

# DES: un recurso para el aprendizaje de bases de datos deductivas

Fernando Sáenz Pérez

Dept. Ingeniería del Software e Inteligencia Artificial

Universidad Complutense de Madrid

Facultad de Informática, c/ Profesor José García Santesmases, s/n,  
28040 Madrid

[fernan@sip.ucm.es](mailto:fernan@sip.ucm.es)

# Índice de la presentación

1. Introducción
2. Características
3. Limitaciones
4. Uso del sistema
5. Implementación
6. Utilidad
7. Conclusiones
8. Enlaces

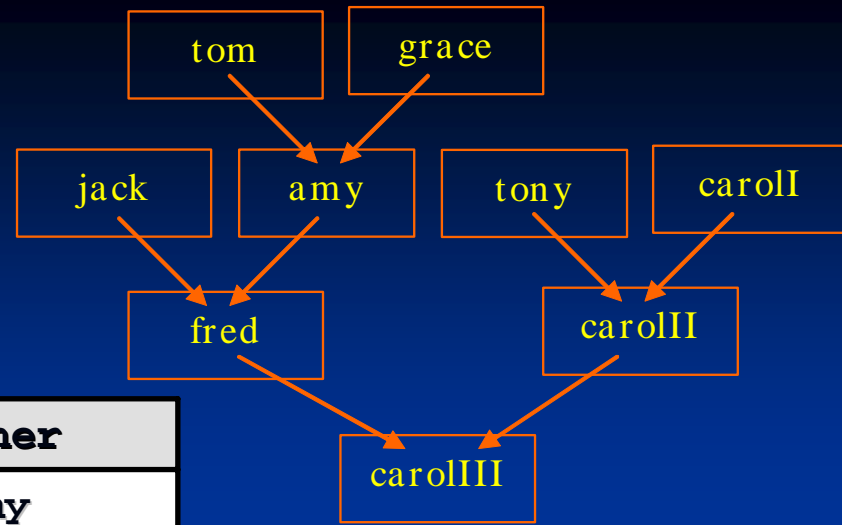
# 1. Introducción

- Bases de datos:  
de las relacionales a las deductivas
- Lenguajes de bases de datos:  
de SQL a Datalog

# 1. Introducción: Datalog

- Lenguaje de consulta de bases de datos deductivas derivado de Prolog
- Predicado (Prolog) – Relación (Datalog)
- Objetivo (Prolog) – Consulta (Datalog)
- Significado de un predicado: conjunto de hechos deducidos:
  - Intensionalmente (reglas o cláusulas)
  - Extensionalmente (hechos)

# 1. Introducción: Datalog



```

father( tom , amy ) .
father( jack , fred ) .
father( tony , carolII ) .
father( fred , carolIII ) .
  
```

```

mother( graceI , amy ) .
mother( amy , fred ) .
mother( carolII , carolIII ) .
mother( carolII , carolIII ) .
  
```

```

parent( X , Y ) :- father( X , Y ) .
parent( X , Y ) :- mother( X , Y ) .
  
```

```

ancestor( X , Y ) :-
  parent( X , Y ) .
ancestor( X , Y ) :-
  parent( X , Z ) ,
  ancestor( Z , Y ) .
  
```

father	
tom	amy
jack	Fred
tony	carolII
fred	carolIII

```

ancestor( tom , X ) =
{
  ancestor( tom , amy ) ,
  ancestor( tom , carolIII ) ,
  ancestor( tom , fred )
}
  
```

ancestor	
tom	amy
tom	carolIII
tom	fred

# 1. Introducción: sistemas

- Sistemas de bases de datos deductivas:  
LDL++, XSB, SDS, Declare, ConceptBase, ...
- Un nuevo sistema, ¿por qué?

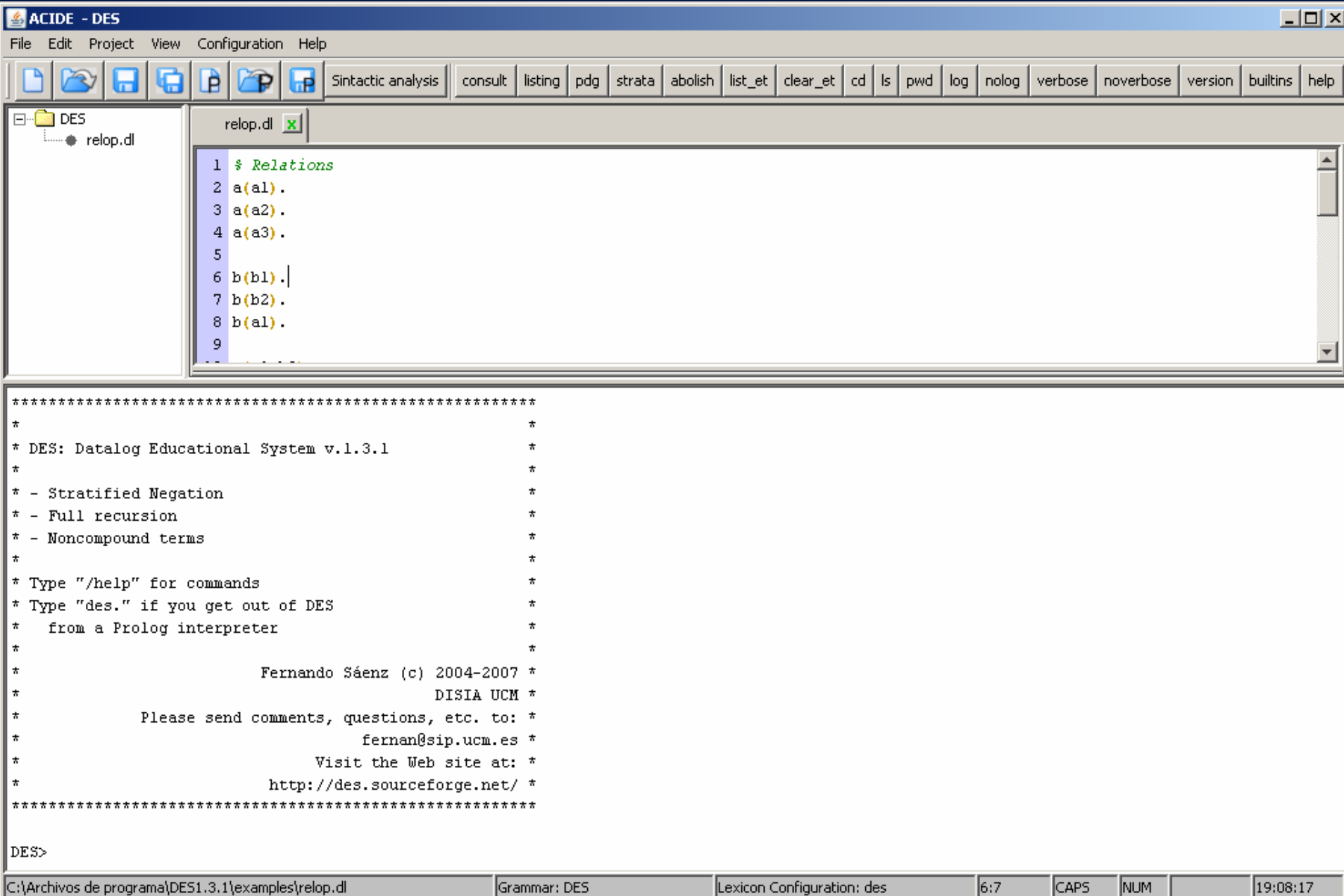
## 2. Características de DES

- Uso simple
- Instalación inmediata
- Mínimos requisitos
- Multiplataforma
- Vistas transitorias
- Recursión no lineal (vs. SQL)
- Negación estratificada
- BD en memoria
- Eficiencia (tablas de extensión)

# 3. Limitaciones de DES

- Agregados (SUM, AVG, ...)
- Aritmética
- Términos compuestos (OO)
- Non-disjunctive
- Depuración
- IDE ... ¿limitación?

# 4. Uso del sistema



The screenshot shows the ACIDE - DES software interface. The top window is a file editor for 'relop.pl' containing the following Prolog code:

```
1 % Relations
2 a(a1).
3 a(a2).
4 a(a3).
5
6 b(b1).
7 b(b2).
8 b(a1).
9
```

The bottom window is a command-line interface for the Datalog Educational System (DES). It displays the following text:

```
*****
*
* DES: Datalog Educational System v.1.3.1
*
* - Stratified Negation
* - Full recursion
* - Noncompound terms
*
* Type "/help" for commands
* Type "des." if you get out of DES
*   from a Prolog interpreter
*
*           Fernando Sáenz (c) 2004-2007
*           DISIA UCM
*
* Please send comments, questions, etc. to:
*           fernan@sip.ucm.es
*
* Visit the Web site at:
*           http://des.sourceforge.net/
*****
DES>
```

The status bar at the bottom of the window shows the following information:

C:\Archivos de programa\DES1.3.1\examples\relop.pl	Grammar: DES	Lexicon Configuration: des	6:7	CAPS	NUM	19:08:17
--	--------------	----------------------------	-----	------	-----	----------

```

relop.dl x
1  % Relations
2  a(a1).
3  a(a2).
4  a(a3).
5
6  b(b1).
7  b(b2).
8  b(a1).
9
10 c(a1,b2).
11 c(a1,a1).
12 c(a2,b2).
13
14 % Relational Operations
15
16 % pi(X)(c(X,Y))
17 projection(X) :- c(X,Y).
18
19 % sigma(X=a2)(a)
20 selection(X) :- a(X), X=a2.
21
22 % a X b
23 cartesian(X,Y) :- a(X), b(Y).
24
25 % a |x| b
26 join(X) :- a(X), b(X).
27
28 % a U b
29 union(X) :- a(X).
30 union(X) :- b(X).
31
32 % a - b
33 difference(X) :- a(X), not(b(X)).
34

```

```

join(X):- a(X),b(X).
union(X):-a(X).
union(X):-b(X).
difference(X):-a(X),not(b(X)).
Info: Computing predicate dependency graph...
Info: Computing strata...

DES> projection(X)
{
  projection(a1),
  projection(a2)
}

DES> /list_et
Tabla de extensión

Answers:
{
  projection(a1),
  projection(a2),
  c(a1,a1),
  c(a1,b2),
  c(a2,b2)
}

Calls:
{
  projection(_36120),
  c(_36115,_36116)
}

DES> |

```

C:\Archivos de programa\DES1.3.1\ Grammar: DES Lexicon Configurati 6:7 CAP: NUI 19:52

# Vistas transitorias

```
relop.dl x
1  % Relations
2  a(a1).
3  a(a2).
4  a(a3).
5
6  b(b1).
7  b(b2).
8  b(a1).
9
10 c(a1,b2).
11 c(a1,a1).
12 c(a2,b2).
13
14 % Relational Operations
15
16 % pi(X)(c(X,Y))
17 projection(X) :- c(X,Y).
18
19 %sigma(X=a2)(a)
20 selection(X) :- a(X), X=a2.
21
22 % a X b
23 cartesian(X,Y) :- a(X), b(Y).
24
25 % a |x| b
26 join(X) :- a(X), b(X).
27
28 % a U b
29 union(X) :- a(X).
30 union(X) :- b(X).
31
32 % a - b
33 difference(X) :- a(X), not(b(X)).
34
```

```
DES> d(X) :- a(X), not(b(X))

Info: Computing predicate dependency graph...
Info: Computing strata...
Info: Computing by stratum of [b(_62518)].
{
  d(a2),
  d(a3)
}

DES> a(X) :- b(X)

Info: Computing predicate dependency graph...
Info: Computing strata...
{
  a(a1),
  a(a2),
  a(a3),
  a(b1),
  a(b2)
}

DES> |

C:\Archivos de program. Grammar: Lexicon Conf 6:7 CA NL 3:38
```

# Depuración declarativa

- Motivación: dificultad en la depuración de programas
- Depuración procedimental vs. Declarativa ([Shaphiro82], *Algorithmic Program Debugging*)
- Orientada al significado, no al procedimiento de cómputo

# Depuración declarativa: ejemplo

```
between(X,Z):- br(X),br(Y),br(Z),X<Y,Y<Z .
```

← Pairs of non-consecutive elements in the sequence

```
next(X,Y) :- br(X), br(Y), X<Y, not(between(X,Y)).
```

```
next(nil,X) :- br(X), not(has_preceding(X)).
```

← Consecutive elements in a sequence (starting at nil)

```
has_preceding(X) :- br(X), br(Y), X > Y.
```

← Elements having preceding values in the sequence

```
even(nil).
```

```
even(X) :- odd(Z), next(Z,X). ← Elements in an even position+nil
```

```
odd(Y) :- even(Z), next(Z,Y). ← Elements in an odd position
```

```
br_is_even :- even(X), not(next(X,Y)).
```

← Succeeds if the cardinality is even

```
br(a).
```

```
br(b).
```

} Base relation (sequence of elements)

# Depuración declarativa: ejemplo

```
between(X,Z):- br(X),br(Y),br(Z),X<Y,Y<Z .
```

← Pairs of non-consecutive elements in the sequence

```
next(X,Y) :- br(X), br(Y), X<Y, not(between(X,Y)).
```

```
next(nil,X) :- br(X), not(has_preceding(X)).
```

← Consecutive elements in a sequence (starting at nil)

```
has_preceding(X) :- br(X), br(Y), X < Y.
```

← Elements having preceding values in the sequence

```
even(nil).
```

```
even(X) :- odd(Z), next(Z,X).
```

← Elements in an even position+nil

```
odd(Y) :- even(Z), next(Z,Y).
```

← Elements in an odd position

```
br_is_even :- even(X), not(next(X,Y)).
```

← Succeeds if the cardinality is even

```
br(a).
```

```
br(b).
```

} Base relation (sequence of elements)

# Depuración declarativa: sesión práctica

```
DES> /debug br_is_even
```

```
Debugger started ...
```

```
Is br(b) = {br(b)} valid(v)/non-valid(n) [v]? v
```

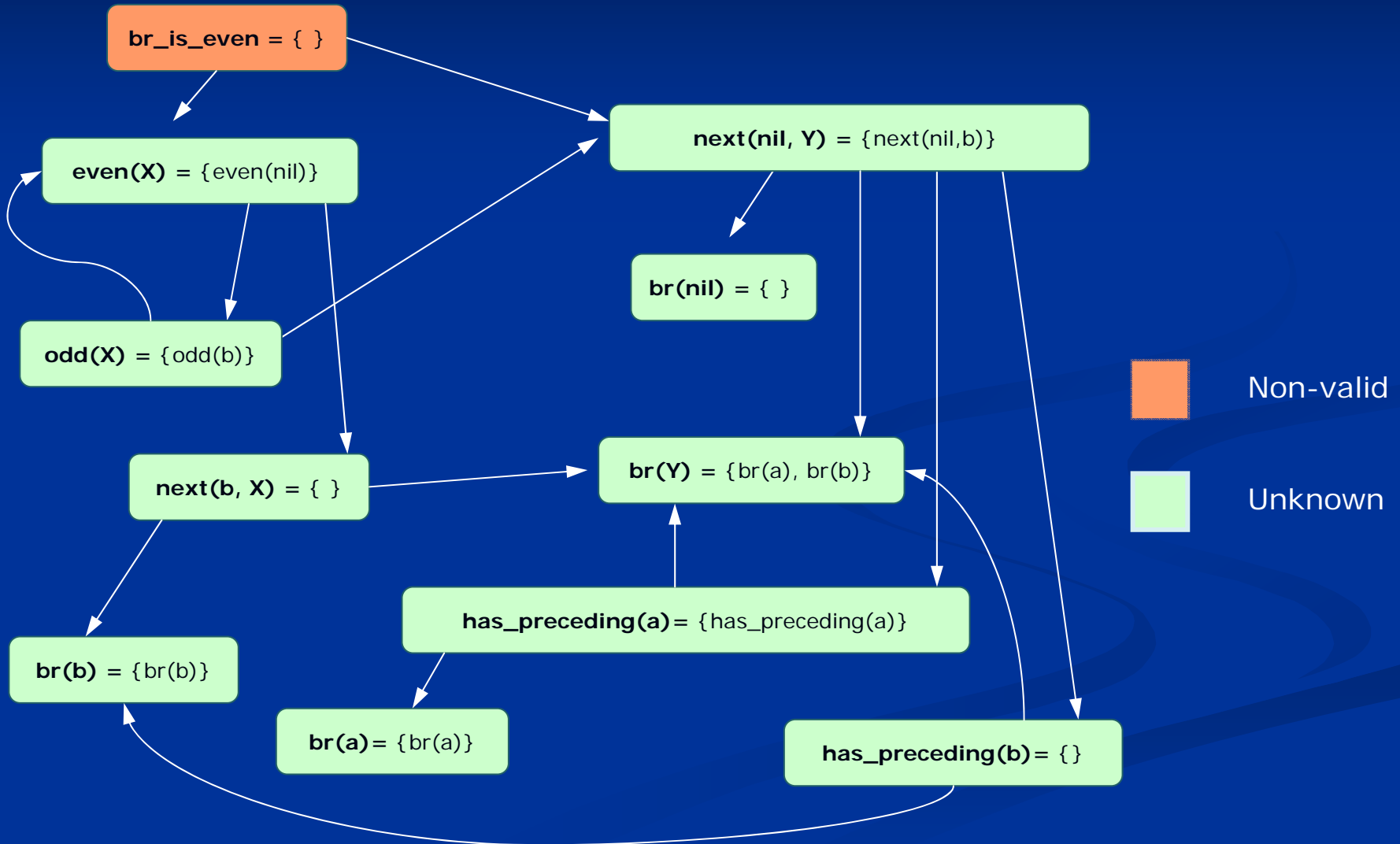
```
Is has_preceding(b) = {} valid(v)/non-valid(n) [v]? n
```

```
Is br(X) = {br(b),br(a)} valid(v)/non-valid(n) [v]? v
```

```
! Error in relation: has_preceding/1
```

```
! Witness query: has_preceding(b) = { }
```

# Depuración declarativa



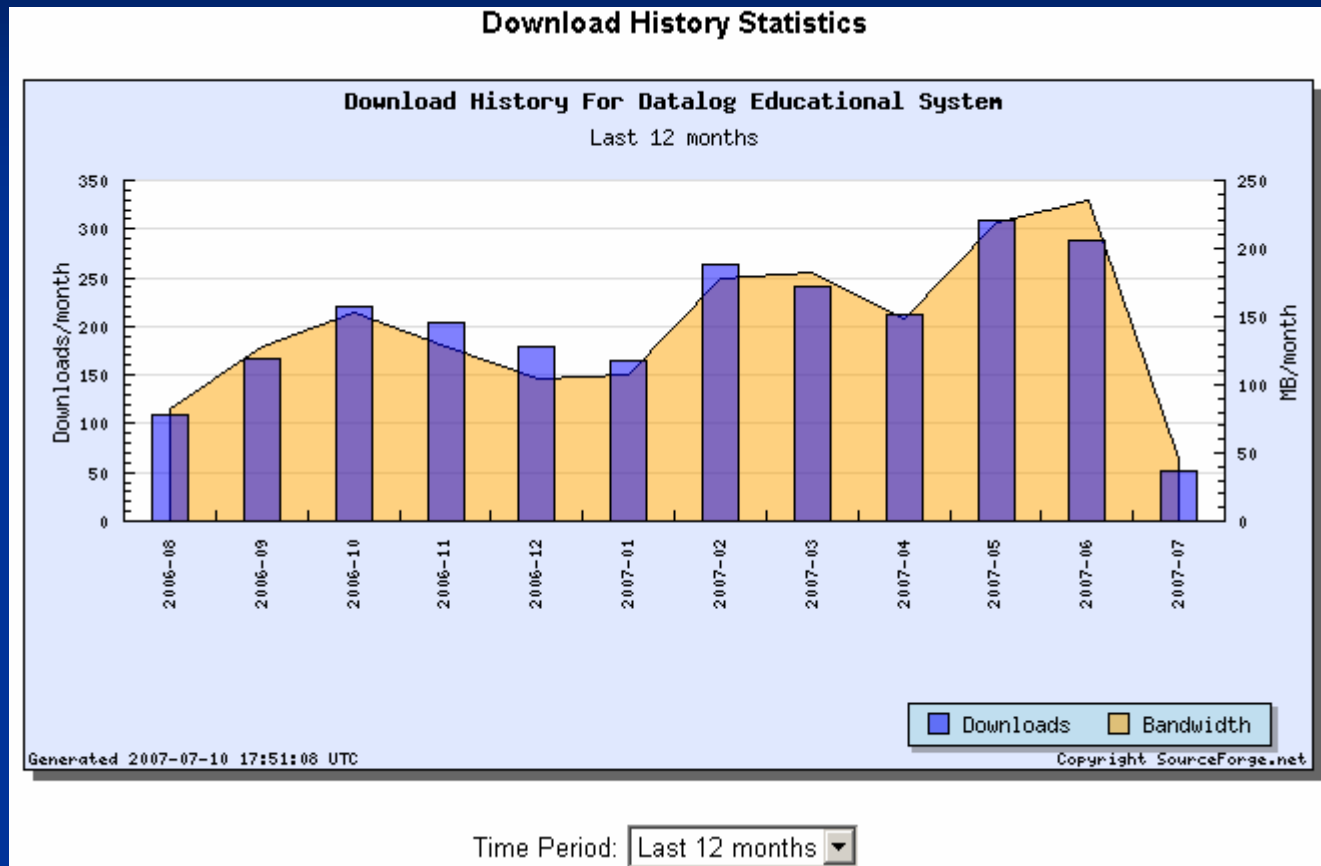
# Instalación

- Distribución GPL en Sourceforge:
  - Fuentes
  - Ejecutables (Windows, Linux, Solaris)
- Instalación e inicio
  - Intérprete Prolog (Ciao, GNU, Sicstus, SWI)
  - Ejecutable Sicstus

# 5. Implementación

- Núcleo de DES: Prolog
  - Cálculo de punto fijo con tablas de extensión
  - Saturación por estratos en presencia de negación
- Depurador: Prolog + Java
  - R. Caballero, Y. García-Ruiz and F. Sáenz-Pérez.  
*A new proposal for debugging datalog programs.*  
In 16th International Workshop on Functional and  
(Constraint) Logic Programming (WFLP'07), June.
- ACIDE: Java
  - Proyecto de Sistemas Informáticos 2006/2007

# 6. Utilidad del sistema



6.000 descargas desde su lanzamiento en 2004

## 6. Utilidad del sistema

- Producto de interés indicado por SIGMOD de la ACM
- Universidades:
  - EE.UU. (California, Búfalo, ...)
  - Canadá (Ottawa)
  - Europa (Alemania, Francia, Bélgica, ...)
- Entre los 10 sistemas Prolog más activos de Sourceforge
- “Mirrors”
- Usado también en investigación

# 7. Conclusiones y trabajo futuro

- Herramienta:
  - Útil
  - Sencilla
  - Aplicable a la enseñanza de BDD
- Trabajo futuro:
  - Aritmética
  - Agregados
  - Depurador gráfico

# 8. Enlaces

- Sistema DES:  
<http://des.sourceforge.net>
- ACIDE:  
<http://acide.sourceforge.net>  
<http://pide.berlios.de>
- Grupo de programación declarativa:  
<http://gpd.sip.ucm.es>