

Bases de Datos I

18/09/2020

Resumen Cap 3 18/09

IS-501

José Inestroza

Modelado de datos usando la entidad, Modelo de relación (ER)

Generalmente, el término aplicación de base de datos se refiere a una base de datos en particular y los programas asociados que implementan las consultas y actualizaciones de la base de datos. Estos programas proporcionarían interfaces gráficas de usuario (GUI) fáciles de usar utilizando formularios y menús para los usuarios finales de la aplicación. Por tanto, una parte importante de la aplicación de la base de datos requerirá diseño, implementación y prueba de estos programas de aplicación. Tradicionalmente, el diseño y la prueba de programas de aplicación se ha considerado parte de ingeniería de software en lugar de diseño de bases de datos. El diseño de programas de aplicación generalmente se cubre en cursos de ingeniería de software. Nosotros presentamos los conceptos de modelado del modelo entidad-relación (ER), que es un popular modelo de datos conceptual de alto nivel. Este modelo y sus variaciones se utilizan con frecuencia para el diseño conceptual de aplicaciones de bases de datos y muchas herramientas de diseño de bases de datos emplean sus conceptos.

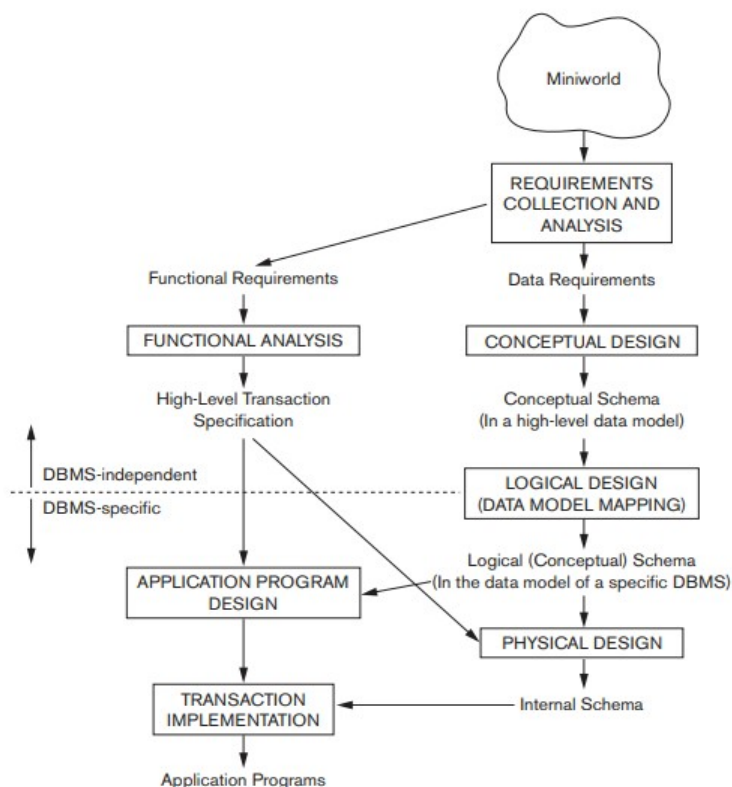
Metodologías de modelado de objetos como el lenguaje de modelado unificado (UML)

son cada vez más populares en el diseño de bases de datos y software. En los diagramas de clases, se especifican operaciones sobre objetos, además especificando la estructura del esquema de la base de datos. Las operaciones se pueden utilizar para especificar los requisitos funcionales durante el diseño de la base de datos.

Uso de modelos de datos conceptuales de alto nivel para el diseño de bases de datos

El primer paso se muestra la recopilación y el análisis de requisitos. Durante este paso, la base de datos Los diseñadores entrevistan a posibles usuarios de bases de datos para comprender y documentar sus requerimientos de datos. El resultado de este paso es un conjunto de requisitos de los usuarios escrito concisamente. Estos requisitos deben especificarse en un formulario detallado y completo. como sea posible. Paralelamente a la especificación de los requisitos de datos, es útil especificar los requisitos funcionales conocidos de la aplicación. Estos consisten en las operaciones (o transacciones) definidas por el usuario que se aplicarán a la base de datos, incluyendo tanto recuperaciones como actualizaciones. Una vez recopilados y analizados los requisitos, el siguiente paso es crear un esquema conceptual para la base de datos, utilizando un modelo de datos conceptual de alto nivel.

Este paso se llama diseño conceptual. El esquema conceptual es una descripción concisa de los requisitos de datos de los usuarios e incluye descripciones detalladas de la entidad tipos, relaciones y limitaciones; estos se expresan utilizando los conceptos proporcionados por el modelo de datos de alto nivel. El esquema conceptual de alto nivel también puede utilizarse como referencia para garantizar que se cumplan todos los requisitos de datos de los usuarios y que los requisitos no entran en conflicto. Este enfoque permite a los diseñadores de bases de datos concentrarse en especificar las propiedades de los datos, sin preocuparse por detalles de almacenamiento e implementación, lo que facilita la creación de un buen diseño de base de datos conceptual. Durante o después del diseño del esquema conceptual, las operaciones



básicas del modelo de datos pueden utilizarse para especificar las consultas y operaciones de usuario de alto nivel identificadas durante análisis funcional. Esto también sirve para confirmar que el esquema conceptual cumple todos los requisitos funcionales identificados. Modificaciones al esquema conceptual puede introducirse si algunos requisitos funcionales no se pueden especificar utilizando el esquema inicial. El siguiente paso en el diseño de la base de datos es la implementación real la base de datos, utilizando un DBMS comercial. La mayoría

de

de los DBMS comerciales actuales utilizan una implementación modelo de datos, como el modelo relacional (SQL), por lo que el esquema conceptual es transformado del modelo de datos de alto nivel al modelo de datos de implementación. Este paso se denomina diseño lógico o mapeo del modelo de datos; su resultado es una base de datos esquema en el modelo de datos de implementación del DBMS.

El mapeo del modelo de datos es a menudo automatizado o semiautomatizado dentro de las herramientas de diseño de bases de datos. El último paso es la fase de diseño físico, durante la cual las estructuras de almacenamiento interno, las organizaciones de archivos, los índices, las rutas de acceso y los parámetros de diseño físico para se especifican los archivos de la base de datos. Paralelamente a estas actividades, se diseñan e implementan programas de aplicación como transacciones de base de datos correspondientes a las especificaciones de transacciones de alto nivel.

Una aplicación de base de datos de muestra

Aplicación de base de datos de muestra, llamada EMPRESA, La base de datos de la EMPRESA realiza un seguimiento de los empleados, departamentos y proyectos. Suponga que después de la fase de recopilación y análisis de requisitos, Los diseñadores de bases de datos proporcionan la siguiente descripción del minimundo: la parte de la empresa que estará representada en la base de datos. La empresa está organizada en departamentos. Cada departamento tiene un nombre, un número único y un empleado en particular que administra el departamento. Llevamos un registro de la fecha de inicio cuando ese empleado comenzó a administrar el Departamento. Un departamento puede tener varias ubicaciones.

* Un departamento controla una serie de proyectos, cada uno de los cuales tiene un nombre, un número único y una única ubicación.

* La base de datos almacenará el nombre de cada empleado, el número de seguro social, dirección, salario, sexo (género) y fecha de nacimiento. Un empleado está asignado a uno departamento, pero puede trabajar en varios proyectos, que no son necesariamente controlado por el mismo departamento. Se requiere realizar un seguimiento del número actual de horas por semana que un empleado trabaja en cada proyecto, como así como el supervisor directo de cada empleado (que es otro empleado).

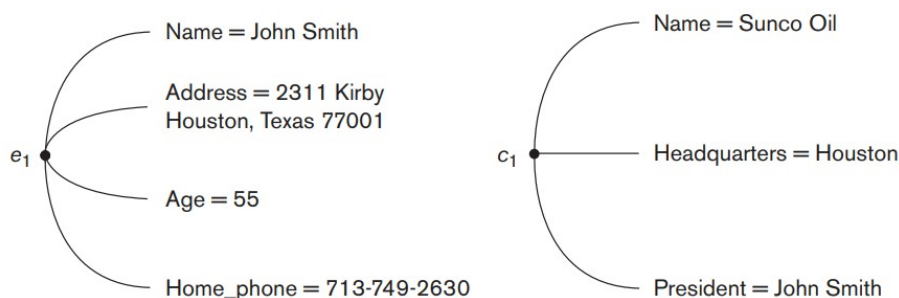
* La base de datos hará un seguimiento de los dependientes de cada empleado a efectos del seguro, incluido el nombre, sexo, fecha de nacimiento y relación con el empleado.

Tipos de entidad, conjuntos de entidades, atributos, y llaves

El modelo ER describe los datos como entidades, relaciones y atributos.

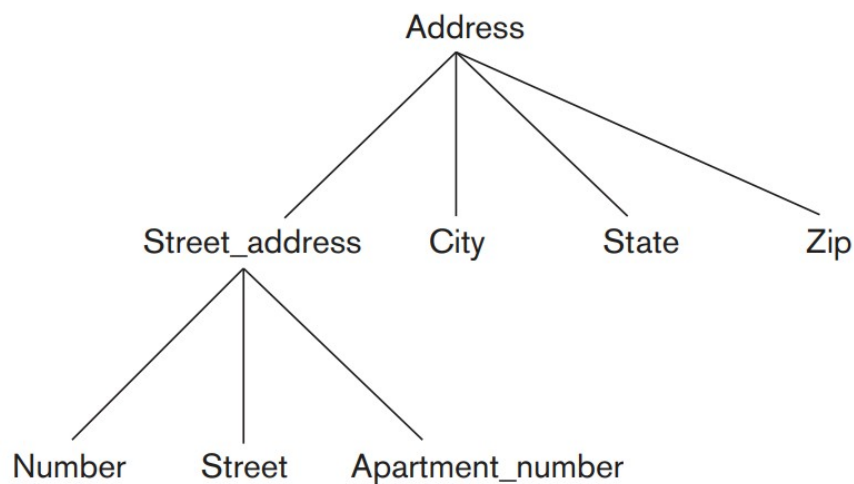
Entidades y atributos

El concepto básico que representa el modelo ER es una entidad, que es una cosa u objeto en el mundo real con una existencia independiente. Una entidad puede ser un objeto con existencia física (por ejemplo, una persona, automóvil, casa o empleado en particular) o puede ser un objeto con existencia conceptual (por ejemplo, una empresa, un trabajo o un curso universitario). Cada entidad tiene atributos: las propiedades particulares que lo describen. Por ejemplo, una entidad EMPLEADO puede ser descrita por el nombre, edad, dirección, salario y trabajo del empleado. Una entidad en particular tendrá un valor para cada uno de sus atributos. Los valores de atributo que describen cada una entidad se convierte en una parte importante de los datos almacenados en la base de datos. El empleado la entidad e1 tiene cuatro atributos: nombre, dirección, edad y teléfono residencial; sus valores son "John Smith", "2311 Kirby, Houston, Texas 77001", "55" y "713-749-2630", respectivamente. La entidad EMPRESA c1 tiene tres atributos: Nombre, Sede y Presidente; sus valores son "Sunco Oil", "Houston" y "John Smith", respectivamente. En el modelo ER se producen varios tipos de atributos: simples frente a compuestos, de valor único frente a valores múltiples y almacenados frente a derivados. Primero definimos estos atributos tipos e ilustrar su uso mediante ejemplos.



Atributos compuestos versus simples (atómicos).

Los atributos compuestos pueden ser divididos en subpartes más pequeñas, que representan atributos más básicos con significados independientes. Atributos que no son divisibles se denominan atributos simples o atómicos. Los atributos compuestos pueden formar una jerarquía; por ejemplo, Street_address se puede subdividir en tres componentes simples atributos: Número, Calle y Número de apartamento. El valor de un atributo compuesto es la concatenación de los valores de su componente simple atributos. Los atributos compuestos son útiles para modelar situaciones en las que un usuario a veces se refiere al atributo compuesto como una unidad, pero en otras ocasiones se refiere específicamente a sus componentes. Si se hace referencia al atributo compuesto solo como un todo, no hay es necesario subdividirlo en atributos de componentes. Por ejemplo, si no es necesario referirse a los componentes individuales de una dirección (código postal, calle, etc.), luego la dirección completa se puede designar como un atributo simple.



Atributos de valor único frente a atributos de valor múltiple.

La mayoría de los atributos tienen un solo valor para una entidad en particular; tales atributos se denominan de valor único. En algunos casos, un atributo puede tener un conjunto de valores para la misma entidad, por ejemplo, un atributo de colores para un automóvil o un Atributo College_degrees para una persona. Los coches de un color tienen un valor único, mientras que los coches de dos tonos tienen dos valores de color. Del mismo modo, una persona puede no tener ningún título universitario, otra persona puede tener uno y una tercera persona puede tener dos o más grados; por lo tanto, diferentes personas pueden tener diferentes números de valores para el Atributo College_degrees. Estos atributos se denominan multivalor. El atributo puede tener límites inferior y superior para restringir el número de valores permitidos para cada entidad individual.

Atributos almacenados versus derivados.

En algunos casos, dos (o más) valores de atributo están relacionados, por ejemplo, los atributos Edad y Fecha de nacimiento de una persona. Para entidad persona en particular, el valor de Edad se puede determinar a partir de la (de hoy) y el valor de la fecha de nacimiento de esa persona. El atributo Edad es por tanto llamado atributo derivado y se dice que es derivable del atributo Birth_date, que se denomina atributo almacenado. Algunos valores de atributo pueden derivarse de entidades; por ejemplo, un atributo Number_of_employees de una entidad DEPARTMENT puede derivarse contando el número de empleados relacionados con (trabajando para) que Departamento.

Valores NULL.

En algunos casos, una entidad en particular puede no tener un valor aplicable para un atributo. Por ejemplo, se aplica el atributo Apartment_number de una dirección sólo a direcciones que se encuentran en edificios de apartamentos y no a otros tipos de residencias, como viviendas unifamiliares. Del mismo modo, se aplica un atributo College_degrees solo para personas con títulos universitarios. Para tales situaciones, un valor especial llamado NULL es creado. La dirección de una vivienda unifamiliar tendría NULL para su Apartment_number, y una persona sin título universitario habría NULL para College_degrees. NULL también se puede utilizar si no conocemos el valor de un atributo para una entidad en particular, por ejemplo, si no conocemos el teléfono de casa número de "John Smith" El significado del primer tipo de NULL es no aplicable, mientras que se desconoce el significado de este último. La categoría desconocida de NULL se puede

clasificar en dos casos. El primer caso surge cuando se conoce que el valor del atributo existe pero falta, por ejemplo, si el atributo Altura de una persona aparece como NULL. El segundo caso surge cuando no se sabe si el valor del atributo existe, por ejemplo, si el atributo Home_phone de una persona es NULL.

Atributos complejos.

Observe que, en general, los atributos compuestos y multivalor pueden anidarse arbitrariamente. Podemos representar un anidamiento arbitrario agrupando componentes de un atributo compuesto entre paréntesis () y separando los componentes con comas, y mostrando atributos multivalor entre llaves {}. Estos atributos se denominan atributos complejos. Por ejemplo, si una persona puede tener más de una residencia y cada residencia puede tener una sola dirección y varios teléfonos, se puede especificar un atributo Address_phone para una persona. Tanto el teléfono como la dirección son atributos compuestos en sí mismos.

Tipos de entidad, conjuntos de entidades, claves y conjuntos de valores

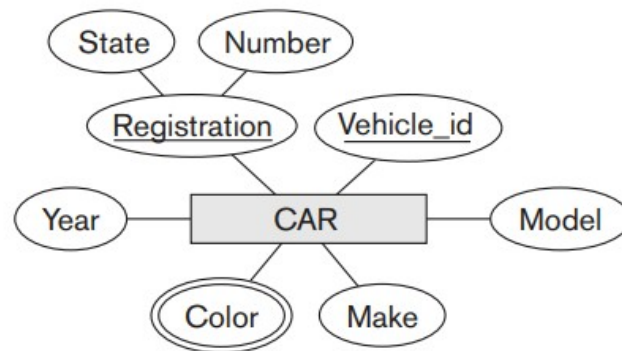
Tipos de entidad y conjuntos de entidades. Una base de datos generalmente contiene grupos de entidades que son similares. El tipo de entidad define una colección (o conjunto) de entidades que tienen los mismos atributos. Cada tipo de entidad en la base de datos se describe por su nombre y atributos. Muestra dos tipos de entidad: EMPLEADO y EMPRESA, y una lista de algunos de los atributos para cada. También se ilustran algunas entidades individuales de cada tipo, junto con valores de sus atributos. La colección de todas las entidades de un tipo de entidad en particular en la base de datos en cualquier momento se denomina conjunto de entidades o colección de entidades; la entidad establecida generalmente se hace referencia al uso del mismo nombre que el tipo de entidad, aunque son dos conceptos separados. Por ejemplo, EMPLEADO se refiere tanto a un tipo de entidad así como la colección actual de todas las entidades de empleados en la base de datos. Ahora es más común para dar nombres separados al tipo de entidad y colección de entidad; por ejemplo en modelos de datos relacionales de objetos y objetos. Un tipo de entidad se representa en diagramas ER5 como una caja rectangular adjuntando el nombre del tipo de entidad. Los nombres de los atributos están encerrados en óvalos y son adjunta a su tipo de entidad mediante líneas rectas. Los atributos compuestos se adjuntan a sus atributos componentes mediante líneas rectas. Los atributos multivalor se muestran en óvalos dobles. Un tipo de entidad describe el esquema o la intención de un conjunto de entidades que comparten la misma estructura. La colección de entidades de un tipo de entidad en particular se agrupa en un conjunto de entidades, que también se denomina extensión del tipo de entidad.

Atributos clave de un tipo de entidad.

Una restricción importante sobre las entidades de un tipo de entidad es la clave o restricción de unicidad de los atributos. Un tipo de entidad generalmente tiene uno o más atributos cuyos valores son distintos para cada entidad individual en el conjunto de entidades. Dicho atributo se denomina atributo clave y sus valores se pueden utilizar para identificar cada entidad de forma única. Por ejemplo, el atributo Name es una clave del Tipo de entidad EMPRESA porque no se permite que dos empresas tengan el mismo nombre. Para el tipo de entidad PERSON, un atributo clave típico es Ssn (número de seguro social). A veces, varios atributos juntos forman una clave, lo que significa que la combinación de los valores de los atributos debe ser distinta para cada entidad. Si un conjunto de atributos posee esta propiedad, la forma correcta de representar esto en el modelo ER que describimos aquí es definir un atributo compuesto y designarlo como un atributo clave del tipo de entidad. Tenga en cuenta que dicha clave compuesta debe ser mínima; eso es todo. Los atributos del componente deben incluirse en el atributo

compuesto para tener la propiedad de unicidad. Los atributos superfluos no deben incluirse en una clave. En ER notación esquemática, cada atributo clave tiene su nombre subrayado dentro del óvalo, como se ilustra en (a). Especificar que un atributo es una clave de un tipo de entidad significa que lo anterior. La propiedad de unicidad debe ser válida para cada conjunto de entidades del tipo de entidad. Por tanto, es una restricción que prohíbe que dos entidades tengan el mismo valor para la clave atributo al mismo tiempo. No es propiedad de un conjunto de entidades en particular; más bien, es una restricción sobre cualquier conjunto de entidades del tipo de entidad en cualquier momento. Esta restricción clave se deriva de las restricciones de la minimundo que representa la base de datos. Algunos tipos de entidad tienen más de un atributo clave. El atributo de registro es un ejemplo de una clave compuesta formada a partir de dos atributos de componentes simples, Estado y Número, ninguno de los cuales es una clave por sí mismo. Un tipo de entidad también puede no tener clave, en cuyo caso se denomina débil tipo de entidad. En nuestra notación diagramática, si dos atributos están subrayados por separado, entonces cada uno es una clave en sí misma. A diferencia del modelo relacional no existe un concepto de clave primaria en el modelo ER que presentamos aquí; la clave principal será elegida durante el mapeo a un esquema relacional.

(a)



Conjuntos de valores (dominios) de atributos.

Cada atributo simple de un tipo de entidad es asociado con un conjunto de valores (o dominio de valores), que especifica el conjunto de valores que puede asignarse a ese atributo para cada entidad individual si el rango de edades permitidas para los empleados es entre 16 y 70, podemos especificar el valor conjunto del atributo Edad del EMPLEADO como el conjunto de números enteros entre 16 y 70. De manera similar, podemos especificar que el valor establecido para el atributo Name sea el conjunto de cadenas de caracteres alfabéticos separados por caracteres en blanco, etc. Conjuntos de valores no se muestran normalmente en los diagramas ER básicos y son similares a los datos básicos tipos disponibles en la mayoría de los lenguajes de programación, como integer, string, booleano, flotante, tipo enumerado, subrango, etc. Sin embargo, los tipos de datos de atributos se pueden especificar en diagramas de clases UML y en otros diagramas notaciones utilizadas en herramientas de diseño de bases de datos. Tipos de datos adicionales para representar comunes También se emplean tipos de bases de datos, como fecha, hora y otros conceptos. Matemáticamente, un atributo A del conjunto de entidades E cuyo conjunto de valores es V se puede definir como una función de E al conjunto de potencia $P(V)$ de V:

$$A : E \rightarrow P(V)$$

Nos referimos al valor del atributo A para la entidad e como A (e). La definición anterior cubre atributos de valor único y multivalor, así como NULL. Un valor NULO está representado por el conjunto vacío. Para atributos de un solo valor, A (e) está restringido a siendo un conjunto singleton para cada entidad e en E, mientras que no hay restricción sobre atributos multivalor. Para un atributo compuesto A, el conjunto de valores V es el conjunto de potencias el producto cartesiano de P (V1), P (V2), . . . , P (Vn), donde V1, V2, . . . , Vn son los conjuntos de valores de los atributos de componentes simples que forman A:

$$V = P(P(V_1) \times P(V_2) \times \cdots \times P(V_n)) \quad \text{El}$$

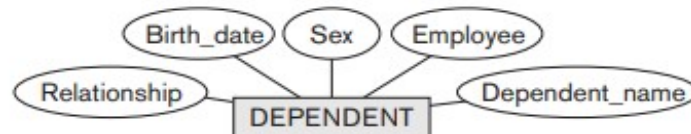
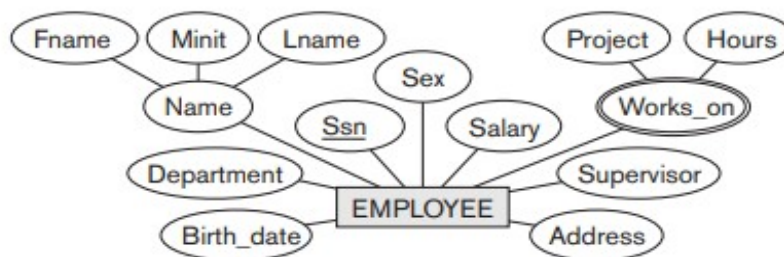
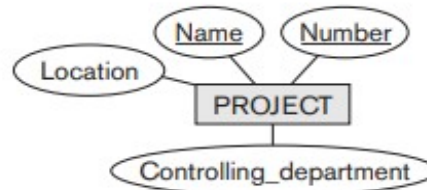
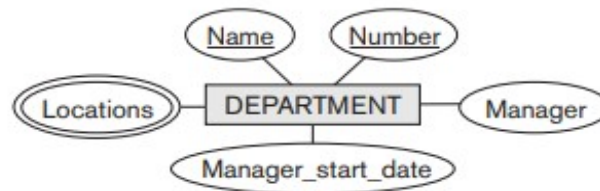
conjunto de valores proporciona todos los valores posibles. Por lo general, solo existe una pequeña cantidad de estos valores en la base de datos en un momento determinado. Esos valores representan los datos de el estado actual del mini mundo y corresponden a los datos tal como existen en realidad en el mini mundo.

Diseño conceptual inicial de la base de datos de la EMPRESA

Ahora podemos definir los tipos de entidad para la base de datos de EMPRESA, basados en los requisitos, podemos identificar cuatro tipos de entidades, una correspondiente a cada uno de los cuatro elementos de la especificación:

1. Un tipo de entidad DEPARTAMENTO con atributos Nombre, Número, Ubicaciones, Manager y Manager_start_date. Ubicaciones es el único atributo de varios valores. Podemos especificar que tanto el Nombre como el Número son atributos clave (separados) porque cada uno fue especificado para ser único.
2. Un tipo de entidad PROYECTO con atributos Nombre, Número, Ubicación y Controlling_department. Tanto el Nombre como el Número son atributos clave (separados).
3. Un tipo de entidad EMPLEADO con atributos Nombre, Ssn, Sexo, Dirección, Salario, Fecha de nacimiento, departamento y supervisor. Tanto el nombre como la dirección pueden ser atributos compuestos; sin embargo, esto no se especificó en los requisitos. Debemos volver a los usuarios para ver si alguno de ellos hace referencia al individuo.
4. Un tipo de entidad DEPENDIENTE con atributos Empleado, Nombre_dependiente, Sexo, Fecha de nacimiento y Relación (con el empleado) componentes de Nombre — Nombre, Inicial del segundo nombre, Apellido — o de Dirección. En nuestro ejemplo, el nombre se modela como un atributo compuesto, mientras que la dirección es no, presumiblemente después de consultar con los usuarios.

Otro requisito es que un empleado pueda trabajar en varios proyectos, y la base de datos tiene que almacenar el número de horas por semana que un empleado trabaja en cada proyecto. y se puede representar mediante un atributo compuesto multivalor de EMPLEADO llamado Works_on con los componentes simples (Proyecto, Horas). Alternativamente, se puede representar como un atributo compuesto multivalor de PROYECTO llamado trabajadores con los componentes simples (Empleado, Horas).



Tipos de conjuntos de Roles y estructurales

relaciones, relaciones, restricciones

Hay varias relaciones implícitas entre los diversos tipos de entidades. De hecho, siempre que un atributo de un tipo de entidad se refiere a otro tipo de entidad, algunos existe una relación. Por ejemplo, el atributo Gerente de DEPARTAMENTO se refiere a un empleado que administra el departamento; el atributo Controlling_department de PROYECTO se refiere al departamento que controla el proyecto; el atributo Supervisor de EMPLEADO se refiere a otro empleado (el que supervisa este empleado); el atributo Departamento de EMPLEADO se refiere al departamento de que trabaja el empleado; y así. En el modelo ER, estas referencias no deben representarse como atributos pero como relaciones. En el diseño inicial de los tipos de entidad, las relaciones generalmente se capturan en la forma de los atributos. A medida que se refina el diseño, estos atributos se convierten en relaciones entre tipos de entidad.

Tipos de relaciones, conjuntos e instancias

Un tipo de relación R entre n tipos de entidad E_1, E_2, \dots, E_n define un conjunto de asociaciones, o un conjunto de relaciones, entre entidades de estos tipos de entidad. Similar En el caso de tipos de entidad y conjuntos de entidades, un tipo de relación y su conjunto de relaciones correspondiente se denominan habitualmente con el mismo nombre, R . Matemáticamente, el conjunto de relaciones R es un conjunto de instancias de relación r_i , donde cada r_i asocia n entidades individuales (e_1, e_2, \dots, e_n), y cada entidad e_j en r_i es un miembro del conjunto de entidades E_j , $1 \leq j \leq n$. Por tanto, un conjunto de

relaciones es una relación matemática en E_1, E_2, \dots, E_n ; alternatively, se puede definir como un subconjunto del producto cartesiano de los conjuntos de entidades $E_1 \times E_2 \times \dots \times E_n$. Cada uno de los tipos de entidad E_1, E_2, \dots, E_n se dice que participa en el tipo de relación R ; de manera similar, cada una de las entidades individuales e_1, e_2, \dots , se dice que en participar en la relación instancia $r_i = (e_1, e_2, \dots, e_n)$. De manera informal, cada instancia de relación r_i en R es una asociación de entidades, donde La asociación incluye exactamente una entidad de cada tipo de entidad participante. Cada instancia de relación r_i representa el hecho de que las entidades que participan en r_i , están relacionados de alguna manera en la situación del mini mundo correspondiente. Por ejemplo, considere un tipo de relación **WORKS_FOR** entre los dos tipos de entidad **EMPLEADO** y **DEPARTAMENTO**, que asocia a cada empleado con el departamento para el que trabaja. Cada instancia de relación en la relación set **WORKS_FOR** asocia una entidad **EMPLEADO** y un **DEPARTAMENTO** entidad, donde cada instancia de relación r_i se muestra conectado a las entidades **EMPLEADO** y **DEPARTAMENTO** que participan en r_i . Los empleados e_1, e_3 y e_6 trabajan para el departamento d_1 ; los empleados e_2 y e_4 trabajan para el departamento d_2 ; y los empleados e_5 y e_7 trabajan para el departamento d_3 . En los diagramas ER, los tipos de relaciones se muestran como cuadros en forma de diamante, que están conectados por líneas rectas a las cajas rectangulares que representan los tipos de entidades participantes. El nombre de la relación se muestra en el cuadro en forma de diamante.

Grado de relación, nombres de roles y relaciones recursivas

Grado de un tipo de relación. El grado de un tipo de relación es el número de tipos de entidades participantes. Por tanto, la relación **TRABAJO PARA** es de grado dos. Un tipo de relación de grado dos se llama binario y uno de grado tres se llama ternario. Un ejemplo de una relación ternaria es **SUMINISTRO**, donde cada instancia de relación r_i asocia tres entidades (un proveedor s , una parte p y un proyecto j) siempre que s suministra la parte p al proyecto j . Las relaciones generalmente pueden ser de cualquier grado, pero las más comunes son las relaciones binarias. Las relaciones de grado superior son generalmente más complejas que las relaciones binarias.

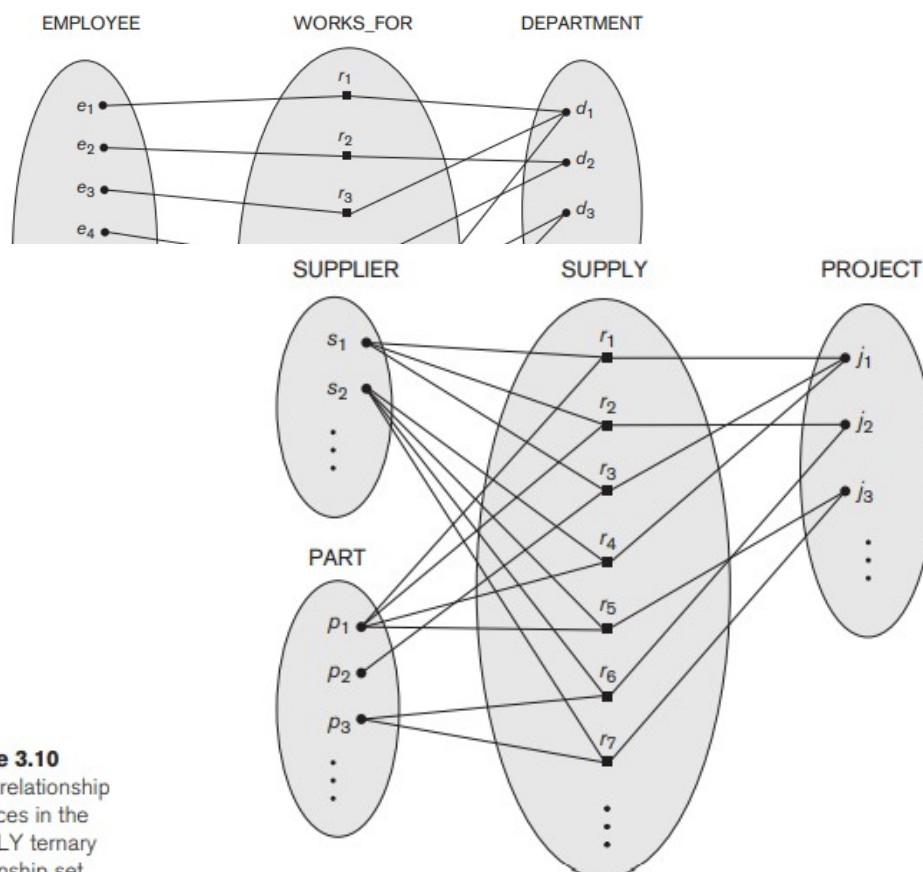


Figure 3.10
Some relationship
instances in the
SUPPLY ternary
relationship set.

Relaciones como atributos.

A veces es conveniente pensar en un tipo de relación binaria en términos de atributos. Considere el tipo de relación `WORKS_FOR`. Se puede pensar en un atributo llamado Departamento del tipo de entidad `EMPLEADO`, donde el valor de Departamento para cada entidad `EMPLEADO` es (una referencia a) la entidad `DEPARTAMENTO` para la que trabaja ese empleado. Por lo tanto, el valor establecido para este atributo de Departamento es el conjunto de todas las entidades `DEPARTAMENTO`, que es el conjunto de entidades `DEPARTAMENTO`. cuando especificamos el diseño inicial del tipo de entidad `EMPLEADO` para la base de datos de `EMPRESA`. Sin embargo, cuando pensamos en una relación binaria como un atributo, siempre tenemos dos opciones o dos puntos de vista. En este ejemplo, el punto de vista alternativo es pensar en un atributo de varios valores Empleados del tipo de entidad `DEPARTAMENTO` cuyo valor para cada entidad `DEPARTAMENTO` es el conjunto de entidades `EMPLEADOS` que trabajan para ese departamento. El conjunto de valores de este atributo Empleados es el conjunto de poder del conjunto de entidades `EMPLEADO`. Cualquiera de estos dos atributos, Departamento de `EMPLEADO` o Empleados de `DEPARTAMENTO`, puede representar el tipo de relación `TRABAJOS_PARA`. Si ambos están representados, están obligados a ser inversos entre sí.

Nombres de roles y relaciones recursivas.

Cada tipo de entidad que participa en un tipo de relación juega un papel particular en la relación. El nombre del rol significa el rol que juega una entidad participante del tipo de entidad en cada instancia de relación, y ayuda a explicar qué significa la relación. Por ejemplo, en el tipo de relación `TRABAJOS_PARA`, `EMPLEADO` desempeña el papel de empleado o trabajador y `DEPARTAMENTO` desempeña el papel de departamento o empleador. Los nombres de roles no son técnicamente necesarios en los tipos de relación donde todos los tipos de entidades participantes son distintos, ya que cada nombre de tipo de entidad participante puede usarse como nombre de rol. Sin embargo, en algunos casos, el mismo tipo de entidad participa más de una vez en un tipo de relación en diferentes roles. En tales casos, el nombre del rol se vuelve esencial para distinguir el significado del rol que desempeña cada entidad participante. Estos tipos de relaciones se denominan relaciones recursivas o relaciones de autorreferencia. El tipo de relación `SUPERVISIÓN` relaciona a un empleado con un supervisor, donde tanto el empleado como las entidades supervisoras son miembros del mismo conjunto de entidades `EMPLEADO`. Por lo tanto, el tipo de entidad `EMPLEADO` participa dos veces en `SUPERVISIÓN`: una vez en el rol de supervisor (o jefe) y una vez en el rol de supervisado (o subordinado). Cada instancia de relación r_i en `SUPERVISIÓN` asocia dos entidades de

empleados diferentes e_j y e_k , una de las cuales desempeña el papel de supervisor y la otra el papel de supervisado. las líneas marcadas con "1" representan el rol de supervisor y las marcadas con "2" representan el rol de supervisado; por tanto, e_1 supervisa e_2 y e_3 , e_4 supervisa e_6 y e_7 , y e_5 supervisa e_1 y e_4 . En este ejemplo, cada instancia de relación debe estar conectada con dos líneas, una marcada con "1" (supervisor) y la otra con "2" (supervisado).

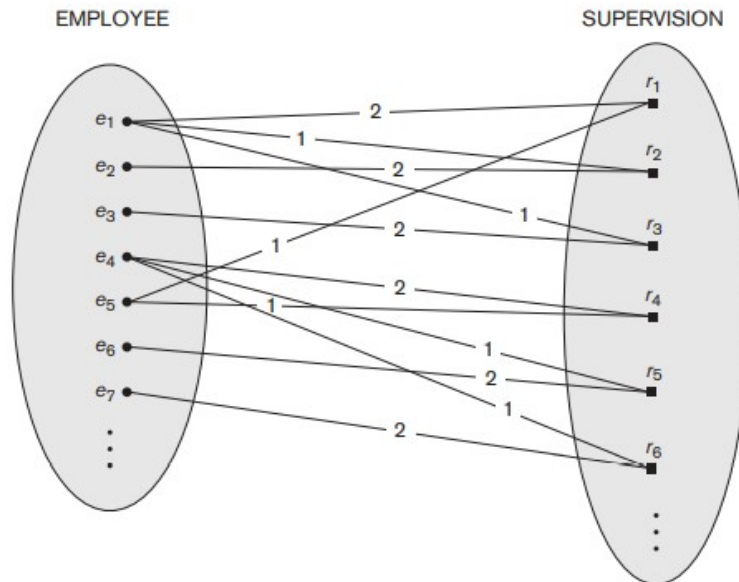


Figure 3.11

A recursive relationship SUPERVISION between EMPLOYEE in the supervisor role (1) and EMPLOYEE in the subordinate role (2).

Restricciones sobre tipos de relaciones binarias

Los tipos de relación suelen tener ciertas restricciones que limitan las posibles combinaciones de entidades que pueden participar en el conjunto de relaciones correspondiente. Estas limitaciones se determinan a partir de la situación del mini mundo que representan las relaciones. Si la empresa tiene una regla según la cual cada empleado debe trabajar para exactamente un departamento, entonces nos gustaría describir esta restricción en el esquema. Podemos distinguir dos tipos principales de restricciones de relación binaria: razón de cardinalidad y participación.

Razones de cardinalidad para relaciones binarias.

La relación de cardinalidad para una relación binaria especifica el número máximo de instancias de relación en las que una entidad puede participar. Por ejemplo, en el tipo de relación binaria WORKS_FOR, DEPARTMENT: EMPLOYEE tiene una relación de cardinalidad 1: N, lo que significa que cada departamento puede estar relacionado con (es decir, emplea) cualquier número de empleados (N), pero un empleado puede estar relacionado con (trabajar para) como máximo un departamento (1). Esto significa que para este tipo de relación en particular WORKS_FOR, una entidad de departamento en particular puede estar relacionada con cualquier número de empleados (N indica que no hay un número máximo). Por otro lado, un empleado puede estar relacionado con un máximo de un departamento. Las posibles razones de cardinalidad para los tipos de relaciones binarias son 1: 1, 1: N, N: 1 y M: N. Un ejemplo de una relación binaria 1: 1 es GESTIONES, que relaciona una entidad de departamento con el empleado que administra ese departamento. Esto representa las limitaciones del mini mundo que, en cualquier momento, un empleado puede administrar como máximo un departamento y un departamento puede tener como máximo un gerente. El tipo de relación WORKS_ON es de razón de cardinalidad M: N, porque la regla del mini mundo es que un empleado puede trabajar en varios proyectos y un proyecto puede tener varios empleados. Las razones de cardinalidad para las relaciones binarias se representan en los diagramas ER mostrando 1, M y N en los diamantes. Observe que en esta notación, podemos especificar un

máximo (N) o un máximo de uno (1) en la participación. Una notación alternativa permite al diseñador especificar un número máximo específico de participación, como 4 o 5.

Figure 3.12

A 1:1 relationship,
MANAGES.

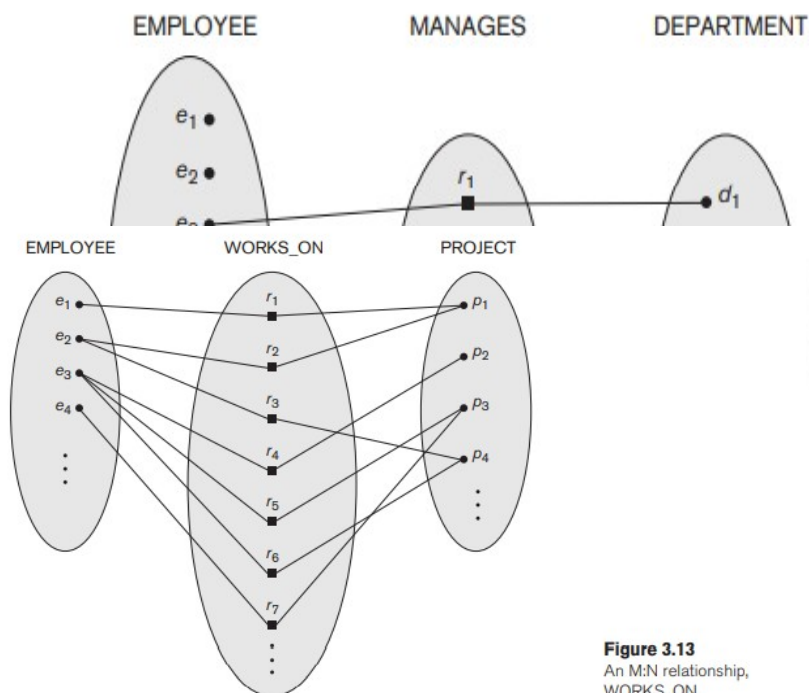


Figure 3.13

An M:N relationship,
WORKS_ON.

Restricciones de participación y dependencias de existencia.

La restricción de participación especifica si la existencia de una entidad depende de que esté relacionada con otra entidad a través del tipo de relación. Esta restricción especifica el número mínimo de instancias de relación en las que cada entidad puede participar y, a veces, se denomina restricción de cardinalidad mínima. Hay dos tipos de restricciones de participación, totales y parciales, que ilustramos con un ejemplo. Si una política de la empresa establece que todos los empleados deben trabajar para un departamento, entonces una entidad de empleados sólo puede existir si participa en al menos una instancia de relación WORKS_FOR. Por lo tanto, la participación de EMPLEADO en WORKS_FOR se denomina participación total, lo que significa que cada entidad en el conjunto total de entidades de empleados debe estar relacionada con una entidad de departamento a través de WORKS_FOR. La participación total también se llama dependencia de existencia. No esperamos que todos los empleados administren un departamento, por lo que la participación del EMPLEADO en el tipo de relación ADMINISTRA es parcial, lo que significa que una parte o parte del conjunto de entidades de los empleados está relacionada con alguna entidad del departamento a través de ADMINISTROS, pero no necesariamente todos. Nos referiremos a la razón de cardinalidad y las restricciones de participación, tomadas en conjunto, como las restricciones estructurales de un tipo de relación. En los diagramas ER, la participación total (o dependencia de existencia) se muestra como una línea doble que conecta el tipo de entidad participante a la relación, mientras que la participación parcial se representa mediante una sola línea. Observe que en esta notación, podemos especificar ningún mínimo (participación parcial) o un mínimo de uno (participación total). Una notación alternativa permite al diseñador especificar un número mínimo específico de participación en la relación, como 4 o 5.

Atributos de los tipos de relaciones

Los tipos de relación también pueden tener atributos, similares a los de los tipos de entidad. Por ejemplo, para registrar el número de horas por semana que un empleado en particular trabaja en un proyecto en particular, podemos incluir un atributo Horas para el tipo de relación WORKS_ON en la Figura 3.13. Otro ejemplo es incluir la fecha en la que un gerente comenzó a administrar un departamento a través de un atributo Fecha_inicio para el tipo de relación ADMINISTRA en la Figura 3.12. Tenga en cuenta que los atributos de tipos de relación 1: 1 o 1: N se pueden migrar a uno de los tipos de entidades participantes. Por ejemplo, el atributo Start_date para MANAGES. La relación puede ser un atributo de EMPLEADO (gerente) o DEPARTAMENTO, aunque conceptualmente pertenece a GERENCIA. Esto se debe a que GESTIONA es una relación de 1: 1, por lo que cada departamento o entidad de empleado participa como máximo en una instancia de relación. Por lo tanto, se puede determinar el valor del atributo Start_date por separado, ya sea por la entidad departamental participante o por la entidad empleado (gerente) participante.

Para un tipo de relación 1: N, un atributo de relación solo se puede migrar al tipo de entidad en el lado N de la relación. Por ejemplo, en la Figura 3.9, si la relación WORKS_FOR también tiene un atributo Start_date que indica cuándo un empleado comenzó a trabajar para un departamento, este atributo puede incluirse como un atributo de EMPLOYEE. Esto se debe a que cada empleado trabaja como máximo para un departamento y, por lo tanto, participa como máximo en una instancia de relación en WORKS_FOR, pero un departamento puede tener muchos empleados, cada uno con una fecha de inicio diferente. En los tipos de relación 1: 1 y 1: N, la decisión de dónde colocar un atributo de relación, como un atributo de tipo de relación o como un atributo de un tipo de entidad participante, la determina subjetivamente el diseñador del esquema.

Para los tipos de relación M: N (muchos a muchos), algunos atributos pueden ser determinados por la combinación de entidades participantes en una instancia de relación, no por una sola entidad. Estos atributos deben especificarse como atributos de relación. Un ejemplo es el atributo Horas de la relación M: N WORKS_ON (Figura 3.13); el número de horas por semana que un empleado trabaja actualmente en un proyecto está determinado por una combinación de empleado-proyecto y no por separado por ninguna entidad.

Tipos de entidades débiles

Los tipos de entidad que no tienen atributos clave propios se denominan tipos de entidad débil. Por el contrario, los tipos de entidades regulares que tienen un atributo clave, que incluyen todos los ejemplos discutidos hasta ahora, se denominan tipos de entidades fuertes. Las entidades que pertenecen a un tipo de entidad débil se identifican por estar relacionadas con entidades específicas de otro tipo de entidad en combinación con uno de sus valores de atributo. Llamamos a este otro tipo de entidad el tipo de entidad identificadora o propietaria, y al tipo de relación que relaciona un tipo de entidad débil con su propietario la relación de identificación del tipo de entidad débil.

Un tipo de entidad débil siempre tiene una restricción de participación total (dependencia de existencia) con respecto a su relación de identificación porque una entidad débil no puede identificarse sin una entidad propietaria. Sin embargo, no todas las dependencias de existencia dan como resultado un tipo de entidad débil. Por ejemplo, una entidad DRIVER_LICENSE no puede existir a menos que esté relacionada con una entidad PERSON, aunque tenga su propia clave (License_number) y, por lo tanto, no sea una entidad débil. Considere el tipo de entidad DEPENDIENTE, relacionado con EMPLEADO, que se usa para realizar un seguimiento de los dependientes de cada empleado mediante una relación 1: N. En nuestro ejemplo, los atributos de DEPENDIENTE son Nombre (el nombre del dependiente), Fecha de nacimiento, Sexo y Relación (con el empleado). Dos

dependientes de dos empleados distintos pueden, por casualidad, tener los mismos valores para Nombre, Fecha de nacimiento, Sexo y Relación, pero siguen siendo entidades distintas. Se identifican como entidades distintas solo después de determinar la entidad de empleado particular con la que cada dependiente está relacionado. Se dice que cada entidad de empleados es propietaria de las entidades dependientes que están relacionadas con ella. Un tipo de entidad débil normalmente tiene una clave parcial, que es el atributo que puede identificar de manera única las entidades débiles que están relacionadas con la misma entidad propietaria. En nuestro ejemplo, si asumimos que dos dependientes del mismo empleado nunca tienen el mismo nombre, el atributo Nombre de DEPENDIENTE es la clave parcial. En el peor de los casos, un atributo compuesto de todos los atributos de la entidad débil será la clave parcial. En los diagramas ER, tanto un tipo de entidad débil como su relación de identificación se distinguen por rodear sus cajas y rombos con líneas dobles. El atributo de clave parcial está subrayado con una línea discontinua o punteada. Los tipos de entidades débiles a veces se pueden representar como atributos complejos (compuestos, multivalor). En el ejemplo anterior, podríamos especificar un atributo de varios valores Dependientes para EMPLEADO, que es un atributo compuesto de varios valores con los atributos de componente Nombre, Fecha de nacimiento, Sexo y Relación. El diseñador de la base de datos decide qué representación utilizar. Un criterio que puede utilizarse es elegir la representación del tipo de entidad débil si el tipo de entidad débil participa independientemente en tipos de relación distintos de su tipo de relación de identificación. En general, se puede definir cualquier número de niveles de tipos de entidades débiles; un tipo de entidad propietaria puede ser en sí mismo un tipo de entidad débil. Además, un tipo de entidad débil puede tener más de un tipo de entidad de identificación y un tipo de relación de identificación de grado superior a dos.

Refinando el diseño de ER para la base de datos de la EMPRESA

Si no se puede determinar alguna razón de cardinalidad o dependencia a partir de los requisitos, los usuarios deben ser interrogados más para determinar estas restricciones estructurales. En nuestro ejemplo, especificamos los siguientes tipos de relaciones:

- **ADMINISTRA**, que es un tipo de relación 1: 1 (uno a uno) entre EMPLEADO y DEPARTAMENTO. La participación del EMPLEADO es parcial. La participación del DEPARTAMENTO no se desprende de los requisitos. Cuestionamos a los usuarios, quienes dicen que un departamento debe tener un gerente en todo momento, lo que implica una participación total. A este tipo de relación se le asigna el atributo Fecha_inicio.
- **WORKS_FOR**, un tipo de relación 1: N (uno a varios) entre DEPARTAMENTO y EMPLEADO. Ambas participaciones son totales.
- **CONTROLES**, un tipo de relación 1: N entre DEPARTAMENTO y PROYECTO. La participación de PROYECTO es total, mientras que la de DEPARTAMENTO se determina que es parcial, previa consulta con los usuarios indica que algunos departamentos pueden no controlar ningún proyecto.
- **SUPERVISIÓN**, un tipo de relación 1: N entre EMPLEADO (en el rol de supervisor) y EMPLEADO (en el rol de supervisado). Se determina que ambas participaciones son parciales, después de que los usuarios indiquen que no todos los empleados son supervisores y no todos los empleados tienen supervisor.
- **WORKS_ON**, determinado como un tipo de relación M: N (muchos a muchos) con el atributo Horas, después de que los usuarios indican que un proyecto puede tener varios empleados trabajando en él. Ambas participaciones están determinadas a ser totales.
- **DEPENDENTS_OF**, un tipo de relación 1: N entre EMPLOYEE y DEPENDIENTE, que también es la relación de identificación para el tipo de entidad débil

DEPENDIENTE. La participación del EMPLEADO es parcial, mientras que la del DEPENDIENTE es total.

Es importante tener la menor redundancia posible cuando diseñamos el esquema conceptual de una base de datos. Si se desea alguna redundancia en el nivel de almacenamiento o en el nivel de vista del usuario, se puede introducir más adelante.

Diagramas ER, convenciones de nomenclatura y problemas de diseño

las instancias de entidad individuales en un conjunto de entidades y las instancias de relación individuales en un conjunto de relaciones. En los diagramas ER, el énfasis está en representar los esquemas en lugar de las instancias. Esto es más útil en el diseño de bases de datos porque el esquema de una base de datos rara vez cambia, mientras que el contenido de la entidad los conjuntos pueden cambiar con frecuencia. Además, el esquema es obviamente más fácil de mostrar, porque es mucho más pequeño. Ahora revisamos la notación completa del diagrama ER. Los tipos de entidad regulares (fuertes) como EMPLEADO, DEPARTAMENTO y PROYECTO se muestran en cuadros rectangulares. Los tipos de relaciones como WORKS_FOR, MANAGES, CONTROLS y WORKS_ON se muestra en cajas en forma de diamante adjuntas a los tipos de entidades participantes con líneas rectas. Los atributos se muestran en óvalos y cada atributo se adjunta mediante una línea recta a su tipo de entidad o tipo de relación. Los atributos de componente de un atributo compuesto se adjuntan al óvalo que representa el atributo compuesto, como lo ilustra el Atributo de nombre del EMPLEADO. Los atributos multivalor se muestran en óvalos dobles, como lo ilustra el atributo Ubicaciones de DEPARTAMENTO. Los atributos clave tienen sus nombres subrayados. Los atributos derivados se muestran en óvalos punteados, como lo ilustra el atributo Number_of_employees de DEPARTMENT. Los tipos de entidades débiles se distinguen por estar colocados en rectángulos dobles y por tener su relación de identificación colocada en diamantes dobles, como lo ilustra el tipo de entidad DEPENDENT y el tipo de relación de identificación DEPENDENTS_OF. La clave parcial del tipo de entidad débil está subrayada con una línea de puntos.

La razón de cardinalidad de DEPARTMENT: EMPLOYEE en MANAGES es 1: 1, mientras que es 1: N para DEPARTMENT: EMPLOYEE en WORKS_FOR y M: N para WORKS_ON. La restricción de participación se especifica con una línea única para la participación parcial y con líneas dobles para la participación total (dependencia de existencia). Observe que la razón de cardinalidad es 1: N de supervisor a supervisado porque cada empleado en el rol de supervisado tiene como máximo un supervisor directo, mientras que un empleado en el rol de supervisor puede supervisar a cero o más empleados. Es importante tener en cuenta que hay muchas otras notaciones esquemáticas alternativas.

Nomenclatura adecuada de construcciones de esquema

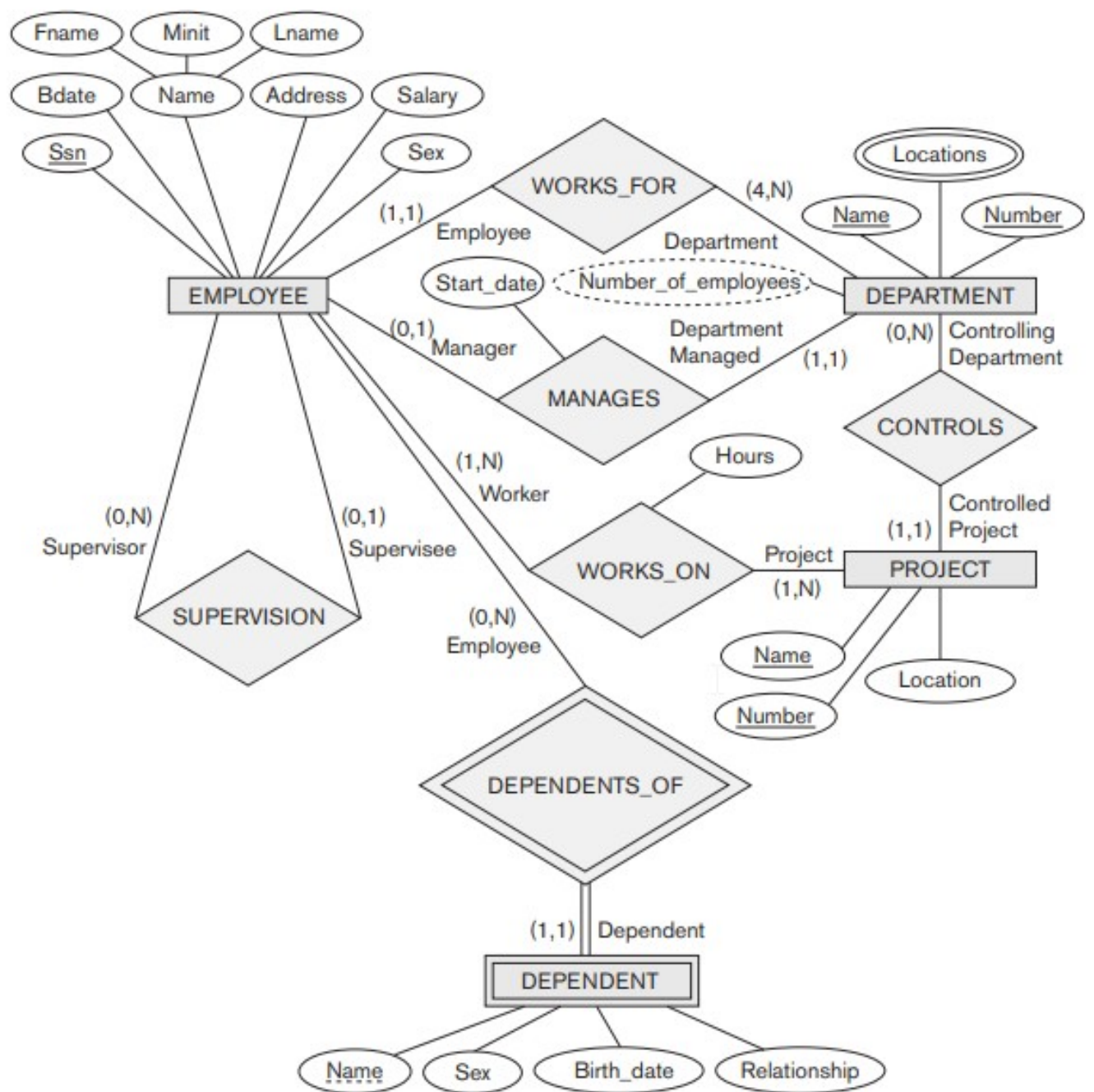
Al diseñar un esquema de base de datos, la elección de nombres para tipos de entidad, atributos, tipos de relación y (en particular) roles no siempre es sencilla. Uno debe elegir nombres que transmitan, en la medida de lo posible, los significados adjuntos a los diferentes constructos en el esquema. Elegimos usar nombres en singular para los tipos de entidad, en lugar de en plural, porque el nombre del tipo de entidad se aplica a cada entidad individual que pertenece a ese tipo de entidad. En nuestros diagramas ER, usaremos la convención de que los nombres de tipo de entidad y tipo de relación están en letras mayúsculas, los nombres de atributos tienen su letra inicial en mayúscula y los nombres de roles están en letras minúsculas. Como práctica general, dada una descripción narrativa de los requisitos de la base de datos, los sustantivos que aparecen en la narrativa tienden a dar lugar a nombres de tipos de entidades, y los verbos tienden a

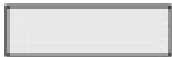
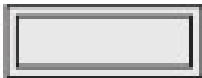
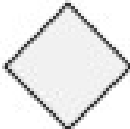
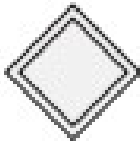


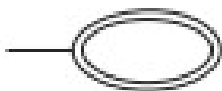
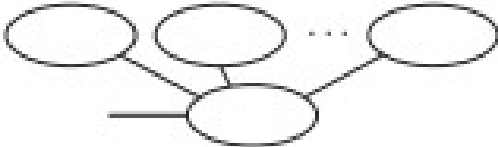

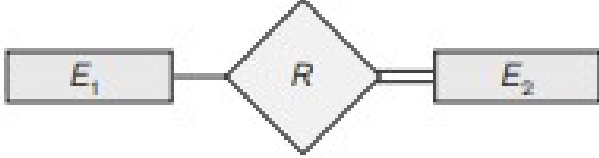
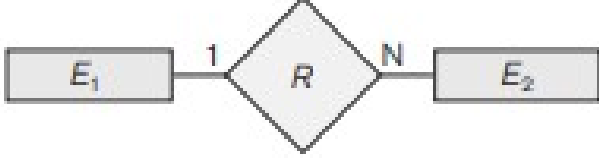

indicar nombres de tipos de relaciones. Los nombres de los atributos generalmente surgen de sustantivos adicionales que describen los sustantivos correspondientes a los tipos de entidad. Otra consideración de denominación implica elegir nombres de relaciones binarias para que el diagrama ER del esquema sea legible de izquierda a derecha y de arriba a abajo.

El tipo de relación `DEPENDENTS_OF`, que se lee de abajo hacia arriba. Cuando describimos esta relación, podemos decir que las entidades `DEPENDIENTES` (tipo de entidad inferior) son `DEPENDENTS_OF` (nombre de la relación) un `EMPLEADO` (tipo de entidad superior). Para cambiar esto para leer de arriba a abajo, podríamos cambiar el nombre del tipo de relación a `HAS_DEPENDENTS`, que luego se leería de la siguiente manera: Una entidad `EMPLEADO` (arriba tipo de entidad) `HAS_DEPENDENTS` (nombre de la relación) de tipo `DEPENDIENT` (tipo de entidad inferior). Tenga en cuenta que este problema surge porque cada relación binaria se puede describir a partir de cualquiera de los dos tipos de entidades participantes,

Notaciones alternativas para diagramas ER

Hay muchas notaciones esquemáticas alternativas para mostrar diagramas ER. El Apéndice A da algunas de las notaciones más populares, presentamos la notación del Lenguaje de modelado unificado (UML) para diagramas de clases, que se ha propuesto como estándar para el modelado de objetos conceptuales. En esta sección, describimos una notación ER alternativa para especificar restricciones estructurales en las relaciones, que reemplaza la razón de cardinalidad ($1:1$, $1:N$, $M:N$) y notación de una o dos líneas para las restricciones de participación. Esta notación implica asociar un par de números enteros (mínimo, máximo) con cada participación de un tipo de entidad E en una relación tipo R , donde $0 \leq \text{mínimo} \leq \text{máximo}$ y $\text{máximo} \geq 1$. Los números significan que para cada entidad e en E , Debemos participar en al menos un mínimo y un máximo de casos de relación en R en cualquier momento. En este método, $\text{min} = 0$ implica participación parcial, mientras que $\text{min} > 0$ implica participación total. Por lo general, se utiliza la notación de razón de cardinalidad / una línea / línea doble o la notación (mínimo, máximo). La notación (min, max) es más precisa y podemos usarla para especificar algunas restricciones estructurales para tipos de relación de mayor grado. Sin embargo, no es suficiente para especificar algunas restricciones clave en las relaciones de grado superior.



Symbol	Meaning
	Entity
	Weak Entity
	Relationship
	Identifying Relationship
	Attribute
	Key Attribute
	Multivalued Attribute
	Composite Attribute
	Derived Attribute
	Total Participation of E_2 in R
	Cardinality Ratio 1 : N for $E_1 : E_2$ in R
	Structural Constraint (min, max) on Participation of E in R

Ejemplo de otra notación: diagramas de clases UML

La metodología UML se utiliza ampliamente en el diseño de software y tiene muchos tipos de diagramas para diversos fines de diseño de software. Aquí solo presentamos brevemente los conceptos básicos de los diagramas de clases UML y los comparamos con los diagramas ER. De alguna manera, los diagramas de clases se pueden considerar como una notación alternativa a los diagramas ER. Una entidad en ER corresponde a un objeto en UML. En los diagramas de clases UML, una clase (similar a un tipo de entidad en ER) se muestra como un cuadro que incluye tres secciones: La sección superior da el nombre de la clase (similar al nombre del tipo de entidad); la sección central incluye los atributos; y la última sección incluye operaciones que se pueden aplicar a objetos individuales (similares a entidades individuales en un conjunto de entidades) de la clase. Las operaciones no se especifican en los diagramas ER.

Los tipos de relación se denominan asociaciones en terminología UML y las instancias de relación se denominan enlaces. Una asociación binaria (tipo de relación binaria) se representa como una línea que conecta las clases participantes (tipos de entidad) y, opcionalmente, puede tener un nombre. Un atributo de relación, llamado atributo de enlace, se coloca en un cuadro que está conectado a la línea de la asociación mediante una línea discontinua. La notación (min, max). Las multiplicidades se especifican en la forma min..max, y un asterisco (*) indica que no hay límite máximo de participación. Sin embargo, las multiplicidades se colocan en los extremos opuestos de la relación cuando se comparan con la notación (mínima, máxima). En UML, un solo asterisco indica una multiplicidad de 0 .. *, y un solo 1 indica una multiplicidad de 1..1. Un tipo de relación recursiva se denomina asociación reflexiva en UML, y los nombres de los roles, como las multiplicidades, se colocan en los extremos opuestos de una asociación cuando se comparan con la colocación de nombres de roles. En UML, hay dos tipos de relaciones: asociación y agregación. La agregación está destinada a representar una relación entre un objeto completo y sus partes componentes, y tiene una notación diagramática distinta. En el modelo ER, ambos se representan como relaciones. UML también distingue entre asociaciones (o agregaciones) unidireccionales y bidireccionales. En el caso unidireccional, la línea que conecta las clases se muestra con una flecha para indicar que solo se necesita una dirección para acceder a los objetos relacionados. Si no se muestra ninguna flecha, se asume el caso bidireccional, que es el predeterminado. Por ejemplo, si siempre esperamos acceder al gerente de un departamento a partir de un objeto DEPARTAMENTO, trazaríamos la línea de asociación que representa la asociación GESTIONES con una flecha de DEPARTAMENTO a EMPLEADO. Además, se pueden especificar instancias de relación para que se ordenen. Por ejemplo, podríamos especificar que los objetos de empleado relacionados con cada departamento a través de la asociación WORKS_FOR (relación) deben ordenarse por su valor de atributo Start_date. Los nombres de asociación (relación) son opcionales en UML, y los atributos de relación se muestran en un cuadro adjunto con una línea discontinua a la línea que representa la asociación / agregación. Las operaciones dadas en cada clase se derivan de los requisitos funcionales de la aplicación. Por lo general, es suficiente especificar los nombres de las operaciones inicialmente para las operaciones lógicas que se espera aplicar. A medida que se perfecciona el diseño, se agregan más detalles, como los tipos de argumentos exactos (parámetros) para cada operación, además de una descripción funcional de cada operación. tiene descripciones de funciones y diagramas de secuencia para especificar algunos de los detalles de la operación, pero estos están más allá del alcance de nuestra discusión. Las entidades débiles se pueden modelar usando la construcción UML llamada asociación calificada (o agregación calificada); esto puede representar tanto la relación de identificación como la clave parcial, que se coloca en una casilla adjunta a la clase de propietario. Esto se ilustra mediante la clase DEPENDIENTE

y su agregación calificada a EMPLEADO. En terminología UML, el atributo de clave parcial `Dependent_name` se denomina discriminador, porque su valor distingue los objetos asociados con (relacionados con) la misma entidad `EMPLOYEE`. Las asociaciones calificadas no se limitan a modelando entidades débiles, y pueden usarse para modelar otras situaciones en UML. Esta sección no pretende ser una descripción completa de los diagramas de clases UML, sino más bien ilustrar un tipo popular de notación esquemática alternativa que se puede utilizar para representar conceptos de modelado ER.

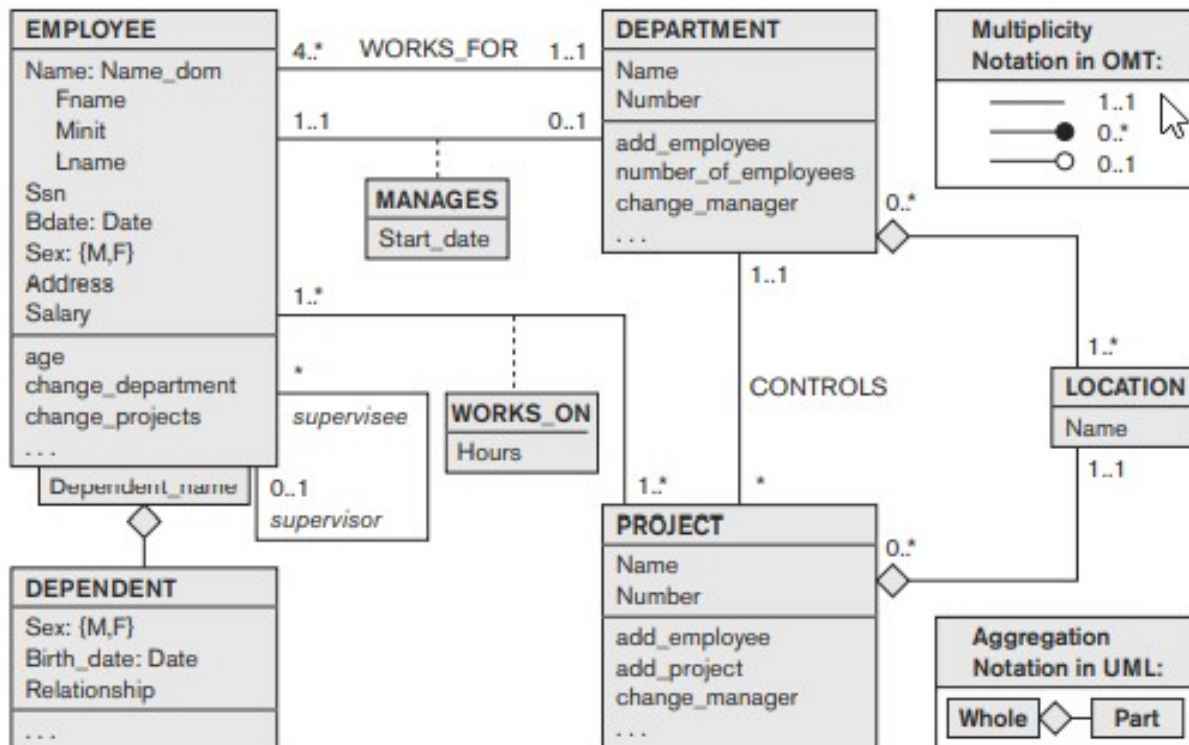


Figure 3.16
The COMPANY conceptual schema in UML class diagram notation.

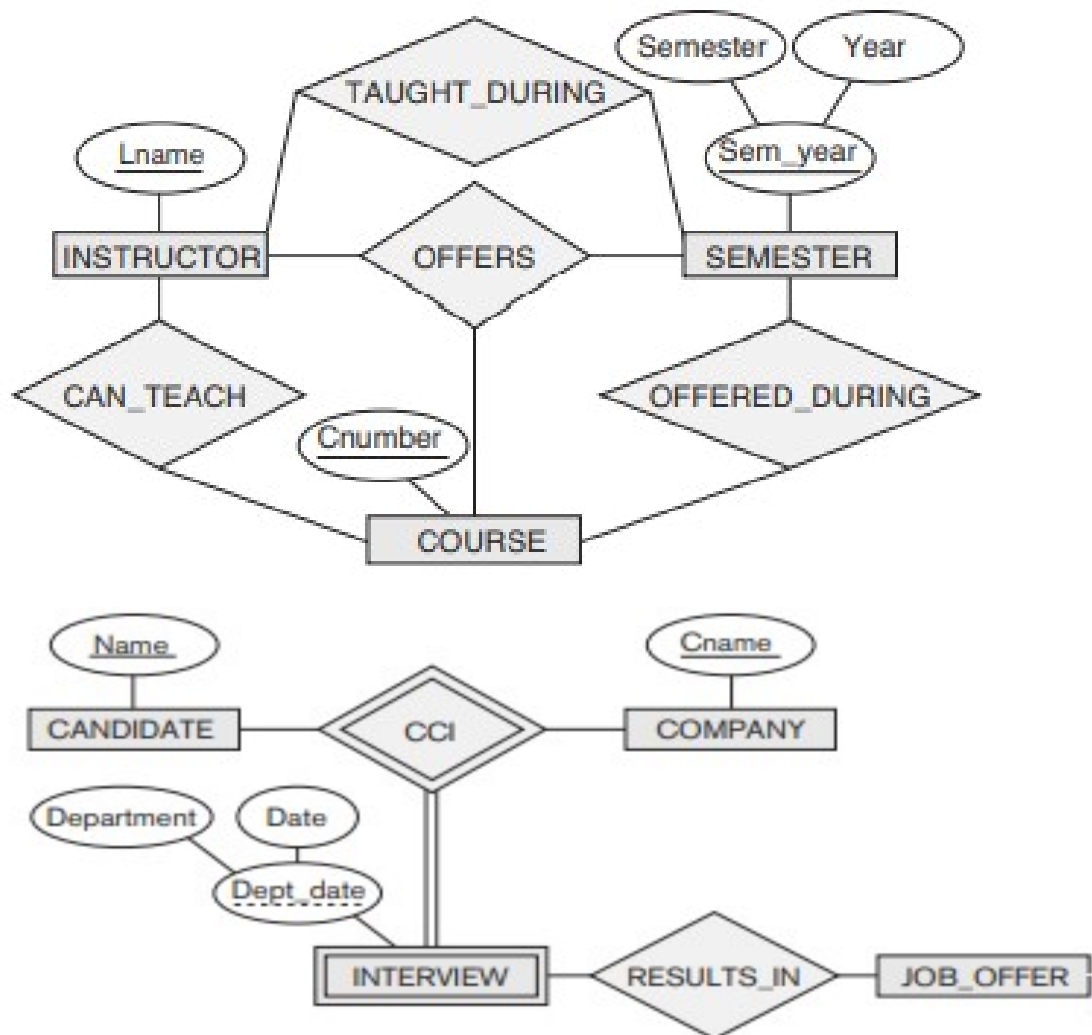
Tipos de relación de grado superior a dos
Definimos el grado de un tipo de relación como el número de tipos de entidades participantes y llamamos un tipo de relación de grado dos binario y un tipo de relación de grado tres ternario. En esta sección, elaboramos las diferencias entre las relaciones binarias y de grado superior, cuándo elegir relaciones de grado superior frente a relaciones binarias y cómo especificar restricciones en las relaciones de grado superior.

Elegir entre relaciones binarias y ternarias (o de grado superior)

La notación del diagrama ER para un tipo de relación ternaria, que muestra el esquema para el tipo de relación SUMINISTRO que se mostró en el nivel de instancia Recuerde que el conjunto de relaciones de SUMINISTRO es un conjunto de instancias de relación (s, j, p), donde el significado es que s es un PROVEEDOR que actualmente está suministrando una PARTE p a un PROYECTO j. En general, un tipo de relación R de grado n tendrá n bordes en un diagrama ER, uno que conecta R a cada tipo de entidad participante. En general, un tipo de relación ternaria representa información diferente a la de tres tipos de relación binaria. Considere los tres binarios tipos de relación `CAN_SUPPLY`, `USES` y `SUPPLIES`. Suponer que `CAN_SUPPLY`, entre `SUPPLIER` y `PART`, incluye una instancia (s, p) siempre que los proveedores s pueden suministrar la parte p (a cualquier proyecto); `USOS`, entre `PROYECTO` y `PARTE`, incluye una instancia (j, p) siempre que el proyecto j use la parte p;

y SUMINISTROS, entre el PROVEEDOR y el PROYECTO, incluye una instancia (s, j) siempre que el proveedor suministra alguna parte para proyectar j. La existencia de tres instancias de relación (s, p), (j, p) y (s, j) en CAN_SUPPLY, USES y SUPPLIES, respectivamente, no implica necesariamente que exista una instancia (s, j, p) en la relación ternaria SUMINISTRO, porque el significado es diferente. A menudo es complicado decidir si una relación en particular debe representarse como un tipo de relación de grado n o debe dividirse en varios tipos de relación de grados más pequeños. El diseñador debe basar esta decisión en la semántica o el significado de la situación particular que se representa. La solución típica es incluir la relación ternaria más una o más de las relaciones binarias, si representan significados diferentes y si todas son que necesita la aplicación. Algunas herramientas de diseño de bases de datos se basan en variaciones del modelo ER que solo permiten relaciones binarias. En este caso, una relación ternaria como SUMINISTRO debe representarse como un tipo de entidad débil, sin clave parcial y con tres identificadores relaciones. Los tres tipos de entidades participantes PROVEEDOR, PARTE y PROYECTO son juntos los tipos de entidad propietaria. Por lo tanto, una entidad en el tipo de entidad débil SUMINISTRO se identifica por la combinación de sus tres entidades propietarias de PROVEEDOR, PARTE y PROYECTO. También es posible representar la relación ternaria como un tipo de entidad regular introduciendo una clave artificial o sustituta. En este ejemplo, se podría usar un atributo clave Supply_id para el tipo de entidad de suministro, convirtiéndolo en un tipo de entidad regular. Tres relaciones binarias N: 1 relacionan el SUMINISTRO con cada uno de los tres tipos de entidades participantes. El tipo de relación ternaria OFERTAS representa información sobre instructores que ofrecen cursos durante semestres particulares; por lo tanto, incluye una instancia de relación (i, s, c) siempre que el INSTRUCTOR i ofrezca el CURSO c durante el SEMESTRE s. Los tres tipos de relaciones binarias tienen los siguientes significados: CAN_TEACH relaciona un curso con los instructores que pueden impartir ese curso, ENSEÑAN_DURANTE relaciona un semestre con los instructores que impartieron algún curso durante ese semestre y OFERTA_DURANTE relaciona un semestre con los cursos ofrecido durante ese semestre por cualquier instructor. Estas relaciones ternarias y binarias representan información diferente, pero ciertas las restricciones deben mantenerse entre las relaciones. Por ejemplo, una instancia de relación (i, s, c) no debe existir en OFERTAS a menos que exista una instancia (i, s) en TAUGHT_DURING, una instancia (s, c) exista en OFFERED_DURING y una instancia (i, c) exista en CAN_TEACH. Sin embargo, lo contrario no siempre es cierto; podemos tener instancias (i, s), (s, c) y (i, c) en los tres tipos de relaciones binarias sin una instancia correspondiente (i, s, c) en OFERTAS. Tenga en cuenta que en este ejemplo, basándonos en los significados de las relaciones, podemos inferir las instancias de TAUGHT_DURING y OFFERED_DURING a partir de las instancias en OFERTAS, pero no podemos inferir las instancias de CAN_TEACH; por lo tanto, TAUGHT_DURING y OFFERED_DURING son redundantes y pueden omitirse. Aunque en general tres relaciones binarias no pueden reemplazar una relación ternaria, pueden hacerlo bajo ciertas restricciones adicionales. En nuestro ejemplo, si la relación CAN_TEACH es 1: 1 (un instructor puede enseñar solo un curso y un curso solo puede ser impartido por un instructor), entonces la relación ternaria OFERTAS puede omitirse porque se puede inferir de las tres relaciones binarias CAN_TEACH, TAUGHT_DURING y OFFERED_DURING. El diseñador de esquemas debe analizar el significado de cada situación específica para decidir cuáles de los tipos de relación binaria y ternaria son necesarios. Observe que es posible tener un tipo de entidad débil con un ternario identificando el tipo de relación. En este caso, el tipo de entidad débil puede tener varios tipos de entidad propietaria. se muestra un ejemplo. Este ejemplo muestra parte de una base de datos que realiza un seguimiento de los candidatos que se entrevistan para puestos de trabajo en varias empresas, que pueden ser parte de una base de datos de una agencia de empleo. En los

requisitos, un candidato puede tener múltiples entrevistas con la misma empresa (por ejemplo, con diferentes departamentos de la empresa o en fechas separadas), pero se realiza una oferta de trabajo basada en una de las entrevistas. Aquí, ENTREVISTA se representa como una entidad débil con dos propietarios CANDIDATO y EMPRESA, y con la clave parcial Fecha_dep. Una entidad de ENTREVISTA se identifica de forma única por un candidato, una empresa y la combinación de la fecha y el departamento de la entrevista.



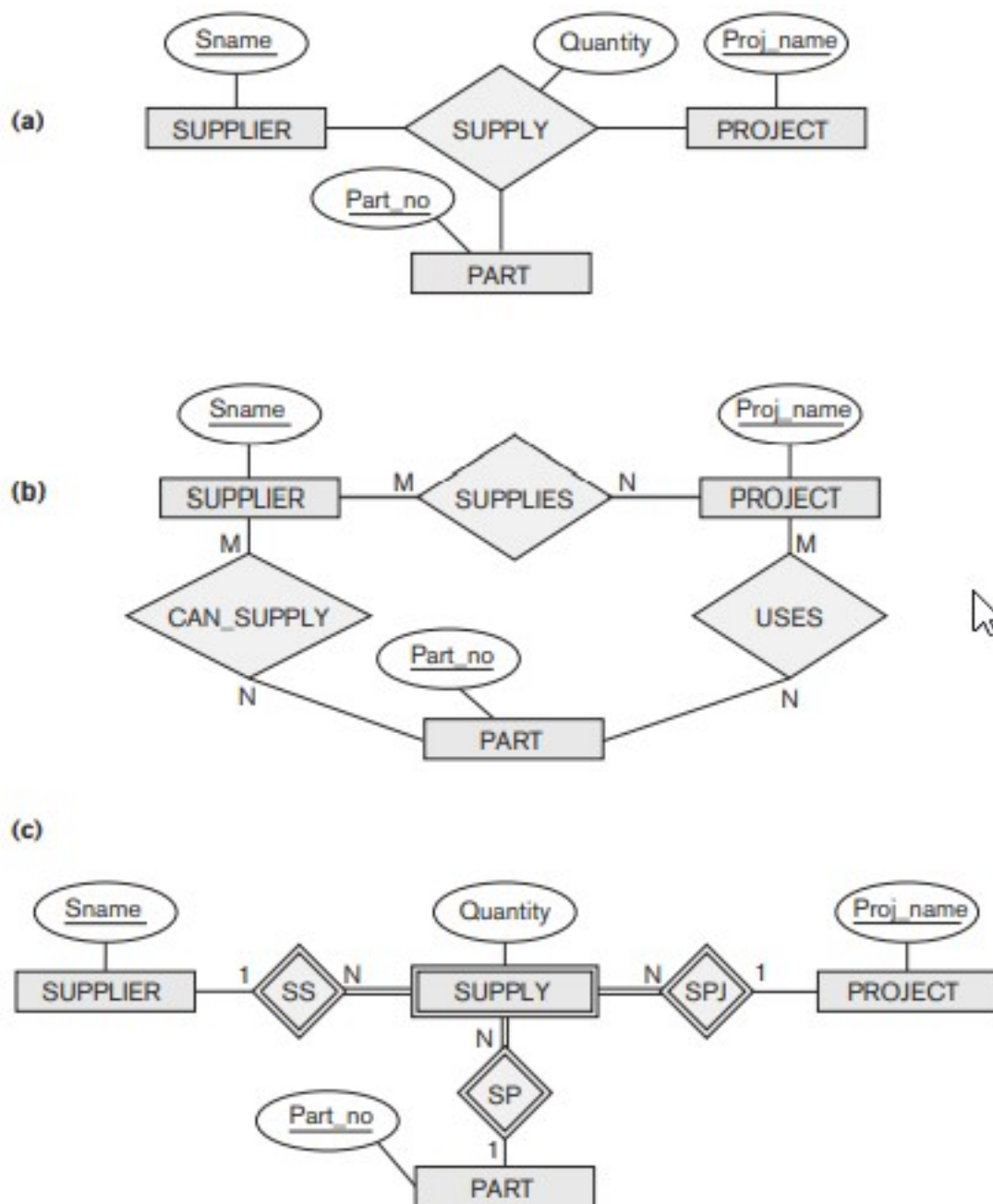


Figura 3.17

Restricciones en ternario (o grado superior)

Relaciones Hay dos notaciones para especificar restricciones estructurales en relaciones n-arias, y especifican restricciones diferentes. Por lo tanto, ambos deben usarse si es importante especificar completamente las restricciones estructurales en una relación ternaria o de grado superior. La primera notación se basa en la notación de razón de cardinalidad de las relaciones binarias. Aquí, se especifica un 1, M o N en cada arco de participación (los símbolos M y N representan muchos o cualquier número). Suponga que existe la restricción de que para una combinación particular de proyecto y parte, solo se utilizará un proveedor (solo un proveedor suministra una parte particular a un proyecto en particular). En este caso, colocamos 1 en la participación del PROVEEDOR y M, N en las participaciones PROYECTO. Esto especifica la restricción de que una combinación particular (j, p) puede aparecer como máximo una vez en el conjunto de relaciones porque cada uno de ellos (PROYECTO, PARTE) la combinación determina de forma única un único proveedor. Por tanto, cualquier instancia de relación (s, j, p) se identifica

unívocamente en el conjunto de relaciones por su combinación (j, p), lo que hace que (j, p) sea una clave para el conjunto de relaciones. En esta notación, las participaciones que tienen un 1 especificado en ellas no están obligadas a formar parte de la identificación clave para el conjunto de relaciones.¹⁶ Si las tres cardinalidades son M o N, entonces la clave será la combinación de los tres participantes. La segunda notación se basa en la notación (mínimo, máximo) para las relaciones binarias. Un (mínimo, máximo) en una participación aquí especifica que cada entidad está relacionada con al menos un mínimo y un máximo de instancias de relación en el conjunto de relaciones. Estas restricciones no influyen en la determinación de la clave de una relación n-aria, donde $n > 2$,¹⁷ pero especifican un tipo diferente de restricción que impone restricciones sobre cuántas instancias de relación puede participar cada entidad.

Otro ejemplo: una base de datos UNIVERSITARIA

Ahora presentamos otro ejemplo, una base de datos de la UNIVERSIDAD, para ilustrar los conceptos de modelado de ER. Suponga que se necesita una base de datos para realizar un seguimiento de las inscripciones de los estudiantes en las clases y las calificaciones finales de los estudiantes. Después de analizar las reglas del mini mundo y las necesidades de los usuarios, se determinó que los requisitos para esta base de datos eran los siguientes (por brevedad, mostramos los nombres de tipo de entidad elegidos y los nombres de atributo para el esquema conceptual entre paréntesis a medida que describimos los requisitos; tipo de relación los nombres solo se muestran en el diagrama de esquema ER):

- La universidad está organizada en colegios (COLLEGE), y cada colegio tiene un nombre único (CName), una oficina principal (COffice) y un teléfono (CPhone), y un miembro de la facultad en particular que es decano del colegio. Cada universidad administra una serie de departamentos académicos (DEPT). Cada departamento tiene un nombre único (DName), un número de código único (DCode), una oficina principal (DOffice) y un teléfono (DPhone), y un miembro de la facultad en particular que preside el departamento. Llevamos un registro de la fecha de inicio (CStartDate) cuando ese miembro de la facultad comenzó a presidir el departamento. Un departamento ofrece una serie de cursos (CURSO), cada uno de los cuales tiene un nombre de curso único (CoName), un número de código único (CCode), un nivel de curso (Nivel: esto se puede codificar como 1 para el nivel de primer año, 2 para el segundo, 3 para junior, 4 para senior, 5 para nivel de maestría y 6 para nivel de doctorado), un curso de horas de crédito (créditos) y una descripción del curso (CDesc). La base de datos también realiza un seguimiento de los instructores (INSTRUCTOR); y cada instructor tiene un identificador único (Id), nombre (IName), oficina (IOffice), teléfono (IPhone) y rango (Rank); además, cada instructor trabaja para un departamento académico primario.

- La base de datos mantendrá los datos de los estudiantes (ESTUDIANTE) y almacenará el nombre de cada estudiante (SName, compuesto por el nombre (FName), segundo nombre (MName), apellido (LName)), identificación del estudiante (Sid, único para cada estudiante), dirección (Dirección), teléfono (Phone), código principal (Major) y fecha de nacimiento (DoB). Un estudiante está asignado a un departamento académico primario. Es necesario realizar un seguimiento de las calificaciones del estudiante en cada sección que haya completado.

- Los cursos se ofrecen por secciones (SECCIÓN). Cada sección está relacionada con un solo curso y un solo instructor y tiene un identificador de sección único (SecId). Una sección también tiene un número de sección (SecNo: este está codificado como 1, 2, 3,... Para múltiples secciones ofrecidas durante el mismo semestre / año), semestre (Sem), año (Año), aula (CRoom: este está codificado como una combinación de código de construcción (Bldg) y número de habitación (RoomNo) dentro del edificio), y días / horas (DaysTime: por ejemplo, 'MWF 9 am-9.50am' o 'TR 3.30pm-5.20pm' - restringido a solo

los valores de días / horas permitidos). (Nota: La base de datos hará un seguimiento de todas las secciones ofrecidas durante los últimos años, además de ofertas actuales. El SecId es único para todas las secciones, no solo para las secciones de un semestre en particular). La base de datos realiza un seguimiento de los estudiantes en cada sección, y la calificación se registra cuando está disponible (esta es una relación de muchos a muchos entre los estudiantes y las secciones). Una sección debe tener al menos cinco estudiantes.

Observe que para el tipo de entidad SECTION, solo mostramos SecID como una clave subrayada, pero debido a las restricciones del minimundo, varias otras combinaciones de valores tienen que ser únicas para cada entidad de sección. Por ejemplo, cada una de las siguientes combinaciones debe ser única en función de las limitaciones típicas del mini mundo:

1. (SecNo, Sem, Year, CCode (del CURSO relacionado con la SECCIÓN)): Esto especifica que los números de sección de un curso en particular deben ser diferentes durante cada semestre y año en particular.
2. (Sem, Year, CRoom, DaysTime): Esto especifica que en un semestre y año en particular, un salón de clases no puede ser utilizado por dos secciones diferentes al mismo día / hora.
3. (Sem, Year, DaysTime, Id (del INSTRUCTOR que enseña la SECCIÓN)): Esto especifica que en un semestre y año en particular, un instructor no puede enseñar dos secciones al mismo día / hora. Tenga en cuenta que esta regla no se aplicará si un instructor puede enseñar dos secciones combinadas juntas en la universidad en particular.