

# Evolución Flexible de Aplicaciones de BD mediante Wrappers

Fco Javier Aranda Granda (alumno)  
Jorge Lloret Gazo (director)

Máster en Ingeniería de Sistemas e Informática  
Programa Oficial de Posgrado de Ingeniería Informática

Septiembre 2010

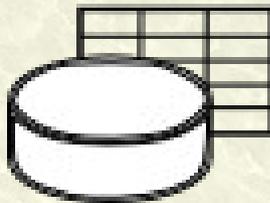
# Definición del problema

Esquema



El esquema de BD contiene la definición de qué datos se almacenan y de qué manera se almacenan.

Datos



Los datos almacenados en la BD se ajustan a la definición del esquema.

Programa



Los programas que acceden a la BD también deben ajustarse al esquema.

# Definición del problema

- El esquema de BD es susceptible de evolucionar
  - Necesidad de nuevos datos
  - Necesidad de reestructurar los datos
- Los elementos asociados deberán cambiar con el esquema

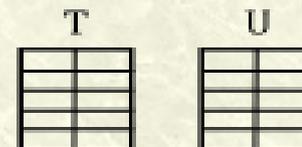
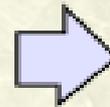
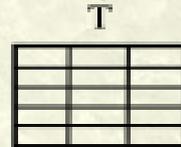
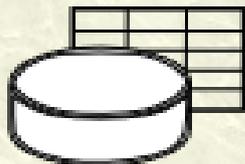
Programa



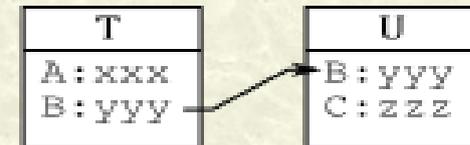
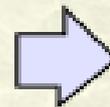
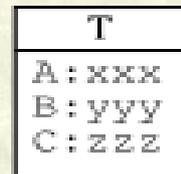
```
select A,B,C from T
```

```
select A,T.B,C from T join U
```

Datos



Esquema



# Caso de estudio: $\mu$ Wiki

- Se ha desarrollado una aplicación de ejemplo ( $\mu$ Wiki) (sencilla y fácil de trabajar, pero que contiene un escenario verosímil)
- El esquema está basado en MediaWiki
  - Es utilizado por otros autores [Curino]
  - Supone un desafío
- Es interesante porque otros pueden experimentar sobre el mismo ejemplo

# Caso de estudio: $\mu$ Wiki

## $\mu$ Wiki: Page Directory

You are logged as **aranda** [[Log out](#)]

### Commands

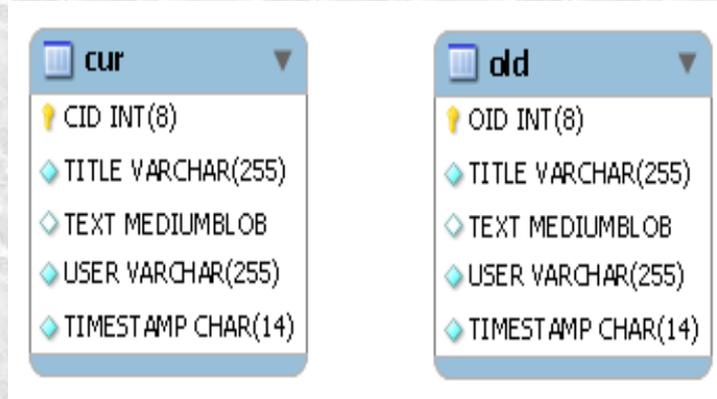
[Create new page](#)

[List Full History](#)

Title	User	Last Modified	Commands
Deimos	aranda	12/11/2009 16:29	<a href="#">View</a> <a href="#">Edit</a>
Europa	aranda	12/11/2009 16:31	<a href="#">View</a> <a href="#">Edit</a>
Ganimedes	aranda	12/11/2009 16:32	<a href="#">View</a> <a href="#">Edit</a>
Io	aranda	23/10/2009 03:59	<a href="#">View</a> <a href="#">Edit</a>
Júpiter	aranda	23/10/2009 03:58	<a href="#">View</a> <a href="#">Edit</a>
Luna	aranda	16/10/2009 03:53	<a href="#">View</a> <a href="#">Edit</a>
Marte	aranda	26/10/2009 17:36	<a href="#">View</a> <a href="#">Edit</a>
Mercurio	editor1	01/09/2009 00:00	<a href="#">View</a> <a href="#">Edit</a>
Phobos	aranda	26/10/2009 17:37	<a href="#">View</a> <a href="#">Edit</a>
Tierra	aranda	16/10/2009 01:24	<a href="#">View</a> <a href="#">Edit</a>
Venus	aranda	12/11/2009 12:56	<a href="#">View</a> <a href="#">Edit</a>

# Caso de estudio: $\mu$ Wiki

## CUR



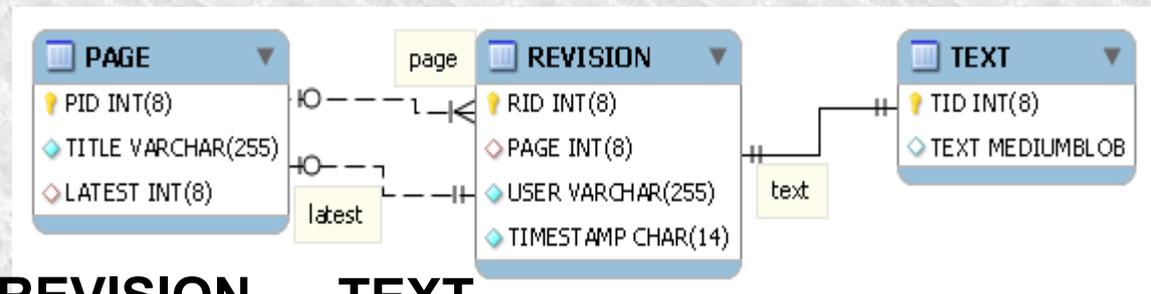
**Esquema inicial (v.41)**

CID	TITLE	TEXT
1	Mercurio	Primer planeta
3	Venus	Segundo planeta

## OLD

CID	TITLE	TEXT
2	Venus	Segundo

**Esquema final (v.42)**



## PAGE

PID	TITLE	LATEST
P1	Mercurio	1
P2	Venus	3

## REVISION

PAGE	RID
P1	1
P2	2
P2	3

## TEXT

TID	TEXT
1	Primer planeta
2	Segundo
3	Segundo planeta

# Caso de estudio: $\mu$ Wiki

```
select CID, TITLE, USER, TEXT, TIMESTAMP from CUR where TITLE=?
```



```
select RID AS CID, TITLE, USER, TEXT, TIMESTAMP from REVISION r, PAGE p,  
TEXT t where r.RID=p.LATEST and r.RID=t.TID and p.TITLE=?
```

---

```
[1] insert into OLD (OID, TITLE, USER, TEXT, TIMESTAMP) select CID, TITLE,  
USER, TEXT, TIMESTAMP from CUR where TITLE=$title;  
[2] delete from CUR where TITLE=$title;  
[3] insert into CUR (TITLE, USER, TEXT, TIMESTAMP) values($title,$u,$tx,$tm);
```



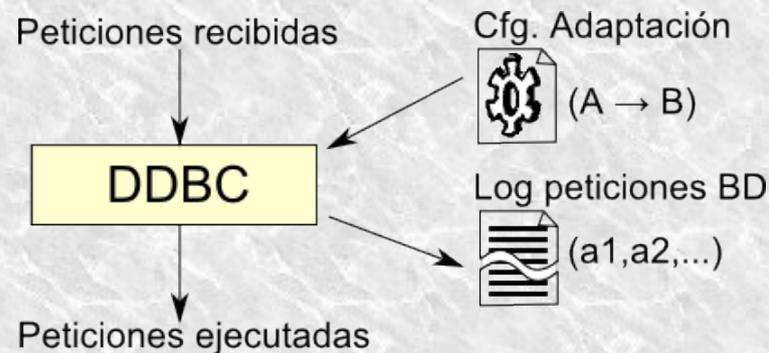
```
[1,2] (omitir)  
[3a] insert into REVISION (USER, TIMESTAMP, PAGE) values($u,$tm,NULL);  
(returning rid into $rid)  
[3b] insert into PAGE (LATEST, TITLE) values($rid, $title) on duplicate key  
update LATEST=$rid;  
(returning pid into $pid)  
[3c] insert into TEXT (TID, TEXT) values ($rid, $tx);  
[3d] update REVISION set PAGE=$pid where RID = $rid;
```

# Descripción de DDBC

- Arquitectura de la implementación
- Descripción de los componentes
- Demo de adaptación
- Conclusiones

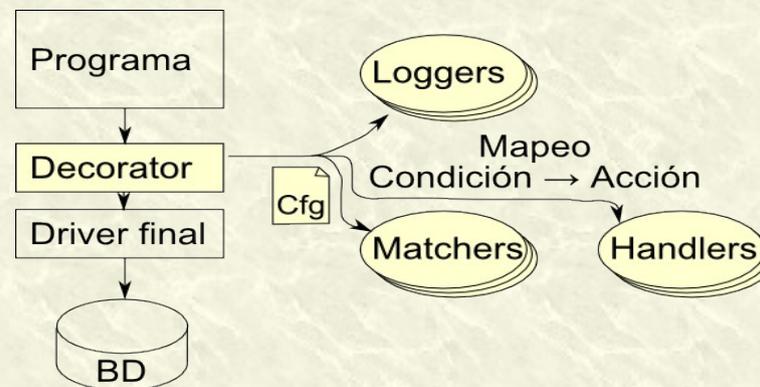
# DDBC: Arquitectura

- DDBC es Decorator DB Connection (Decorator por el patrón de diseño)
- Decorator=Wrapper encadenable
- Funcionalidad realizada:
  - Traducción de las peticiones (según configuración)
  - Escritura de log



# DDBC: Arquitectura

- DDBC es altamente configurable
  - Se pueden elegir y configurar los componentes que realizarán las tareas concretas
  - Se permite incluso al usuario que pueda definir y utilizar sus componentes personalizados
    - Loggers: Registran los accesos a BD
    - Matchers: Identifican la operación realizada para seleccionar el tratamiento correspondiente
    - Handlers: realizan el tratamiento en sí



# DDBC: Matchers y Handlers

## Traducción literal

- Se identifican las sentencias que coincidan con la especificada (es tolerante respecto a mayúsculas y espacios)
- Es útil cuando no hay partes variables

```
<mapping id="select title from cur">  
  <matcher type="equals">select CID, TITLE from CUR where title=?  
  </matcher>  
  <handler type="literal">select LATEST as CID, TITLE from PAGE  
    where TITLE=?</handler>  
</mapping>
```

# DDBC: Matchers y Handlers

## Traducción por expresiones regulares

- Se identifican las sentencias que cumplan la expresión regular dada
- Permite realizar la traducción mediante un conjunto de reglas expresión buscada-expresión sustituta

```
select CID, * from CUR where *
```



```
select RID as CID, * from PAGE P, REVISION R, TEXT T WHERE  
P.LATEST=R.RID AND R.RID=T.TID AND *
```

```
<matcher type="regexMatcher">select CID, (.*) from CUR where (.*)</matcher>  
<handler type="regexHandler" >  
  <rx-rule>  
    <rx-match>select CID, (.*) from CUR where (.*)</rx-match>  
    <rx-subst>select PID AS CID, $1 from PAGE p, REVISION r, TEXT t  
      where p.LATEST=r.RID and r.RID=t.TID and $2</rx-subst>  
  </rx-rule>  
</handler>
```

# DDBC: Matchers y Handlers

## Otros elementos predefinidos

- Condición otherwise: Permite asociar un tratamiento a las sentencias que no cumplan las reglas indicadas
- Handler passthrough: “pasa” la petición inalterada (para sentencias no afectadas, previsiblemente muchas)
- Handler fail: No se ejecutará, y lanzará un error
  - Útil para cubrirnos en salud ante lo inesperado
  - Útil si no hemos implementado (todavía) el tratamiento

```
<matcher type="literal">select 1 from DUAL</matcher>  
<handler type="passThrough" unexpected="false" />
```

```
<otherwise>  
  <handler type="fail">Sentencia inesperada</handler>  
</otherwise>
```

# DDBC: Matchers y Handlers

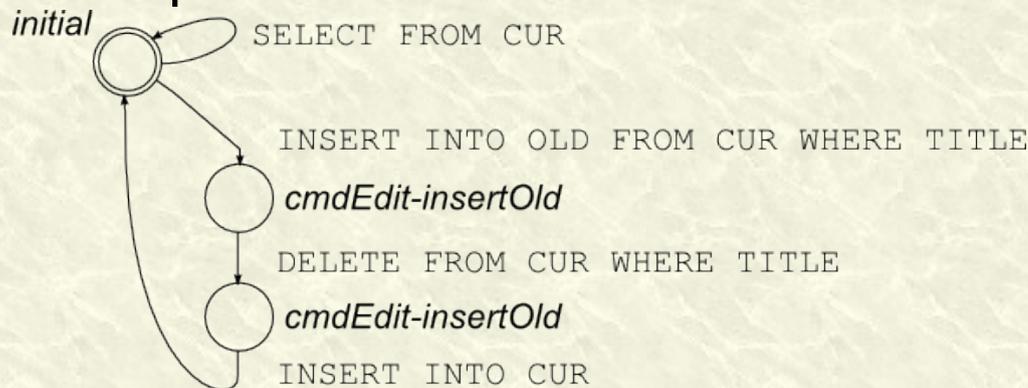
## Matchers y Handlers personalizados

- Para las traducciones complejas no cubiertas por los elementos predefinidos
- Los programadores deberán desarrollarlos. Se pretende ofrecer funciones de utilidad y ejemplos variados

```
<def type="cmdEdit" class="uwiki.evol.HandlerCmdEdit410in412" />
  . . . . .
<mapping id="delete cur where title">
  <matcher type="equals" fromState="cmdEdit-deleteCur">
    DELETE FROM cur WHERE title=?</matcher>
  <handler type="cmdEdit" toState="cmdEdit-insertCur" />
</mapping>
```

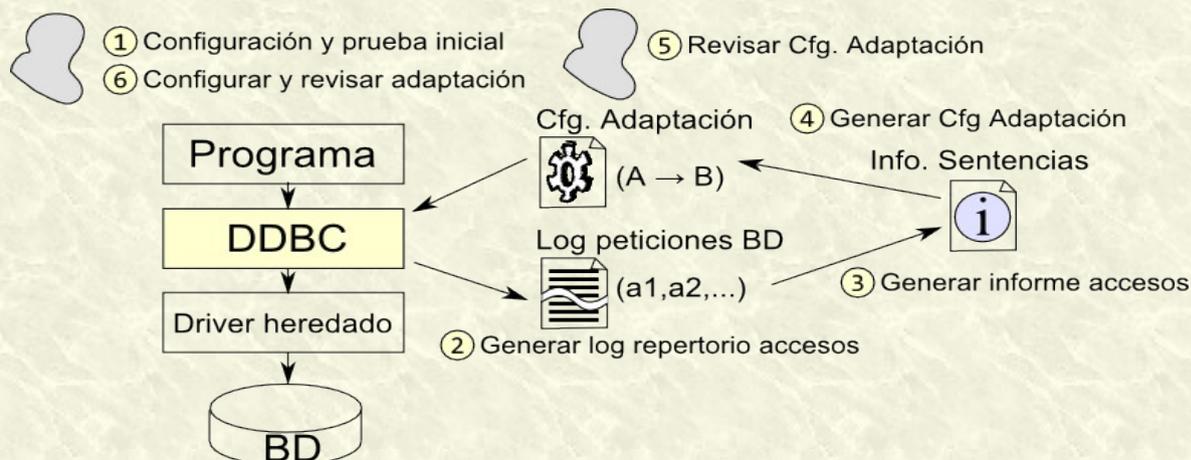
# DDBC: Matchers y Handlers

- Tratamientos dependientes del contexto (estado) (e.g. el “delete from CUR”)
  - Puede depender de las sentencias ejecutadas antes (el “insert into OLD” previo hace que no sea eliminación)
  - Puede depender de las sentencias posteriores (el “insert into CUR” posterior haría innecesario establecer la columna latest a NULL)
- Por ello se mantiene una variable de estado
  - Depende de las sentencias previas, se establece con `toState`
  - Se comprueba con `fromState`

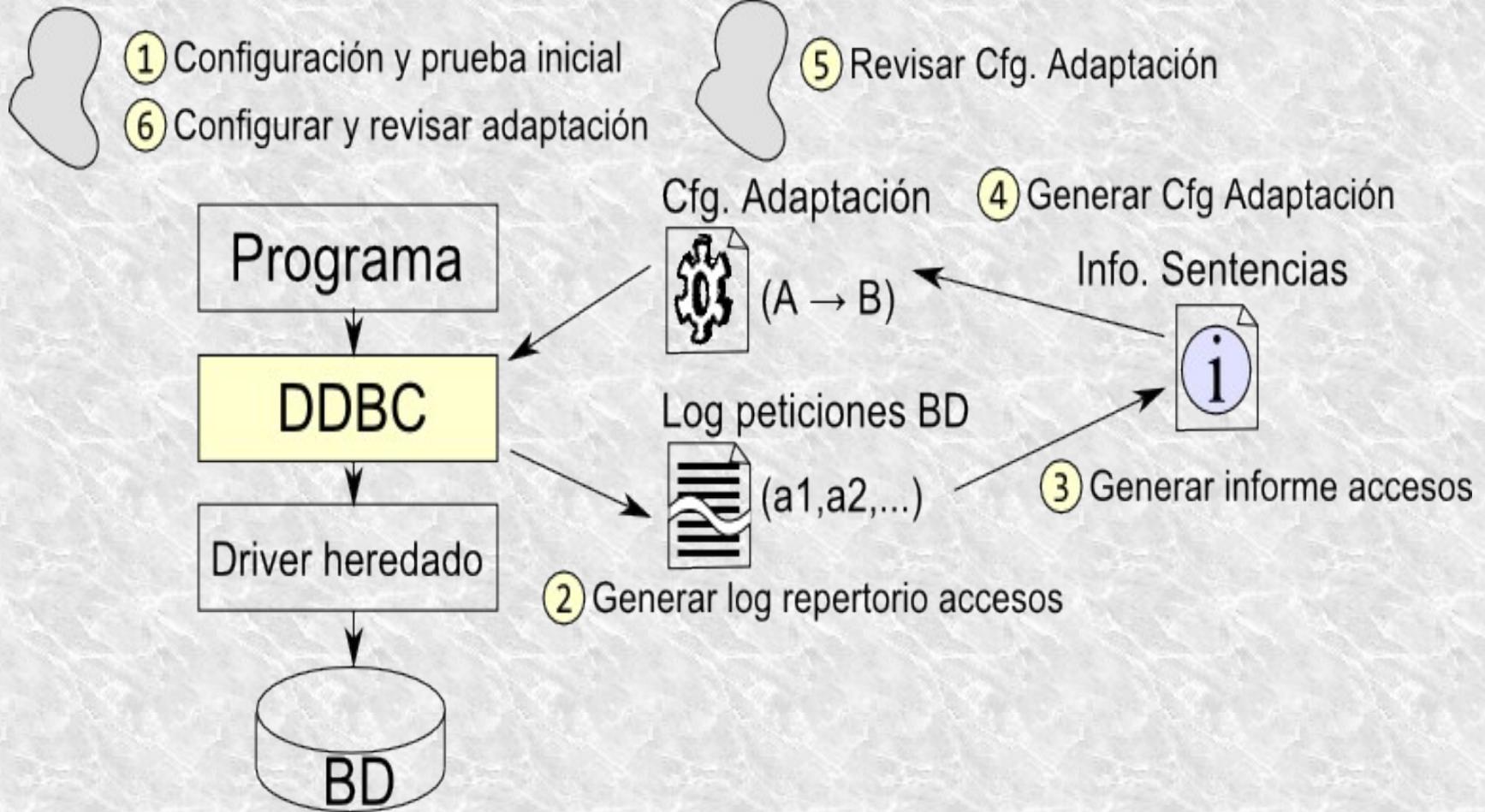


# DDBC: Proceso de adaptación

- En primer lugar se recopila la información de los accesos a datos del programa
  - El repertorio ejecutado debe ser suficientemente amplio
  - Recopilamos los datos involucrados y las ubicaciones
- Elaboramos un informe de sentencias
  - Se resumen las sentencias repetidas
  - Se detectan las dependencias entre sentencias
- Elaboramos “esqueleto” de la adaptación
  - Deberá ser revisado y probado por un técnico



# DDBC: Proceso de adaptación



# DDBC: Conclusiones, aportaciones

- Desarrollado un wrapper JDBC que permite la adaptación de programas que utilizan BD
  - Sin necesidad de analizar o modificar el código
  - Permitiendo la adaptación de operaciones complejas como la del caso de estudio.  
(pero alguien tiene que hacer la especificación)
- No obstante todavía queda mucho por hacer ...

# Otros enfoques

## Cuestiones abordadas por otros enfoques

- ¿Cómo obtener la información necesaria?
- ¿Cómo especificamos la modificación del esquema?
- ¿Qué podemos hacer por los usuarios?

# Otros Enfoques: Análisis estático

- Inspeccionar el código para determinar qué sentencias SQL se ejecutan y dónde
  - Partiendo de las expresiones de llamada al API de BD, se determinan los valores posibles de las variables relevantes (la sentencia SQL) mediante un análisis de dependencia de datos.

```
selectT("where a='1'");
```

```
selectT(where) {  
  query=" ";  
  query="select A,B,C from T " + where;  
  exec(query);  
}
```

- ④ Continuar en parametros actuales
- ③ Buscar valor 'where' (determina 'query')
- ② Buscar valor 'query' (variable relevante)
- ① Localizar llamadas API

# Otros: Especificación primitivas

- Las modificaciones del esquema se pueden especificar como una secuencia de operaciones primitivas (ADD COLUMN, RENAME COLUMN, SPLIT TABLE, ...)
- Si cada primitiva está asociada a cambios específicos en el programa, entonces la modificación completa se podría realizar mediante la secuencia de modificaciones de cada primitiva
  - (hay dificultades, como en la modificación de  $\mu$ Wiki)
- Referencias: [Curino, Hainaut]

# Otros Enfoques: Mapeo de datos

- Se basa en establecer la asociación de las columnas/campos de un esquema con las columnas/campos otro esquema (E.g. CUR.TITLE del esquema simulado corresponde con PAGE.TITLE del esquema actual)
- Adecuado para la generación de vistas SQL, y la traducción de sentencias [Curino]

# Otros enfoques: Vistas SQL

- Las vistas son tablas virtuales definidas sobre una consulta sobre las tablas actuales (u otras vistas)
- Una tabla de un esquema simulado se puede definir como una vista
- Referencias: [Hainaut, Curino]

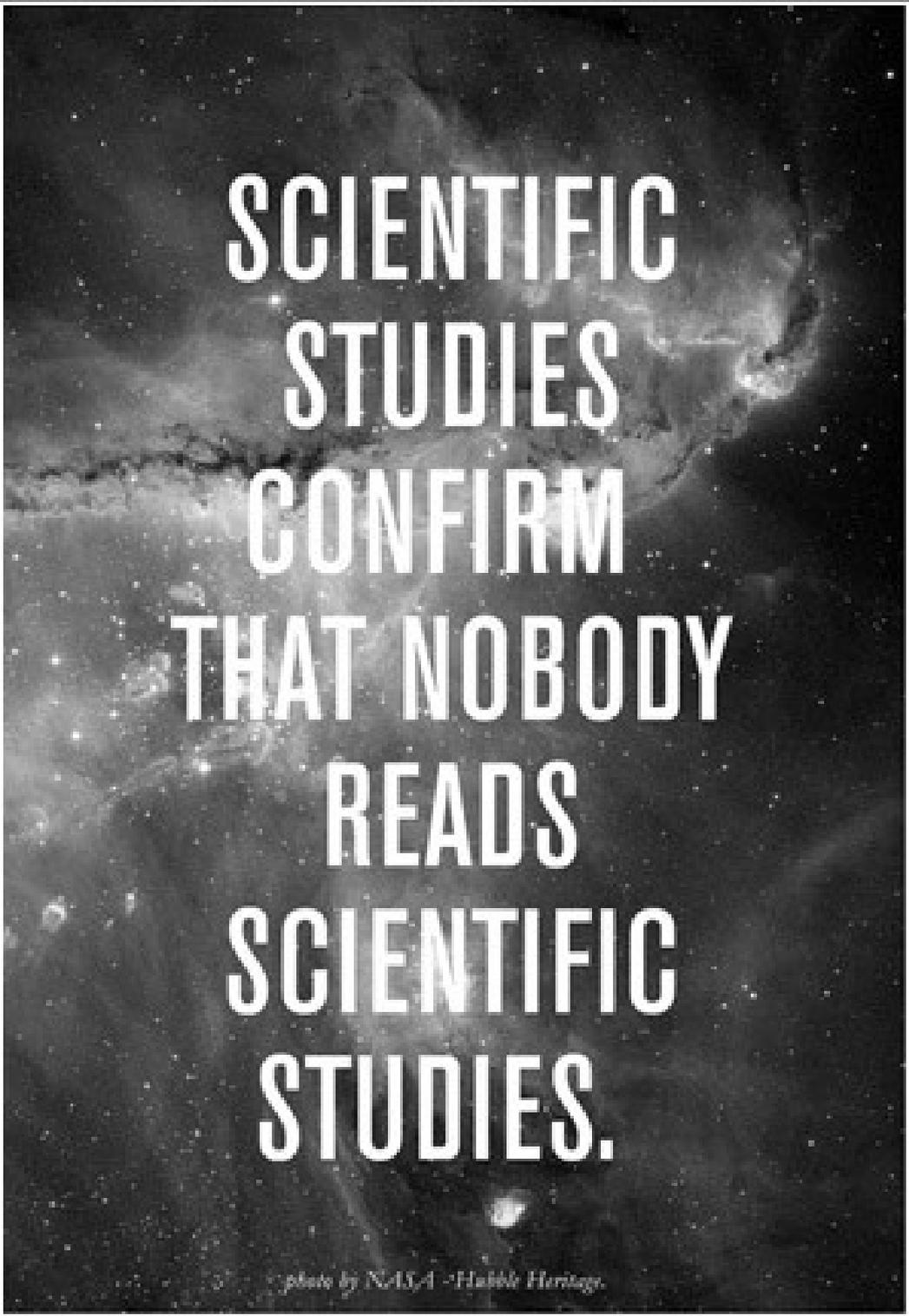
```
create view V41.CUR as select RID as CID, TITLE,  
TEXT, USER, TIMESTAMP from V42.PAGE, V42.REVISION,  
V42.TEXT where LATEST=RID and RID=TID
```

# Enfoques: Impacto en el programa

- Conocido el programa y la modificación se puede elaborar un informe sobre:
  - Los puntos del programa afectados
  - La dificultad/coste estimados del cambio
  - Modificación propuesta
- Este informe puede interesar para
  - Decidir a priori sobre si realizar una modificación
  - Orientar a los técnicos que modifican el programa
- Referencias: [Maule]

# Enfoques: Traducción de SQL

- Conocida la modificación del esquema, existen herramientas [Curino] para traducir consultas SQL sobre dicho esquema
  - Dichas traducciones pueden ser susceptibles de corrección/optimización



**SCIENTIFIC  
STUDIES  
CONFIRM  
THAT NOBODY  
READS  
SCIENTIFIC  
STUDIES.**

*photo by NASA - Hubble Heritage.*

# Bibliografía

## Carlo Curino et al.

- C. Curino, H. Moon, C. Zaniolo  
“*Graceful Database Schema Evolution: the PRISM Workbench*”  
2008,PVLDB 1,1 p761-772

## J.L. Hainaut et al.

- J.M. Hick, J.M. Hainaut,  
“*Database Application Evolution. A transformational Approach*”  
2006,DKE 59,3 p534-558

## A.Maule et al.

- A.Maule, W. Emmerich, D.S.Rosenblum  
“*Impact Analysis of Database Schema Changes*”  
ISCE 2008

Muchas Gracias por la atención  
¿Preguntas?

