

Resumen de teórico Base de Datos

Agustin Curto, agucurto95@gmail.com

2016

Índice general

1. Modelado Entidad - Relación	3
1.1. Conjunto de Entidades	3
1.2. Claves	3
1.3. Correspondencia de cardinales	4
1.4. Formas de participación	4
1.5. Conjunto de entidades débiles	4
1.6. Diseño de una DB de calidad	5
1.6.1. Eliminación de información redundante	5
1.6.2. Uso de CE vs Atributos	5
1.6.3. Errores comunes	6
1.6.4. Uso de CEs vs uso de CR	6
1.6.5. CR de grado > 2 vs CR binarias	6
1.6.6. CE débiles vs Atributo multivalorado compuesto	6
1.7. Especialización - Generalización	6
1.8. Esquemas e instancias relacionales	8
1.9. Tipos de atributos	8
1.10. DB relacionales	8
1.10.1. Superclaves	9
1.10.2. Claves candidatas y primarias	9
1.10.3. Claves foráneas	9
1.11. Diseño de una DB relacional	10
1.12. Reducción a esquemas relacionales	10
1.12.1. Regla CEF1	10
1.12.2. Regla CEF2	10
1.12.3. Regla CEF3	11
1.12.4. Regla CR1	11
1.12.5. Regla CR2	12
1.12.6. Regla CR3	12
1.12.7. Regla CED	13
1.12.8. Reducción de generalización	13
2. Algebra Relacional	15
2.1. Selección	15
2.2. Proyección	16
2.3. Proyección generalizada	16
2.4. Unión, diferencia e intersección	17
2.5. Producto cartesiano	17
2.6. Renombramiento	17
2.7. Definición del AR	18
2.8. Vínculo de AR con SQL	19

2.9. Reunión natural	19
2.10. División	20
2.11. Agregación	22
2.12. Reuniones externas	23
2.13. Manejo de valores nulos	24
2.14. Asignación	25

Capítulo 1

Modelado Entidad - Relación

1.1. Conjunto de Entidades

- Una **entidad** es un objeto que existe y es distinguible de los otros objetos. Las entidades tienen atributos.

Ejemplo: una persona tiene nombres y direcciones.

- Un **conjunto de entidades** es un conjunto de entidades del mismo tipo (i.e. con los mismos atributos) que comparte las mismas propiedades.

Ejemplo: conjunto de todas las personas con los atributos del ejemplo anterior.

- Un **dominio** es el conjunto de valores permitidos para cada atributo.
- Tipos de atributos:
 - Atributos simples y compuestos.
 - Atributos uni-valorados y multi-valorados
 - Atributos derivados, pueden computarse de otros atributos.

Notaciones de diagrama

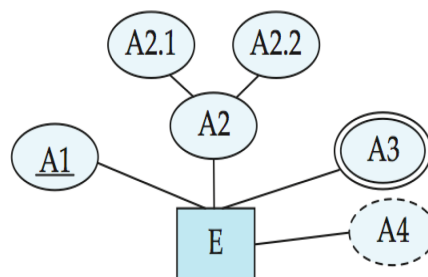


Figura 1.1: Entidad de nombre E, Atributo simple A1, Atributo compuesto A2, Atributo multivalorado A3, Atributo derivado A4, Clave primaria A1.

1.2. Claves

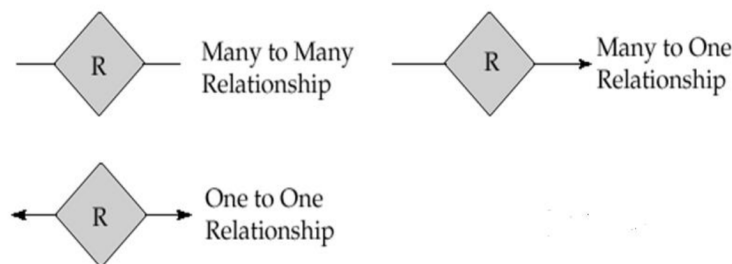
- Una **superclave** de un CE es un conjunto de uno o más atributos cuyos valores unívocamente determinan cada entidad.

- Una **clave candidata** de un CE es una superclave minimal (i.e. si se quita atributo dejamos de tener superclave, por lo general los ID).
- Una **clave primaria** es aquella seleccionada de entre todas las claves candidatas que puedan existir.

1.3. Correspondencia de cardinales

Sea R un conjunto de relaciones de conjunto de entidades $E1$ a conjunto de entidades $E2$.

- **Conjuntos de relaciones uno-uno:** una entidad de **E1** está asociada con a lo más una entidad de **E2** vía la relación **R**. Una entidad de **E2** está asociada con a lo más una entidad de **E1** vía **R**.
- **Conjuntos de relaciones uno-varios / varios-uno:** una entidad de **E1** está asociada con ninguna o varias entidades de **E2** vía la relación **R**. Una entidad de **E2** está asociada con a lo más una entidad de **E1** vía **R**.
- **Conjuntos de relaciones varios-varios:** una entidad de **E1** está asociada con ninguna o varias entidades de **E2** vía **R**. Una entidad de **E2** está asociada con a lo más una entidad de **E1** vía **R**.



1.4. Formas de participación

- **Participación total:** toda entidad en el conjunto de entidades participa en al menos una relación en el conjunto de relaciones. Es indicada por una línea doble.

Ejemplo: participación de caja de ahorro en cliente es total. Toda caja de ahorro debe tener clientes asociados.

- **Participación parcial:** algunas entidades no participan en alguna relación en el conjunto de relaciones.

Ejemplo: participación de instructor en supervisor es parcial.

1.5. Conjunto de entidades débiles

Un CE que no tiene una clave primaria en el conjunto de sus atributos, se llama **conjunto de entidades débiles**. La existencia de un CE débiles depende de la existencia de un CE fuertes llamado **CE identificador**.

Ejemplo: ¿En el caso de un libro, con solo su número de inventario como atributo, y una biblioteca cuál sería el CE identificador?

Hay un CR varios-uno entre CE débil y CE identificador, donde el CE débil tiene participación total, a este CR se le llama **CR de identificación**. El mismo se representa con un diamante doble.

El **discriminador** de un CE débiles es un conjunto de atributos que permite distinguir entre todas las entidades de un CE débiles asociadas a la misma entidad fuerte.

Para identificar las entidades débiles se forma la clave primaria del CE con la clave primaria del CE identificador más el discriminador del CE débiles.

1.6. Diseño de una DB de calidad

1.6.1. Eliminación de información redundante

Una vez que se han decidido los CE del problema y los atributos de los CE se pasa a definir los CR, esos CR pueden señalar que algunos atributos en CE son redundantes y por lo tanto necesitan ser removidos.

Ejemplo:

- CE **aula** con atributos (edificio, número y capacidad)
- CE **sección** con atributos (idCurso, idSec, semestre, año, edificio, número)
- CR **sec-aula** que relaciona secciones con aulas

¿Qué pasa con esta propuesta?

- edificio y número aparecen en ambos CE pero son clave primaria de aula
- Luego edificio y número son redundantes en **sección** y deben ser removidos de allí

1.6.2. Uso de CE vs Atributos

A veces tenemos dos alternativas para modelar un objeto perteneciente a un CE E, como un CE o como un atributo de E.

- Cuando el CE E tiene un solo objeto que tiene un solo valor (atómico), conviene modelar al objeto como un atributo simple.
- Cuando el CE E tiene cero o más objetos que tienen un solo valor (atómico), se puede modelar al objeto como un atributo multivalorado.
- Cuando el objeto tiene varios atributos y es poseído por una sola entidad del CE E, conviene modelar el objeto como CE. La razón es que “ser poseído por una sola entidad” no se puede representar usando un atributo.
- Cuando el objeto tiene varios atributos algunos de los cuales no son de clave primaria del objeto y es poseído por varias entidades del CE E, conviene modelar el objeto como CE. La razón es que los atributos del objeto que no se usan para identificarlo generan redundancia de información, cosa que no pasa cuando el objeto se representa un como CE.

1.6.3. Errores comunes

Un error muy común es usar la clave primaria de un CE como atributo de otro CE en lugar de usar un CR. La forma correcta sería usar un CR que vincule los dos CE, ya que hace la conexión entre los dos CE explícita, en lugar de implícita vía atributos de la clave primaria de uno de los CE.

1.6.4. Uso de CEs vs uso de CR

Muchas veces son posibles dos alternativas para modelar un asunto, como un CE o como un CR.

- Si CR es **uno-varios** o **varios-uno**: si hay atributos, no de clave primaria en el CE A, entonces los atributos de A que no son de clave primaria dan lugar a redundancia de información si se usa alternativa CE. Por lo que conviene usar la opción CR en lugar de CE.
- Si el CR es **varios – varios**: si en al menos un CE hay atributos que no son de clave primaria, entonces ocurre igual que antes, los atributos del CE que no son de clave primaria dan lugar a redundancia de información si se usa la opción CE. Por lo que conviene usar CR en lugar de CE.

1.6.5. CR de grado > 2 vs CR binarias

A veces tenemos las alternativas de modelar una situación como un CR de grado > 2 o como varios CR binarios. Si con los CR binarios puedo capturar los datos del CR de grado > 2 y puedo expresar más restricciones de integridad que con el CR de grado > 2 , entonces usar los CR binarios.

1.6.6. CE débiles vs Atributo multivalorado compuesto

A veces para modelar una situación tenemos dos alternativas un CE débiles o un atributo multi-valorado compuesto. Si el objeto contemplado tiene su complejidad e involucra varios atributos y es de suma importancia para la organización, es poco conveniente tenerla como un atributo de un CE fuerte, por lo tanto, conviene modelarlo como un CE débiles.

Si los objetos siendo considerados están relacionados con entidades de un CE fuerte que no es de identificación, entonces es necesario modelar dichos objetos como CE débiles.

1.7. Especialización - Generalización

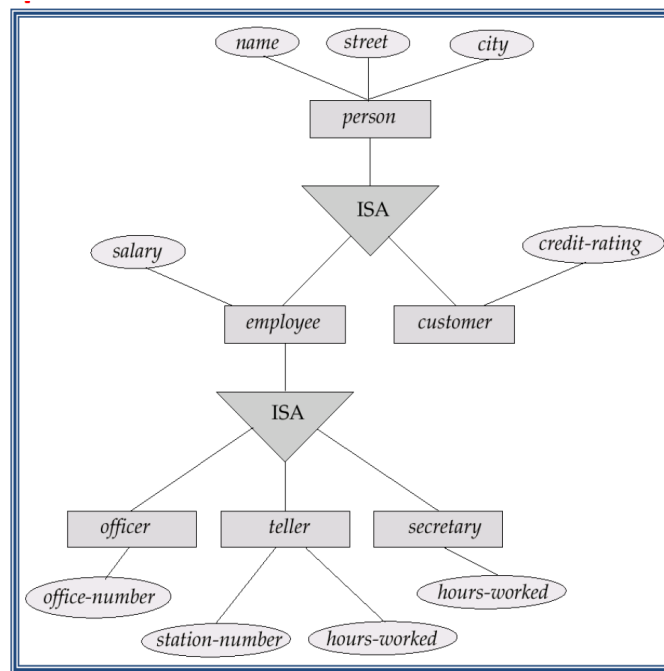
Cuando en un diseño ER hay varios CE que son bastante similares en el sentido que comparten varios atributos en común, que tienen las mismas claves primarias y que participan en los mismos CR, aparecen problemas de repetición.

Esto es malo ya que estos modelos ocupan demasiado espacio. Muchos CR hacen el diagrama más intrincado, esto se ve agravado cuando el esquema de la BD tiene muchos CE y CR. Al cambiar un CE o CR muchas veces hay que propagar el cambio a otros CE o CR.

La solución a este problema es utilizar especialización-generalización.

- **Especialización:** hace referencia a un proceso de diseño *top-down* donde designamos subgrupos dentro de un CE que son distintivos de otras entidades en el CE. Estos subgrupos son CE de más bajo nivel que tienen atributos específicos o participan de CR

que no aplican al CE de más alto nivel. Una especialización se denota con un triángulo etiquetado ISA. La relación ISA se llama también relación de superclase – subclase.



■ **Herencia de atributos:** un CE de más bajo nivel hereda:

- todos los atributos
- la clave primaria
- participaciones en CR

del CE de más alto nivel con el cual está relacionado.

■ **Generalización:** hace referencia a un proceso de diseño *bottom-up* que generaliza unos cuantos CE que comparten las mismas propiedades en un CE de más alto nivel.

■ **Restricciones de integridad:**

Para indicar si una entidad pertenece o no a más de un CE de nivel más bajo dentro de la generalización.

- Disjunto: una entidad puede pertenecer a solo un CE de nivel más bajo. Usar palabra reservada *disj*.
- Solapado: una entidad puede pertenecer a más de un CE de nivel más bajo.

■ **Restricciones de completitud:**

Para indicar si una entidad en el CE de nivel más alto debe pertenecer a al menos uno de los CE de nivel más bajo en la generalización.

- Total: una entidad debe pertenecer a un CE de nivel más bajo. Usar línea doble.
- Parcial: una entidad puede no pertenecer a un CE de nivel más bajo.

1.8. Esquemas e instancias relacionales

Definiciones:

- A_1, A_2, \dots, A_n son **atributos**
- $R = (A_1, A_2, \dots, A_n)$ es un **esquema de relación**
Dado un enunciado en lenguaje natural, se puede identificar atributos primero, y luego se pueden proponer esquemas de relación para distintos conceptos del problema actual.
- Dados conjuntos D_1, D_2, \dots, D_n una **relación r** es un subconjunto de D_1, D_2, \dots, D_n . Las relaciones son conjuntos (el orden entre las tuplas no importa).
- Se puede pensar una relación como una tabla con columnas para los dominios D_1, D_2, \dots, D_n respectivamente.
- Un **elemento** $t \in r$ es una tupla representado mediante una fila de la tabla.

Notar que el concepto de relación en el modelo relacional no tiene nada que ver con el concepto de relación en el modelo ER. Es más bien como están acostumbrados a trabajar en matemática.

Se puede expresar la información manejada por una organización como un conjunto de relaciones o tablas, en cada columna se deciden los dominios con los que se trabaja para los valores.

Notación: $r(R)$ significa r es una relación con esquema de relación R , es decir, las columnas de r tienen como nombres los atributos de R .

- Los conceptos del problema que se asociaban a esquemas de relación ahora se terminan asociando también a relaciones o tablas de la BD.
- Los dominios de las tablas pasan a ser los dominios de los atributos del esquema relacional.
- Además es más fácil de comprender una tabla donde las columnas tienen los nombres de un esquema de relación.

1.9. Tipos de atributos

El conjunto de valores permitidos para cada atributo se llama **dominio del atributo**. Se requiere que los valores de los atributos sean atómicos; esto es, indivisibles.

A veces no sabemos el valor de un atributo o el mismo aún no existe, para poder reflejar esto en la base de datos se utiliza el valor especial **null**, el cual es un miembro de todo dominio y significa que el valor es desconocido o no existe. Si para una tupla no tenemos el valor de un atributo por algún motivo, podemos poner null como valor para ese atributo.

1.10. DB relacionales

Definiciones:

- Un **esquema de una BD relacional** es un conjunto de esquemas de relación $R_1, R_2 \dots R_n$.

- **Una base de datos relacional** consiste de múltiples relaciones $r_1(R_1), r_2(R_2) \dots r_n(R_n)$
Primero definimos los conceptos del problema mediante esquemas de relación, luego definimos los datos como tablas asociadas a esos esquemas, por último podemos poblar las tablas, consultarlas y alterarlas.
- nombre de relación, expresado en minúscula, $r, s, u, r1, r2, \dots$
- nombre de esquema de relación, expresado en mayúscula $R, S, U, R1, R2, \dots$
- sea $t \in r, r(R), A \in R$. $t[A]$ es el valor de t en A .
- sea $t \in r, r(R)$. $t[i]$ es el valor de t en atributo i -ésimo de R .

1.10.1. Superclaves

Definición: Sea $K \subseteq R$, R esquema de relación; K es una **superclave** de R si los valores para K son suficientes para identificar una tupla única en cada posible relación $r(R)$. Observar que se habla de relaciones para el problema o situación del mundo real que está siendo considerando.

1.10.2. Claves candidatas y primarias

Una superclave K es una **clave candidata** si K es minimal, es decir, para todo atributo de K si se lo quito dejo de tener una superclave.

No confundir clave candidata con superclave de cardinalidad mínima. Una de las claves candidatas es elegida para ser la clave primaria.

Notación: Se indican los atributos de una clave primaria para un esquema de relación R subrayando los atributos de R que forman la clave primaria.

1.10.3. Claves foráneas

Sea R_1 una **relación referenciante** y R_2 una **relación referenciada**, entonces:

- Restricción de clave foránea: Los valores de uno o más atributos en una tupla de R_1 aparecen en uno o más atributos de una tupla en R_2 .
- Los atributos referenciados en R_1 suelen formar una **clave primaria** del esquema de R_2 .
- Los atributos referenciados de R_1 suelen formar una **clave candidata** del esquema de R_2 .

Representación gráfica:

- Los esquemas se representan con rectángulos conteniendo nombre de esquema y nombres de atributos.
- Los atributos de clave primaria se subrayan.
- Las restricciones de clave foránea se representan mediante flechas que van de atributos referenciantes (de esquema referenciante) a atributos referenciados (de esquema referenciado).

1.11. Diseño de una DB relacional

Existen dos formas de diseñar una buena base de datos:

- Aprovechar un buen diseño de esquema ER: si se tiene un buen diseño de ER, se lo puede transicionar a un buen diseño relacional.
- Obtener un buen diseño relacional a partir de los atributos y las restricciones de integridad del problema. La **teoría de normalización** trabaja con esta idea y trata con cómo diseñar buenos esquemas de BD relacionales.

1.12. Reducción a esquemas relacionales

Se tiene un buen diseño de esquema de BD usando modelado-ER que se va a traducir a un conjunto esquemas relacionales con claves primarias y restricciones de clave foránea. Hay reglas de traducción que suelen resolver un porcentaje alto del problema pero esas reglas no contemplan todos los casos.

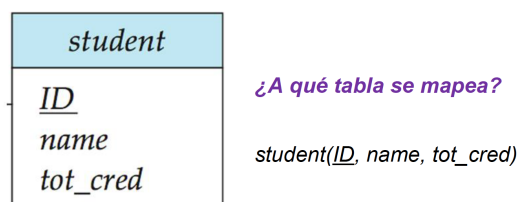
Una propiedad de la traducción de diagramas ER a tablas que se debe respetar es que para cada CE y CR hay un esquema relacional único al cual se asigna el CE / CR.

Además de los esquemas de relación obtenidos por la traducción automática se debe obtener:

- Claves primarias de esquemas de relación: para esto se usa la información de claves primarias de CE.
- Restricciones de clave foránea: para esto se usa el tipo de elemento de modelo ER a mapear (p.ej. CR, CE débiles, atributo multivalorado, etc.).

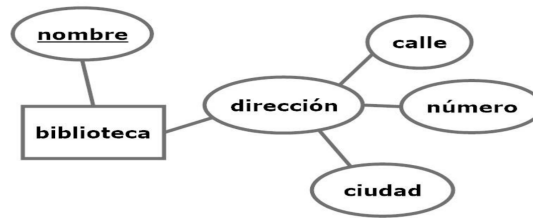
1.12.1. Regla CEF1

Un CE fuerte que no involucra atributos compuestos ni atributos multivalorados se mapea a un esquema relacional con los mismos atributos. La clave primaria del CE se convierte en la clave primaria del esquema relacional.



1.12.2. Regla CEF2

Un CE fuerte que no involucra atributos/subatributos multivalorados se mapea a un esquema relacional con los mismos atributos simples y los subatributos hoja de los atributos compuestos.



¿A qué tabla se mapea?

biblioteca(nombre, calle, número, ciudad)

1.12.3. Regla CEF3

Un atributo multivalorado M simple de un CE E es representado por un esquema separado EM. EM tiene atributos correspondientes a la clave primaria de E y un atributo correspondiente al atributo multivalorado M. Todos los atributos de EM forman su clave primaria. Se pone una restricción de clave foránea desde EM que referencia a la clave primaria de E.



¿A qué tablas se mapea?

libro(título, ISBN, editorial, edición)

libro-autor(ISBN, autor)

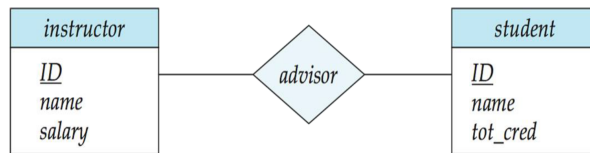
Aclaración: Cada valor del atributo multivalorado mapea a una tupla separada en la tabla del esquema EM.

Por ejemplo un libro con ISBN 970-17-0256-5 y autores con DNIs: 11122256 y 15345678 se mapea a dos tuplas: (970-17-0256-5, 11122256) y (970-17-0256-5, 15345678)

Decisión: Los atributos derivados no son explícitamente representados en el modelo de datos relacional. Se verá que si se los necesita una forma de computarlos es por medio de consultas.

1.12.4. Regla CR1

Un CR **varios a varios** es representado con un esquema con atributos para las claves primarias de los dos CE participantes y todos los atributos descriptivos del CR (que no son multivalorados). La clave primaria del esquema del CR es la unión de las claves primarias de los CE que participan en el CR. Para cada CE que participa en el CR se crea restricción de clave foránea que referencia clave primaria de CE.

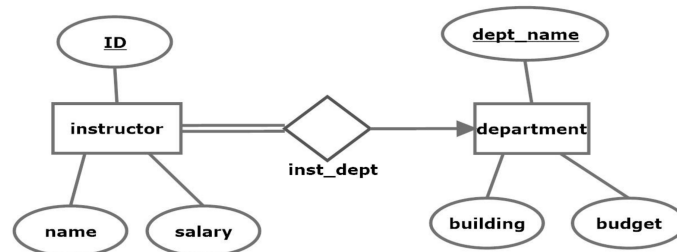


¿A qué esquema se mapea el CR?

$Advisor = (i-id, s-id)$

1.12.5. Regla CR2

Un CR **varios a uno** o **uno a varios** que es total en el lado varios puede ser representado agregando atributos extra en el CE del lado varios, conteniendo la clave primaria del lado uno. La clave primaria del CR es la clave primaria del CE del lado varios. Se crea restricción de clave foránea de CR que referencia a clave primaria de CE de lado varios.



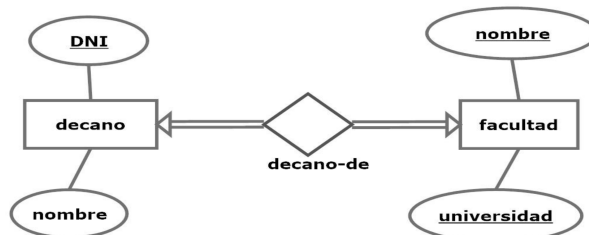
¿A qué tabla se mapea inst_dept?

$instructor(\underline{ID}, name, salary, dept_name)$

Si la participación es parcial en el lado varios, aplicar la regla anterior puede resultar en valores nulos. Esto sucede cuando a una entidad del CE del lado varios no le corresponde ninguna entidad del CE del lado uno.

1.12.6. Regla CR3

Un CR **uno a uno** puede ser mapeado agregando al esquema resultante de traducir uno de los CE participantes los atributos de la clave primaria del otro CE. La clave primaria de cualquier CE puede ser elegida como la clave primaria del CR. Se crea restricción de clave foránea de esquema relacional asociado al CR que referencia clave primaria de otro CE (el que no se tomo de base para hallar el esquema asociado al CR).



¿A qué tabla se mapea decano-de?

Las siguientes respuestas son igualmente válidas:

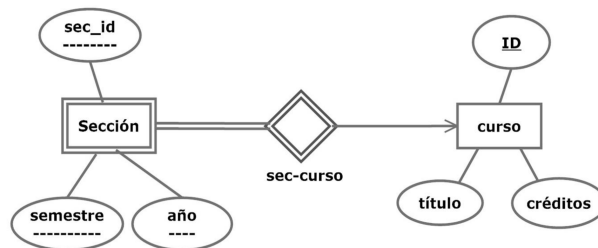
$decano(\underline{DNI}, nombre, nombreFacultad, universidad)$

y

$facultad(nombre, \underline{universidad}, DNI)$

1.12.7. Regla CED

Un CE débiles se mapea a una tabla que incluye columnas para la clave primaria del CE identificador más los atributos (no multi-valorados) del CE débiles (achatando jerarquías de atributos compuestos si es necesario). La clave primaria del CE identificador más el discriminador del CE débil forman la clave primaria del esquema relacional de la traducción. Para atributos de esquema de CE débil que provienen de CE identificadora se agrega restricción de clave foránea desde esquema de CE débil a CE identificador. El CR identificador no se mapea.



¿A qué tabla se mapea el CE débil?

sección(ID, sec_id, semestre, año)

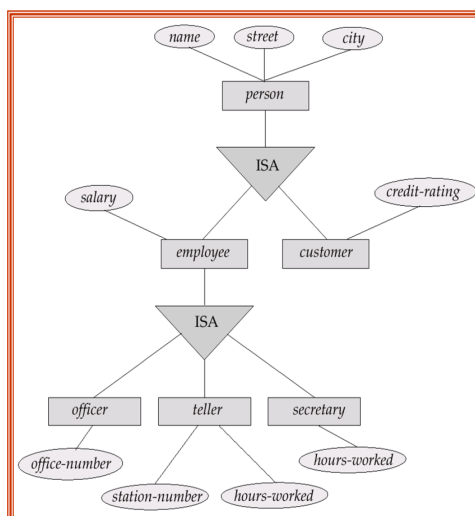
1.12.8. Reducción de generalización

Para traducir generalización iremos proponiendo y evaluando ideas de traducción. Al final decimos bajo qué circunstancias las ideas de traducción son reglas a usar.

Idea G1:

Traducción de generalización a tablas.

- Formar una tabla para el CE de nivel más alto (la generalización).
- Formar una tabla para cada CE especialización que incluye la clave primaria del CE generalización y los atributos locales.



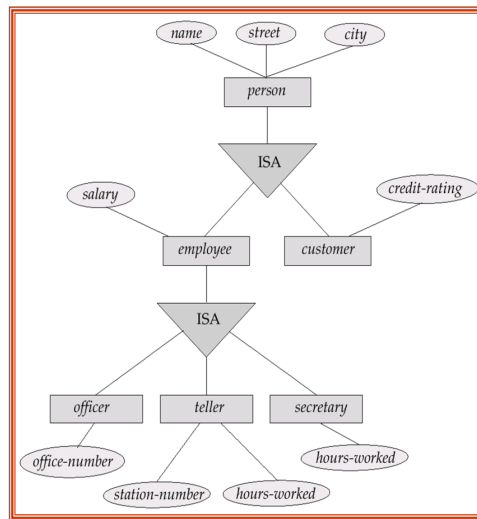
¿A qué tablas se mapea la generalización person?

person(name, street, city)
customer(name, credit-rating)
Employee(name, salary)

Evaluación de Idea G1: obtener toda la información de CE de especialización requiere acceder a dos tablas.

Idea G2:

Formar una tabla para cada CE especialización con los atributos locales y heredados del CE generalización.



¿A qué tablas se mapea la generalización person usando idea G2?

*person(name, street, city)
customer(name, street, city,
credit-rating)
Employee(name, street, city,
salary)*

Evaluación de Idea G2: si dos o más especializaciones contienen la misma entidad, entonces los atributos que no son clave primaria del CE generalización son almacenados redundantemente.

Si la generalización es total con la traducción usando esta idea, la tabla del CE generalización no se requiere para almacenar información.

Idea G3:

Si la generalización es total formar una tabla para cada CE especialización con los atributos locales y heredados; no formar tabla para la generalización. Esta idea es óptima para cuando la generalización es total y disjunta.

En resumen:

- Generalización total y disjunta: usar idea G3.
- Generalización no total y disjunta: usar idea G2.
- Generalización no disjunta: usar idea G1 (para evitar tener que tratar con redundancia de información).

Capítulo 2

Algebra Relacional

Un lenguaje de consultas es un lenguaje en el que un usuario solicita información de la base de datos. Estos lenguajes suelen ser de un nivel superior que el de los lenguajes de programación habituales.

En los lenguajes procedimentales el usuario instruye al sistema para que lleve a cabo una serie de operaciones, en la base de datos para calcular el resultado deseado. Por ejemplo, **álgebra relacional**. Otros lenguajes de consulta a estudiar: SQL (en modelo relacional), XPath y XQuery (en XML).

El AR consiste de un conjunto de operadores sobre relaciones que se pueden componer. Se parte de las relaciones de la BD, a esas relaciones se les aplican operadores del AR.

2.1. Selección

El operador **selección** sirve para definir por comprensión subconjuntos de una relación. Se usa la sintaxis $\sigma_p(r)$, donde

$$\sigma_p(r) := \{t \in r \mid p(t)\}$$

Donde p es predicado de selección.

- p consistente de términos conectados por conectivos proposicionales $\{\wedge, \vee, \neg\}$
- Una relación de comparación rc satisface: $rc \in \{=, <, >, \neq, \geq, \leq\}$
- Dada una rc , un **término** se forma de la siguiente manera:

$$\langle atributo \rangle rc \langle atributo \rangle,$$

$$\langle atributo \rangle rc \langle constante \rangle,$$

$$\langle constante \rangle rc \langle atributo \rangle,$$

A	B	C	D
α	α	1	7
α	β	5	7
β	β	12	3
β	β	23	10

Figura 2.1: Relación r

A	B	C	D
α	α	1	7
β	β	23	10

Figura 2.2: $\sigma_{A=B \wedge D>5}(r)$

2.2. Proyección

Notación: $\Pi_{A_1, A_2 \dots A_k}(r)$

Semántica:

- relación de k columnas obtenida de borrar columnas que no son listadas
- se borran filas duplicadas del resultado

A	B	C
α	10	1
α	20	1
β	30	1
β	40	2

Figura 2.3: Relación r

A	C
α	1
α	1
β	1
β	2

=

A	C
α	1
β	1
β	2

Figura 2.4: $\Pi_{A,C}(r)$

2.3. Proyección generalizada

Notación: $\Pi_{F_1, F_2 \dots F_n}(E)$

Donde los $F_1, F_2 \dots F_n$ son expresiones aritméticas involucrando constantes y atributos de E. Puedo permutar orden de atributos de r .

Ejemplo: obtener el salario mensual de un instructor

$\text{instructor}(\text{ID}, \text{name}, \text{dept_name}, \text{salary})$
 salary: es salario anual

$\sqcap_{\text{ID}, \text{name}, \text{dept_name}, \text{salary} \div 12} (\text{instructor})$

2.4. Unión, diferencia e intersección

Operaciones de conjuntos del mismo nombre.

- $r \cup s = \{t \mid t \in r \vee t \in s\}$
- $r - s = \{t \mid t \in r \wedge t \notin s\}$
- $r \cap s = \{t \mid t \in r \wedge t \in s\}$

Condiciones para que estas operaciones tengan sentido.

1. Ambas relaciones deben tener el mismo número de atributos.
2. Los dominios de atributos de la misma posición deben ser iguales.

2.5. Producto cartesiano

Definición: $r \times s := \{t \mid t \in r \wedge t \in s\}$

Esquema de $r \times s$ ($r(R)$ y $s(S)$):

- Si R y S son disjuntos: $R \cdot S$
- Si R y S no son disjuntos, se debe usar renombramiento.

Ejemplo:

$\text{instructor}(\text{ID}, \text{name}, \text{dept_name}, \text{salary})$
 $\text{teaches}(\text{ID}, \text{course_id}, \text{sec_id}, \text{semester}, \text{year})$

El esquema de $r = \text{instructor} \times \text{teaches}$ es:

$(\text{instructor.ID}, \text{instructor.name}, \text{instructor.dept_name}, \text{instructor.salary}, \text{teaches.ID},$
 $\text{teaches.course_id}, \text{teaches.sec_id}, \text{teaches.semester}, \text{teaches.year})$

El esquema anterior se puede simplificar a:

$(\text{instructor.ID}, \text{name}, \text{dept_name}, \text{salary}, \text{teaches.ID}, \text{course_id}, \text{sec_id}, \text{semester}, \text{year})$

Como consecuencias de la convención anterior, no se puede aplicar a $r \times r$ y tampoco se puede aplicar a $r \times E$, con E expresión de consulta arbitraria.

2.6. Renombramiento

Se quiere poder aplicar operador de AR a expresión E arbitraria de AR, sin embargo para operadores selección y proyección es necesario tener acceso a nombres de atributos, es decir, que se necesita que expresiones arbitrarias de AR tengan nombres para las columnas de las tablas que representan.

A	B			
α	1			
β	2			
<i>r</i>				

C	D	E		
α	10	a		
β	10	a		
β	20	b		
γ	10	b		
<i>s</i>				

Figura 2.5: Relaciones r s

A	B	C	D	E
α	1	α	10	a
α	1	β	10	a
α	1	β	20	b
α	1	γ	10	b
β	2	α	10	a
β	2	β	10	a
β	2	β	20	b
β	2	γ	10	b

Figura 2.6: $r \times s$

¿Cómo dar nombres de atributo a columnas de tabla asociada a expresión de consulta de AR?

La solución está en asociar un esquema a una expresión de ER mediante el mecanismo de renombramiento. Si la tabla de una expresión de AR E tiene aridad n , entonces:

$$\rho_x(A_1, A_2 \dots A_n)(E)$$

retorna la tabla de E con nombre X y con atributos renombrados a $A_1, A_2 \dots A_n$.

Ejemplo: $\sigma_{x.A_1 = r.A_1}(\rho_x(A_1, A_2 \dots A_n)(E) \times r)$, r tabla de la BD.

Notar que necesito dar nombre X a E para poder decir a cuál atributo A_1 me refiero.

A veces solo necesito darle nombre a la tabla de una expresión de AR, por lo que asociamos nombre a una expresión de AR mediante el mecanismo de renombramiento simplificado.

- $\rho_x(E)$ retorna la tabla de E con el nombre X .
- Por ejemplo: para r relación de BD, puedo expresar $(\rho_s(r)) \times r$.

2.7. Definición del AR

Una expresión básica del AR consiste de una de las siguientes:

- Una relación de la BD
- Una relación constante

Si E_1 y E_2 son expresiones del AR, entonces las siguientes son expresiones del AR:

- $E1 \cup E2$
- $E1 - E2$
- $E1 \times E2$
- $\sigma_p(E1)$, p predicado en atributos de E.
- $\pi_s(E1)$, S lista con algunos de los atributos de E1.
- $\rho_x(E1)$, x es un nombre para el resultado de E1.
- $\rho_{x(A_1, \dots, A_n)}(E1)$, A_1, \dots, A_n , son nombres de atributos y x es un nombre para el resultado de E1.

2.8. Vínculo de AR con SQL

En el proyecto de taller trabajaremos con expresiones **select-from-where** de SQL con sintaxis:

- select distinct A_1, \dots, A_n
- from r_1, \dots, r_n
- where P

Son equivalentes a:

$$\pi_{A_1, A_2, \dots, A_n}(\sigma_P(r_1 \times r_2 \times \dots \times r_n))$$

Esto es cierto si r_1, r_2, \dots, r_n son relaciones, sino hay que definir AR para multiconjuntos.

2.9. Reunión natural

Se puede operar con esquemas de relación, para eso podemos definir algunos operadores sobre esquemas. Se comportan como operaciones sobre listas de atributos.

- $S-R$: a S le quito los atributos que también están en R.
- $S.R$: la lista de S seguida de la lista de R (concatenación).
- $R \cup S := R.(S-R)$.

Notación: $r \bowtie s$. Sean $r(R)$ y $s(S)$, entonces $r \bowtie s (R \cup S)$.

$r \bowtie s$ se obtiene como sigue:

- Sean $t_r \in r$ y $t_s \in s$ tal que t_r y t_s tienen los mismos valores en $R \cap S$ (o sea t_r y t_s cazan)
- Si t concuerda con t_r en R y t concuerda con t_s en S, entonces $t \in r \bowtie s$.

Ejemplo:

- $R = (A, B, C, D)$
- $S = (E, B, D)$
- $R \cup S = (A, B, C, D, E)$

A	B	C	D
α	1	α	a
β	2	γ	a
γ	4	β	b
α	1	γ	a
δ	2	β	b

r

B	D	E
1	a	α
3	a	β
1	a	γ
2	b	δ
3	b	ϵ

s

Figura 2.7: Relaciones r y s

A	B	C	D	E
α	1	α	a	α
α	1	α	a	γ
α	1	γ	a	α
α	1	γ	a	γ
δ	2	β	b	δ

Figura 2.8: $r \bowtie s$

$$\blacksquare r \bowtie s = \Pi_{r.A, r.B, r.C, r.D, s.E}(\sigma_{r.B=s.B \wedge r.D=s.D}(r \times s))$$

■ En general:

$$r \bowtie s = \Pi_{R \cup S}(\sigma_{r.A_1=s.A_1 \wedge \dots \wedge r.A_n=s.A_n}(r \times s)), \text{ donde } R \cap S = \{A_1, \dots, A_n\}$$

Algunas propiedades interesantes: $r(R)$, $s(S)$, $u(U)$:

$$\blacksquare (r \bowtie s) \bowtie u = (r \bowtie (s \bowtie u))$$

$$\blacksquare R \cap S = \emptyset \Rightarrow r \bowtie s = r \times s$$

$$\blacksquare R = S \Rightarrow r \bowtie s = r \cap s$$

Ejemplo:

- persona(DNI, nombre, apellido)
- libro(título, ISBN, editorial, edición)
- libroBib(numInv, ISBN, nombreBib)
- prestadoA(numInv, nombreBib, DNI)
- socio(posición, DNI)
- inscriptoEn(DNI, nombreBib)

Obtener los ISBN de los libros prestados a Juan Pérez en la biblioteca de “FaMAF”.

2.10. División

La operación división, resulta adecuada para muchas consultas que incluyen la frase “para todo/s”.

Notación: $r \div s$.

Definición: Sea $r(R)$ y $s(S)$ donde

- $R = (A_1, \dots A_m, B_1, \dots B_n)$
- $S = (B_1 \dots B_n)$
- $r \div s(R - S)$ y $R - S = (A_1, \dots A_m)$
- $r \div s := \{t \in \Pi_{R-S}(r) \mid \forall u \in s(t.u \in r)\}$

A	B	C	D	E
α	a	α	a	1
α	a	γ	a	1
α	a	γ	b	1
β	a	γ	a	1
β	a	γ	b	3
γ	a	γ	a	1
γ	a	γ	b	1
γ	a	β	b	1

r

D	E
a	1
b	1

s

Figura 2.9: Relaciones r y s

A	B	C
α	a	γ
γ	a	γ

Figura 2.10: $r \div s$

Sea $q = r \div s$ entonces q es la relación más grande que satisface $q \times s \subseteq r$.

Ejemplo:

- libro(título, ISBN, editorial, edición)
- libroBib(numInv, ISBN, nombreBib)
- biblioteca(nombreBib, calle, número)

Encontrar los títulos de los libros que están en todas las bibliotecas.

Proceso para hacer una consulta con división:

1. Identificar tablas a usar
2. Reescribir consulta en español de modo que quede más claro dónde se usa división y la división sea más evidente.
3. Identificar partes de la división
 - Escribirlas primero en español
 - Luego pasarlas a AR
4. Usar proyecciones para que partes cumplan con esquemas exigidos por división.
5. Escribir división, chequearla usando la definición de división y corregirla.
6. Hacer la consulta completa en AR que involucra división.

2.11. Agregación

Función de Agregación: toma una colección de valores y retorna un valor simple como resultado. En AR se consideran las siguientes funciones de agregación: avg, min, max, sum, count.

Muchas veces ocurre que debemos eliminar múltiples ocurrencias de un valor antes de computar una función de agregación. Si queremos eliminar duplicados usamos los mismos nombres de función, con la adición de *distinct* al final.

Operación de agregación en AR

$$G_{F_1(A_1), F_2(A_2), \dots, F_n(A_n)}(E)$$

- E expresión de AR
- Cada F_i es una función de agregación
- Cada A_i es un nombre de atributo

Semántica: el resultado tiene una tupla (a_1, a_2, \dots, a_n) donde, para cada i , a_i es el resultado de aplicar la función de agregación F_i al multiconjunto de valores del atributo A_i en E.

Otra operación de agregación en AR más compleja

$$G_1, G_2, \dots, G_n \ G_{F_1(A_1), F_2(A_2), \dots, F_n(A_n)}(E)$$

- E expresión de AR
- G_1, G_2, \dots, G_n lista de atributos en los cuales agrupar (puede ser vacío)
- Cada F_i es una función de agregación
- Cada A_i es un nombre de atributo

Semántica:

- Las tuplas en el resultado de la expresión E se dividen en grupos tales que:
 1. Todas las tuplas del grupo tienen los mismos valores para G_1, G_2, \dots, G_n .
 2. Las tuplas de grupos diferentes tienen valores diferentes para G_1, G_2, \dots, G_n .
- Para cada grupo asociado a (g_1, g_2, \dots, g_n) el resultado tiene una tupla $(g_1, g_2, \dots, g_n, a_1, a_2, \dots, a_m)$, donde, para cada i , a_i es el resultado de aplicar la función de agregación F_i al multiconjunto de valores del atributo A_i en el grupo asociado a (g_1, g_2, \dots, g_n) .

Ejemplo:

- bibliotecario(antigüedad, DNI)
- trabajaEn(DNI, nombreBib)

Obtener el promedio de las antigüedades de los bibliotecarios que trabajan en cada biblioteca.

2.12. Reuniones externas

Sea $r(R)$ y $s(S)$, se quiere expresar información de casamiento de tuplas de r y de s y también información de tuplas de r -tuplas de s que no cazan con tuplas de la otra tabla. Esto no es posible mediante $r \bowtie s$ ya que solo informa para cada tupla de r con que tuplas de s caza.

Para resolver este problema se definen operadores especiales que son extensiones de la reunión natural que evitan pérdida de informaciones de tuplas que no cazan (ya sea en una o ambas de las tablas), para ello se computa la reunión natural y luego se agregan tuplas de una relación que no cazan con las tuplas en la otra relación. **Usa valores nulos.**

Ejemplo:

<i>ID</i>	<i>name</i>	<i>dept_name</i>
10101	Srinivasan	Comp. Sci.
12121	Wu	Finance
15151	Mozart	Music

Figura 2.11: Relación Instructor

<i>ID</i>	<i>course_id</i>
10101	CS-101
12121	FIN-201
76766	BIO-101

Figura 2.12: Relación Teaches

- Reunión natural

<i>ID</i>	<i>name</i>	<i>dept_name</i>	<i>course_id</i>
10101	Srinivasan	Comp. Sci.	CS-101
12121	Wu	Finance	FIN-201

Figura 2.13: Instructor \bowtie Teaches

- Reunión externa a la izquierda

<i>ID</i>	<i>name</i>	<i>dept_name</i>	<i>course_id</i>
10101	Srinivasan	Comp. Sci.	CS-101
12121	Wu	Finance	FIN-201
15151	Mozart	Music	<i>null</i>

Figura 2.14: Instructor $_ \bowtie$ Teaches

- Reunión externa a la derecha
- Reunión externa full

Las reuniones externas pueden ser expresadas usando operadores básicos, por ejemplo $r \bowtie s$ puede escribirse como:

$$(r \bowtie) \cup (r - \sqcap_R(r \bowtie s)) \times \{null, null \dots null\}$$

<i>ID</i>	<i>name</i>	<i>dept_name</i>	<i>course_id</i>
10101	Srinivasan	Comp. Sci.	CS-101
12121	Wu	Finance	FIN-201
76766	null	null	BIO-101

Figura 2.15: Instructor \bowtie Teaches

<i>ID</i>	<i>name</i>	<i>dept_name</i>	<i>course_id</i>
10101	Srinivasan	Comp. Sci.	CS-101
12121	Wu	Finance	FIN-201
15151	Mozart	Music	null
76766	null	null	BIO-101

Figura 2.16: Instructor $_ \bowtie _$ Teaches

2.13. Manejo de valores nulos

El resultado de una expresión aritmética involucrando null es null. Las funciones de agregación ignoran valores nulos (decisión arbitraria). Para eliminación de duplicados y agrupamientos a null se lo trata como a cualquier otro valor.

Comparaciones (como $\geq, \leq, =, >, \neq$) con valores null retorna el valor de verdad *unknown*. Las comparaciones que incluyan nulos pueden aparecer dentro de expresiones booleanas con conectivos.

Una lógica de tres valores de verdad se usa con el valor de verdad unknown.

- OR: (*unknown* or true) = true,
(*unknown* or false) = *unknown*,
(*unknown* or unknown) = *unknown*
- AND: (true and *unknown*) = *unknown*,
(false and *unknown*) = false,
(*unknown* and *unknown*) = *unknown*
- NOT: (not *unknown*) = *unknown*
- **Selección:** evalúa el predicado P en $\sigma_P(E)$ sobre cada tupla de E. Si P(t) devuelve el valor cierto, se añade t al resultado, caso contrario, si P(t) devuelve desconocido o falso, t no se añade al resultado.
- **Reunión natural** $r \bowtie s$: si dos tuplas $t_r \in r$ y $t_s \in s$, tienen un valor nulo en un atributo común, entonces las tuplas no casan.
- **Proyección:** trata los nulos como cualquier otro valor al eliminar duplicados.
- **Unión, intersección y diferencia:** tratan las tuplas que tienen los mismos valores en todos los campos como duplicados incluso si algunos de los campos tienen valores nulos en ambas tuplas.
- **Agregación:** Cuando hay valores nulos en los atributos agregados, la función de agregación a ser aplicada borra los valores nulos antes de aplicarse. Si el multiconjunto resultante está vacío, el resultado de la función de agregación es null.

2.14. Asignación

La asignación (\leftarrow) provee una manera de expresar consultas complejas. También sirve para expresar alteraciones a la BD.

Escribir la consulta como un programa secuencial consistente de:

1. una serie de asignaciones
2. una expresión cuyo valor se muestra como resultado de la consulta

La asignación se hace sobre una variable relacional temporal, es decir, su valor es una tabla.
Ejemplo:

- $\text{persona}(\text{DNI}, \text{nombre}, \text{apellido})$
- $\text{biblioteca}(\text{nombreBib}, \text{calle}, \text{número})$
- $\text{bibliotecario}(\text{antigüedad}, \text{DNI})$
- $\text{trabajaEn}(\text{DNI}, \text{nombreBib})$

Dar las bibliotecas en las que trabajan los bibliotecarios de apellido Pérez.

- $r \leftarrow \sigma_{\text{apellido}=\text{"Perez"}}(\text{persona} \bowtie \text{bibliotecario})$
- $s \leftarrow \sqcap_{\text{antigüedad}, \text{DNI}}(r)$
- $\sqcap_{\text{nombreBib}}(s \bowtie \text{trabajaEn})$