

Teoría de Bases de Datos

Juan L. Reutter

Pontificia Universidad Católica de Chile

Center for **Semantic Web** Research



Las bases de datos están presentes en
todo lo que hacemos

Las bases de datos están presentes en todo lo que hacemos

¿Qué película están pasando hoy?

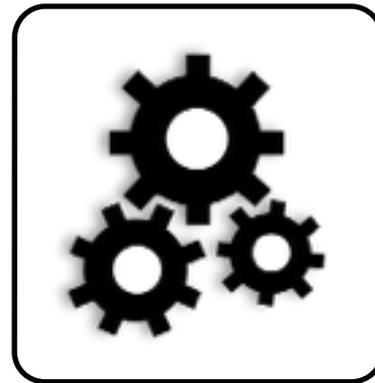


Buscando a Dory
Civil War
X-Men Apocalypse

...

Las bases de datos están presentes en todo lo que hacemos

¿Qué película están pasando hoy?



Buscando a Dory
Civil War
X-Men Apocalypse

...

¡Y todo funciona (casi siempre) muy bien!

La investigación en bases de datos ha contribuido (y sigue contribuyendo) a los avances en este campo

La investigación en bases de datos ha contribuido (y sigue contribuyendo) a los avances en este campo

Un poco de historia

Problemática actual

La investigación en bases de datos ha contribuido (y sigue contribuyendo) a los avances en este campo

Bases de datos relacionales

Un poco de historia

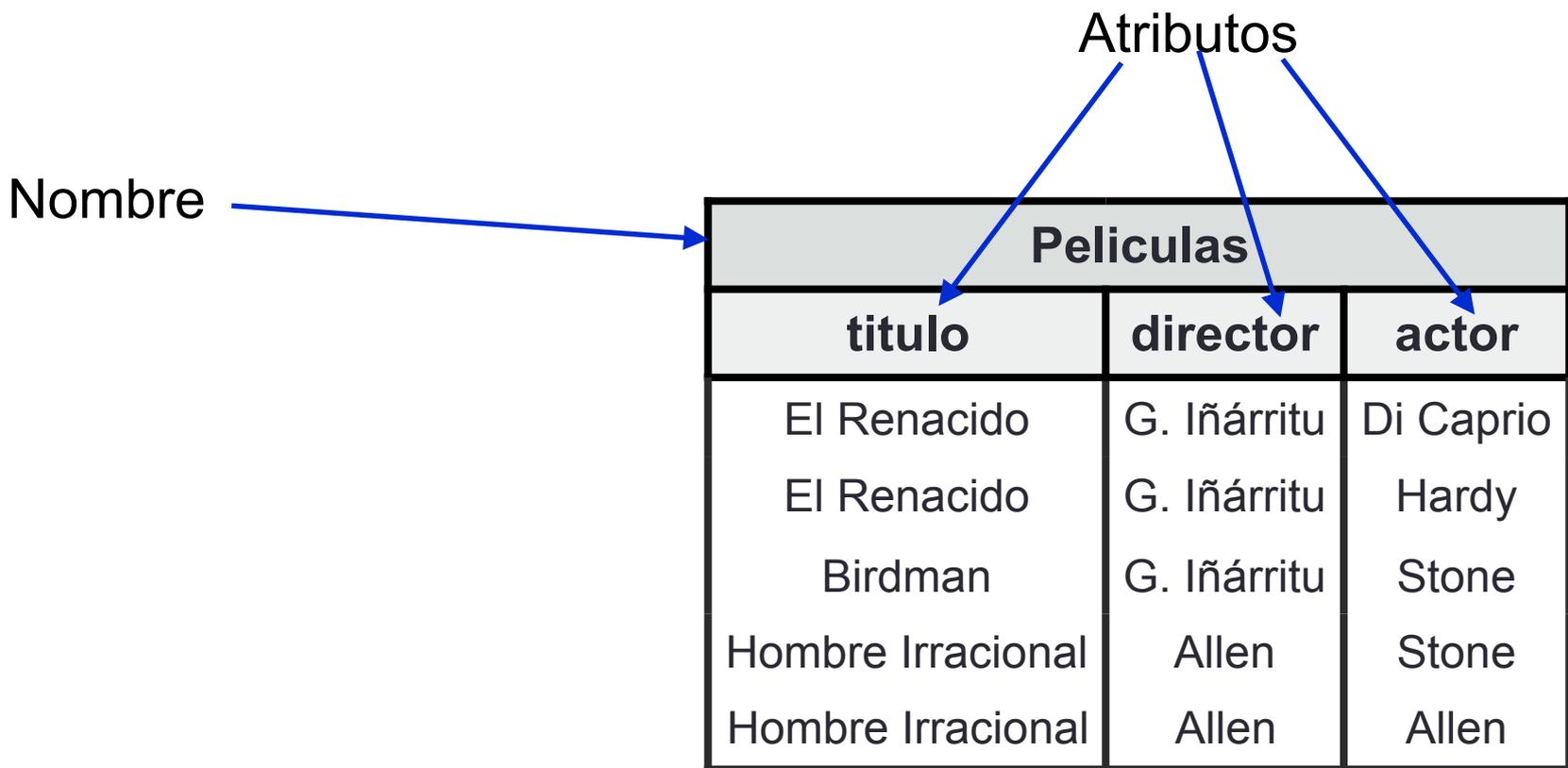
Desafío: Optimización de consultas

Solución: Modelación en lógica

Modelo relacional: información en relaciones (tablas)

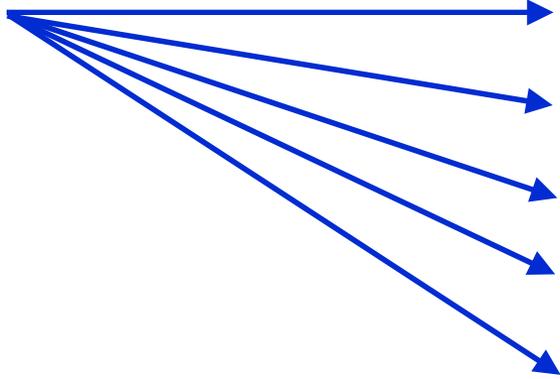
Peliculas		
titulo	director	actor
El Renacido	G. Iñárritu	Di Caprio
El Renacido	G. Iñárritu	Hardy
Birdman	G. Iñárritu	Stone
Hombre Irracional	Allen	Stone
Hombre Irracional	Allen	Allen

Modelo relacional: información en relaciones (tablas)



Modelo relacional: información en relaciones (tablas)

Tuplas



Películas		
titulo	director	actor
El Renacido	G. Iñárritu	Di Caprio
El Renacido	G. Iñárritu	Hardy
Birdman	G. Iñárritu	Stone
Hombre Irracional	Allen	Stone
Hombre Irracional	Allen	Allen

En la práctica:

Sistemas de manejo de Bases de datos relacionales (SMBD)

¿Qué película están
pasando hoy?



Buscando a Dory
Civil War
X-Men Apocalypse
...

En la práctica: Sistemas de manejo de Bases de datos relacionales (SMBD)

¿Qué película están
pasando hoy?



```
SELECT pelicula  
FROM Programacion  
WHERE fechaEstreno <= date()
```



Buscando a Dory
Civil War
X-Men Apocalypse
...

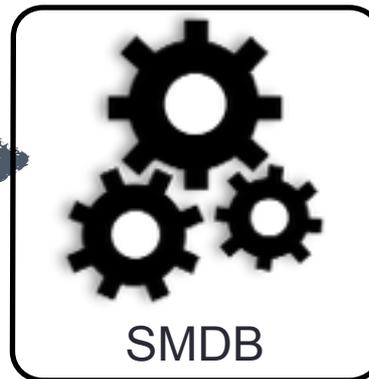
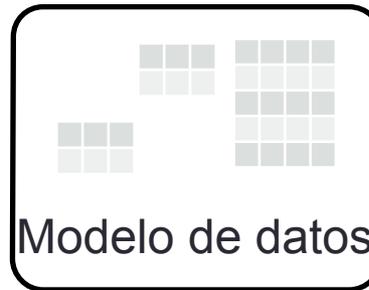


En la práctica: Sistemas de manejo de Bases de datos relacionales (SMDB)

¿Qué película están
pasando hoy?



```
SELECT pelicula  
FROM Programacion  
WHERE fechaEstreno <= date()
```



Buscando a Dory
Civil War
X-Men Apocalypse
...



La investigación en bases de datos ha contribuido (y sigue contribuyendo) a los avances en este campo

Bases de datos relacionales

Un poco de historia

Desafío: Optimización de consultas

Solución: Modelación en lógica

A veces los usuarios escriben consultas muy largas,
siendo que hay consultas cortas que hacen lo mismo

A veces los usuarios escriben consultas muy largas,
siendo que hay consultas cortas que hacen lo mismo

Pelicula(titulo,director,actor)

Programacion(pelicula,cine,sala,hora)

A veces los usuarios escriben consultas muy largas,
siendo que hay consultas cortas que hacen lo mismo

Pelicula(titulo,director,actor)

Programacion(pelicula,cine,sala,hora)

```
SELECT pelicula
FROM Programacion, Peliculas
WHERE fechaEstreno <= date()
      AND Programacion.pelicula NOT IN (SELECT titulo FROM Peliculas)
UNION
SELECT titulo AS pelicula
FROM Peliculas
WHERE titulo IN
      (SELECT pelicula FROM Programacion)
```

A veces los usuarios escriben consultas muy largas,
siendo que hay consultas cortas que hacen lo mismo

Pelicula(titulo,director,actor)

Programacion(pelicula,cine,sala,hora)

```
SELECT pelicula
FROM Programacion, Peliculas
WHERE fechaEstreno <= date()
      AND Programacion.pelicula NOT IN (SELECT titulo FROM Peliculas)
UNION
SELECT titulo AS pelicula
FROM Peliculas
WHERE titulo IN
      (SELECT pelicula FROM Programacion)
```

```
SELECT pelicula
FROM Programacion
WHERE fechaEstreno <= date()
```

Los sistemas son capaces de optimizar estas consultas

Los sistemas son capaces de optimizar estas consultas

Mundo ideal:

- El usuario ingresa una consulta
- El sistema elige la forma más eficiente de computarla

Los sistemas son capaces de optimizar estas consultas

Mundo ideal:

- El usuario ingresa una consulta
- El sistema elige la forma más eficiente de computarla

¿Cómo hago esto?

¿Cómo funcionan los sistemas?

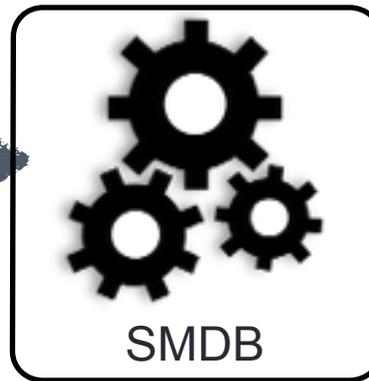
Los sistemas no procesan SQL directamente,
si no que transforman a Álgebra Relacional

¿Qué película están pasando hoy?



Los sistemas no procesan SQL directamente, si no que transforman a Álgebra Relacional

```
SELECT pelicula
FROM Programacion
WHERE fechaEstreno <= date()
```



¿Qué película están pasando hoy?

Los sistemas no procesan SQL directamente, si no que transforman a Álgebra Relacional



```
SELECT pelicula  
FROM Programacion  
WHERE fechaEstreno <= date()
```

Traducción a
álgebra relacional



Procesamiento

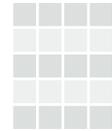
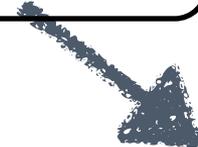


Tabla Respuesta

Álgebra Relacional:

Formalismo usado en el interior de los sistemas,
para el cómputo de consultas

Conjunto de operadores, que operan sobre tablas, y que crean otras tablas.

Álgebra Relacional: Proyección

Películas		
titulo	director	actor
El Renacido	G. Iñárritu	Di Caprio
El Renacido	G. Iñárritu	Hardy
Birdman	G. Iñárritu	Stone
Hombre Irracional	Allen	Stone
Hombre Irracional	Allen	Allen

Elimina atributos de una tabla

Actrices y Actores en mi tabla:

$$\pi_{\text{actor}}(\textit{Películas})$$

Resultado
actor
Di Caprio
Hardy
Stone
Stone
Allen

Álgebra Relacional: Selección

Películas		
titulo	director	actor
El Renacido	G. Iñárritu	Di Caprio
El Renacido	G. Iñárritu	Hardy
Birdman	G. Iñárritu	Stone
Hombre Irracional	Allen	Stone
Hombre Irracional	Allen	Allen

Elimina tuplas de acuerdo a condiciones

Tuplas donde actuó Emma Stone:

$$\sigma_{\text{actor} = \text{'Stone'}}(\textit{Películas})$$

Resultado		
titulo	director	actor
Birdman	G. Iñárritu	Stone
Hombre Irracional	Allen	Stone

Álgebra Relacional: Componiendo Operadores

Películas		
titulo	director	actor
El Renacido	G. Iñárritu	Di Caprio
El Renacido	G. Iñárritu	Hardy
Birdman	G. Iñárritu	Stone
Hombre Irracional	Allen	Stone
Hombre Irracional	Allen	Allen

Películas donde actuó Emma Stone:

$$\pi_{\text{titulo}}(\sigma_{\text{actor} = \text{'Stone'}}(\textit{Películas}))$$

Resultado
titulo
Birdman
Hombre Irracional

Álgebra Relacional: Unión / Intersección

Películas		
titulo	director	actor
El Renacido	G. Iñárritu	Di Caprio
El Renacido	G. Iñárritu	Hardy
Birdman	G. Iñárritu	Stone
Hombre Irracional	Allen	Stone
Hombre Irracional	Allen	Allen

Actrices y Actores que han trabajado con Allen y con G. Iñárritu:

$$\pi_{\text{actor}}(\sigma_{\text{director} = \text{'G. Inarritu'}}(\textit{Películas}))$$

∩

$$\pi_{\text{actor}}(\sigma_{\text{director} = \text{'Allen'}}(\textit{Películas}))$$

Actrices y Actores que han trabajado con Allen o con G. Iñárritu:

$$\pi_{\text{actor}}(\sigma_{\text{director} = \text{'G. Inarritu'}}(\textit{Películas}))$$

∪

$$\pi_{\text{actor}}(\sigma_{\text{director} = \text{'Allen'}}(\textit{Películas}))$$

Álgebra Relacional: Joins

Películas		
titulo	director	actor
El Renacido	G. Iñárritu	Di Caprio
El Renacido	G. Iñárritu	Hardy
Birdman	G. Iñárritu	Stone
Hombre Irracional	Allen	Stone
Hombre Irracional	Allen	Allen

Cartelera	
titulo	cine
El Renacido	Cinemark
El Renacido	Cinepolis
Birdman	Cinepolis
El Marciano	Cinemark

Actrices y Actores en cartelera:

$$\pi_{\text{actor}}(\mathit{Películas} \bowtie \mathit{Cartelera})$$

Álgebra Relacional: Diferencia

Películas		
titulo	director	actor
El Renacido	G. Iñárritu	Di Caprio
El Renacido	G. Iñárritu	Hardy
Birdman	G. Iñárritu	Stone
Hombre Irracional	Allen	Stone
Hombre Irracional	Allen	Allen

Todas las tuplas en una tabla pero no en otra

Actrices y Actores que han trabajado con G. Iñárritu pero no con Allen:

$$\pi_{\text{actor}}(\sigma_{\text{director} = \text{'G. Inarritu'}}(\textit{Películas}))$$

—

$$\pi_{\text{actor}}(\sigma_{\text{director} = \text{'Allen'}}(\textit{Películas}))$$

Álgebra Relacional

El operador más problemático es un join:

Casi todas las consultas usan joins,
Pero son muy lentos para calcular

Nuestra meta es, entonces,
encontrar una consulta que minimize la cantidad de joins.

Los sistemas son capaces de optimizar estas consultas

Mundo ideal:

- El usuario ingresa una consulta
- El sistema elige la forma más eficiente de computarla

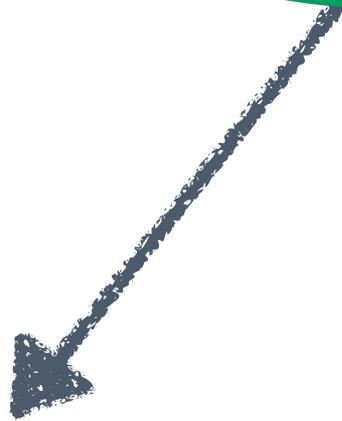
Los sistemas son capaces de optimizar estas consultas

Mundo ideal:

- El usuario ingresa una consulta
- El sistema elige la forma más eficiente de computarla

Digamos:

consulta equivalente que usa la menor cantidad de joins



Los sistemas son capaces de optimizar estas consultas

Mundo ideal:

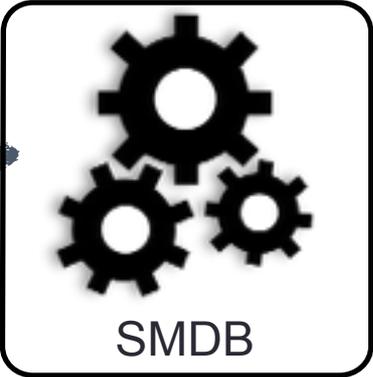
- El usuario ingresa una consulta
- El sistema la transforma a álgebra relacional
- El sistema busca la forma más óptima de computarla

Los sistemas son capaces de optimizar estas consultas

¿Qué película están pasando hoy?



```
SELECT pelicula  
FROM Programacion  
WHERE fechaEstreno <= date()
```



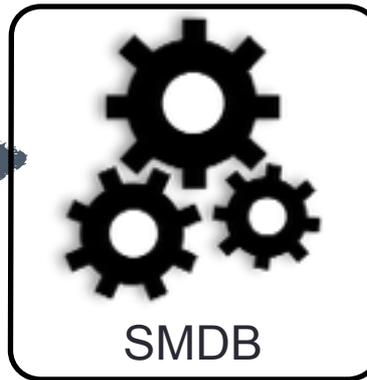
Los sistemas son capaces de optimizar estas consultas

¿Qué película están pasando hoy?



1. Se transforma SQL en álgebra relacional
2. Se trata de escoger la consulta más optima
3. Se procesa esa consulta

```
SELECT pelicula  
FROM Programacion  
WHERE fechaEstreno <= date()
```



Para esto necesitamos resolver este problema:

Dada una consulta en álgebra relacional

¿Cómo puedo encontrar la consulta mínima equivalente?

Sea S un esquema relacional, y Q una consulta sobre S .

El resultado de ejecutar la consulta Q sobre una instancia D
se escribe como $Q(D)$

Definición: Equivalencia

Q es **equivalente** a Q' si para toda instancia D sobre S ,
 $Q(D) = Q(D')$

El Problema de Equivalencia

consiste en decidir si dos consultas son equivalentes

Definición: Minimización

Q es la **minimización** de Q' si

- Q es equivalente a Q'
- para toda consulta Q^* que es equivalente a Q' , la cantidad de joins en Q^* es mayor o igual que los de Q

El Problema de Minimización

consiste en decidir si una consulta es la minimización de otra

No todo es tan simple:

El problema de equivalencia es indecidible para el álgebra relacional (y por tanto el de minimización también)

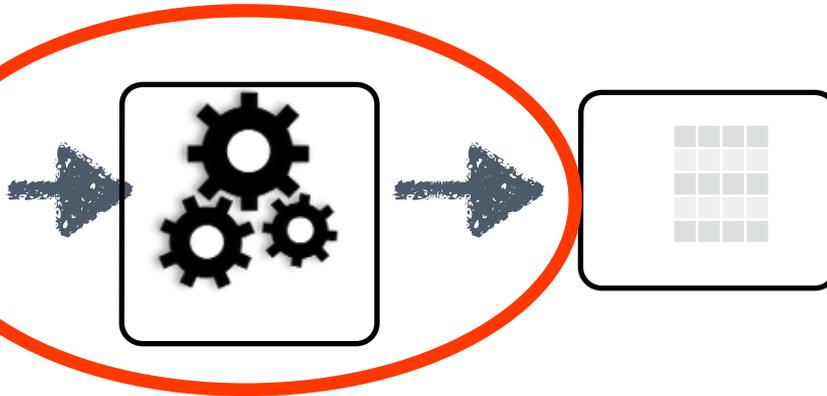
No todo es tan simple:

El problema de equivalencia es indecidible para el álgebra relacional (y por tanto el de minimización también)

¿Qué película están pasando hoy?



```
SELECT pelicula
FROM Programacion
WHERE fechaEstreno <= date()
```

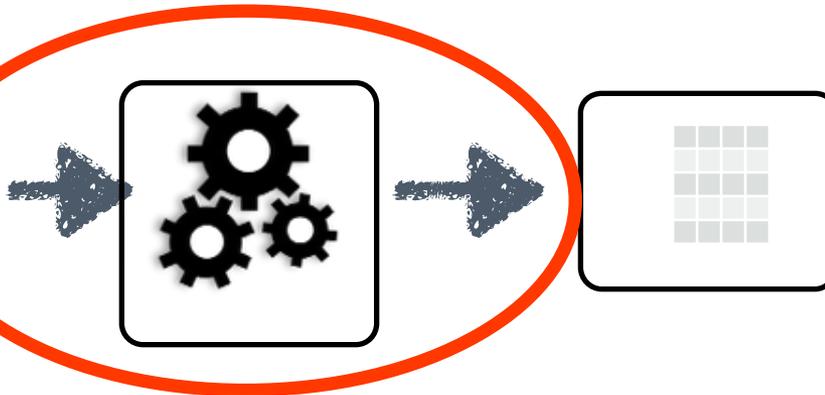


En general, no vamos a poder optimizar las consultas de los usuarios

¿Qué película están pasando hoy?



```
SELECT pelicula  
FROM Programacion  
WHERE fechaEstreno <= date()
```

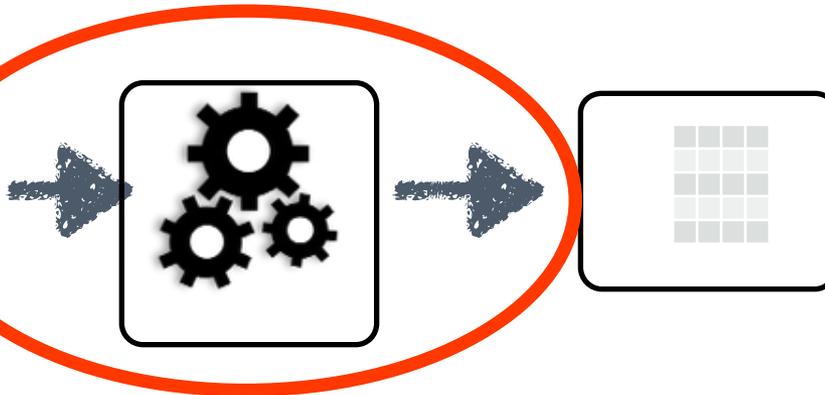


Si optimizamos la mayoría de las consultas, es un gran avance.

¿Qué película están pasando hoy?



```
SELECT pelicula
FROM Programacion
WHERE fechaEstreno <= date()
```



Si optimizamos la mayoría de las consultas, es un gran avance.

Y en la práctica,

la mayoría de las consultas suelen ser simples.

El fragmento **selección - proyección - join** del álgebra relacional corresponde a las consultas en SQL que usan:

- SELECT
- FROM
- WHERE, y donde sólo se mencionan igualdades

El fragmento **selección - proyección - join** del álgebra relacional corresponde a las consultas en SQL que usan:

- SELECT
- FROM
- WHERE, y donde sólo se mencionan igualdades

La gran mayoría de las consultas caen en este fragmento:

¿Que películas hay en cartelera?

¿Cuál es mi balance actual?

¿Hay vuelos para esta fecha?

...

...

La investigación en bases de datos ha contribuido (y sigue contribuyendo) a los avances en este campo

Bases de datos relacionales

Un poco de historia

Desafío: Optimización de consultas

Solución: Modelación en lógica

Usamos lógica para modelar consultas

Construimos fórmulas en lógica usando:

- nombres de las relaciones
- variables y constantes
- igualdades
- cuantificadores existenciales

Lenguaje recibe el nombre de **conjunctive queries (CQ)**

Ejemplos de CQs

Películas		
titulo	director	actor
El Renacido	G. Iñárritu	Di Caprio
El Renacido	G. Iñárritu	Hardy
Birdman	G. Iñárritu	Stone
Hombre Irracional	Allen	Stone
Hombre Irracional	Allen	Allen

Liste todos los actores:

$$\{x \mid \exists y \exists z \text{ Peliculas}(y,z,x)\}$$

Liste todas las películas y sus directores:

$$\{x, y \mid \exists z \text{ Peliculas}(x,y,z)\}$$

Ejemplos de CQs (con igualdad)

Películas		
titulo	director	actor
El Renacido	G. Iñárritu	Di Caprio
El Renacido	G. Iñárritu	Hardy
Birdman	G. Iñárritu	Stone
Hombre Irracional	Allen	Stone
Hombre Irracional	Allen	Allen

Liste todas las películas donde actúa Emma Stone:

$$\{x \mid \exists y \exists z \text{ Peliculas}(x,y,z) \wedge z = \text{'Stone'}\}$$

Liste todos los directores que se dirigieron a si mismos:

$$\{x \mid \exists y \exists z \text{ Peliculas}(y,x,z) \wedge x = z\}$$

Ejemplos de CQs (con joins)

Películas		
titulo	director	actor
El Renacido	G. Iñárritu	Di Caprio
El Renacido	G. Iñárritu	Hardy
Birdman	G. Iñárritu	Stone
Hombre Irracional	Allen	Stone
Hombre Irracional	Allen	Allen

Cartelera	
titulo	cine
El Renacido	Cinemark
El Renacido	Cinepolis
Birdman	Cinepolis
El Marciano	Cinemark

Liste todos los actores en cartelera:

$$\{x \mid \exists y \exists z \exists w \text{ Películas}(y,z,x) \wedge \text{Programacion}(y,w)\}$$

Usamos lógica para modelar consultas

Teorema.

Para cada consulta **select-project-join** de álgebra relacional podemos **encontrar una CQ** tal que ambas son **equivalentes**

Para **cada CQ** podemos **encontrar una consulta select-project-join** de álgebra relacional tal que ambas son **equivalentes**

(Hay lógicas para cada fragmento de álgebra relacional)

¿Dónde estamos?

Tenemos una manera alternativa de lidiar con consultas, basada en lógica.

Vamos a ver una aplicación de este formalismo: analizar algoritmos para resolver el problema de minimización

Definición: Contención

Q está **contenida** en Q' si para toda instancia D sobre S ,
 $Q(D)$ es **subconjunto** $Q(D')$

El Problema de Contención

consiste en decidir si una consulta está contenida en otra

Definición: Contención

Q está **contenida** en Q' si para toda instancia D sobre S ,
 $Q(D)$ es **subconjunto** $Q(D')$

Notar que Q **es equivalente a** Q' si

- Q está **contenida** en Q'
- Q' está **contenida** en Q

Nos concentramos en contención
(es más simple)

Dadas consultas Q y Q' ,
un homomorfismo h de Q a Q'
es una función $h : \text{Var}(Q') \rightarrow \text{Var}(Q)$ tal que:

1. Para cada igualdad $x = y$ en Q' se tiene que $h(x) = h(y)$, y
2. Para cada átomo $R(x_1, \dots, x_n)$ en Q' ,
el átomo $R(h(x_1), \dots, h(x_n))$ está en Q .

Dadas consultas Q y Q' ,
un homomorfismo h de Q a Q'
es una función $h : \text{Var}(Q') \rightarrow \text{Var}(Q)$ tal que:

1. Para cada igualdad $x = y$ en Q' se tiene que $h(x) = h(y)$, y
2. Para cada átomo $R(x_1, \dots, x_n)$ en Q' ,
el átomo $R(h(x_1), \dots, h(x_n))$ está en Q .

Q

$\text{Películas}(x,y,z) \wedge \text{Programación}(y,w)$

Q'

$\text{Películas}(x,y,z)$

Dadas consultas Q y Q' ,
un homomorfismo h de Q a Q'
es una función $h : \text{Var}(Q') \rightarrow \text{Var}(Q)$ tal que:

1. Para cada igualdad $x = y$ en Q' se tiene que $h(x) = h(y)$, y
2. Para cada átomo $R(x_1, \dots, x_n)$ en Q' ,
el átomo $R(h(x_1), \dots, h(x_n))$ está en Q .

Q

$\text{Películas}(x,y,z) \wedge \text{Programación}(y,w)$

$\text{Var}(Q): \{x,y,z,w\}$

Q'

$\text{Películas}(x,y,z)$

$\text{Var}(Q'): \{x,y,z\}$

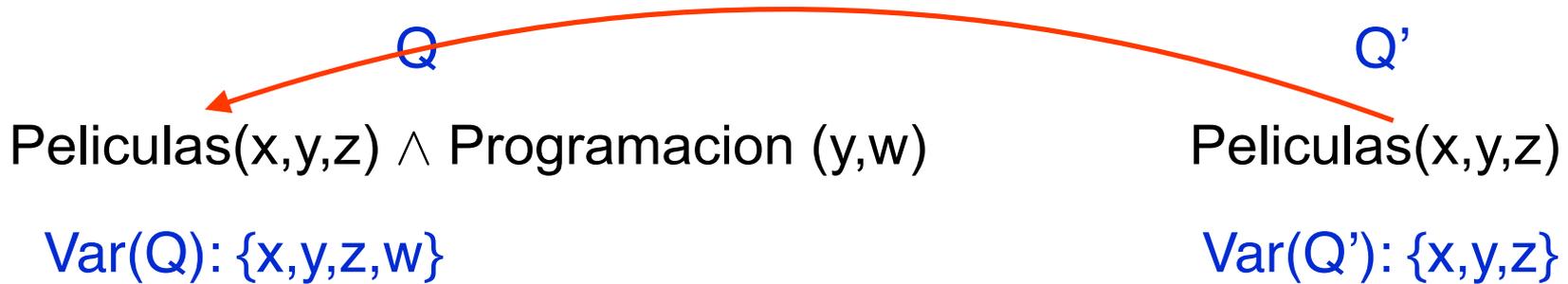
$h: x \rightarrow x$

$y \rightarrow y$

$z \rightarrow z$

Dadas consultas Q y Q' ,
un homomorfismo h de Q a Q'
es una función $h : \text{Var}(Q') \rightarrow \text{Var}(Q)$ tal que:

1. Para cada igualdad $x = y$ en Q' se tiene que $h(x) = h(y)$, y
2. Para cada átomo $R(x_1, \dots, x_n)$ en Q' ,
el átomo $R(h(x_1), \dots, h(x_n))$ está en Q .



$$\begin{aligned} h: x &\rightarrow x \\ y &\rightarrow y \\ z &\rightarrow z \end{aligned}$$

Dadas consultas Q y Q' ,
un homomorfismo h de Q a Q'
es una función $h : \text{Var}(Q') \rightarrow \text{Var}(Q)$ tal que:

1. Para cada igualdad $x = y$ en Q' se tiene que $h(x) = h(y)$, y
2. Para cada átomo $R(x_1, \dots, x_n)$ en Q' ,
el átomo $R(h(x_1), \dots, h(x_n))$ está en Q .

Q

Películas(x,y,y)

Q'

Películas(x,y,z)

Dadas consultas Q y Q' ,
un homomorfismo h de Q a Q'
es una función $h : \text{Var}(Q') \rightarrow \text{Var}(Q)$ tal que:

1. Para cada igualdad $x = y$ en Q' se tiene que $h(x) = h(y)$, y
2. Para cada átomo $R(x_1, \dots, x_n)$ en Q' ,
el átomo $R(h(x_1), \dots, h(x_n))$ está en Q .

Q

Películas(x,y,y)

$\text{Var}(Q): \{x,y\}$

Q'

Películas(x,y,z)

$\text{Var}(Q'): \{x,y,z\}$

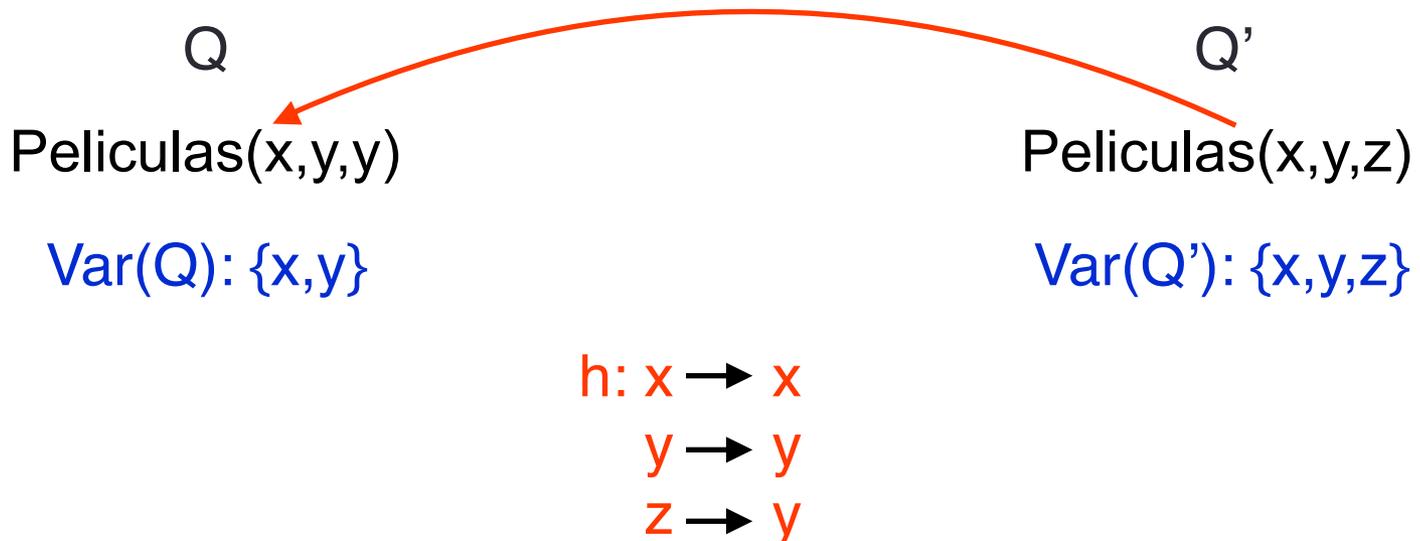
$h: x \rightarrow x$

$y \rightarrow y$

$z \rightarrow y$

Dadas consultas Q y Q' ,
un homomorfismo h de Q a Q'
es una función $h : \text{Var}(Q') \rightarrow \text{Var}(Q)$ tal que:

1. Para cada igualdad $x = y$ en Q' se tiene que $h(x) = h(y)$, y
2. Para cada átomo $R(x_1, \dots, x_n)$ en Q' ,
el átomo $R(h(x_1), \dots, h(x_n))$ está en Q .



Teorema.

Sean Q y Q' dos CQs.

Q está **contenida** en Q'

si y solo si

existe un **homomorfismo** de Q' a Q

$\text{Películas}(x,y,z) \wedge \text{Programación}(y,w)$ está contenida en

$\text{Películas}(x,y,z)$

$\text{Películas}(x,y,y)$ está contenida en

Teorema.

Sean Q y Q' dos CQs.

Q está **contenida** en Q'

si y solo si

existe un **homomorfismo** de Q' a Q

Algoritmo para razonar sobre algebra relacional:

- Pasar de álgebra relacional a lógica
- Buscar un homomorfismo

Teorema.

Sean Q y Q' dos CQs.

Q es la **minimización** de Q'

si y solo si:

- Q es **equivalente** a Q'
- Q' es la consulta con **menor número de átomos** que satisface esa condición

Algoritmo para minimizar:

- Pasar de álgebra relacional a lógica
- Búsqueda exhaustiva de la consulta mínima

El problema de minimización de consultas es NP-hard,
creemos que no hay un algoritmo eficiente para resolver esto

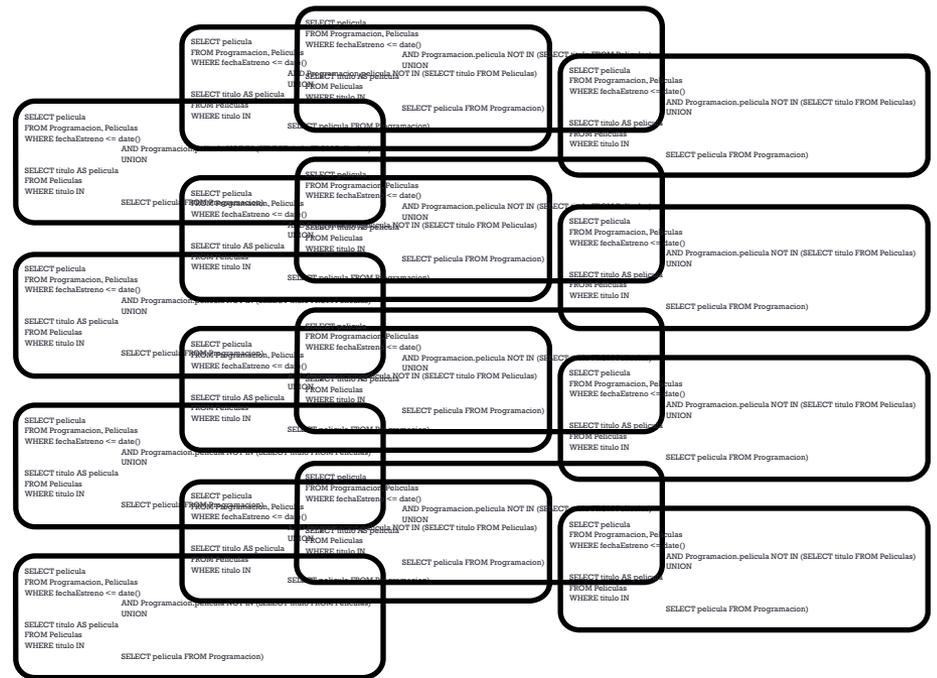
El problema de minimización de consultas es NP-hard, creemos que no hay un algoritmo eficiente para resolver esto

Solución práctica: búsqueda exhaustiva con heurísticas.

El problema de minimización de consultas es NP-hard, creemos que no hay un algoritmo eficiente para resolver esto

Solución práctica: búsqueda exhaustiva con heurísticas.

```
SELECT pelcula
FROM Programacion, Pelculas
WHERE fechaEstreno <= date()
AND Programacion.pelcula NOT IN (SELECT titulo FROM Pelculas)
UNION
SELECT titulo AS pelcula
FROM Pelculas
WHERE titulo IN
      SELECT pelcula FROM Programacion)
```



No buscamos todas las posibles consultas mínimas

El problema de minimización de consultas es NP-hard, creemos que no hay un algoritmo eficiente para resolver esto

Solución práctica: búsqueda exhaustiva con heurísticas.

```
SELECT pelucia
FROM Programacion, Pelucas
WHERE fechaEstreno <= date()
      AND Programacion.pelucia NOT IN (SELECT titulo FROM Pelucas)
UNION
SELECT titulo AS pelucia
FROM Pelucas
WHERE titulo IN (SELECT pelucia FROM Programacion)
```



Usamos heurísticas para reducir el espacio

El problema de minimización de consultas es NP-hard, creemos que no hay un algoritmo eficiente para resolver esto

Solución práctica: búsqueda exhaustiva con heurísticas.

```
SELECT pelicula
FROM Programacion, Peliculas
WHERE fechaEstreno <= date()
AND Programacion.pelicula NOT IN (SELECT titulo FROM Peliculas)
UNION
SELECT titulo AS pelicula
FROM Peliculas
WHERE titulo IN
      SELECT pelicula FROM Programacion)
```



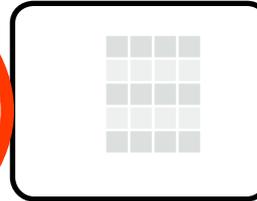
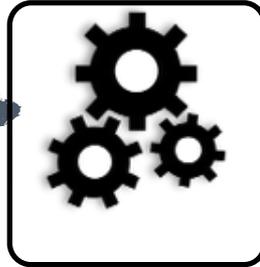
Puede que el resultado no sea óptimo, pero es suficiente para fines prácticos

Los sistemas hoy pueden optimizar muchas clases de consultas

¿Qué película están
pasando hoy?

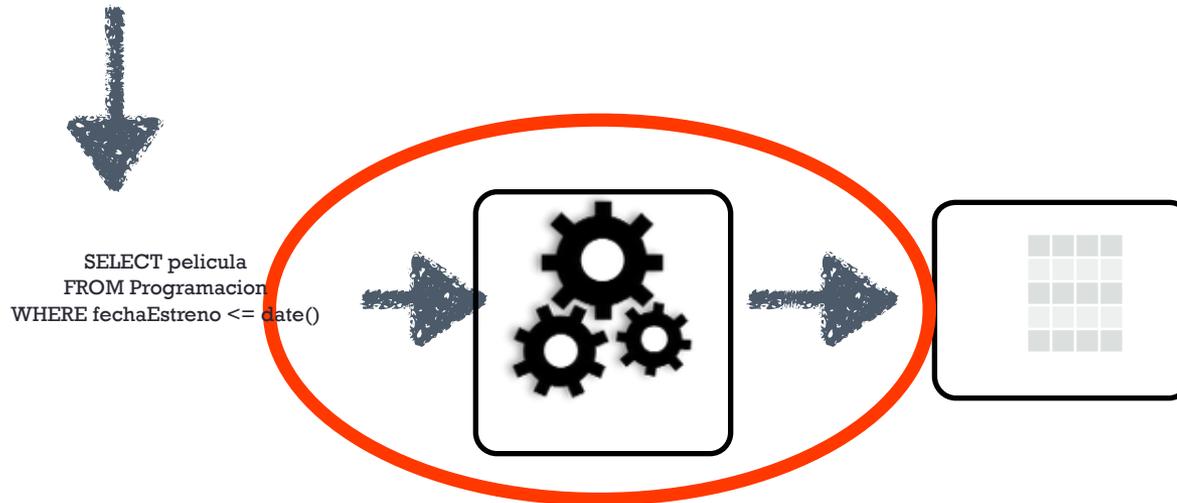


```
SELECT pelicula  
FROM Programacion  
WHERE fechaEstreno <= date()
```



Los sistemas hoy pueden optimizar muchas clases de consultas

¿Qué película están
pasando hoy?



Se usan una variedad de técnicas, algoritmos muy sofisticados, pero todos han surgido de estas bases teóricas

La investigación en bases de datos ha contribuido (y sigue contribuyendo) a los avances en este campo

Un poco de historia

Problemática actual

La investigación en bases de datos ha contribuido (y sigue contribuyendo) a los avances en este campo

NoSQL y nuevos modelos de datos

Problemática actual

Datos en la Web, Web Semántica

Integración de datos

Sistema de bases de datos relacionales no están preparados para entornos altamente distribuidos

- WWW, facebook, twitter, etc
- Integración de muchas bases de datos diferentes

Necesitamos nuevos modelos y nuevos sistemas, mejor preparados para estos desafíos.

Todos estos reciben el nombre de NoSQL

La irrupción de NoSQL trae consigo miles de nuevos desafíos en teoría de bases de datos

- Nuevos modelos de datos (JSON, XML, grafos, CSV, ...)
- Nuevos sistemas (Hadoop, MongoDB, Neo4j, ...)

Lenguajes de consulta deben ser estandarizados

Nuevos algoritmos de procesamiento

Herramientas para entender los nuevos modelos

Comunicación, acceso a datos guardados de distinta forma

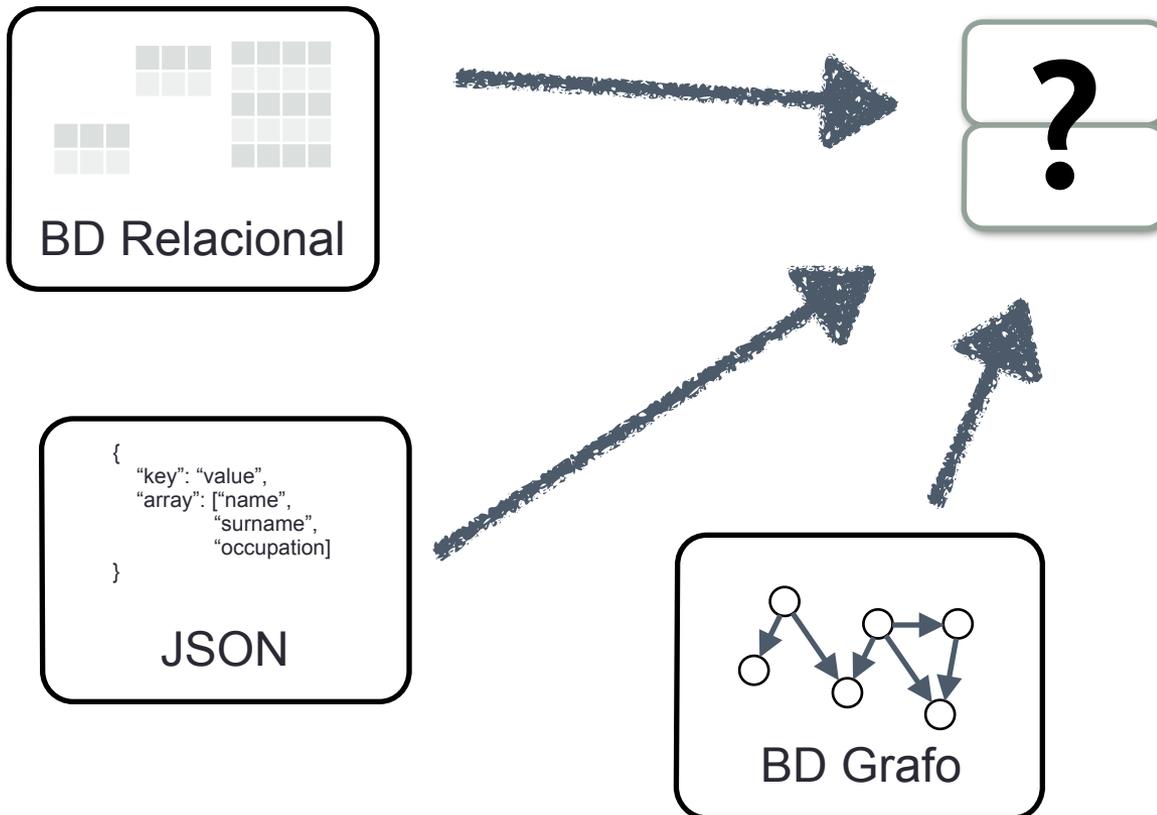
WWW:

La base de datos más grande y más completa

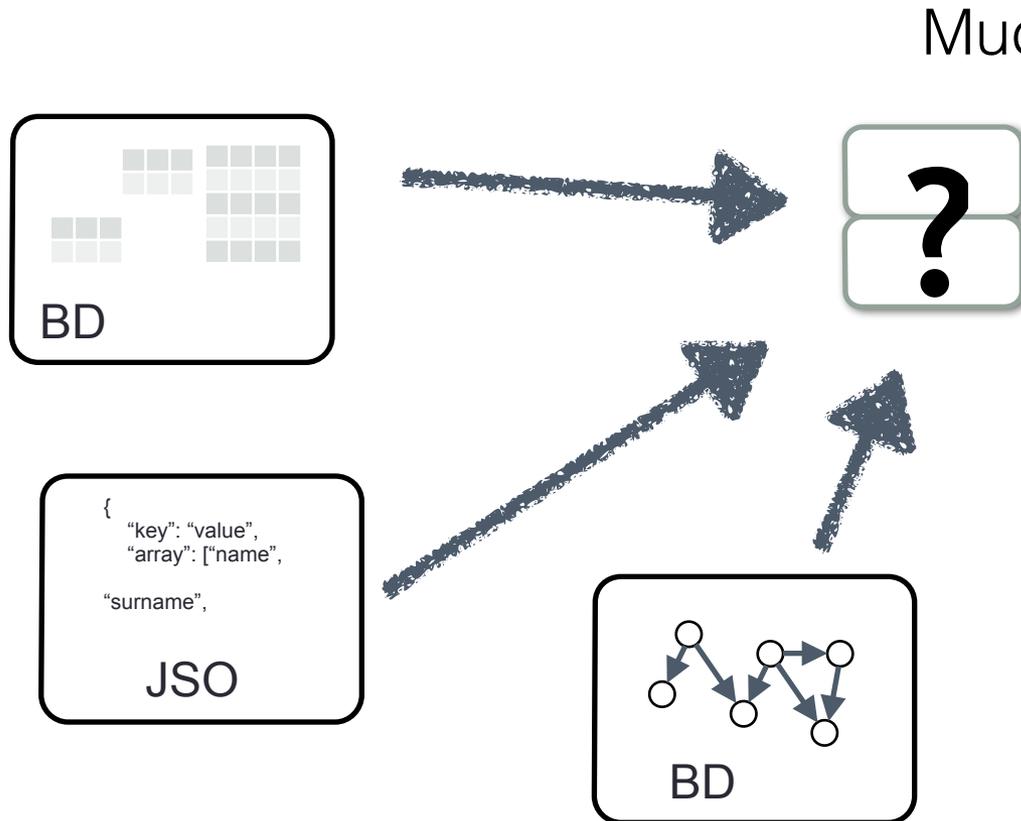
- Web Semántica
- Búsqueda y extracción de información en la Web (en la tarde)

Integración de datos:

¿Cómo extraer información de muchas fuentes distintas?



Integración de datos



Muchos avances
usando ontologías:

- Mapeo cada modelo a una ontología común
- Proceso consultas usando la ontología

Más información: en la tarde!

Centro de investigación de la Web Semántica

Center for **Semantic Web** Research



10 Profesores en universidades chilenas
30+ alumnos

Centro de investigación de la Web Semántica

Estandarización de lenguajes:

Lenguaje de consulta para grafos
(junto con grandes compañías)

SPARQL, JSON, CSV

Análisis de sistemas de manejo de datos:

Web semántica, grafos e incluso sistemas relacionales

Extracción de información en la Web:

Modelos de detección de terremotos via Twitter

Búsqueda de información en la Web Semántica