos y no entre los de la clave principal



Ejemplo:

Esquema-1

Matricula (dni, asignatura#, apellidos, nombre, nota, curso, aula, lugar)

El atributo *curso* representa el curso en el que se imparte la docencia de una asignatura. Por tanto, se va a suponer que existe una dependencia funcional entre los atributos *asignatura#* y *curso*, de la forma: *Matricula.asignatura#* → *Matricula.curso*, que representa que, si bien en un curso se puede impartir docencia para varias asignaturas, una asignatura está asignada a la docencia de un único curso.

Se puede apreciar que esta relación no se encuentra en FN2, puesto que existe una dependencia funcional no completa entre atributos de la relación que no forman parte de la clave y la clave de la relación.

La relación *Matricula* algunos inconvenientes debidos a que no satisface la FN2, inconvenientes que se ponen de manifiesto en:

Inserción de tuplas: No se pueden conocer algunas asignaturas que se imparten hasta que no exista algún alumno matriculado en esas asignaturas, debido a que se violaría la integridad de la clave al existir valores nulos.

Borrado de tuplas: Si se eliminan las tuplas correspondientes a los alumnos matriculados en una asignatura, se pierde la información que representa que una asignatura forma parte de la docencia de un curso.

Actualización de tuplas: Si se realiza un cambio de asignación de curso para una asignatura, sería necesario actualizar todas las tuplas correspondientes a todos los alumnos matriculados de esa asignatura, debido a que existe una gran redundancia de información.

Para eliminar todos estos inconvenientes es necesario realizar un proceso de descomposición de la relación *Matricula* en dos nuevas relaciones, las cuales satisfacen la FN2:

Esquema-2

Imparte (asignatura#, curso)

Matricula-2 (dni, asignatura#, apellidos, nombre, nota, aula, lugar)

Con este nuevo esquema, podemos apreciar:

- La relación *Imparte*, se encuentra en FN2. La clave de esta relación es el atributo *asignatura#*, y el único atributo no primo

(*curso*) depende funcionalmente de forma completa de la clave de la relación.

- Al eliminarse el atributo *curso* de la relación *Matricula-2*, se ha eliminado la dependencia funcional no completa entre el atributo *curso* y la clave de la relación.
- No se ha producido pérdida de información, puesto que el atributo *asignatura#* debe definirse como clave foránea de la relación *Imparte*. De esta forma, el atributo *asignatura#* en la relación *Matricula-2* para cualquier tupla, sólo podrá tomar valores existentes en alguna tupla de la relación *Imparte*. Así, basándose en esta referencia podrá conocerse en cada momento en qué curso es impartida cada asignatura para cada uno de los alumnos matriculados.

Hacemos el mismo razonamiento con la dependencia funcional no completa existente entre la clave de la relación *Matricula-2* y el agregado (*apellidos + nombre*), puesto que el agregado sólo depende del atributo *dni* (*identificador del alumno*) y no de las asignaturas (*valores del atributo asignatura#*) en las que se encuentra matriculado. Si se aplica, por tanto, a la relación *Matricula-2* la FN2 para eliminar esta dependencia, se obtiene:

Esquema-3

Imparte (asignatura#, curso)

Alumno-2 (dni, apellidos, nombre)

Matricula-3 (dni, asignatura#, nota, aula, lugar)

TERCERA FORMA NORMAL. FN3

Una relación *R* satisface la tercera forma normal si, y sólo si, satisface la FN2 y cada atributo no primo de la relación no depende funcionalmente de forma transitiva de la clave primaria de esa relación. Es decir, no pueden existir dependencias entre los atributos que no forman parte de la clave primaria de la relación *R*.

Un **determinante funcional** es un atributo simple o compuesto el cual determina funcionalmente de forma completa a cualquier otro atributo de la relación.

En el ejemplo anterior, se observa que existe una dependencia funcional entre los atributos no primos *lugar* y *aula*, además de las dependencias funcionales completas entre los atributos *lugar* y *aula* con la clave de la relación *Matricula-3*. Problemas en:



Inserción de tuplas: No se pueden conocer las aulas de cada uno de los lugares utilizadas para la docencia de asignaturas hasta que no haya algún alumno matriculado en esa asignatura.

Borrado de tuplas: Si se borran todos los alumnos matriculados en una asignatura, se pierde información sobre las aulas de cada lugar.

Modificación de tuplas: Existe una gran redundancia de información puesto que se almacena el lugar en el que se encuentra ubicada un aula para cada uno de los alumnos que cursan una asignatura que se imparte en dicha aula.

Estos problemas se deben a la dependencia funcional transitiva. La relación *Matrícula-3* presenta las siguientes dependencias:

$\text{Matrícula-3}(\underline{\text{dni}}, \text{asignatura\#}) \rightarrow \text{Matrícula-3.aula}$
 $\text{Matrícula-3}(\underline{\text{dni}}, \text{asignatura\#}) \rightarrow \text{Matrícula-3.lugar}$
 $\text{Matrícula-3}(\text{aula}) \rightarrow \text{Matrícula-3.lugar}$

Y, como se observa, el atributo *lugar* es dependiente de la clave de la relación de forma transitiva, debido a que es dependiente funcionalmente de otro atributo no primo (*aula*), que también depende de la clave de la relación. Para eliminar los problemas, se divide la relación:

Esquema-4	
Imparte	(<u>asignatura#</u> , curso)
Alumno-2	(<u>dni</u> , apellidos, nombre)
Ubicación	(<u>aula</u> , lugar)
Matrícula-4	(<u>dni</u> , <u>asignatura#</u> , nota, aula)

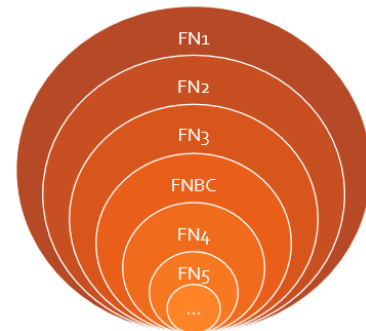
Donde las relaciones *Ubicación* y *Matrícula-4* se encuentran en FN3, puesto que todas las dependencias funcionales existentes son completas y entre atributos no primos y la clave de la relación.

De nuevo, el atributo *aula*, de la relación *Matrícula-4* deberá definirse como clave foránea de la relación *Ubicación*, de forma que para cualquier tupla de la relación *Matrícula-4*, este atributo sólo pueda tomar valores nulos o existentes en alguna tupla de *Ubicación*.

FORMA NORMAL DE BOYCE-CODD. FNBC

Las relaciones que satisfacen la FNBC satisfacen la FN2 y FN3. Esta forma normal se basa en el concepto de **Determinante funcional** (uno o varios atributos de una relación *R* del cual depende funcionalmente de forma completa algún otro atributo de la misma relación)

La FN de Boyce-Codd se expresaría como: Una relación *R* que se encuentra en FN1, y cada determinante funcional es una clave candidata de la relación *R*.



EL PROCESO DE DESCOMPOSICIÓN

El proceso de descomponer una relación no es un proceso trivial y debe ser realizado siguiendo los siguientes principios:

- **Descomposición por aplicación de la FN2:** Dada una relación *R* cuya clave es un agregado de la forma (*R.x*, *R.y*), y en esta relación existe una dependencia funcional de la forma *R.y* → *R.z*, donde *R.z* es un atributo no primo de la relación *R*:

- Eliminamos de *R* el atributo *R.z*, quedando igual el resto.
- Construimos *R₁*, cuya intención es:
 - *R.z* es atributo no primo de *R₁*
 - *R.y* es clave primaria de *R₁*
- Definimos *R.y* como clave foránea de la relación *R₁* (*R₁.y*)

Luego la descomposición se debe hacer en base a la dependencia funcional incompleta, y nunca por cualquier otra dependencia entre atributos de la relación. En este proceso tanto *R.x*, *R.y*, *R.z* pueden ser atributos simples o agregados de datos.

- **Descomposición por aplicación de la FN3:** Dada una relación *R* de clave *R.x* y dos atributos no primos *R.y* y *R.z* y en esta relación existen las dependencias funcionales *R.x* → *R.y*, *R.y* → *R.z* y por tanto la dependencia transitiva *R.x* → *R.z*, realizamos el siguiente proceso de descomposición:

- De *R* eliminamos el atributo que mantiene una dependencia funcional transitiva con la clave de la relación *R.z* y dejando el resto de la relación igual.



- Construimos R_1 , cuya intención es:
 - $R.z$ que mantenía una dependencia transitiva.
 - $R.y$ con el cual $R.z$ mantenía una dependencia funcional completa
 - La clave de R_1 será $R.y$ y en la relación sólo existirá una dependencia funcional completa de la forma $R.y \rightarrow R.z$
- En R se define $R.y$ como clave foránea de R_1 .

Luego la descomposición que se debe realizar por aplicación de la FN3 se debe hacer basándose en la dependencia funcional transitiva y nunca por cualquier otra dependencia entre los atributos de la relación.

- **Descomposición por aplicación de la FNBC:**

Dada una relación R en la cual están presentes uno o más determinantes funcionales $R.x, R.y, \dots, R.z$ formados por un agregado de datos de la forma $R.[i]=\{R.a, R.b, \dots, R.j\}$, y en esta relación existen dependencias funcionales distintas a dependencias completas de la forma $R.[i] \rightarrow R.n$, donde $R.n$ es cualquier atributo de la relación R , el proceso de descomposición se realizará de la forma siguiente:

- De la relación R se elimina $R.n$, quedando igual el resto de la intención.
- Construimos R_1 , cuya intención es:
 - $R.n$ es atributo no primo de R_1
 - El atributo $R.m$ es clave primaria, donde $R.m \in R.[i]$, $R.m \subseteq R.[i]$ está formado por un subconjunto de agregados de datos que forman parte del determinante funcional $R.[i]$, y por el que existe una dependencia funcional de la forma $R.m \rightarrow R.n$
- Se define $R..m$ como clave foránea de la clave de la relación R_1 .



Tema 4: El álgebra relacional