

Introducción al Uso de WinRDBI

versión 2.0.5

Introducción:

WinRDBI es una herramienta educativa, desarrollada por la Arizona State University, para aprender los lenguajes relacionales formales practicándolos. Para ello, funciona con un interfaz de usuario amigable en entorno Windows que posibilita la creación de bases de datos relacionales, la inserción de contenidos en dichas bases de datos, y la formulación de consultas en los siguientes lenguajes relacionales:

- Álgebra Relacional (AR),
- Cálculo Relacional de Tuplas (CRT),
- Cálculo Relacional de Dominios (CRD),
- SQL (versión simplificada de SQL-92).

La versión actual (2.0.5) está desarrollada en Java (es necesario haber instalado antes el Java Runtime Environment 1.3) y utiliza Amzi! Prolog para interpretar las consultas declarativas.

La página oficial de esta herramienta está en <http://www.eas.asu.edu/~cse412/winrdbi.html>.

Se recomienda a los alumnos de la asignatura Bases de Datos, de 3º curso, que utilicen esta herramienta para resolver los ejercicios propuestos en el tema 4 (Lenguajes Relacionales).

Resumen de la sintaxis:

WinRDBI utiliza una sintaxis para los lenguajes formales relacionales (AR, CRT y CRD) muy parecida a la explicada en el tema 4, pero los operadores matemáticos están cambiados por palabras clave y la escritura es lineal (no existen subíndices y superíndices en las expresiones).

A continuación presentamos un resumen de estas diferencias (leer la *User Guide* para información más detallada):

1) Consultas y asignaciones:

En cualquiera de los cuatro lenguajes permitidos, una consulta acaba un punto y coma “;”.

Las asignaciones tienen la forma:

<relación> := <consulta>
ejemplo: em := empleados;

Se pueden renombrar atributos mediante:

<relación>(<lista de atributos>) := <consulta>
ejemplo: em(dni,nom,edad) := empleados;

2) Operadores del Álgebra relacional:

En la tabla siguiente se resume la sintaxis de los operadores algebraicos:

Álgebra Relacional (apuntes)	WinRDBI
$\sigma_p(R)$	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

Tabla 1. Operadores del Álgebra Relacional

3) Sintaxis del Cálculo Relacional:

Los nombres de variables (tupla o de dominio) deben empezar por una letra mayúscula. Pueden utilizarse variables y constantes anónimas mediante un símbolo de subrayado “_”. Los operadores lógicos y cuantificadores se indican en la tabla siguiente.

Cálculo Relacional (apuntes)	WinRDBI
$F \vee G$	F or G
$F \wedge G$	F and G
$\neg F$	not (F)
$F \rightarrow G$	no incorporado
$\exists xF$	(exists X) (F)
$\forall xF$	(forall X) (F)

Tabla 2. Operadores y Cuantificadores del Cálculo Relacional

Ejemplos de consultas:

A continuación se muestran varios ejemplos de consultas, formuladas en cada uno de los cuatro lenguajes. Estas consultas están basadas en el siguiente esquema:

empleados (dni, apel1, apel2, nombre, fechanac, dirección, sexo, salario, supervisor, dep)
departamentos (dnumero, dnombre, jefe, jefefecha)
dep_ciudades (dep, dciudad)
proyectos (pnumero, pnombre, pciudad, dep)
trabaja_en (emp, pro, horas)
parientes (emp, par_nombre, sexo, fechanac, parentesco)

con las siguientes integridades referenciales:

empleados.supervisor → empleados	empleados.dep → departamentos
departamentos.jefe → empleados	
dep_ciudades.dep → departamentos	
proyectos.dep → departamentos	
trabaja_en.emp → empleados	trabaja_en.pro → proyectos
parientes.emp → empleados	

Consulta 1

Recuperar el nombre y dirección de todos los empleados que trabajan para el departamento de 'Investigacion' :

alg1 := project nombre, direccion (select dnumero=dep ((select dnombre='Investigacion' (de)) product em));

crt1 := {E.nombre, E.direccion | em(E) and
(exists D) (de(D) and D.dnombre='Investigacion' and D.dnumero=E.dep)};

crd1 := {Nombre, Direccion | (exists Dnumero)
(de(Dnumero,'Investigacion',_,_) and em(_,_,Nombre,_,Direccion,_,_,Dnumero))};

sql1 := select nombre, direccion
from empleados E, departamentos D
where D.dnombre = 'Investigacion' and E.dep = D.dnumero;

Consulta 2

Para cada proyecto localizado en 'Ciudad Real', listar el numero del proyecto, el numero del departamento que lo controla, y los apellidos, nombre y direccion del jefe del departamento:

% proyectos localizados en 'Ciudad Real'
v21 := project pnumero,dep (select pciudad='Ciudad Real' (pr));
% departamentos que los controlan
v22(pnumero, depart, jefe) := project pnumero, dep, jefe (select dep=dnumero (v21 product de));
% jefes de dichos departamentos
alg2 := project pnumero, depart, apel1, apel2, nombre, direccion (select jefe=dni (v22 product em));

crt2 := {P.pnumero, P.dep, E.apel1, E.apel2, E.nombre, E.direccion |
pr(P) and em(E) and P.pciudad='Ciudad Real' and
(exists D) (de(D) and P.dep=D.dnumero and D.jefe=E.dni)};

crd2 := {Pnumero, Dep, Apel1, Apel2, Nombre, Direccion | (exists Jefe)
(pr(Pnumero,_, 'Ciudad Real', Dep) and de(Dep,_, Jefe,_) and
em(Jefe, Apel1, Apel2, Nombre,_, Direccion,_,_,_))};

```
sql2 := select pnumero, P.dep, apel1, apel2, nombre, direccion
        from proyectos P, departamentos D, empleados E
        where pciudad='Ciudad Real' and P.dep=D.dnumero and D.jefe=E.dni;
```

Consulta 3

Encontrar los nombres de empleados que trabajan en todos los proyectos controlados por el departamento 5 :

```
% proyectos del departamento 5
v31(pn) := project pnumero (select dep=5 (pr));
% empleados que trabajan en algun proyecto
v32(dni, pn) := project emp, pro (te);
v33 := project dni (v32);
% todas las posibles combinaciones de empleados (que trabajan en proyectos) y
% de proyectos del departamento 5
v34 := v31 product v33;
% empleados que NO trabajan en TODOS los proyectos del departamento 5
v35 := project dni (v34 difference v32);
v36 := v33 difference v35;
alg3 := project apel1, apel2, nombre (v36 njoin em);

crt3 := {E.apel1, E.apel2, E.nombre |
        em(E) and (forall P) (not (pr(P)) or (not (P.dep=5) or (exists T)
        (te(T) and T.emp=E.dni and P.pnumero=T.pro)))));
        % F implica G = (not F) or G

crd3 := {Apel1, Apel2, Nombre | (exists Dni)
        ( em(Dni,Apel1,Apel2,Nombre,_,_,_,_,_) and not(exists Pnumero)
        ( pr(Pnumero,_,_,5) and not( te(Dni,Pnumero,_) ))));

sql3 := select apel1, apel2, nombre
        from empleados E
        where not exists ( select *
                          from proyectos P
                          where P.dep=5 and not exists ( select *
                                                          from trabaja_en T
                                                          where T.emp=E.dni and T.pro=P.pnumero));
```

En la página web de la asignatura se incluyen ficheros con estos ejemplos y otros más.