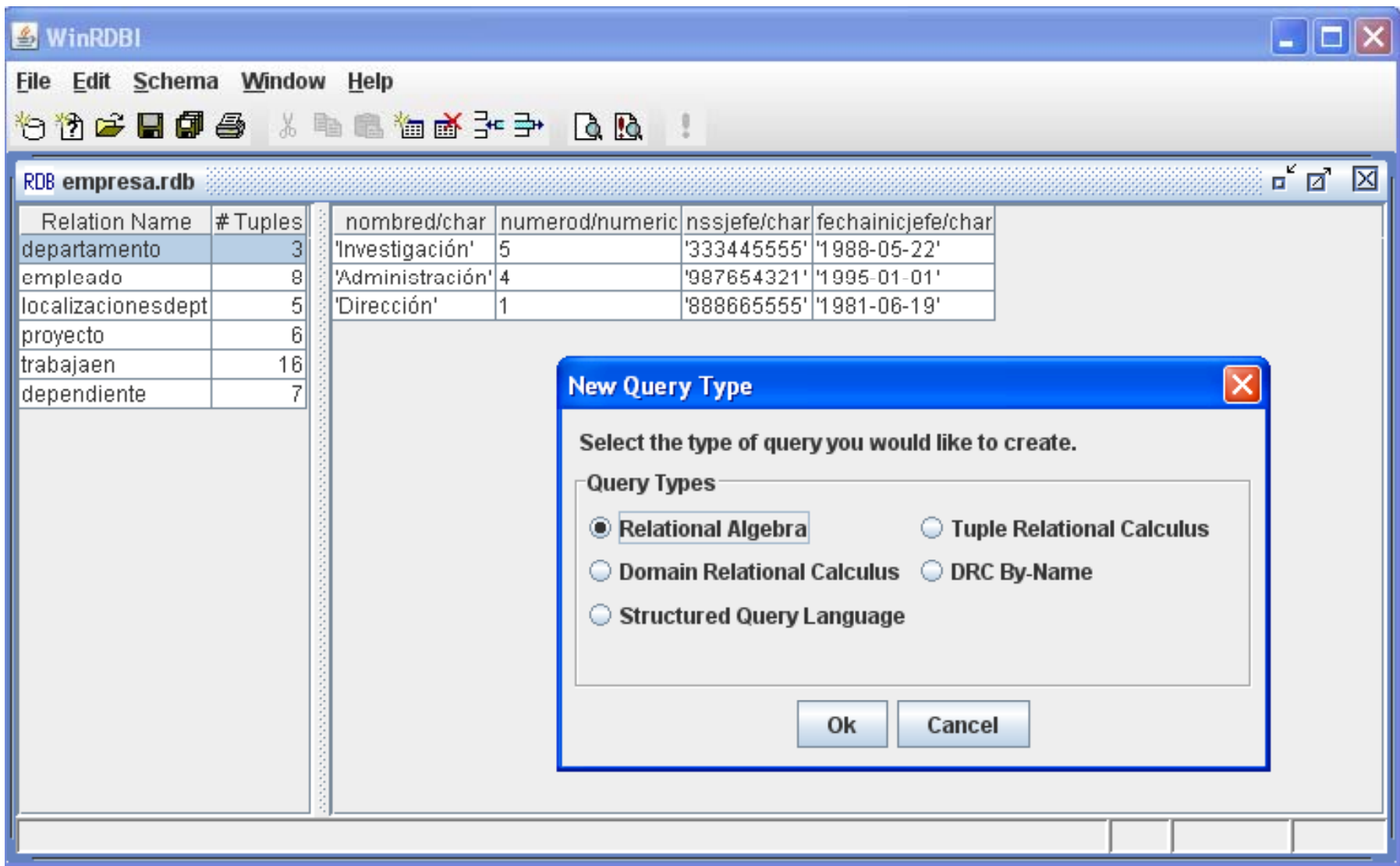


WinRDBI



ALG Untitled1.alg

```
rel1:=project nombred(departamento);
```

Relation Name	# Tuples
rel1	3

nombred/char
'Investigación'
'Administración'
'Dirección'

ALG Untitled1.alg

```
rel1:=select nombred='Investigación'(departamento);
```

Relation Name	# Tuples
rel1	1

nombred/char	numerod/numeric	nssjefe/char	fechainicjefe/char
'Investigación'	5	'333445555'	'1988-05-22'

Mod Line 1 Col 37

# Lenguajes en WinRDBI

# Gramáticas libres de contexto

[http://es.wikipedia.org/wiki/Gram%C3%A1tica\\_libre\\_de\\_contexto](http://es.wikipedia.org/wiki/Gram%C3%A1tica_libre_de_contexto)

- En lingüística e informática, una **gramática libre de contexto** (o **de contexto libre**) es una gramática formal en la que cada regla de producción es de la forma:

$$V \rightarrow w$$

Donde  $V$  es un símbolo no terminal y  $w$  es una cadena de terminales y/o no terminales.

- El término *libre de contexto* se refiere al hecho de que el no terminal  $V$  puede siempre ser sustituido por  $w$  sin tener en cuenta el contexto en el que ocurra

- Las gramáticas libres de contexto permiten describir la mayoría de los [lenguajes de programación](#), de hecho, la sintaxis de la mayoría de lenguajes de programación está definida mediante gramáticas libres de contexto
- Una notación popular en informática es [Backus–Naur Form](#), o *BNF*.

# Backus–Naur Form, o *BNF*.

- Una especificación de BNF es un sistema de reglas de derivación, escrito como:

**<simbolo> ::= <expresión con símbolos>**

- donde <simbolo> es un no terminal, y la expresión consiste en secuencias de símbolos o secuencias separadas por la barra vertical, '|', indicando una opción, el conjunto es una posible substitución para el símbolo a la izquierda.
- Los símbolos que nunca aparecen en un lado izquierdo son terminales. Por otro lado, símbolos que aparecen en la izquierda son no terminales y siempre van dentro del par<>.
- Los '::=' significan que el símbolo en la izquierda ha de ser reemplazado por la expresión a la derecha.

# Álgebra relacional en WinRDBI

- Con excepción de SQL, los lenguajes en WinRDBI son case sensitive.
- Los nombres de las relaciones y los atributos han de empezar con minúscula. Las constantes son strings (single quoted) numéricas.
- Los símbolos terminales van en **negrita**, no-terminales dentro de <>, patrones en cursiva y las llaves ({} ) indican elementos opcionales.



# Consultas en WinRDBI

```

<Query_Definition>      →  <Query> ;
                        |  <Assignment_Statement> ;

<Assignment_Statement>  →  Relation_Name := <Query>
                        |  Relation_Name ( <Attribute_List> ) := <Query>

<Attribute_List>        →  Attribute_Name , <Attribute_List>
                        |  Attribute_Name

```

# BNF álgebra relacional

<Query>	→	<Expression>
<Expression>	→	<i>Identifier</i>
		( <Expression> )
		<Select_Expr>
		<Project_Expr>
		<Binary_Expr>
<Select_Expr>	→	<b>select</b> <Condition> ( <Expression> )
<Project_Expr>	→	<b>project</b> <Attributes> ( <Expression> )
<Binary_Expr>	→	( <Expression> ) <Binary_Op> ( <Expression> )
<Condition>	→	<And_Condition>
		<And_Condition> or <Condition>

<And_Condition>	→	<Rel_Formula>
		<Rel_Formula> and <And_Condition>
<Rel_Formula>	→	<Operand> <Relational_Op> <Operand>
		( <Condition> )
<Binary_Op>	→	<b>union</b>
		<b>njoin</b>
		<b>product</b>
		<b>difference</b>
		<b>intersect</b>
<Attributes>	→	<i>Identifier</i>
		<i>Identifier</i> , <Attributes>
<Operand>	→	<i>Identifier</i>
		<i>Constant</i>
<Relational_Op>	→	=>
		>
		<
		<>
		>=

<b>Álgebra Relacional</b> (apuntes)	<b>WinRDBI</b>
$\sigma_p(R)$	Select p (R)
$\Pi_x(R)$	Project x (R)
$R1 \cup R2$	R1 union R2
$R1 - R2$	R1 difference R2
$R1 \times R2$	R1 product R2
$R_1 \underset{\text{condición}}{\theta} R_2$	no incorporado
$R1 * R2$	R1 njoin R2
$R1 \cap R2$	R1 intersect R2
$R1 : R2$	no incorporado

# Algunos comentarios

- The union, difference and intersect operators require that the operand relations are compatible, i.e., the operand relations have identical schemas.
- The (Cartesian) product operator requires that the operand relations have disjoint attribute names.
- The njoin (natural join) operator does **NOT** require common attribute names in the operand relations; if the operand relations have disjoint attribute names, then the njoin results in a Cartesian product

# Ejemplos

- `researchDept := select dname='Research' (department);`
- `alg1 := project fname, lname, address (employee);`
- `select dnumber=dno (department product employee);`

# Bibliografía

- <http://winrdbi.asu.edu/UserGuide/Content.html>
- [http://es.wikipedia.org/wiki/Gram%C3%A1tica libre de contexto](http://es.wikipedia.org/wiki/Gram%C3%A1tica_libre_de_contexto)