

Análisis de sistemas:
modelización conceptual de
datos. Técnicas descriptivas.
Documentación.

TEMA 50

ABACUS NT

Índice

- 1. Introducción**
- 2. Diseño conceptual: El modelo entidad/Relación**
 - 2.1. Elementos básicos**
 - 2.2. Entidades**
 - 2.3. Atributos, dominios y claves.**
 - 2.4. Relación**
 - 2.5. Grado de la relación**
 - 2.6. Cardinalidad de la relación (tipo de correspondencia)**
 - 2.7. Cardinalidades de las entidades**
 - 2.8. Entidades débiles vs fuertes**
- 3. Modelo E/R extendido**
 - 3.1. Cardinalidades de un tipo de entidad:**
 - 3.2. Relaciones exclusivas:**
 - 3.3. Dependencias débiles:**
 - 3.4. Generalización y herencia:**
 - 3.5. Relaciones ternarias,**
- 4. Reglas para la construcción de diagramas entidad-relación**
 - 4.1. Añadir tipos de objetos adicionales**
 - 4.2. Eliminar tipos de objetos**
- 5. Diccionario de datos**
 - 5.1. Extensiones al diccionario de datos para diagramas E-R**
- 6. Conclusión.**
 - 6.1. Relación con el sistema educativo**
- 7. Bibliografía**

1. Introducción

En el análisis de sistemas es fundamental la correcta modelización conceptual de los datos que se almacenan y se transforman en el sistema.

Partiendo de los documentos generados en la fase anterior, especialmente del diccionario de datos, se realizan una serie de esquemas que relacionan las distintas entidades entre sí.

En este capítulo se explora una notación gráfica para modelar datos. El **diagrama de entidad-relación** (también conocido como **DER**, o **diagrama E-R**) es un modelo de red que describe con un alto nivel de abstracción la distribución de datos almacenados en un sistema. Es muy diferente del DFD, que modela las funciones que lleva a cabo un sistema; y es diferente del diagrama de transición de estados, que modela el comportamiento dependiente del tiempo de un sistema.

Para el analista, el DER representa un gran beneficio: enfatiza las relaciones entre almacenes de datos en el DFD que de otra forma se hubieran visto sólo en la especificación de proceso. Cada una de las cajas rectangulares corresponde a un almacén de datos en un DFD, y puede verse que hay relaciones (conexiones) que normalmente no se aprecian en un DFD. Esto se debe a que el DFD enfoca la atención del lector a las funciones que el sistema efectúa, no a los datos que ocupa.

2. Diseño conceptual: El modelo entidad/Relación

El modelo E/R fue propuesto por **Peter P. Chen**, en los años 70. Presenta una vista unificada de los datos, centrándose en la estructura lógica y abstracta de los mismos, como representación del mundo real.

Inicialmente (en la propuesta de Chen) sólo se incluían los conceptos de entidad, relación y atributos. Después se añadieron otras propuestas (atributos compuestos, generalizaciones...) que forman el llamado **modelo entidad relación extendido**.

2.1. Elementos básicos

Los elementos con los que se construye un diagrama E/R son:

Entidad: Es un objeto del que nos interesa almacenar información, por ejemplo, puede ser una persona. Se representa mediante un rectángulo.

Relaciones: Se define como la asociación entre dos o más entidades. Se representa con un líneas, rombo, e indicando el **tipo de correspondencia**, la cual puede ser 1:1 o 1:N, por ejemplo.

Atributos: Es cada una de las propiedades que tiene un tipo de entidad o relación, por ejemplo, nombre, o precio. Se representa mediante una elipse o un círculo. Aquellos que son utilizados para identificar únicamente las instancias de una entidad (las claves) se representan sombreados o subrayados.

2.2. Entidades

Una entidad, representa cualquier persona, suceso, evento o concepto **sobre el que queramos almacenar información**.

La **representación gráfica** de un tipo de entidad es un rectángulo etiquetado con el nombre del tipo de entidad:



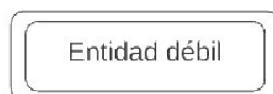
Una entidad cumple las siguientes propiedades según Tardieu:

1. Tiene existencia propia. Es decir, desde el punto de vista en el cual se estudia el sistema y al nivel de abstracción en que es considerado, la entidad existe como un elemento que interviene en el comportamiento global del sistema.
2. Es diferente del resto de las entidades (objetos) que intervienen en el sistema de información.
3. Todas las ocurrencias de un tipo de entidad deben tener los mismos tipos de características (atributos).

Las entidades las podemos clasificar en:

- **Entidades Fuertes, o regulares:** tienen existencia por sí mismas.
- **Entidades Débiles:** Su existencia depende de la existencia de otra entidad.

Las entidades débiles se representan mediante dos rectángulos concéntricos con su nombre en el interior:



2.3. Atributos, dominios y claves.

Cada una de las propiedades o características que tiene un tipo de entidad o un tipo de relación se denomina **atributo**. La representación gráfica de un atributo consiste en una elipse con el nombre del atributo en su interior.



Otra forma de representarlo, puede ser mediante un círculo y una etiqueta:



El conjunto de posibles valores que puede tomar un atributo recibe el nombre de dominio. El **dominio** tiene un nombre y una existencia propia con independencia de cualquier entidad o atributo.

Del conjunto de los atributos de una entidad, debemos elegir uno (o varios) que actúen como clave primaria. Estos atributos se representarán siempre con el nombre subrayado:

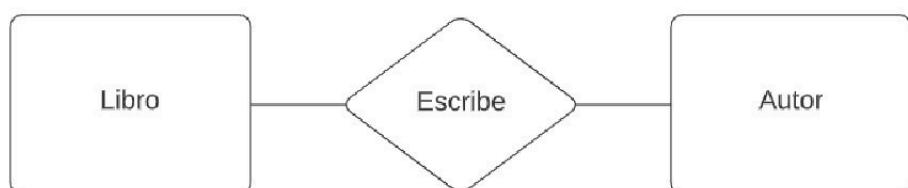


2.4. Relación

Se entiende por relación aquella *asociación o correspondencia existente entre entidades*.

Se llama tipo de relación a la estructura genérica del conjunto de relaciones existentes entre dos o más tipos de entidad.

El tipo de relación se representa mediante un rombo etiquetado con el nombre de la misma, unido mediante arcos a los tipos de entidad que asocia.



Para definir una relación debemos tener en cuenta los siguientes elementos:

Nombre de la relación: Nombre de la relación: cada relación tiene un nombre que la distingue claramente del resto y mediante el cual ha de ser referenciada. Habitualmente se utiliza un verbo en forma singular

Grado de la relación: Es el número de entidades que participan en una relación.

Cardinalidad de la relación: Es el número máximo de instancias de cada entidad que pueden intervenir.

Cardinalidades de las entidades: para el modelo E/R extendido.

Veamos con detalle, estas propiedades y alguna más:

2.5. Grado de la relación

Es el número de entidades que participan en una relación. Normalmente las relaciones son **binarias**, pero también podrían ser **ternarias** o de n miembros (**n-arias**). Existe un grado especial que el de las relaciones **unarias**, que se utiliza para relaciones reflexivas de una entidad consigo misma.

2.6. Cardinalidad de la relación (tipo de correspondencia)

Es el número máximo de instancias de cada entidad que pueden intervenir. También recibe el nombre de “*tipo de correspondencia*”. En la representación gráfica aparece como una etiqueta con 1:1, 1:n, n:1 o n:m, que se leen respectivamente como uno a uno, uno a muchos, muchos a uno y muchos a muchos.

Cardinalidad 1:1 (uno a uno): cuando en la interrelación sólo puede aparecer, como máximo, una ocurrencia del tipo de entidad por cada ocurrencia del otro.

Cardinalidad 1:n (uno a muchos): si para uno de los tipos de entidad puede haber un número indefinido (mayor que uno) de ocurrencias

Cardinalidad n:m (muchos a muchos): Si para cada ocurrencia de un tipo de entidad hay un número indefinido y mayor que uno de ocurrencias, y viceversa para el otro tipo de entidad.

2.7. Cardinalidades de las entidades

Esta propiedad se corresponde al modelo E/R extendido como veremos más adelante.

2.8. Entidades débiles vs fuertes

Al igual que las entidades, las relaciones se clasifican también en:

Fuertes. Asocian dos entidades fuertes.

Débiles. Asocian una entidad débil con otra fuerte.

3. Modelo E/R extendido

Varios autores han realizado diversas extensiones al modelo E/R definido por Chen, profundizando en aspectos que aportan aún más significado y relevancia a esta herramienta para el diseño conceptual de la BD.

El potencial expresivo del modelo E/R ha sido ampliado con nuevos elementos tales como:

3.1. Cardinalidades de un tipo de entidad:

Se define como el número máximo y mínimo de ocurrencias de una entidad que participan en el tipo de relación.

Las cardinalidades de las entidades se definen como el número máximo y mínimo de instancias de una entidad que pueden estar relacionadas con una instancia de otra u otras entidades que participan en la relación. Su representación gráfica es una etiqueta del tipo (0,1), (1,1), (0,N) o (1,N), según corresponda. El significado de cada una es:

(0,1) Para una ocurrencia determinada de los otros tipos de entidades que participan en la relación, el valor mínimo de ocurrencias de la entidad que se trata es 0, y el número máximo es 1

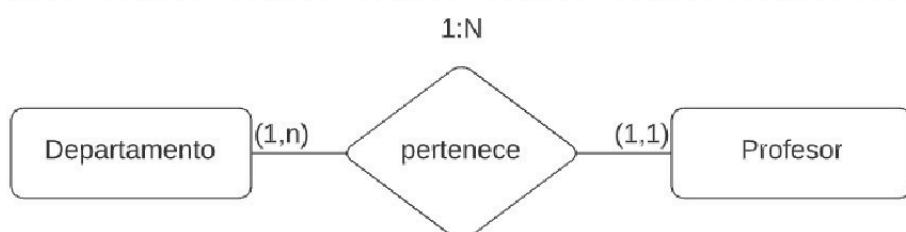
(1,1) Para una ocurrencia determinada de los otros tipos de entidades que participan en la relación, debe existir **una y sólo una** ocurrencia de la entidad que se trata.

(0,n) Para una ocurrencia determinada de los otros tipos de entidades que participan en la relación, el valor mínimo de ocurrencias de la entidad que se trata es 0, y el número máximo es n.

(1,n) Para una ocurrencia determinada de los otros tipos de entidades que participan en la relación, debe existir **como mínimo una** ocurrencia de la entidad que se trata, y el número máximo es n.

Aunque en este momento parezca que los conceptos "cardinalidad de una relación" y "cardinalidad de una entidad" son muy similares, su existencia se hace necesaria para la transformación del diagrama E/R al modelo relacional.

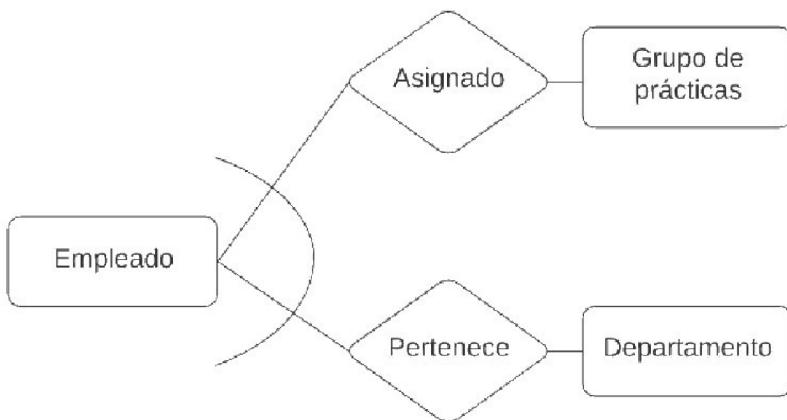
Ejemplo:



En el ejemplo, un profesor pertenece a un único departamento, mientras que un departamento está formado por mínimo un profesor y máximo n. La relación "pertenece" es de uno a muchos.

3.2. Relaciones exclusivas:

Cuando cada ocurrencia de un tipo de entidad sólo puede pertenecer a un tipo de relación y sólo a uno simultáneamente. Esto se representa mediante un semicírculo:



3.3. Dependencias débiles:

Dependencia en existencia: Se produce cuando una entidad (entidad débil) no pueden existir sin la ocurrencia de la entidad de la que dependen (entidad regular).

Dependencia en identificación: Se da cuando, además de la dependencia en existencia, las ocurrencias del tipo de entidad débil necesitan añadir la clave de la ocurrencia de la entidad regular para identificarse.

3.4. Generalización y herencia:

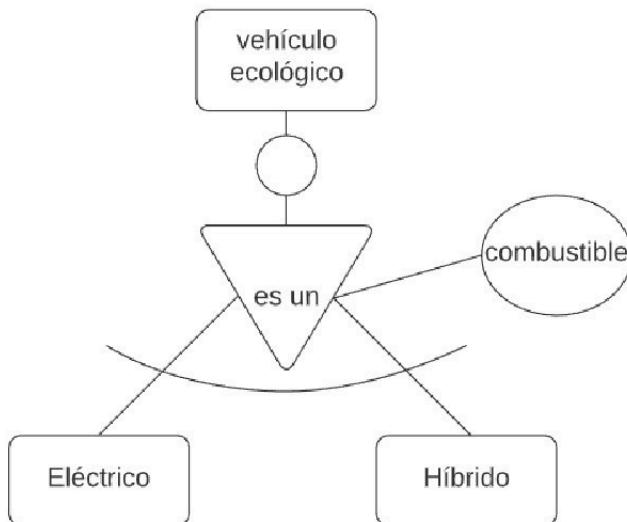
Relación que existe entre un tipo de entidad (supertipo) y los tipos de entidad más específicos (subtipos), en donde los atributos del supertipo son heredados por los subtipos.

La subdivisión es una necesidad bastante habitual en las BD. Esta relación especial se representa mediante un triángulo invertido con la base paralela al rectángulo que representa al supertipo.

Además, será necesario tener en cuenta y representar:

- El atributo del supertipo que no es heredado, será el que haga de discriminante, y se representa como un **atributo** de la generalización.
- Si los subtipos son de tipo exclusivo (pueden pertenecer a un subtipo u otro, pero no a los dos al la vez) se representan mediante un **arco**.
- Si el toda entidad del supertipo debe obligatoriamente ser de uno de los subtipos, esto se representa mediante un círculo o **elipse vacías**,

tal como se aprecia en la imagen:

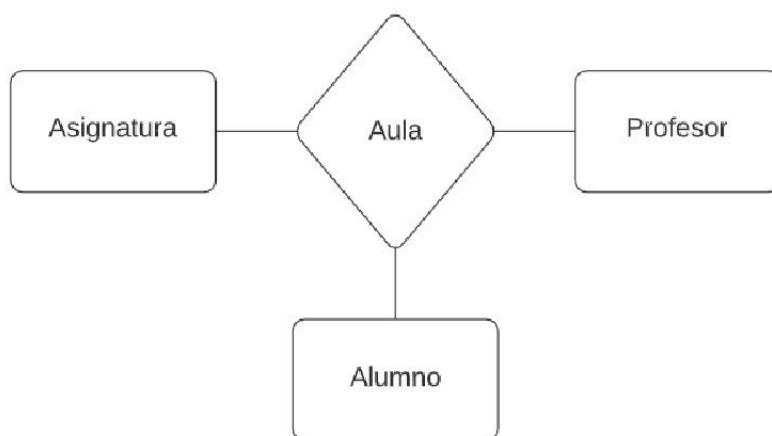


Siempre se verifica que toda ocurrencia de un subtipo es ocurrencia del supertipo: esto implica que las cardinalidades serán siempre (1,1) en el supertipo y (0,1) o (1,1) en el subtipo.

3.5. Relaciones ternarias,

O de orden superior: además de las relaciones habituales entre dos entidades (binarias) pueden existir otras que se consideran en el modelo E/R extendido.

Las relaciones ternarias se pueden descomponer en tres relaciones binarias. En general una relación n-aria se puede descomponer en n relaciones binarias, siendo esta última forma mucho más clara y fácil de interpretar en un diagrama E/R.



4. Reglas para la construcción de diagramas entidad-relación

La notación que se muestra en la sección anterior es suficiente para construir diagramas E-R arbitrariamente complejos. Sin embargo, podría estar pensando en este momento: "¿Cómo descubrir qué son, para comenzar, los objetos y las relaciones?". El *modelo inicial de objetos y relaciones* usualmente se derivará de

- 1) su comprensión de la aplicación del usuario,
- 2) entrevistas con el usuario y
- 3) cualquier otro tipo de investigación y recolección de información que pueda usar.

No espere que el primer diagrama E-R que haga sea el final, que revisará con la comunidad usuaria o que entregará a los diseñadores del sistema. Como los diagramas de flujo de datos y todas las demás herramientas de modelado, los diagramas E-R deben revisarse y mejorarse muchas veces; la primera versión típicamente no será más que un borrador, y las versiones subsecuentes se producirán utilizando una serie de reglas de refinamiento que se presentan en esta sección. Algunas de las reglas de refinamiento llevan a la creación de tipos adicionales de objeto, mientras que otras llevarán a la eliminación de objetos y/o relaciones.

4.1. Añadir tipos de objetos adicionales

Como se indicó anteriormente, el primer diagrama E-R típicamente se creará a partir de entrevistas iniciales con el usuario, y de su conocimiento de la materia en cuanto al negocio del usuario. Esto, desde luego, le dará una buena pista respecto a la identidad de los principales objetos y relaciones.

Después de haber desarrollado el primer DER, el siguiente paso es *asignar los datos del sistema a los diversos tipos de objetos*. Se supone, desde luego, que sabe cuáles son los datos. Esto puede suceder en cualquiera de tres maneras:

1. Si el modelo del proceso (el DFD) ya se ha desarrollado o se está desarrollando paralelamente al modelo de datos, entonces el diccionario de datos ya existirá. Pudiera no estar completo aún, pero lo que haya será suficiente para comenzar el proceso de asignación.
2. Si el modelo del proceso no se ha desarrollado (o en el caso extremo, si no tiene intención de desarrollar uno), entonces pudiera tener que empezar por entrevistar a todos los usuarios apropiados para construir una lista exhaustiva de datos (y sus definiciones). Al hacer esto, puede asignar los datos a los objetos en el diagrama de E-R. (Sin embargo, note que esto es un proceso ascendente que consume tiempo, y que pudiera ocasionar retrasos y frustración.).
3. Si está trabajando con un grupo activo de administración de datos, hay una buena probabilidad de que ya exista un diccionario de datos, que podría obtenerse pronto durante el proyecto, de manera que en ese momento ya pudiera comenzar el proceso de asignación.

El proceso de asignación puede ofrecer una de tres *razones para crear nuevos tipos de objetos*:

1. Es posible descubrir datos que se pueden asignar a algunas instancias de un tipo de objeto, pero no a otras.
2. Pudieran descubrirse datos aplicables a todas las instancias de dos objetos distintos.
3. Podría descubrirse que algunos datos describen relaciones entre otros tipos de objetos.

Si durante el proceso de asignar datos a tipos de objetos encuentra que algunos datos no se pueden aplicar a todas las instancias de algún tipo de objeto dado, necesitará crear un conjunto de **subtipos** bajo el tipo de objeto con el que ha estado trabajando, y asignar los datos específicos a los subtipos apropiados.

Suponga que, por ejemplo, se está desarrollando un sistema de personal, y se ha identificado (con gran perspicacia y creatividad) un tipo de objeto llamado empleado. Al revisar los datos disponibles se encuentra que muchos de ellos (edad, estatura, fecha de contratación, etc.) se aplican a todas las instancias de un empleado. Sin embargo, luego se descubre un dato llamado número-de-embarazos; se trata obviamente de un dato relevante para las empleadas, pero no para los empleados varones. Esto llevaría a crear empleados-varones y empleadas como subtipos de la categoría general de empleado.

Obviamente, no se está sugiriendo que todos los sistemas de personal deban hacer seguimiento del número de embarazos que cada empleada ha tenido; el ejemplo se escogió meramente porque existe un consenso general de que los empleados varones no se pueden embarazar. Compare esto, sin embargo, con el dato nombre-del-cónyuge: hay varias instancias de empleado para quienes no se aplicaría esto (porque son solteros), pero es una situación muy distinta a la de un dato que no se puede aplicar definitivamente.

En la mayoría de los casos el proceso de crear nuevos subtipos y asignarles datos de manera apropiada es bastante directo. Sin embargo, debe tenerse siempre en mente una situación excepcional: pudiera suceder que todos los datos relevantes se atribuyan a uno de los subtipos, y que ninguno de los datos se pueda asignar al objeto supertipo; es decir, puede suceder que los datos sean mutuamente excluyentes, perteneciendo a un subtipo o a otro, pero no a ambos. Suponga, por ejemplo, que los únicos datos que se puede asignar a los empleados son número-de-embarazos y años-de-experiencia-jugando-con-el-equipo-de-baloncesto. Podría decidirse (tras preguntarnos a qué lunático usuario pudiera habérsele ocurrido tal sistema) que el supertipo general empleado no se aplica.

También puede ocurrir la situación inversa: los datos pueden describir instancias de dos (o más) tipos distintos de objetos de la misma manera. Si esto ocurre, debe crearse un **supertipo** nuevo y asignarle los datos comunes al supertipo. Por ejemplo, tal vez se identificó cliente-de-contado y cliente-a-crédito como dos tipos de objetos distintos cuando se creó el DER para un sistema de pedidos (tal vez porque el usuario señaló que eran dos categorías distintas). Sin embargo, pronto se hace evidente que los datos nombre-del-cliente y domicilio-del-cliente describen ambos tipos de cliente de la misma forma, lo cual apoyaría la creación de un supertipo.

Similarmente, si un dato describe la interacción de dos o más tipos de objetos, entonces debería reemplazarse la relación "desnuda" entre los dos objetos con un tipo asociativo de objeto. Por ejemplo, en el primer borrador de DER podría existir una relación de compra entre cliente y artículo. Durante la asignación de datos pudiera encontrarse con que hay un dato llamado fecha-de-compra que 1) parece pertenecer a la relación compra y 2) obviamente describe, o proporciona datos acerca de la interacción de un cliente con un artículo. Esto sugiere que debe sustituirse la relación compra por un tipo asociativo de objeto.

A veces el diagrama E-R inicial tendrá un tipo de objeto que, visto de cerca, claramente parece ser un tipo asociativo de objeto.

Este proceso de eliminar objetos incluidos en otros es parte de una actividad de refinamiento más general llamada normalización. El objetivo de la normalización es producir tipos de objetos, en los que cada instancia (o miembro) consiste en un valor llave primaria que identifica a alguna entidad, junto con un conjunto de valores de atributo independientes que describen a la entidad de alguna manera.

4.2. Eliminar tipos de objetos

La sección anterior trató los refinamientos del DER que crean objetos y/o relaciones adicionales. Sin embargo, existe un buen número de situaciones en las que los refinamientos del DER llevan a la eliminación de tipos de objetos y relaciones redundantes o erróneas. Examinaremos cuatro situaciones comunes:

1. Tipos de objetos que consisten sólo en un identificador.
2. Tipos de objetos para los cuales existe una sola instancia.
3. Tipos asociativos de objetos flotantes.
4. Relaciones derivadas.

Si se tiene un diagrama E-R en el cual uno de los tipos de objeto tiene sólo un identificador asignado como dato, existe la oportunidad de eliminar el tipo de objeto y asignar el identificador, como dato, a un tipo de objeto relacionado.

5. Diccionario de datos

Debe existir una verdadera arquitectura de datos que facilite la integración. El núcleo de dicha arquitectura será el **diccionario de recursos de información (iRD)**, que contendrá las descripciones de todos los datos que constituyen el sistema de información.

Las finalidades principales de los DRI son precisamente ayudar a la gestión de la información como recurso y conseguir la integración de la semántica de las aplicaciones de forma centralizada, con lo que se aumentan los beneficios estratégicos, operacionales y de control de las empresas.

Para Codd (1990), uno de los requisitos para que un SGBD pueda considerarse como verdaderamente relacional es que soporte un catálogo dinámico en línea en el que la descripción de la base de datos se represente como cualquier otro dato, permitiendo a los usuarios autorizados aplicar el mismo lenguaje relacional tanto a la descripción de la base de datos como a los datos regulares.

Los recursos informáticos de las instituciones se gestionan almacenando, administrando y controlando los denominados **metadatos**, esto es, los datos que definen y describen los datos de la institución. El término más difundido para estos metadatos es el de **Diccionario de Datos (DD)** o catálogo del sistema.

El directorio de datos ha sido desde siempre un componente del SGBD, encargado de describir dónde y cómo se almacenan los datos de la base, el modo de acceso y otras características físicas de los mismos, atendiendo de este modo las peticiones de los programas. Contiene, en definitiva, las especificaciones necesarias para pasar de la representación externa de los datos a la representación interna de los mismos. Su objetivo principal es transmitir al SGBD la información necesaria para poder acceder a los datos contenidos en la base.

Cuando los SGBD eran más limitados, los usuarios empleaban una lista "manual" donde anotaban la descripción narrativa y técnica (tamaño, tipo...) de los datos, así como los permisos de acceso a cada uno de ellos, el uso que las aplicaciones hacían de ellos, etc. Esta lista se denominaba diccionario de datos. Consistía en información sobre los datos, o, en otras palabras, datos sobre los datos, también conocidos como metadatos.

Por tanto, el diccionario de datos se considera como un conjunto de metadatos (datos sobre los datos) que contiene las características lógicas de los datos que se van a utilizar en el sistema que se programa, incluyendo nombre, descripción, alias, contenido y organización.

Si el diccionario de datos lo usan sólo los administradores, usuarios y diseñadores, y no el software del SGBD.

Por último, a partir de los 80, el diccionario de datos se incorporó a los SGBD como un componente más e integrándose junto al directorio. Al conjunto se le suene denominar catálogo o repositorio. El catálogo es el responsable de establecer la correspondencia entre los tres niveles de abstracción. Los módulos del SGBD usan y leen el catálogo con mucha frecuencia; por ello es importante implementar su acceso de la forma más eficiente posible.

Con la amplia difusión del modelo relacional el catálogo se almacena junto al resto de los datos de la base, en forma de tablas. La información almacenada en el catálogo de un SGBD relacional incluye, entre otros aspectos:

- Descripciones de los nombres de las relaciones (tablas).
- Nombres de los atributos.
- Tipos de datos de los atributos (dominios).
- Claves primarias.
- Claves ajenas.
- Obligación de establecer los datos como no nulos en algunas columnas.
- Vistas.

- Índices.
- Estructuras de almacenamiento.
- Permisos.
- Información sobre el propietario de cada tabla.

Dado que los datos del catálogo se almacenan en forma de tablas, se puede acceder a los mismos mediante instrucciones de SQL.

Directorio de datos: Es un subsistema del SGBD, encargado de describir dónde y cómo se almacenan los datos de la base, el modo de acceso y otras características físicas de los datos, atendiendo de este modo las peticiones de los programas y procesos. Un directorio de datos contiene, en definitiva, las especificaciones necesarias para pasar de la representación externa de los datos a su representación interna. Ha de estar siempre en un formato legible para la máquina. Su objetivo principal es transmitir al sistema la información necesaria para poder acceder a los datos contenidos en la base. El directorio, a diferencia del diccionario, es un instrumento del SGBD, y está orientado a facilitar a éste la información que necesita para su funcionamiento.

Un desarrollo importante en los SGBD relacionales es la práctica común de almacenar el directorio como un conjunto de relaciones. Esto permite el uso de los lenguajes de manipulación de datos de los SGBD para consultar, actualizar y mantener el diccionario de datos.

Recientemente, ha aparecido un nuevo concepto, el **diccionario de recursos de información (dri)**, que pretende ser el eslabón final en la evolución de los almacenes de datos. El DRI constituye el depósito integrado de todos los datos sobre la organización, automatizados o no, que son utilizados para efectuar las labores de planificación, control y operación que permitan a la empresa cumplir sus objetivos. Éstos engloban de algún modo las capacidades y funciones de todos los almacenes de datos anteriores.

5.1. Extensiones al diccionario de datos para diagramas E-R

Finalmente, observamos que el diccionario de datos necesita extenderse para tomar en cuenta la notación de Diagrama E-R. En general, los objetos del DER corresponden con almacenes del DFD. Esto significa que en la definición sacada del diccionario de datos que se da a continuación, cliente es tanto definición del tipo de objeto como instancia del almacén clientes.

CLIENTES = {CLIENTE}

Cliente = @nombre-del-cliente + domicilio + número-telefónico

Nótese también que la definición de un cliente incluye la especificación del campo llave, que es el dato (o atributo) que diferencia una instancia de un cliente de cualquier otra. El signo de arriba (@) se utiliza para indicar el o los campos llave.

Sin embargo, también hay que incluir en el diccionario de datos una definición de todas las relaciones que se muestran en el DER. La definición de relación debe incluir una descripción de su significado en el contexto de la aplicación; y debe indicar los objetos que forman la asociación. Los límites superiores e inferiores apropiados deben especificarse para indicar si la asociación es de *uno a uno*, *de uno a muchos* o *de muchos a muchos*. Por ejemplo, la relación compras que se muestra en la figura 50.4. puede definirse en el diccionario de datos de la siguiente forma:

compras = *la asociación de un cliente y uno o más artículos*

@identidad-del-cliente + {@identidad-del-artículo + cantidad-comprada}

6. Conclusión.

En este tema hemos tratado el análisis y la modelización del sistema desde el enfoque de desarrollo orientado a grandes volúmenes de datos.

Existen por supuesto otras visiones, y todas son válidas, ya que cada tipo de desarrollo implica unas características y herramientas particulares.

6.1. Relación con el sistema educativo

- GS – DAW - DAM –Entornos de Desarrollo

7. Bibliografía

- Pressman, R. S. **Ingeniería del Software. Un enfoque práctico**, 3^a edición. Ed. McGraw-Hill, 2000.
- Sommerville, I.: **Ingeniería de Software**. 6^a Edición. Addison-Wesley Iberoamericana, 2002.
- Piattini, M. y otros: **Análisis y diseño detallado de Aplicaciones Informáticas de Gestión**. RA-MA, 1996.
- Cerrada, J. A.: **Introducción a la Ingeniería del Software**. 1^a Edición de 2000. Editorial Centro de Estudios Ramón Areces, S.A.

