DECSAI
Departamento de Ciencias de la Computación e I.A.
Universidad de Granada



# Integración de datos - Esquemas

Fernando Berzal, berzal@acm.org

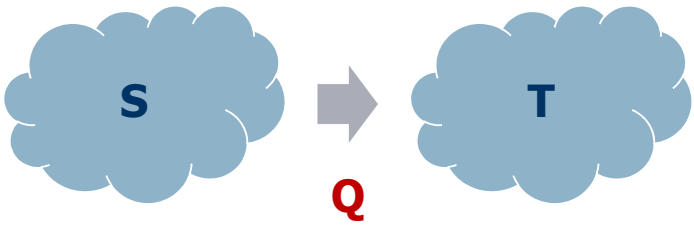
## Integración de datos

- Descripción de fuentes de datos
- Integración de esquemas
  - Emparejamiento de esquemas [schema matching]
  - Correspondencias entre esquemas [schema mapping]
  - Gestión de modelos
- Emparejamiento de datos [data matching]
- Wrappers
- Apéndices:
  - Emparejamiento de cadenas [string matching]
  - Procesamiento de consultas

# Integración de esquemas

Establecer la correspondencia más adecuada entre distintos esquemas es el problema más difícil en integración de datos (heterogeneidad semántica).

Dados dos esquemas relacionales S y T,
una correspondencia semántica es una expresión (una consulta) que relaciona el esquema S con el esquema T.






# Integración de esquemas

**Ejemplo**

DVD-VENDOR
**Movies**(id, title, year)
**Products**(mid, releaseDate, releaseCompany, basePrice, rating, saleLocID)
**Locations**(lid, name, taxRate)

AGGREGATOR
**Items**(name, releaseInfo, classification, price)

**Movies.title**

```
SELECT name as title
FROM Items
```

**Items.price**

```
SELECT (basePrice * (1 + taxRate)) AS price
FROM Products, Locations
WHERE Products.saleLocID = Locations.lid
```

# Integración de esquemas

**Ejemplo**

DVD-VENDOR
**Movies**(id, title, year)
**Products**(mid, releaseDate, releaseCompany, basePrice, rating, saleLocID)
**Locations**(lid, name, taxRate)

AGGREGATOR
**Items**(name, releaseInfo, classification, price)

**Items**

```
SELECT title AS name,
       releaseDate AS releaseInfo,
       rating AS classification,
       basePrice * (1 + taxRate) AS price
FROM Movies, Products, Locations
WHERE Movies.id = Products.mid
   AND Products.saleLocID = Locations.lid
```



# Integración de esquemas

Imaginemos que estamos diseñando el esquema integrado de un sistema de integración de datos con varias fuentes de datos:

- **GAV [Global-as-View]:** Describimos el esquema integrado como consultas sobre las fuentes de datos.
- **LAV [Local-as-View]:** Describimos las fuentes de datos como consultas sobre el esquema integrado.
- **GLAV [Global-and-Local-as-View]**: Hay que establecer correspondencias en ambos sentidos.

# Integración de esquemas

**Fase 1: SEMANTIC MATCHING**

Relaciones de conjuntos de elementos de un esquema S con conjuntos de elementos de otro esquema T (elicitadas usando conocimiento del dominio).

- Emparejamientos uno a uno:

  Movies.title = Items.name
  Products.rating = Items.classification

- Emparejamientos uno a muchos:

  Items.price = Products.basePrice * (1 + Locations.taxRate)



# Integración de esquemas

**Fase 2: SEMANTIC MAPPING**

A partir de las relaciones funcionales identificadas, se elaboran las consultas necesarias para establecer la correspondencia entre esquemas (p.ej. usando SQL).

- Emparejamiento:

  Items.price = Products.basePrice * (1 + Locations.taxRate)

- Correspondencia:

```
SELECT (basePrice * (1 + taxRate)) AS price
FROM Product, Location
WHERE Product.saleLocID = Location.lid
```

# Integración de esquemas

## Recordatorio: Heterogeneidad Semántica

Hay que reconciliar las diferencias entre esquemas:

- Distintos nombres para el mismo concepto.
  rating vs. classification
- Distinta representación en esquemas diferentes.
  basePrice & taxRate vs. Price
- Diferente organización del esquema
  1 tabla en Aggregator vs. 3 tablas en DVD-Vendor
- Distinto grado de detalle en las diferentes fuentes
  DVD-Vendor incluye releaseDate & releaseCompany



# Integración de esquemas

## Observación clave

Se necesitan múltiples heurísticas para establecer el emparejamiento entre dos esquemas:

- Emparejando nombres, podríamos inferir las correspondencias releaseInfo=releaseDate o releaseInfo=releaseCompany, pero no establecer cuál de las dos sería la correcta.
- Emparejando datos, podríamos inferir que releaseInfo=releaseDate o releaseInfo=year, pero sigue existiendo ambigüedad.
- Combinando ambos resultados, se puede inferir que releaseInfo=releaseDate.

# Integración de esquemas

***realestate.com***

| listed-price | contact-name | contact-phone | office | comments |
|---|---|---|---|---|
| $250K<br>$320K<br>..... | James Smith<br>Mike Doan<br>..... | (305) 729 0831<br>(617) 253 1429<br>..... | (305) 616 1822<br>(617) 112 2315<br>..... | Fantastic house<br>Great location<br>..... |

***homes.com***

| sold-at | contact-agent | extra-info |
|---|---|---|
| $350K<br>$230K | (206) 634 9435<br>(617) 335 4243 | Beautiful yard<br>Close to Seattle |

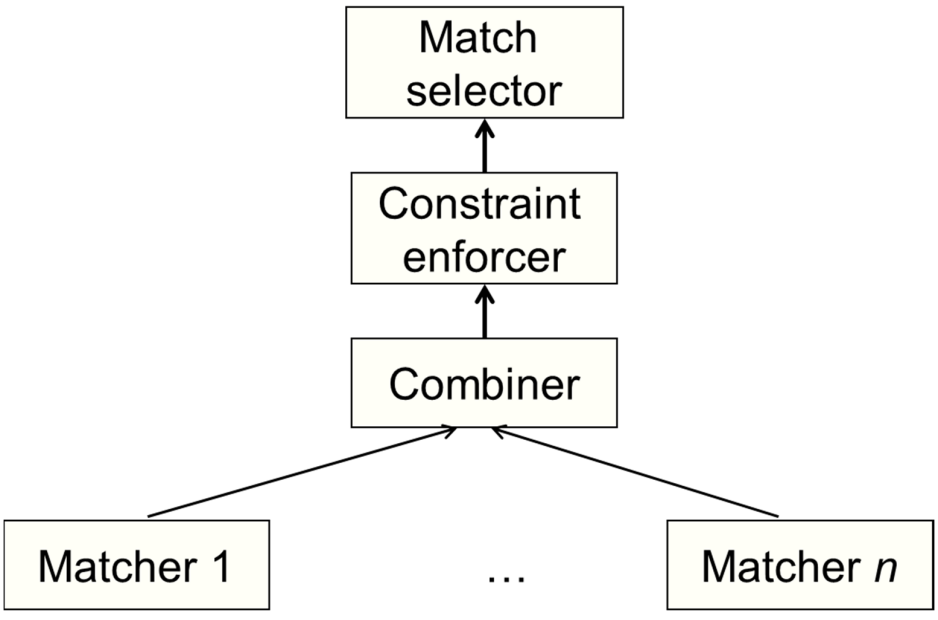
- Emparejando nombres, contact-agent puede casar con contact-name o contact-phone.
- Emparejando datos, contact-agent puede casar con contact-phone u office.
- Combinando ambos resultados, contact-agent=contact-phone.



# Integración de esquemas

**Schema matching**

Arquitectura de un sistema de emparejamiento

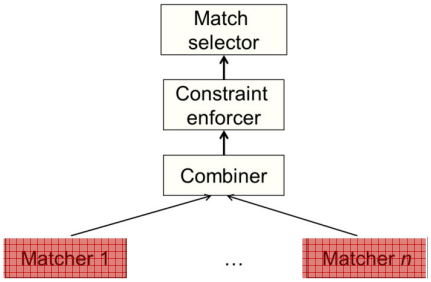


# Integración de esquemas

**Schema matching**

TÉCNICAS DE EMPAREJAMIENTO

Esquemas ➔ Matriz de similitud




## Entradas

Dos esquemas S y T

+ Información potencialmente útil (datos reales, descripciones textuales…)

## Salida

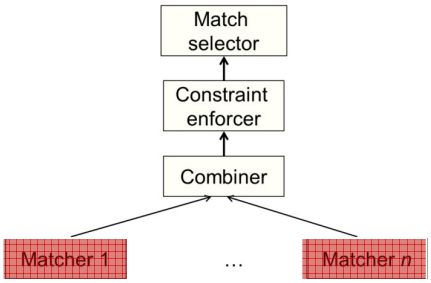
Matriz de similitud que asigna a cada par de SxT un número entre 0 y 1 en función de su nivel de emparejamiento.



# Integración de esquemas

**Schema matching**

TÉCNICAS DE EMPAREJAMIENTO




Familias de técnicas:

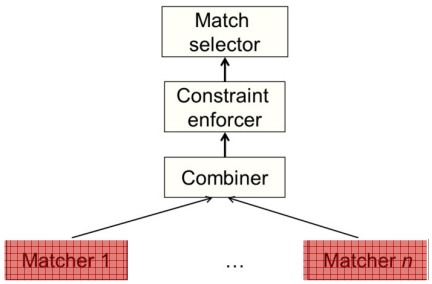
- Emparejamiento de nombres
- Emparejamiento de datos

# Integración de esquemas

## Schema matching

TÉCNICAS DE EMPAREJAMIENTO

### Emparejamiento de nombres




- Algoritmos sobre cadenas
  p.ej. Distancia de edición

d("data mining", "data minino") = 1
d("efecto", "defecto") = 1
d("poda", "boda") = 1
d("night","natch") = d("natch","noche") = 3



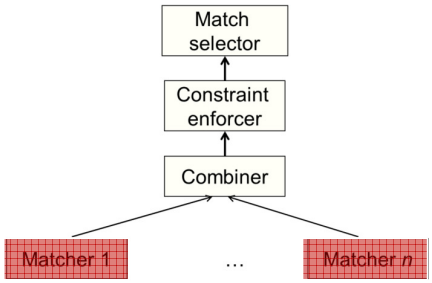
# Integración de esquemas

## Schema matching

TÉCNICAS DE EMPAREJAMIENTO

### Emparejamiento de nombres

Técnicas de preprocesamiento




- División de términos compuestos:
  saleLocID ➔ sale, Loc, ID
- Expansión de abreviaturas y acrónimos:
  loc ➔ location, cust ➔ customer
- Expansión con sinónimos, hipónimos e hiperónimos:
  price ➔ cost, product ➔ book, dvd, cd
- Eliminación de "stop words"
  the, a, in, at, of, and…

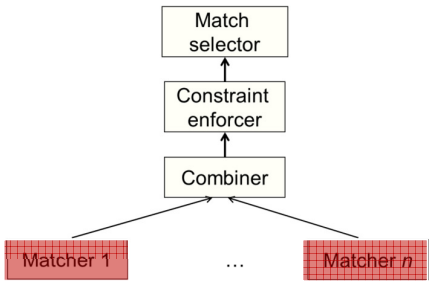
# Integración de esquemas

## Schema matching

TÉCNICAS DE EMPAREJAMIENTO

### Emparejamiento de nombres

EJEMPLO




DVD-VENDOR
**Movies**(id, title, year)
**Products**(mid, releaseDate, releaseCompany, basePrice, rating, saleLocID)
**Locations**(lid, name, taxRate)

AGGREGATOR
**Items**(name, releaseInfo, classification, price)

**name-base matcher:**
name = <name: 1, title: 0.2>
releaseInfo = <releaseDate: 0.5, releaseCompany: 0.5>
price = <basePrice: 0.8>

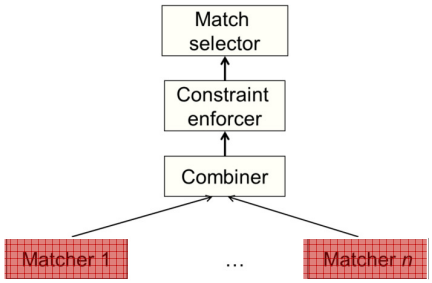


# Integración de esquemas

## Schema matching

TÉCNICAS DE EMPAREJAMIENTO

### Emparejamiento de datos




Cuando disponemos de datos, éstos pueden ser muy útiles en la integración de esquemas.

Algunas técnicas:

- Reconocedores (diccionarios, regexps y reglas)
- Solapamiento (mismos valores en los atributos)
- Clasificadores (técnicas de aprendizaje supervisado)

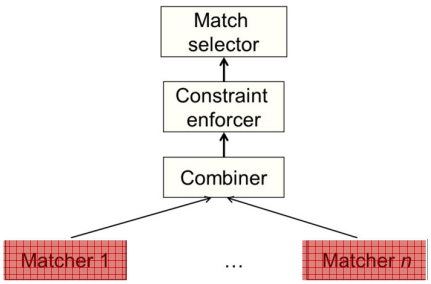
# Integración de esquemas

**Schema matching**

TÉCNICAS DE EMPAREJAMIENTO

**Emparejamiento de datos**

**Reconocedores**




Uso de diccionarios, expresiones regulares y reglas simples para reconocer valores de ciertos tipos de atributos.

Útiles para nombres propios (países, ciudades, personas…), apellidos, colores, teléfonos, DNIs, direcciones de correo electrónico, URLs, códigos postales, ratings, medicamentos, genes, proteínas…



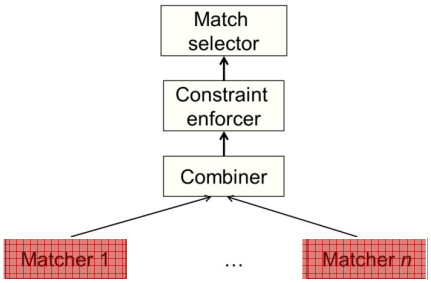
# Integración de esquemas

**Schema matching**

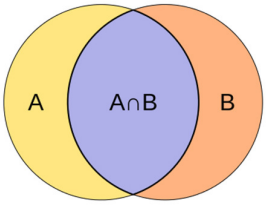
TÉCNICAS DE EMPAREJAMIENTO

**Emparejamiento de datos**

**Solapamiento**




Útil para atributos cuyos valores pertenecen a un dominio finito (p.ej. países, provincias, títulos, ratings…).




Habitualmente se utiliza el coeficiente de Jaccard:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

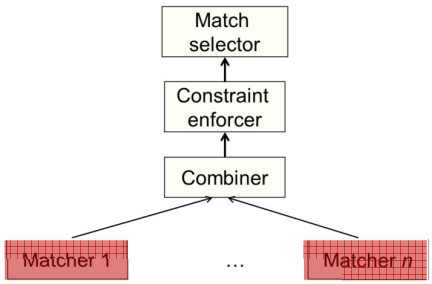
# Integración de esquemas

**Schema matching**

TÉCNICAS DE EMPAREJAMIENTO

**Emparejamiento de datos**

EJEMPLO




DVD-VENDOR
**Movies**(id, title, year)
**Products**(mid, releaseDate, releaseCompany, basePrice, rating, saleLocID)
**Locations**(lid, name, taxRate)

AGGREGATOR
**Items**(name, releaseInfo, classification, price)

**data-based matcher:**
name = <name: 0.2, title: 0.5>
releaseInfo = <releaseDate: 0.7>
classification = <rating: 0.6>
price = <basePrice: 0.2>



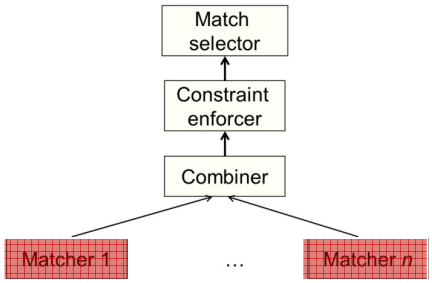
# Integración de esquemas

**Schema matching**

TÉCNICAS DE EMPAREJAMIENTO

**Emparejamiento de datos**

**Clasificadores (I.A.)**




Construcción de clasificadores sobre un esquema para clasificar los elementos del otro esquema.

Múltiples técnicas:
Naive Bayes, árboles de decisión, SVMs…

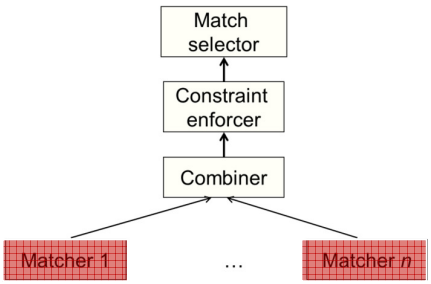
# Integración de esquemas

**Schema matching**

TÉCNICAS DE EMPAREJAMIENTO

**Emparejamiento de datos**

**Clasificadores (I.A.)**




Para cada elemento $s_i$ del esquema S,
se entrena un clasificador $C_i$ para reconocer instancias de $s_i$.

Aprendizaje supervisado: Se necesitan ejemplos etiquetados.

- Ejemplos positivos:
  Instancias disponibles de $s_i$.
- Ejemplos negativos:
  Instancias de otros elementos de S.



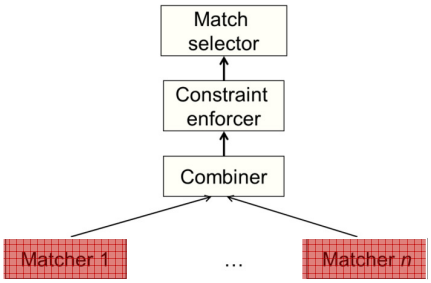
# Integración de esquemas

**Schema matching**

TÉCNICAS DE EMPAREJAMIENTO

**Emparejamiento de datos**

**Clasificadores (I.A.)**




Una vez creado el clasificador $C_i$, se utiliza para determinar la similitud entre $s_i$ y cada elemento $t_j$ del esquema T.

- Para cada instancia de $t_j$, se aplica $C_j$, que indica con un número [0,1] su confianza en que la instancia sea realmente una instancia de $s_i$.
- Los valores asociados a las distintas instancias de $t_j$ se agregan para obtener la similitud entre $s_i$ y $t_j$ (p.ej. valor medio sobre todas las instancias de $t_j$).

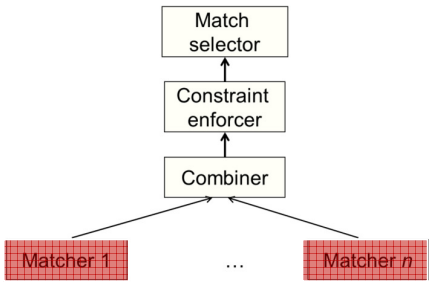
# Integración de esquemas

**Schema matching**

TÉCNICAS DE EMPAREJAMIENTO

**Emparejamiento de datos**

**Clasificadores (I.A.)**




EJEMPLO

SCHEMA S

| current-showing | address | phone |
|---|---|---|
| Lord of the Rings | Madison WI | (608) 695 2311 |
| | Mountain View CA | (650) 277 1358 |

SCHEMA T

| name | location | phone |
|---|---|---|
| … | Milwaukee WI | … |
| | Palo Alto CA | |
| | Philadelphia PA | |

**$s_i$ = address**
**$t_j$ = location**
Sim scores (0.9, 0.7, 0.5)
$sim(s_i,t_j) = 0.7$



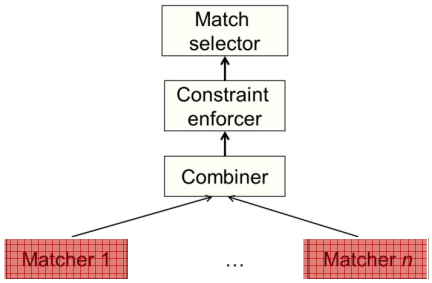
# Integración de esquemas

**Schema matching**

TÉCNICAS DE EMPAREJAMIENTO

**Emparejamiento de datos**

**Clasificadores (I.A.)**



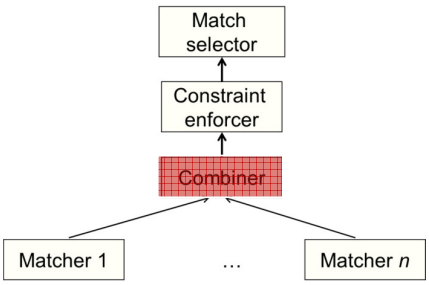

- El diseñador elige qué esquema hace de S (esto es, sobre cuál se construyen los clasificadores).
- Normalmente, será el esquema integrado para poder reutilizar los clasificadores cuando se añadan nuevas fuentes de datos.

NOTA: También se puede hacer en ambos sentidos: Clasificadores sobre S para clasificar instancias de T, y clasificadores sobre T para clasificar instancias de S.

# Integración de esquemas

## Schema matching

COMBINACIÓN DE PREDICCIONES




Una vez calculada la similitud entre los elementos de los esquemas S y T utilizando distintas técnicas,
se pueden combinar las medidas de similitud utilizando distintas funciones de agregación:

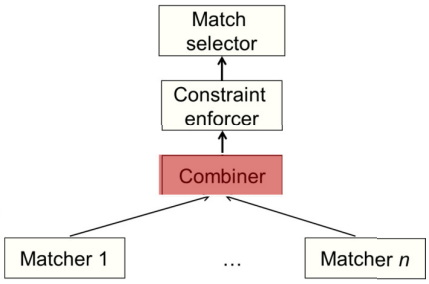
- Media aritmética
- Mínimo
- Máximo
- ...



# Integración de esquemas

## Schema matching

COMBINACIÓN DE PREDICCIONES




DVD-VENDOR
**Movies**(id, title, year)
**Products**(mid, releaseDate, releaseCompany, basePrice, rating, saleLocID)
**Locations**(lid, name, taxRate)

AGGREGATOR
**Items**(name, releaseInfo, classification, price)

**name-base matcher:**
name = <name: 1, title: 0.2>
releaseInfo = <releaseDate: 0.5, releaseCompany: 0.5>
price = <basePrice: 0.8>

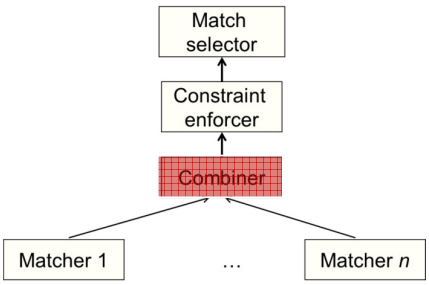
**data-based matcher:**
name = <name: 0.2, title: 0.5>
releaseInfo = <releaseDate: 0.7>
classification = <rating: 0.6>
price = <basePrice: 0.2>

**average-combiner:**
name = <name: 0.6, title: 0.5>
releaseInfo = <releaseDate: 0.6, releaseCompany: 0.25>
classification = <rating: 0.3>
price = <basePrice: 0.5>

# Integración de esquemas

## Schema matching

COMBINACIÓN DE PREDICCIONES




¿Cuándo usar cada función de agregación?

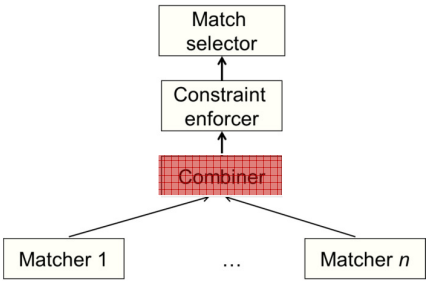
- **Media aritmética:**
  Cuando no tenemos razones para creer que un algoritmo de emparejamiento es mejor que otro.
- **Máximo:**
  Cuando confiemos en una señal "fuerte" de una de las técnicas de emparejamiento.
- **Mínimo:**
  Cuando queramos ser conservadores.



# Integración de esquemas

## Schema matching

COMBINACIÓN DE PREDICCIONES




Alternativas más complejas:

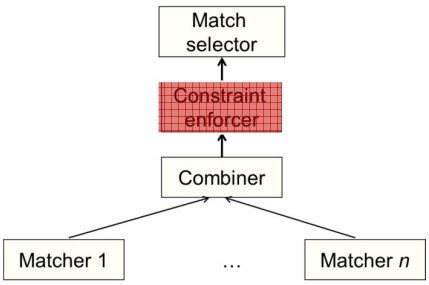
- Criterios "ad hoc", p.ej. utilizar sólo técnicas de emparejamiento de datos cuando se trate de direcciones o números de teléfono.
- Media ponderada (con los pesos ajustados utilizando un conjunto de entrenamiento, p.ej. regresión lineal).
- Algoritmo de aprendizaje (utilizar alguna técnica de I.A. que aprenda cómo combinar las medidas).

# Integración de esquemas

## Schema matching

RESTRICCIONES DE INTEGRIDAD

Incorporación del conocimiento que tengamos acerca de los esquemas:




Expresamos dicho conocimiento en forma de restricciones de integridad, de forma que se eliminen las combinaciones que no satisfagan dichas restricciones.

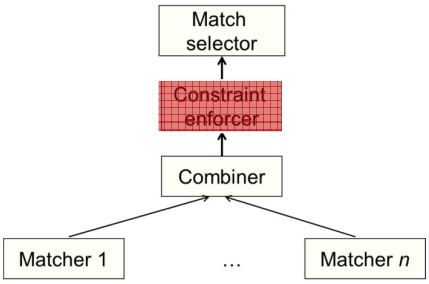
- Algoritmo de búsqueda, p.ej. A*



# Integración de esquemas

## Schema matching

RESTRICCIONES DE INTEGRIDAD




**average-combiner:**
name = <name: 0.6, title: 0.5>
releaseInfo = <releaseDate: 0.6, releaseCompany: 0.25>
classification = <rating: 0.3>
price = <basePrice: 0.5>

Distintas combinaciones posibles

p.ej. $M_1$ = {name = name, 0.6
releaseInfo = releaseDate, 0.6
classification = rating, 0.3
price = basePrice } 0.5

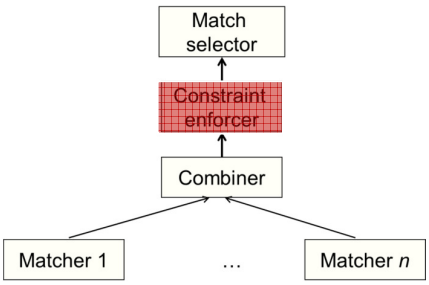
score($M_1$) = 0.6*0.6*0.3*0.5

# Integración de esquemas

**Schema matching**

RESTRICCIONES DE INTEGRIDAD

Ejemplo




Si sabemos que AGGREGATOR.name hace referencia al título de la película y que muchos títulos contienen al menos cuatro palabras, podemos establecer una restricción del tipo:

si un atributo A empareja con AGGREGATOR.name, en cualquier muestra aleatoria de 100 valores de A tiene que haber al menos 10 con 4 palabras o más.

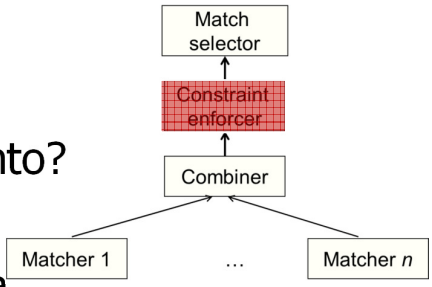


# Integración de esquemas

**Schema matching**

RESTRICCIONES DE INTEGRIDAD

¿Cómo se busca el mejor emparejamiento?




- Idealmente, por orden de preferencia, se van comprobando restricciones hasta encontrar una solución que satisfaga todas las restricciones.

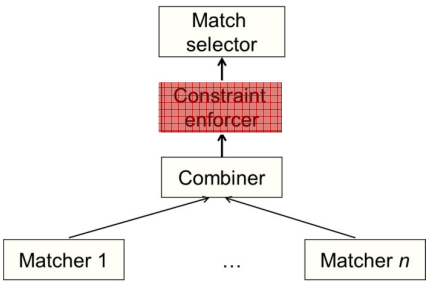
- En la práctica, se debe manejar un conjunto amplio de restricciones, en ocasiones contradictorias, por lo que se debe diseñar un método de búsqueda eficiente…

# Integración de esquemas

## Schema matching

RESTRICCIONES DE INTEGRIDAD

Tipos de restricciones




### Restricciones estrictas [hard]

- Obligatorias.
- Cualquier solución admisible debe respetarlas.

### Restricciones flexibles [soft]

- De tipo heurístico.
- Pueden violarse, aunque intentaremos maximizar su grado de cumplimiento



# Integración de esquemas

## Schema matching

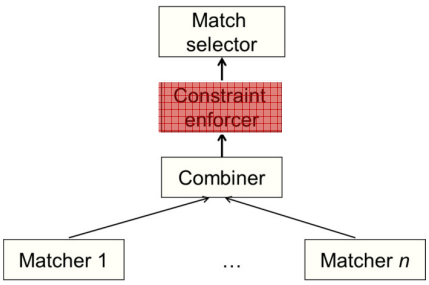
RESTRICCIONES DE INTEGRIDAD

| | Restricciones | Coste |
|---|---|---|
| $c_1$ | If A = Items.code, then A is a key | ∞ |
| $c_2$ | If A = Items.desc, then any random sample of 100 data instances of A must have an average length of at least 10 words | 1.5 |
| $c_3$ | If $A_1 = B_1$, $A_2 = B_2$, $B_2$ is next to $B_1$ in the schema, but $A_2$ is not next to $A_1$, then there is no A* next to $A_1$ such that $\|sim(A^*,B_2) - sim(A_2,B_2)\| \leq t$ for a small pre-specified t | 2 |
| $c_4$ | If more than half of the attributes of Table U match those of Table V, then U = V | 1 |

# Integración de esquemas

**Schema matching**

RESTRICCIONES DE INTEGRIDAD




Algoritmos de búsqueda:

- **Algoritmo A***
  (garantiza la solución óptima, de acuerdo con los costes asociados a las diferentes restricciones).

- **Propagación local**
  (técnica de búsqueda local, más eficiente)

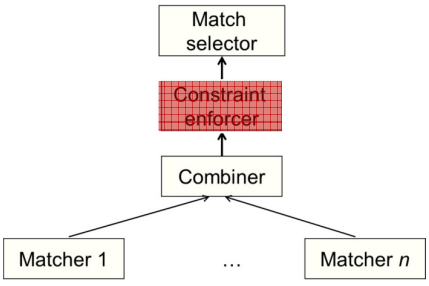


# Integración de esquemas

**Schema matching**

RESTRICCIONES DE INTEGRIDAD

**Algoritmo A***




Técnica de búsqueda heurística:

- **g(n)**
  Coste de llegar al nodo actual (hacia atrás).
- **h(n)**
  Coste estimado para llegar a la solución (hacia adelante).

El algoritmo A* guía la búsqueda utilizando la suma:

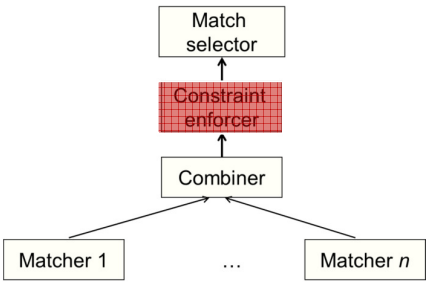
**f(n) = g(n) + h(n)**

# Integración de esquemas

**Schema matching**

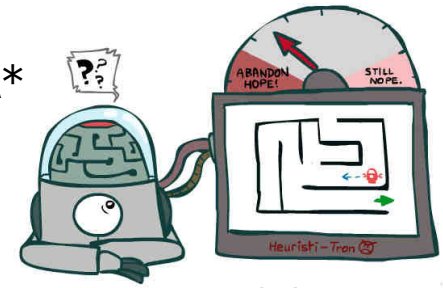
RESTRICCIONES DE INTEGRIDAD

**Algoritmo A***




Para garantizar la solución óptima, la heurística h(n) debe ser admisible (optimista).

- Las heurísticas pesimistas (inadmisibles) impiden la optimalidad del algoritmo A* al descartar buenos planes.
- Las heurísticas optimistas (admisibles) no subestiman la calidad de un buen plan.




Berkeley CS188

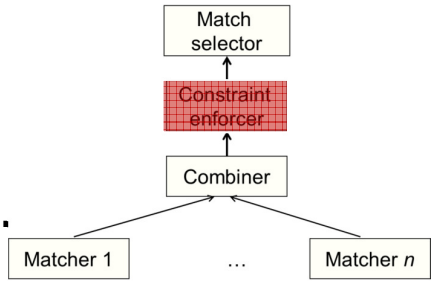


# Integración de esquemas

**Schema matching**

RESTRICCIONES DE INTEGRIDAD

**Algoritmo A***




Emparejamiento de los esquemas S y T.

atributos(S) = $\{s_1, s_2 .. s_n\}$

atributos(T) = $\{t_1, t_2 .. t_m\}$

Descripción de los estados del espacio de búsqueda:

Tupla de tamaño n, en la que el elemento i-ésimo indica el emparejamiento de $s_i$

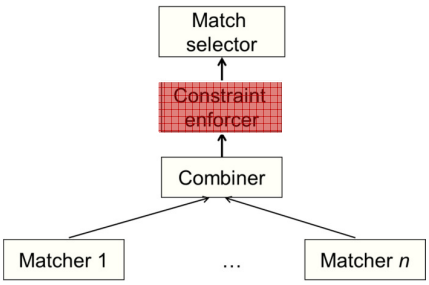
- Un atributo concreto $t_j$ de T, o bien
- Un comodín * si aún no se ha determinado el emparejamiento adecuado para el atributo $s_i$ de S.

# Integración de esquemas

**Schema matching**

RESTRICCIONES DE INTEGRIDAD

**Algoritmo A***




Un estado, p.ej. ($t_2$, *, $t_1$, $t_3$, $t_2$), representa un conjunto de emparejamientos consistente con las restricciones de integridad.

- Estado inicial (*, *, ..., *).
- Estados finales: estados concretos, sin comodines.
- Expansión de estados: Seleccionar un comodín (*) y reemplazarlo con todos sus posibles emparejamientos.

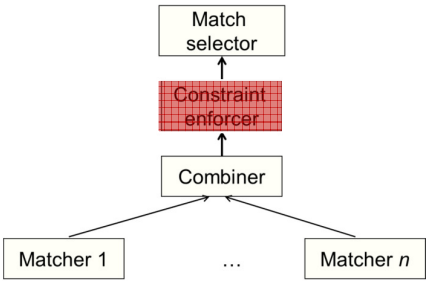


# Integración de esquemas

**Schema matching**

RESTRICCIONES DE INTEGRIDAD

**Algoritmo A***




Coste asociado a los estados:

**cost(M) = -LH(M) + $\Sigma$ cost(M,$c_i$)**

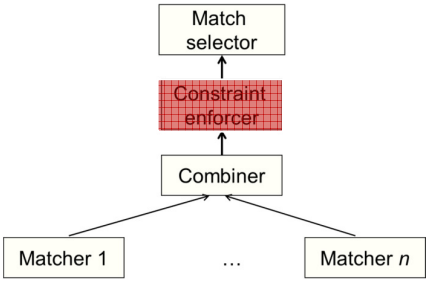
- Verosimilitud de M dadas las matrices de similitud [log-likelihood]: LH(M) = log conf(M)
- Confianza en el emparejamiento: conf(M) = score(M) = $\Pi$ sim ($s_i$, $M_i$)
- Violaciones de las restricciones de integridad: cost(M,$c_i$)

# Integración de esquemas

## Schema matching

RESTRICCIONES DE INTEGRIDAD

### Propagación local




- Algoritmo de optimización local:

  Se propagan las restricciones
  sobre los elementos de un esquema
  hasta que se alcanza un "punto fijo" (óptimo local).



# Integración de esquemas

## Schema matching

RESTRICCIONES DE INTEGRIDAD

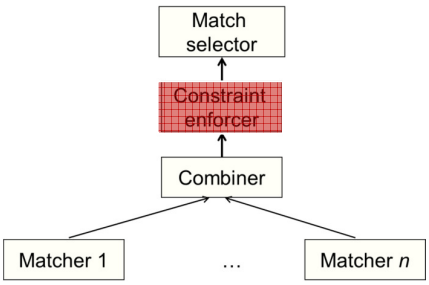
| | Restricciones | Coste |
|---|---|---|
| $c_1$ | If A = Items.code, then A is a key | ∞ |
| $c_2$ | If A = Items.desc, then any random sample of 100 data instances of A must have an average length of at least 10 words | 1.5 |
| $c_3$ | If $A_1 = B_1$, $A_2 = B_2$, $B_2$ is next to $B_1$ in the schema, but $A_2$ is not next to $A_1$, then there is no A* next to $A_1$ such that $\|sim(A^*,B_2) - sim(A_2,B_2)\| \leq t$ for a small pre-specified t | 2 |
| $c_4$ | If more than half of the attributes of Table U match those of Table V, then U = V | 1 |

# Integración de esquemas

**Schema matching**

RESTRICCIONES DE INTEGRIDAD

**Propagación local**




Las restricciones de integridad, p.ej. $c_3$,
y se reescriben como se muestra a continuación:

if $sim(A_1, B_1) \leq 0.9$ and $A_1$ has a neighbor $A_2$
such that $sim(A_2, B_2) \geq 0.75$,
and $B_1$ is a neighbor of $B_2$,
then increase $sim(A_1, B_1)$ by $\alpha$



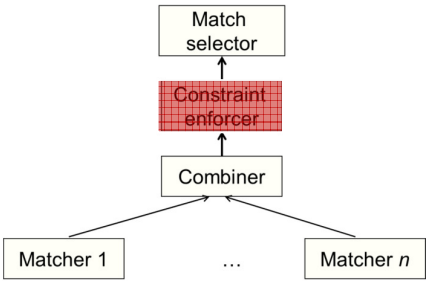
# Integración de esquemas

**Schema matching**

RESTRICCIONES DE INTEGRIDAD

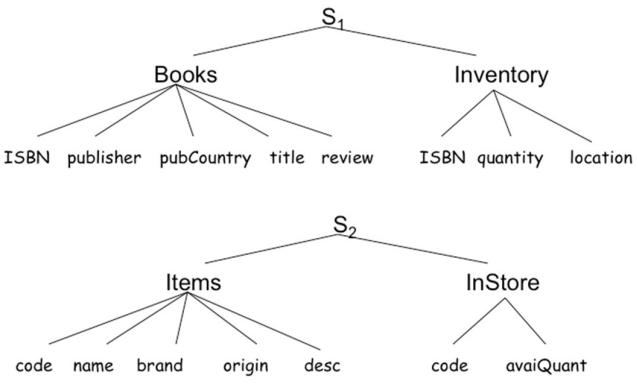
**Propagación local**

Algoritmo de optimización local




1. Inicialización

Matriz de similitud calculada a partir de los resultados del emparejamiento.



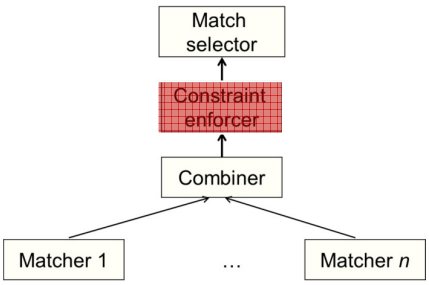
# Integración de esquemas

**Schema matching**

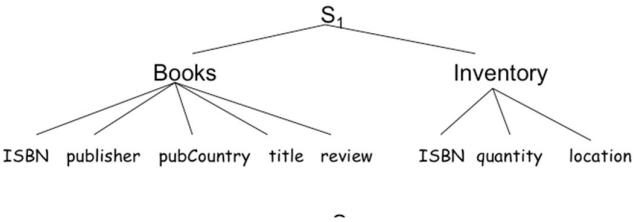
RESTRICCIONES DE INTEGRIDAD

**Propagación local**

Algoritmo de optimización local




2. Iteración



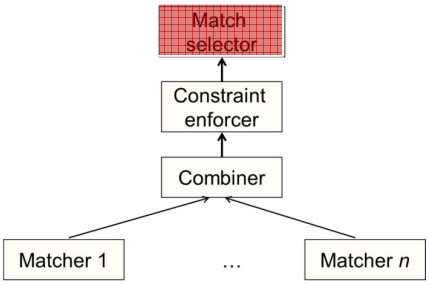

Se selecciona un nodo de S1 y se actualiza la matriz de similitud de sus vecinos de acuerdo con las reglas asociadas a las restricciones de integridad.



# Integración de esquemas

**Schema matching**

SELECCIÓN DEL EMPAREJAMIENTO




La solución más simple:
**Umbralización [thresholding]**

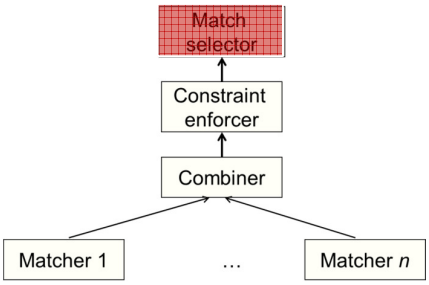
- Se seleccionan elementos de la matriz de similitud.
- Todos los pares de atributos con una similitud no inferior a un umbral se dan por válidos.

# Integración de esquemas

## Schema matching

SELECCIÓN DEL EMPAREJAMIENTO

**Umbralización [thresholding]**




EJEMPLO

Dada la matriz de similitud

name = <title: 0.5>
releaseInfo = <releaseDate: 0.6>
classification = <rating: 0.3>
price = <basePrice: 0.5>

y el umbral 0.5, se genera el emparejamiento

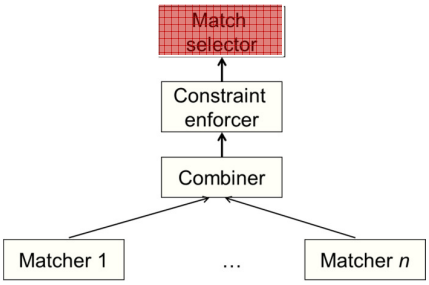
{ (name, title), (releaseInfo,releaseDate) ... }



# Integración de esquemas

## Schema matching

SELECCIÓN DEL EMPAREJAMIENTO




Una estrategia habitual:
**Emparejamientos estables**

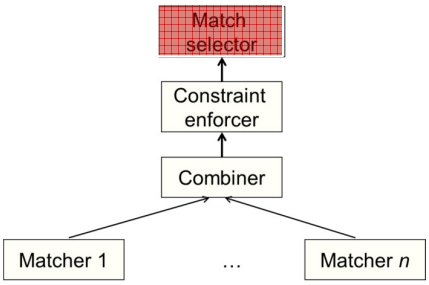
Dados dos conjuntos de elementos S y T
y una matriz que expresa sus preferencias
(la matriz de similitud entre S y T),
encontrar un emparejamiento estable
entre los elementos de S y de T.

# Integración de esquemas

## Schema matching

SELECCIÓN DEL EMPAREJAMIENTO

### Emparejamientos estables




Un emparejamiento es inestable si existe algún par inestable ($s_i$,$t_j$).

Un par ($s_i$,$t_j$) es inestable si:

- $s_i$ prefiere a $t_j$ en lugar del elemento $t_k$ al que ha sido asignado: sim(i,j) > sim(i,k)
- $t_j$ prefiere a $s_i$ en lugar del elemento $s_k$ al que ha sido asignado: sim(i,j) > sim(k,j)

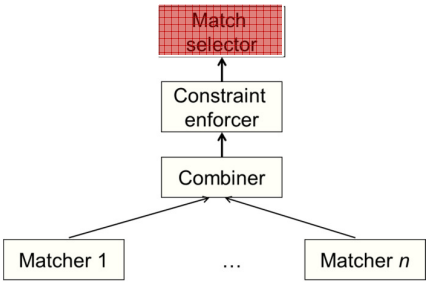


# Integración de esquemas

## Schema matching

SELECCIÓN DEL EMPAREJAMIENTO

### Emparejamientos estables




En otros términos (asumiendo n=m):

Dados n hombres y n mujeres, encontrar un emparejamiento estable de hombres con mujeres.

Cada persona "evalúa" a las personas del sexo opuesto.

- Los hombres ordenan a las mujeres según sus preferencias.
- Las mujeres ordenan a los hombres según sus preferencias.

# Integración de esquemas

**Algoritmo de Gale & Shapley** (1962)

```
while  ( queden hombres sin pareja que no le hayan pedido
         salir a todas las mujeres) {
   m = Uno de esos hombres
   w = Primera mujer en la lista de preferencias de m
       a quien todavía no le haya pedido salir
   if (w está sin pareja)
       Añadir (m,w) a los emparejamientos
   else if (w prefiere a m frente a su pareja actual m')
       Añadir (m,w) a los emparejamientos
       Dejar a m' sin pareja
   else
      w rechaza a m
}
```


# Integración de esquemas

**Algoritmo de Gale & Shapley** (1962)

- Los hombres se declaran a las mujeres en orden decreciente de preferencias.
- Una vez que una mujer se empareja, sólo cambia de estado para mejorar de pareja

El algoritmo de Gale-Shapley termina después de, como mucho, $n^2$ iteraciones.

El algoritmo de Gale-Shapley devuelve un emparejamiento estable.

# Integración de esquemas

**Algoritmo de Gale & Shapley** (1962)

CURIOSIDADES

Si existen varios emparejamientos estables,
¿cuál devuelve el algoritmo de Gale-Shapley?

Un hombre m es una pareja válida para una mujer w si existe un emparejamiento estable en el que estén emparejados.

El algoritmo de Gale-Shapley obtiene un emparejamiento estable que, además, es una asignación óptima para los hombres:

Todos los hombres consiguen a su mejor pareja válida.



# Integración de esquemas

**Algoritmo de Gale & Shapley** (1962)

CURIOSIDADES

Cuando los hombres son los que se declaran,
todos consiguen a su mejor pareja válida.

Ahora bien,
¿la asignación óptima para los hombres
se consigue a costa de las mujeres?

SÍ

¡Todas las mujeres consiguen a su peor pareja válida!

# Integración de esquemas

**Algoritmo de Gale & Shapley** (1962)

CURIOSIDADES

¿Podrían los participantes "engañar"
al algoritmo de Gale-Shapley para salir beneficiados?

Favorita ↓ Menos favorita ↓

| | 1st | 2nd | 3rd |
|---|---|---|---|
| Jorge | Ana | Bea | Clara |
| Luis | Bea | Ana | Clara |
| Mario | Ana | Bea | Clara |

Preferencias de los hombres

Favorito ↓ Menos favorito ↓

| | 1st | 2nd | 3rd |
|---|---|---|---|
| Ana | Luis | Jorge | Mario |
| Bea | Jorge | Luis | Mario |
| Clara | Jorge | Luis | Mario |

Preferencias reales de las mujeres

**Los hombres, obviamente, no.**
**Algunas mujeres, sí**
**(si conocen las preferencias de todos los demás).**

| | 1st | 2nd | 3rd |
|---|---|---|---|
| Ana | Luis | Mario | Jorge |
| Bea | Jorge | Luis | Mario |
| Clara | Jorge | Luis | Mario |

Ana miente para mejorar…



# Integración de esquemas

**Schema matching**

BÚSQUEDA DE EMPAREJAMIENTOS

- Las tareas de emparejamiento suelen ser repetitivas (p.ej. cuando se añaden nuevas fuentes de datos que han de emparejarse con el esquema integrado).
- El rendimiento de un sistema de emparejamiento puede mejorar a lo largo del tiempo utilizando técnicas de aprendizaje automático [machine learning].

# Integración de esquemas

**Schema matching**

BÚSQUEDA DE EMPAREJAMIENTOS

**Aprendizaje multi-estrategia**

- Dado un conjunto de fuentes de datos $S_1..S_n$ y un esquema integrado G.
- Se emparejan manualmente las fuentes $S_1..S_m$ con G, siendo m<<n.
- Se generalizan esos emparejamientos para predecir los emparejamientos más adecuados para $S_{m+1}..S_n$ ¿Cómo?



# Integración de esquemas

**Schema matching**

BÚSQUEDA DE EMPAREJAMIENTOS

**Aprendizaje multi-estrategia**

- Se entrenan clasificadores $L_1..L_k$ para los elementos e del esquema integrado G (usando los datos obtenidos de los emparejamientos de las fuentes $S_1..S_m$ con G).
- Se utiliza un algoritmo de aprendizaje para ajustar los pesos $\mathbf{w_{e,Li}}$ para cada elemento e del esquema integrado y cada clasificador $L_i$ (lo que permitirá combinar las predicciones de los clasificadores $L_i$).

# Integración de esquemas

## Schema matching

BÚSQUEDA DE EMPAREJAMIENTOS

### Aprendizaje multi-estrategia

Dado un nuevo esquema S con atributos $s_1..s_n$, se utilizan los clasificadores $L_1..L_k$ sobre $s_1..s_n$ y se combinan sus predicciones:

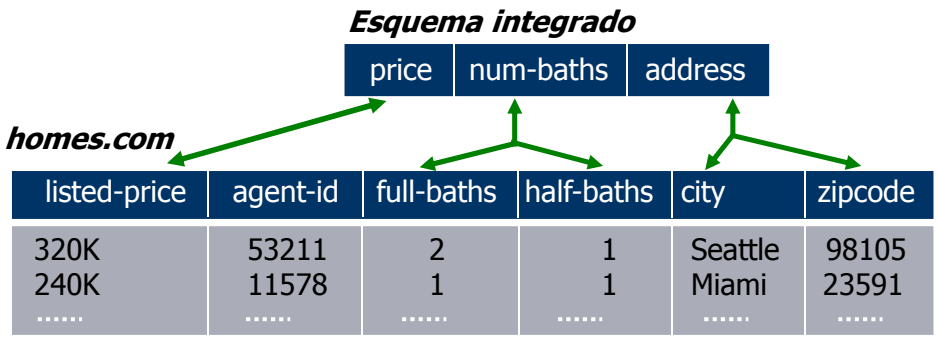
$$score_e(s) = \sum_{i=1}^{k} w_{e,L_i} * score_{e,L_i}(s)$$



# Integración de esquemas

## Schema matching

BÚSQUEDA DE EMPAREJAMIENTOS COMPLEJOS

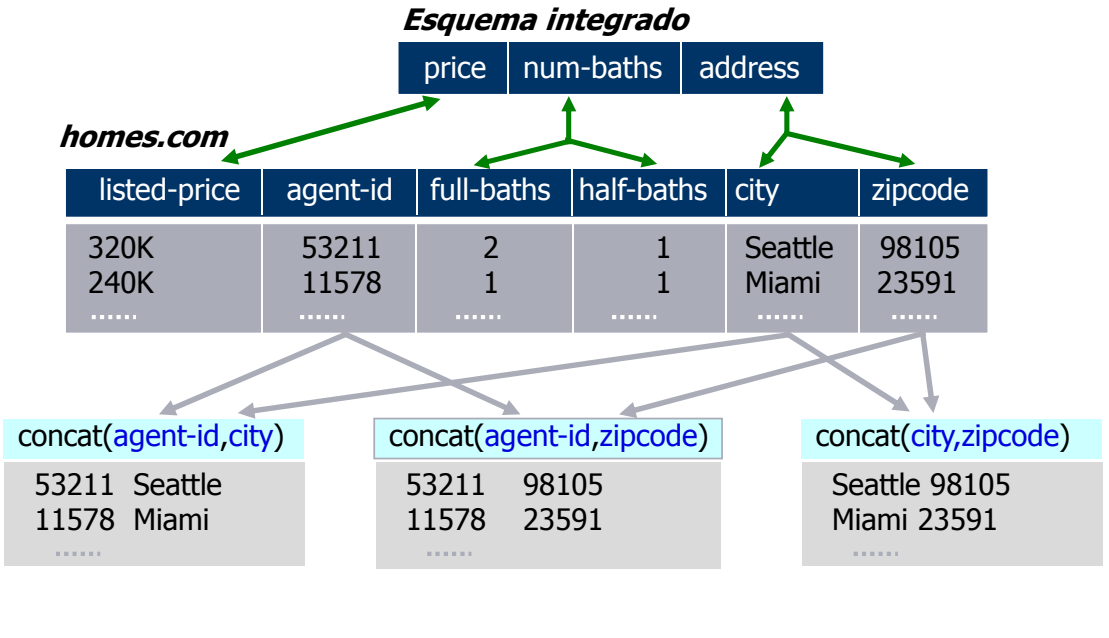



- Textos: Concatenación de columnas.
- Números: Expresiones aritméticas.
- Fechas: Combinaciones de meses/años/días.

# Integración de esquemas

## Schema matching

BÚSQUEDA DE EMPAREJAMIENTOS COMPLEJOS






# Integración de esquemas

## Schema matching

BÚSQUEDA DE EMPAREJAMIENTOS COMPLEJOS

Control del proceso de búsqueda:

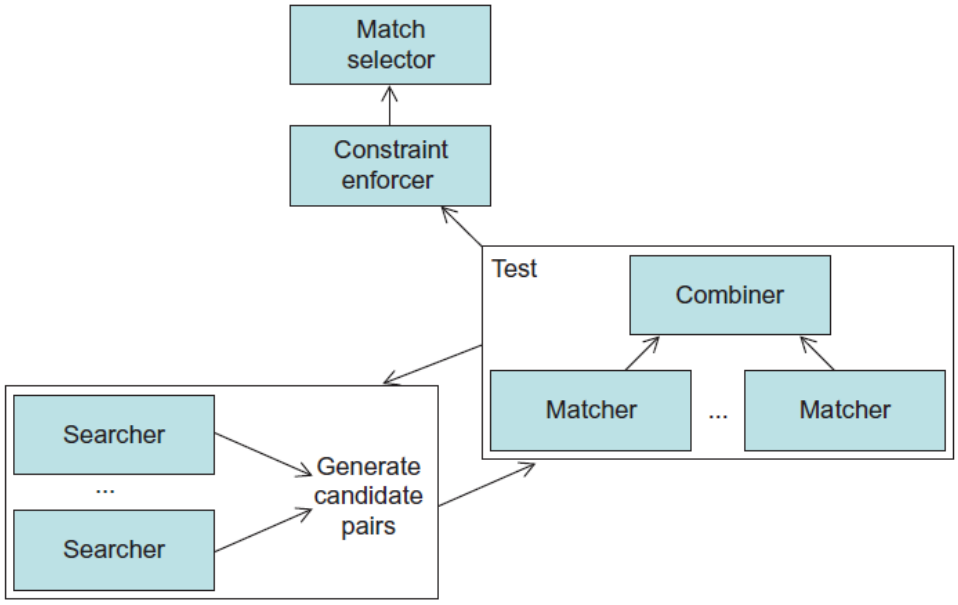
- Buscadores especializados (textos, números, fechas).
- Búsqueda dirigida [beam search],
  i.e. sólo se consideran los k mejores candidatos.

Ejemplo: **iMAP** [Doan et al., SIGMOD'2004]

# Integración de esquemas

## Schema matching

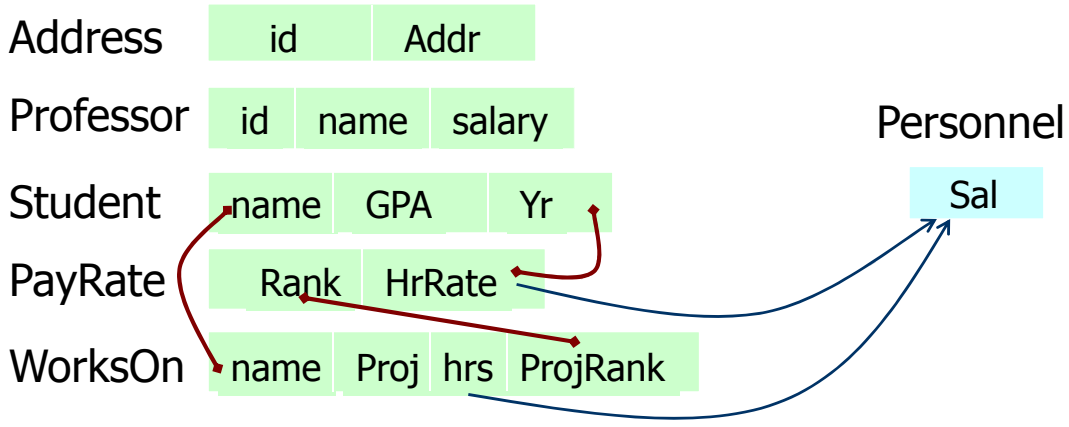
BÚSQUEDA DE EMPAREJAMIENTOS COMPLEJOS






# Integración de esquemas

## Schema mapping




f1: PayRate(HrRate) * WorksOn(Hrs) = Personnel(Sal)

# Integración de esquemas

**Schema mapping**

Posibles consultas:

```
select P.HrRate * W.hrs
from PayRate P, WorksOn W
where P.Rank = W.ProjRank
```

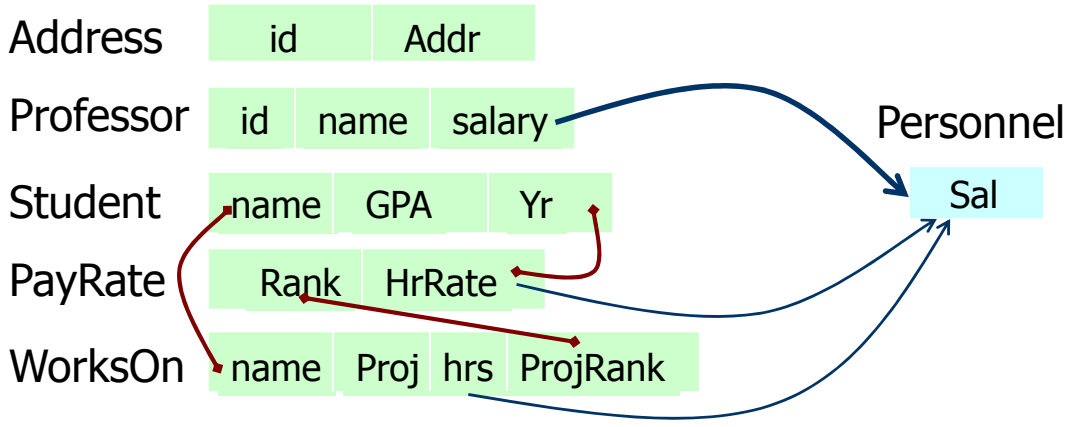
```
select P.HrRate * W.hrs
from PayRate P, WorksOn W, Student S
where W.Name=S.Name and S.Yr = P.Rank
```

f1: PayRate(HrRate) * WorksOn(Hrs) = Personnel(Sal)



# Integración de esquemas

**Schema mapping**




f2: Professor(Sal) → Personnel(Sal)

# Integración de esquemas

**Schema mapping**

```
select P.HrRate * W.hrs
from PayRate P, WorksOn W
where P.Rank = W.ProjRank

UNION ALL

select Sal
from Professor
```



# Integración de esquemas

**Schema mapping**

Decisiones de diseño


- ¿Qué reuniones realizar?
  ➔ Conjuntos de candidatos
- ¿Cómo combinar los resultados de las reuniones?
  ➔ Unión de conjuntos de candidatos

**IBM Clio Project**

http://dblab.cs.toronto.edu/project/clio/

# Integración de esquemas

**Schema mapping**

Identificación de conjuntos de candidatos

- Conjuntos de candidatos obtenidos
  a partir de las claves externas en el esquema (DDL)
  y de las consultas que se suelen realizar sobre los
  datos (DML).

- Seleccionar las reuniones aprovechando claves
  externas, restricciones semánticas y diferencias entre
  reuniones internas y externas [inner & outer joins].



# Integración de esquemas

**Schema mapping**

Selección de candidatos

- Cobertura:
  Conjunto mínimo de candidatos
  que cubre todas las correspondencias.

- Problema de búsqueda heurística:
  Seleccionar la mejor cobertura (prefiriendo candidatos
  con menor número de reuniones que cubran el mayor
  número de atributos del esquema de destino).

NOTA: La interacción con el "usuario" es clave.

# Integración de esquemas

**Gestión de modelos [model management]**

Las correspondencias entre esquemas pueden definirse mediante operadores:

- Operadores de combinación de esquemas.
- Operadores de traducción de esquemas.
- …

En este contexto, **"modelo" ≈ "esquema"**.

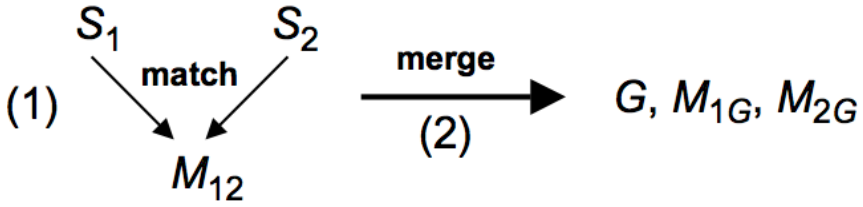
Más concretamente, un modelo es una descripción específica de un conjunto de datos en un modelo de datos dado.



# Integración de esquemas

**Gestión de modelos [model management]**

EJEMPLO




Integración de dos fuentes de datos $S_1$ y $S_2$:

- **Match**: Un operador de emparejamiento de $S_1$ con $S_2$ para establecer la correspondencia entre $S_1$ y $S_2$.
- **Merge**: Creación de un esquema integrado para $S_1$ y $S_2$ (el operador crea el mínimo esquema que incluye tanto a $S_1$ como a $S_2$).

RECORDATORIO: En este contexto, "modelo" ≈ "esquema".

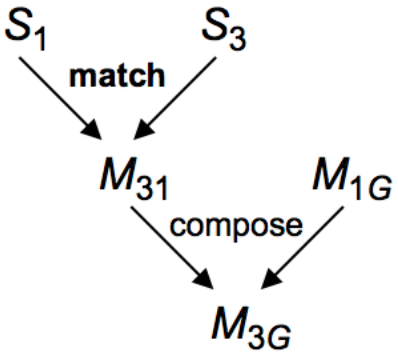
# Integración de esquemas

**Gestión de modelos [model management]**

EJEMPLO

Integración de una tercera fuente de datos $S_3$:

Si $S_3$ es muy similar a $S_1$, podemos emparejar $S_1$ con $S_3$ para luego componer el emparejamiento de $S_3$ con el esquema integrado G.






# Integración de esquemas

**Gestión de modelos [model management]**

OPERADORES GENÉRICOS DE GESTIÓN DE MODELOS

- **Match** (establecer una correspondencia $M_{12}$ entre dos modelos $S_1$ y $S_2$).
- **Merge** (crear un esquema combinado de los modelos $S_1$ y $S_2$ con respecto a la correspondencia $M_{12}$).
- **ModelGen** (crear un modelo equivalente utilizando un modelo de datos diferente, p.ej. relacional ➔ XML).
- **Invert** (dada una correspondencia $M_{12}$, obtener la correspondencia inversa $M_{21}$).
- **Diff** (encontrar las diferencias entre dos modelos).

# Integración de esquemas

## Gestión de modelos [model management]

EL OPERADOR MERGE

Dados

- Dos modelos $M_1$ y $M_2$
- Una correspondencia de $M_1$ a $M_2$

Crear

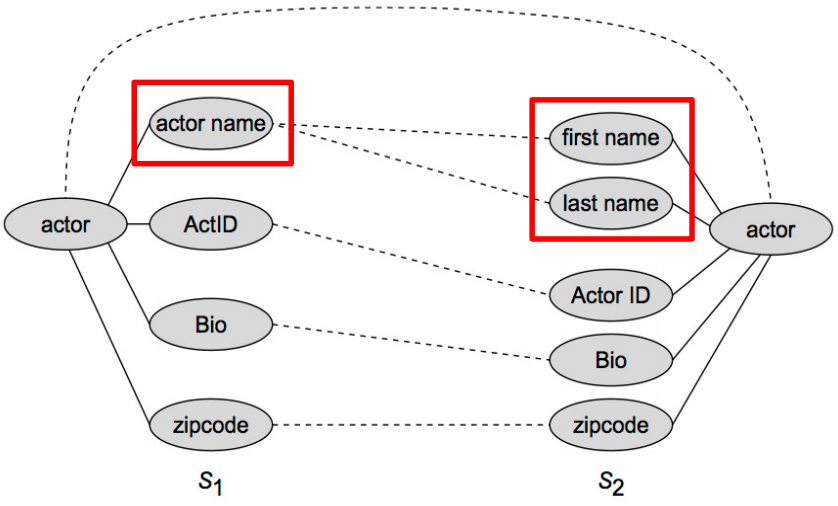
- Un modelo combinado $M_{12}$ que contiene la información de $M_1$ y $M_1$ pero no repite lo que está en ambos.
- Correspondencias de $M_1$ y $M_2$ a $M_{12}$.



# Integración de esquemas

## Gestión de modelos [model management]

EL OPERADOR MERGE

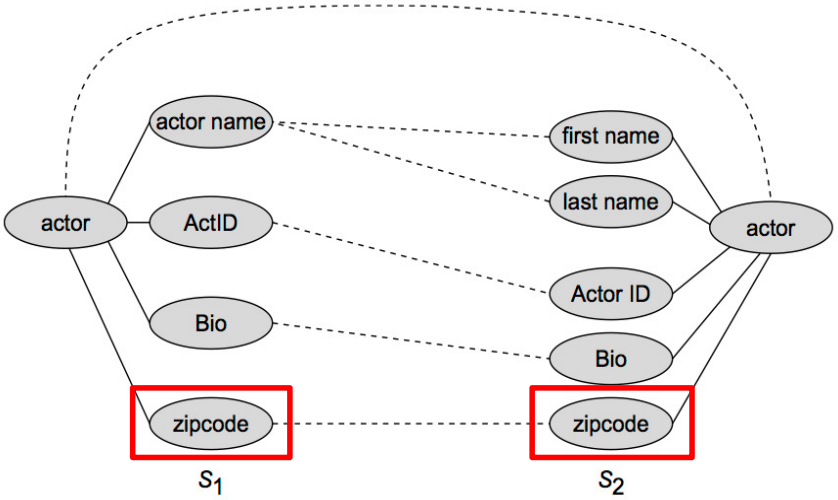



Diferentes representaciones para los mismos atributos.

# Integración de esquemas

## Gestión de modelos [model management]

EL OPERADOR MERGE




Diferentes codificaciones para los atributos.



# Integración de esquemas

## Gestión de modelos [model management]

EL OPERADOR MERGE

**Problemas prácticos**

- Diferentes representaciones para los atributos (la resolución forma parte de la correspondencia).
- Combinación de modelos de distintos modelos de datos (p.ej. atributos compuestos).
- Conflictos irresolubles, p.ej. código postal como entero en un modelo y como cadena en otro (no puede ser ambas cosas en el modelo combinado).

# Integración de esquemas

## Gestión de modelos [model management]

EL OPERADOR MODELGEN

Transformar un esquema de un metamodelo a otro.

En este contexto, un metamodelo es un modelo de datos (p.ej. modelo de clases en Java, esquema relacional para un RDBMS, esquema XML como DTD o XML Schema…).

**Problema práctico:** Existen características en el metamodelo fuente que no existen en el de destino.


# Integración de esquemas

## Gestión de modelos [model management]

EL OPERADOR MODELGEN

De clases en Java a relaciones

```
public class Company {
   public string name;
}

public class Supplier
     extends Company {
   public item[] parts;
}

public class Item {
   public string ISBN;
   public int Cost;
}
```

```
CREATE TABLE Company(
 Name varchar(50),
 oid int NOT NULL PRIMARY KEY)

CREATE TABLE Supplier(
 oid int NOT NULL PRIMARY KEY,
 isSameAs int NOT NULL UNIQUE
 FOREIGN KEY REFERENCES Company
    (oid))

CREATE TABLE PartsArray(
 Supplier int NOT NULL
 FOREIGN KEY REFERENCES Company
    (oid),
 itemISBN varchar(50),
 itemCost int)
```

No existen ni clases ni herencia en el modelo relacional.

# Integración de esquemas

**Gestión de modelos [model management]**

EL OPERADOR MODELGEN

- Es posible diseñar transformaciones específicas de un metamodelo a otro, pero sería preferible una solución general…
- Estrategia: Se diseña un "súper metamodelo" que soporta todas las características que existen en los distintos metamodelos (y sabe qué características están presentes en cada metamodelo).



# Integración de esquemas

**Gestión de modelos [model management]**

EL OPERADOR MODELGEN

Se traduce el modelo $M_1$ al "súper metamodelo" y del "súper metamodelo" ahí al Modelo $M_2$.

# Integración de esquemas

## Gestión de modelos [model management]

EL OPERADOR MODELGEN

**Entrada**

Modelo $M_1$ en el metamodelo $MM_1$.

**Salida**

Modelo $M_2$ en el metamodelo $MM_2$ equivalente a $M_1$.

**Algoritmo**

- Transformar $M_1$ al supermodelo M'.
- Mientras queden características en M' que no estén presentes en $MM_2$, aplicar transformaciones sobre M' para eliminar dichas características.
- Transformar el supermodelo M' en $M_2$.



# Integración de esquemas

## Gestión de modelos [model management]

EL OPERADOR MODELGEN

De clases en Java a relaciones

```
public class Company {
   public string name;
}

public class Supplier
     extends Company {
   public item[] parts;
}

public class Item {
   public string ISBN;
   public int Cost;
}
```

```
CREATE TABLE Company(
 Name varchar(50),
 oid int NOT NULL PRIMARY KEY)

CREATE TABLE Supplier(
 oid int NOT NULL PRIMARY KEY,
 isSameAs int NOT NULL UNIQUE
 FOREIGN KEY REFERENCES Company
    (oid))

CREATE TABLE PartsArray(
 Supplier int NOT NULL
 FOREIGN KEY REFERENCES Company
    (oid),
 itemISBN varchar(50),
 itemCost int)
```

No existen ni clases ni herencia en el modelo relacional.

# Integración de esquemas

**Gestión de modelos [model management]**

EL OPERADOR DE INVERSIÓN

Normalmente, las correspondencias entre esquemas son direccionales (esquema fuente → esquema destino).

¿Se puede encontrar la correspondencia inversa?

La solución dependerá de los metamodelos concretos (no existen algoritmos genéricos).

**PROBLEMA:**
Distintos modelos pueden tener el mismo destino…



# Integración de esquemas

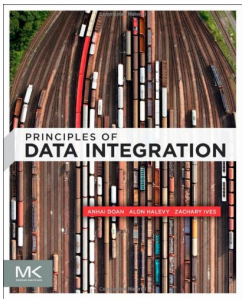
**Gestión de modelos [model management]**

El uso de operadores genéricos de gestión de modelos:

- Puede ahorrarnos mucho trabajo (eliminan la necesidad de escribir código repetitivo).
- Puede servirnos para ser cuidadosos en el diseño del proceso de integración de datos (los operadores describen formalmente el algoritmo utilizado para integrar datos de distintas fuentes).

# Bibliografía recomendada

- Hai Doan, Alon Halevy & Zachary Ives:
  **Principles of Data Integration**
  Morgan Kaufmann, 1st edition, 2012.
  ISBN 0124160441
  http://research.cs.wisc.edu/dibook/




Chapter 5: Schema Matching and Mapping

Chapter 6: General Schema Manipulation Operators