La perspectiva lógica del modelo relacional

Red Docente BDeIS

9 de marzo de 2007

Los lenguajes de especificación en el MR

Codd propone 3 lenguajes teóricos en su teoría del MR

- álgebra relacional
- cálculo relacional de tuplas
- cálculo relacional de dominios

¿para qué?

- base teórica
- declaración de los mínimos a cumplir por todo lenguaje relacional

¿y por qué sólo teóricos

- falta de operadores básicos
- algunos ejemplos “escandalosos”
  - no hay operadores aritméticos simples (sumas, restas, etc.)
  - no hay operadores de manipulación de cadenas de caracteres
  - ...

Objetivos

Conocer dos de los lenguajes propuestos por Codd para el modelo

- Cálculo Relacional de Tuplas
- Cálculo Relacional de Dominios

Pero, antes, necesitamos obtener una visión de una BD desde una perspectiva lógica con la que, mediante Fórmulas Bien Formadas, se podrá

- expresar restricciones de integridad
- interrogar a la base de datos.

el sistema de información: matriculación

ALUMNOS

Pepa Sonia Pepe Manolo

ASIGNATURAS

BD1 BD2

MATRÍCULAS

Pepa BD1
Sonia BD1
Pepe BD1
Manolo BD1

la pregunta: “Alumnos matriculados en BD1”

ALUMNOS
Pepa Sonia Pepe Manolo

ASIGNATURAS
BD1 BD2

MATRÍCULAS
Pepa BD1
Sonia BD1
Pepe BD1
Manolo BD2

YO: “Alumnos matriculados en BD1”

la pregunta: “Alumnos matriculados en BD1”

ALUMNOS
Pepa Sonia Pepe Manolo

ASIGNATURAS
BD1 BD2

MATRÍCULAS
Pepa BD1
Sonia BD1
Pepe BD1
Manolo BD2

“$mat(x, \text{'}BD1\text{'})$”

La necesidad

- Necesitamos un lenguaje formal, preciso, que nos permita consultar BB.DD. relacionales
  - entendible por todos
  - procesable por una máquina
  - eficaz

La necesidad

- Necesitamos un lenguaje formal, preciso, que nos permita consultar BB.DD. relacionales
  - entendible por todos
  - procesable por una máquina
  - eficaz
- Las matemáticas :-(siempre han sido y serán
  - entendibles por todos ???
  - procesables por una máquina
  - eficaces

La elección (de Codd)

- El cálculo de predicados de primer orden
  La Lógica de Primer Orden, es uno de los formalismos más utilizados para representar conocimiento.
- La lógica de predicados está basada en la idea de que las sentencias expresan relaciones entre objetos, así como también cualidades y atributos de tales objetos. Los objetos pueden ser personas, objetos físicos, o conceptos. Tales cualidades, relaciones o atributos, se denominan predicados. Los objetos se conocen como argumentos o términos del predicado.

resolvamos la pregunta: “Alumnos matriculados en BD1”

ALUMNOS
Pepa Sonia Pepe Manolo

ASIGNATURAS
BD1 BD2

MATRÍCULAS
Pepa BD1
Sonia BD1
Pepe BD1
Manolo BD2

- representemos MATRÍCULAS $MATRICULA(x, y)$
- preguntemos $MATRICULA(x, `BD1`)$
- resolvamos $MATRICULA(\Box, `BD1`)$
  - $\text{MATRICULA}(`\text{Pepa}`, `\text{BD1}`) = \text{sí}$
  - $\text{MATRICULA}(`\text{Sonia}`, `\text{BD1}`) = \text{sí}$
  - $\text{MATRICULA}(`\text{Pepe}`, `\text{BD1}`) = \text{sí}$
  - $\text{MATRICULA}(`\text{Manolo}`, `\text{BD1}`) = \text{no}$
  - $\text{MATRICULA}(`\text{BD2}`, `\text{BD1}`) = \text{no}$
  - $\text{resultado} = \{\text{Pepa, Sonia, Pepe}\}$

Una BDR desde el punto de vista del CPPO

ALUMNOS

Pepa Sonia Pepe Manolo

ASIGNATURAS

BD1 BD2

MATRÍCULAS

Pepa BD1
Sonia BD1
Pepe BD1
Manolo BD2

cada tabla va a ser un predicado

ALUMNOS()
ASIGNATURAS()
MATRICULA()

Una BDR desde el punto de vista del CPPO

ALUMNOS

Pepa Sonia Pepe Manolo

ASIGNATURAS

BD1 BD2

MATRÍCULAS

Pepa BD1
Sonia BD1
Pepe BD1
Manolo BD2

los términos (los "parámetros") serán las columnas

ALUMNOS(x)
ASIGNATURAS(y)
MATRICULA(x,y)

Una BDR desde el punto de vista del CPPO

ALUMNOS

Pepa Sonia Pepe Manolo

ASIGNATURAS

BD1 BD2

MATRÍCULAS

Pepa BD1
Sonia BD1
Pepe BD1
Manolo BD2

el predicado se evaluará a CIERTO si los valores sustituidos en los términos están almacenados en la tabla

MATRICULA('Pepa', 'BD1') = sí
MATRICULA('Manolo', 'BD1') = no
ALUMNOS('Sonia') = sí
ALUMNOS('bicicleta') = no
ASIGNATURAS('macramé') = no

Qué podemos hacer con CPPO

- consultar a la BD por la existencia de un valor concreto
- obtener un listado de valores que cumplen cierta condición
- representar propiedades BD que se deben cumplir en cualquier estado de BD: restricciones de integridad

las malas noticias

- tenemos que aprender un nuevo lenguaje
- deberemos ser capaces de expresar lo que queremos con él
- necesitaremos refrescar lo que aprendimos en Lógica

Para poder consultar esta BDR necesitamos:

ALUMNOS
dni: nnnnnnnna
nombre: string(100)
dirección: string(100)
provincia: string(25)

Para poder consultar esta BDR necesitamos:

ALUMNOS

dni: nnnnnnnna
nombre: string(100)
dirección: string(100)
provincia: string(25)

ALUMNOS(....)

- símbolos de predicado

Para poder consultar esta BDR necesitamos:

ALUMNOS

dni: nnnnnnnna
nombre: string(100)
dirección: string(100)
provincia: string(25)

las que necesitemos y declaremos

x: nnnnnnnna
y: nnnnnnnna
z: string(25)

- símbolos de predicado
- símbolos de variable

Para poder consultar esta BDR necesitamos:

ALUMNOS

dni: nnnnnnnna
nombre: string(100)
dirección: string(100)
provincia: string(25)

- símbolos de predicado
- símbolos de variable
- símbolos de constante

de nnnnnnnna ...

00000000A 00000001A 00000002A ...
24123123S ... 99999999Z

de string(25) ...

(cadena vacía) a aa aaa ... Asturias ...
sususú ... dxxyjp.c10b9)ñt0t0+ ...

de string(100) ...

(cadena vacía) a aa aaa ... Asturias ...
sususú ...
dxxyjp.c10b9)ñt0t0hhasklhjkffdjkl+ ...

Símbolos del lenguaje

comparación

$= \neq > \geq < \leq$

puntuación

$(\;) \; .$

conectivas lógicas

$\wedge \; \vee \; \neg \; \rightarrow$

cuantificadores

$\forall \; \exists$

Evaluación de las conectivas lógicas

$\wedge$	**CIERTO**	FALSO
CIERTO	**CIERTO**	FALSO
FALSO	FALSO	FALSO

$\rightarrow$	CIERTO	**FALSO**
CIERTO	CIERTO	**FALSO**
FALSO	CIERTO	CIERTO

$\vee$	CIERTO	**FALSO**
CIERTO	CIERTO	CIERTO
FALSO	CIERTO	**FALSO**

$\neg$	
CIERTO	FALSO
FALSO	CIERTO

Consultando la BD

¿Octavio es un alumno?

$\exists x \exists y \exists z ALUMNO(x, \text{`}Octavio\text{`}, y, z)$

respuesta

CIERTO

ALUMNOS

dni	nombre	dirección	provincia
21	Octavio	C/Tridente	Alicante
22	Segismundo	C/Tridente	Alicante
23	Rodolfo	C/Tridente	Alicante
24	Severo	C/Tridente	Alicante
25	Octavio	C/Vidente	Elche

Consultando la BD

DNI de los alumnos que se llaman octavio
$\exists y \exists z ALUMNO(x, `Octavio`, y, z)$

respuesta

dni
21
25

ALUMNOS

dni	nombre	dirección	provincia
21	Octavio	C/Tridente	Alicante
22	Segismundo	C/Tridente	Alicante
23	Rodolfo	C/Tridente	Alicante
24	Severo	C/Tridente	Alicante
25	Octavio	C/Vidente	Elche

Consultando la BD

Todos los alumnos que viven en la calle Tridente son de Alicante
$\forall x \forall y \exists z (ALUMNO(x, y, `C/Tridente`, z) \rightarrow z = `Alicante`)$

respuesta

CIERTO

ALUMNOS

dni	nombre	dirección	provincia
21	Octavio	C/Tridente	Alicante
22	Segismundo	C/Tridente	Alicante
23	Rodolfo	C/Tridente	Alicante
24	Severo	C/Tridente	Alicante
25	Octavio	C/Vidente	Elche

Primeras conclusiones

$$\exists x \exists y \exists z ALUMNO(x, Octavio, y, z) = CIERTO$$
$$\exists y \exists z ALUMNO(x, `Octavio`, y, z) = \{21, 25\}$$

- puedo obtener 2 tipos de respuesta: CIERTO o FALSO, o un listado

Primeras conclusiones

$$\exists x \exists y \exists z ALUMNO(x, Octavio, y, z) = CIERTO$$
$$\exists y \exists z ALUMNO(x, `Octavio`, y, z) = \{21, 25\}$$

- puedo obtener 2 tipos de respuesta: CIERTO o FALSO, o un listado
- el tipo de respuesta lo determina la existencia o no de variables no cuantificadas ($\forall$ $\exists$)
  - si alguna variable no está cuantificada, genera una columna de la "tabla" resultado
  - las variables cuantificadas sirven para establecer las condiciones que deben cumplir los valores resultado

Primeras conclusiones

$$\exists x \exists y \exists z ALUMNO(x, Octavio, y, z) = CIERTO$$
$$\exists y \exists z ALUMNO(x, `Octavio`, y, z) = \{21, 25\}$$

- puedo obtener 2 tipos de respuesta: CIERTO o FALSO, o un listado
- el tipo de respuesta lo determina la existencia o no de variables no cuantificadas ($\forall$ $\exists$)
  - si alguna variable no está cuantificada, genera una columna de la "tabla" resultado
  - las variables cuantificadas sirven para establecer las condiciones que deben cumplir los valores resultado
- las fórmulas no se escriben de cualquier manera

Primeras conclusiones

$$\exists x \exists y \exists z ALUMNO(x, Octavio, y, z) = CIERTO$$
$$\exists y \exists z ALUMNO(x, `Octavio`, y, z) = \{21, 25\}$$

- puedo obtener 2 tipos de respuesta: CIERTO o FALSO, o un listado
- el tipo de respuesta lo determina la existencia o no de variables no cuantificadas ($\forall$ $\exists$)
  - si alguna variable no está cuantificada, genera una columna de la "tabla" resultado
  - las variables cuantificadas sirven para establecer las condiciones que deben cumplir los valores resultado
- las fórmulas no se escriben de cualquier manera
- el resultado depende del estado de la BD, los símbolos del lenguaje no

Sobre variables y alcance de los cuantificadores

Estas dos expresiones son equivalentes

$$\forall x\ \forall y \forall z(ALUMNO(\mathbf{x}, y, z, `Alicante`) \rightarrow PARTICIPA(\mathbf{x}, `GH28`))$$

$$\forall x\ (\forall y \forall z ALUMNO(\mathbf{x}, y, z, `Alicante`) \rightarrow PARTICIPA(\mathbf{x}, `GH28`))$$

AMBAS devolverían CIERTO o FALSO
3 variables, todas cuantificadas

Sobre variables y alcance de los cuantificadores

Estas dos expresiones son equivalentes

$$\forall x \; \forall y \forall z ALUMNO(\mathbf{x}, y, z, \text{`}Alicante\text{`}) \rightarrow PARTICIPA(\mathbf{x}, \text{`}GH28\text{`})$$

$$\forall x \; \forall y \forall z ALUMNO(\mathbf{x}, y, z, \text{`}Alicante\text{`}) \rightarrow PARTICIPA(\mathbf{t}, \text{`}GH28\text{`})$$

AMBAS devolverían una columna
4 variables, una sin cuantificar

Sobre variables y alcance de los cuantificadores

Ocurrencias de variable **ligadas**

$\exists x \exists y \exists z(ALUMNO(x, y, z, `Alicante`) \wedge PARTICIPA(x, `GH28`))$

Todas aquellas bajo el alcance de un cuantificador

informalmente: variables ligadas

Ocurrencias de variable **libres**

$\exists x(ALUMNO(x, y, z, `Alicante`) \wedge PARTICIPA(x, `GH28`))$

No cuantificadas

informalmente: variables libres

Sobre variables y fórmulas

Fórmulas **cerradas**

$\exists x \exists y \exists z (ALUMNO(x, y, z, \text{`}Alicante\text{`}) \wedge PARTICIPA(x, \text{`}GH28\text{`}))$

Todas las variables son **ligadas**

Fórmulas **abiertas**

$\exists x (ALUMNO(x, y, z, \text{`}Alicante\text{`}) \wedge PARTICIPA(x, \text{`}GH28\text{`}))$

Al menos hay una variable libre

Un esquema de la BDR

ALUMNO(dni:nn,
nombre:string(100),
dirección:string(100),
ciudad:string(25),
CP: (dni))

ASIGNATURA(cod:aaaa,
nombre:string(75),
CP: (cod))

PROGRAMA(siglas:aaaa,
nombre:string(100),
cultura:{0|1|2},
CP: (siglas))

APRUEBA(dni:nn, asig:aaaa,
CP: (dni, asig)
CAj: (dni) → ALUMNO
CAj: (asig) → ASIGNATURA)

PARTICIPA(dni:nn, prog:aaaa,
CP: (dni, prog)
CAj: (dni) → ALUMNO
CAj: (prog) → PROGRAMA)

Un estado de BDR

ALUMNO

dni	nombre	dirección	ciudad
21	Octavio	Tridente	Alicante
22	Segismundo	Tridente	Alicante
24	Severo	Tridente	Alicante
25	Octavio	Vidente	Elche

APRUEBA

dni	asig
25	BD1

PARTICIPA

dni	prog
21	GH28
21	OT34
22	GH28
24	GH28
21	RR

ASIGNATURA

cod	nombre
BD1	BasesDatos1
BD2	BasesDatos2

PROGRAMA

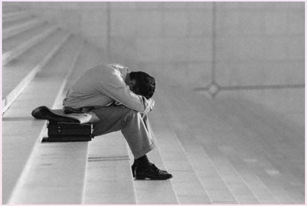
siglas	nombre	c
GH28	GranHermano28	0
OT34	OperaciónTriunfo34	0
RR	Redes	2

Ejercicios 1

1. nombre de los alumnos de Alicante
2. todos los alumnos de Alicante participan en GH28
3. nombre de los alumnos aprobados
4. DNI y nombre de los alumnos aprobados
5. nombre y ciudad de los alumnos participantes en GH28 u OT34
6. nombre de los programas en que ha participado el alumno de DNI 21

¿Cansado de ...

- escribir tantas variables?
- de tener que mirar el esquema constantemente para ver en qué orden están los datos en las tablas?



No problemo

Tenemos 2 lenguajes orientados a su uso en BB.DD. relacionales

- Cálculo relacional de tuplas
- Cálculo relacional de dominios

Facilitando la escritura

Vamos a redefinir la construcción de fórmulas

- Usaremos sólo las variables necesarias
- Permitiremos comparaciones fuera de los predicados

Facilitando la escritura

Vamos a redefinir la construcción de fórmulas

- Usaremos sólo las variables necesarias
- Permitiremos comparaciones fuera de los predicados
- Según el tipo de variable diferenciaremos entre 2 lenguajes

Facilitando la escritura

Vamos a redefinir la construcción de fórmulas

- Usaremos sólo las variables necesarias
- Permitiremos comparaciones fuera de los predicados
- Según el tipo de variable diferenciaremos entre 2 lenguajes
  - CR de dominios: variables escalares
    $x = \text{‘}2006\text{‘}$

$$ALUMNO(dni : x, ciudad : \text{‘}Alicante\text{‘})$$

Facilitando la escritura

Vamos a redefinir la construcción de fórmulas

- Usaremos sólo las variables necesarias
- Permitiremos comparaciones fuera de los predicados
- Según el tipo de variable diferenciaremos entre 2 lenguajes
  - CR de dominios: variables escalares
    $x = \text{‘}2006\text{‘}$

$$ALUMNO(dni : x, ciudad : \text{‘}Alicante\text{‘})$$

  - CR de tuplas: variables tupla
    $x.fecha = \text{‘}2006\text{‘}$

$$ALUMNO(x) \wedge x.ciudad = \text{‘}Alicante\text{‘}$$

Cálculo Relacional de Dominios

Cálculo Relacional de Dominios

- variables dominio

Cálculo Relacional de Dominios

Cálculo Relacional de Dominios

- variables dominio
- construcción de fórmulas

Cálculo Relacional de Dominios

Cálculo Relacional de Dominios

- variables dominio
- construcción de fórmulas
  - acceso a tablas: ALUMNO(dni: x)

Cálculo Relacional de Dominios

Cálculo Relacional de Dominios

- variables dominio
- construcción de fórmulas
  - acceso a tablas: ALUMNO(dni: x)
  - comparaciones: x = ‘21‘

Cálculo Relacional de Dominios

Cálculo Relacional de Dominios

- variables dominio
- construcción de fórmulas
  - acceso a tablas: ALUMNO(dni: x)
  - comparaciones: x = '21'
  - restricciones de integridad: se evalúan a *CIERTO* o *FALSO* expresión: *F*

Cálculo Relacional de Dominios

Cálculo Relacional de Dominios

- variables dominio
- construcción de fórmulas
  - acceso a tablas: ALUMNO(dni: x)
  - comparaciones: x = ‘21‘
  - restricciones de integridad: se evalúan a *CIERTO* o *FALSO*
    expresión: F
  - consultas: devuelven una relación
    expresión: $\{variablesResultado|F\}$

Ejemplos

Restricción de integridad: todas las variables cuantificadas (o constantes)

2. todos los alumnos de Alicante participan en GH28

$$\forall x(ALUMNO(dni : x, ciudad : \text{‘}Alicante\text{‘}) \\ \rightarrow PARTICIPA(dni : x, prog : \text{‘}GH28\text{‘}))$$

$$\forall x(ALUMNO(ciudad : \text{‘}Alicante\text{‘}, dni : x) \\ \rightarrow PARTICIPA(prog : \text{‘}GH28\text{‘}, dni : x))$$

Ejemplos

Consulta: algunas variables no están cuantificadas

5. nombre y ciudad de los alumnos participantes en GH28 u OT34

$$\{y, t | \exists x(ALUMNO(dni : x, nombre : \mathbf{y}, ciudad : \mathbf{t}) \\ \wedge \exists z(PARTICIPA(dni : x, prog : z) \\ \wedge (z = `GH28` \vee z = `OT34`)))\}$$

5. nombre y ciudad de los alumnos participantes en GH28 u OT34 (alternativa)

$$\{y, t | \exists x(ALUMNO(dni : x, nombre : \mathbf{y}, ciudad : \mathbf{t}) \\ \wedge (PARTICIPA(dni : x, prog : `GH28`) \\ \vee PARTICIPA(dni : x, prog : `OT34`)))\}$$

Ejercicios 1

1. nombre de los alumnos de Alicante
2. todos los alumnos de Alicante participan en GH28 (resuelto)
3. nombre de los alumnos aprobados
4. DNI y nombre de los alumnos aprobados
5. nombre y ciudad de los alumnos participantes en GH28 u OT34 (resuelto)
6. nombre de los programas en los que ha participado el alumno de DNI 21

Ejercicios 2

1. si participas en algún programa, no apruebas
2. nombre y dni de aquellos alumnos que han participado en todos los programas
3. nombre y dni de aquellos alumnos que no han participado en ningún programa
4. DNI y nombre de los alumnos aprobados que no han participado en ningún programa
5. ciudades de los alumnos que no han participado en ningún programa
6. participar en programas con un índice cultural 2 asegura el aprobado en BD1

Cálculo Relacional de Tuplas

Cálculo Relacional de Tuplas

- variables tupla

Cálculo Relacional de Tuplas

Cálculo Relacional de Tuplas

- variables tupla
- construcción de fórmulas

Cálculo Relacional de Tuplas

Cálculo Relacional de Tuplas

- variables tupla
- construcción de fórmulas
  - acceso a tablas: ALUMNO(x)

Cálculo Relacional de Tuplas

Cálculo Relacional de Tuplas

- variables tupla
- construcción de fórmulas
  - acceso a tablas: ALUMNO(x)
  - comparaciones: x.DNI = '21'

Cálculo Relacional de Tuplas

Cálculo Relacional de Tuplas

- variables tupla
- construcción de fórmulas
  - acceso a tablas: ALUMNO(x)
  - comparaciones: x.DNI = '21'
  - restricciones de integridad: se evalúan a *CIERTO* o *FALSO*
    expresión: *F*

Cálculo Relacional de Tuplas

Cálculo Relacional de Tuplas

- variables tupla
- construcción de fórmulas
  - acceso a tablas: ALUMNO(x)
  - comparaciones: x.DNI = '21'
  - restricciones de integridad: se evalúan a *CIERTO* o *FALSO*
    expresión: F
  - consultas: devuelven una relación
    expresión: $\{variablesResultado|F\}$

Ejemplos

Restricción de integridad: todas las variables cuantificadas (o constantes)

2. todos los alumnos de Alicante participan en GH28

$$\forall x(ALUMNO(x) \wedge x.ciudad = \text{`}Alicante\text{`} \\ \rightarrow \exists y(PARTICIPA(y) \\ \wedge x.dni = y.dni \wedge y.prog = \text{`}GH28\text{`}))$$

Ejemplos

Consulta: algunas variables no están cuantificadas

5. nombre y ciudad de los alumnos participantes en GH28 u OT34

$$\{x.nombre, x.ciudad | ALUMNO(x)$$
$$\wedge \exists y(PARTICIPA(y)$$
$$\wedge\ y.dni = x.dni \wedge (y.prog = \text{`}GH28\text{`} \vee y.prog = \text{`}OT34\text{`}))\}$$

Ejercicios 3

1. nombre de los alumnos de Alicante
2. todos los alumnos de Alicante participan en GH28 (resuelto)
3. nombre de los alumnos aprobados
4. DNI y nombre de los alumnos aprobados
5. nombre y ciudad de los alumnos participantes en GH28 u OT34 (resuelto)
6. nombre de los programas en los que ha participado el alumno de DNI 21

Ejercicios 4

1. si participas en algún programa, no apruebas
2. nombre y dni de aquellos alumnos que han participado en todos los programas
3. nombre y dni de aquellos alumnos que no han participado en ningún programa
4. DNI y nombre de los alumnos aprobados que no han participado en ningún programa
5. ciudades de los alumnos que no han participado en ningún programa
6. participar en programas con un índice cultural 2 asegura el aprobado en BD1

Fórmulas seguras

- El CPPO es general, su aplicación NO: Bases de Datos Relacionales
- ciertas "preguntas" no tienen sentido en una BD

dni de los alumnos desconocidos

$\neg ALUMNO(dni : x)$

¿Motivo de despido?

00000000A 00000001A 00000002A 00000003A 00000004A 00000005A
00000006A 00000007A 00000008A 00000009A 00000010A 00000011A
00000012A 00000013A 00000014A 00000015A 00000016A 00000017A
00000018A 00000019A 00000020A 00000021A 00000022A 00000023A
00000024A 00000025A 00000026A 00000027A 00000028A 00000029A
00000030A 00000031A 00000032A 00000033A 00000034A 00000035A
00000036A 00000037A 00000038A 00000039A 00000040A 00000041A
00000042A 00000043A 00000044A 00000045A 00000046A 00000047A
00000048A 00000049A 00000050A 00000051A 00000052A 00000053A
00000054A 00000055A 00000056A 00000057A 00000058A 00000059A
00000060A 00000061A 00000062A 00000063A 00000064A 00000065A
00000066A 00000067A 00000068A 00000069A 00000070A 00000071A
00000072A 00000073A 00000074A 00000075A 00000076A 00000077A
00000078A 00000079A 00000080A 00000081A 00000082A 00000083A
00000084A 00000085A 00000086A 00000087A 00000088A 00000089A
00000090A 00000091A 00000092A 00000093A 00000094A 00000095A

Variables ligadas por ∃

ASIGNATURA	
cod	**nombre**
BD1	BasesDatos1
BD2	BasesDatos2

APRUEBA	
dni	**asig**
25	BD1

BasesDatos2 tiene al menos un aprobado

$\exists x(ASIGNATURA(cod : x, nombre : `BasesDatos2`)$
$\rightarrow APRUEBA(asig : x))$

respuesta esperada

FALSO

Variables ligadas por ∃

ASIGNATURA

cod	nombre
BD1	BasesDatos1
BD2	BasesDatos2

APRUEBA

dni	asig
25	BD1

BasesDatos2 tiene al menos un aprobado

$\exists x(ASIGNATURA(cod: x, nombre: \text{‘}BasesDatos2\text{‘})$
$\rightarrow APRUEBA(asig: x))$

supongamos x=‘PUF‘

$$ASIGNATURA(cod: \text{‘}PUF\text{‘}, nombre: \text{‘}BasesDatos2\text{‘}) = FALSO$$

$$FALSO \rightarrow ? = CIERTO$$

Hemos encontrado un valor para el que la fórmula se hace cierta

Variables ligadas por ∃

ASIGNATURA	
cod	**nombre**
BD1	BasesDatos1
BD2	BasesDatos2

APRUEBA	
dni	**asig**
25	BD1

BasesDatos2 tiene al menos un aprobado

$\exists x(ASIGNATURA(cod : x, nombre : `BasesDatos2`)$
$\rightarrow APRUEBA(asig : x))$

respuesta obtenida

CIERTO

Variables ligadas por $\exists$

ASIGNATURA	
cod	**nombre**
BD1	BasesDatos1
BD2	BasesDatos2

APRUEBA	
dni	**asig**
25	BD1

BasesDatos2 tiene al menos un aprobado

$\exists x(ASIGNATURA(cod : x, nombre : `BasesDatos2`)$
$\rightarrow APRUEBA(asig : x))$

lo que pretendíamos escribir

$\exists x(ASIGNATURA(cod : x, nombre : `BasesDatos2`)$
$\wedge APRUEBA(asig : x))$

Variables ligadas por $\forall$

ASIGNATURA

cod	**nombre**
BD1	BasesDatos1
BD2	BasesDatos2

APRUEBA

dni	**asig**
25	BD1

La única asignatura con aprobados es BD1

$$\forall x(APRUEBA(asig : x) \wedge x = `BD1`))$$

respuesta esperada

CIERTO

Variables ligadas por $\forall$

ASIGNATURA	
cod	**nombre**
BD1	BasesDatos1
BD2	BasesDatos2

APRUEBA	
dni	**asig**
25	BD1

La única asignatura con aprobados es BD1

$\forall x(APRUEBA(asig : x) \wedge x = \text{`}BD1\text{`})$

supongamos x=‘BICICLETA‘

$$APRUEBA(asig : \text{`}BICICLETA\text{`}) = FALSO$$

$$FALSO \wedge ? = FALSO$$

Hemos encontrado un valor para el que la fórmula se hace falsa (no es cierta “para todo x”)

Variables ligadas por $\forall$

ASIGNATURA

cod	nombre
BD1	BasesDatos1
BD2	BasesDatos2

APRUEBA

dni	asig
25	BD1

La única asignatura con aprobados es BD1

$\forall x(APRUEBA(asig : x) \wedge x = \text{‘}BD1\text{‘}))$

respuesta obtenida

FALSO

Variables ligadas por $\forall$

ASIGNATURA

cod	**nombre**
BD1	BasesDatos1
BD2	BasesDatos2

APRUEBA

dni	**asig**
25	BD1

La única asignatura con aprobados es BD1

$$\forall x(APRUEBA(asig : x) \wedge x = \text{‘}BD1\text{‘}))$$

lo que pretendíamos escribir

$$\forall x(APRUEBA(asig : x) \rightarrow x = \text{‘}BD1\text{‘}))$$

Casos

No deseamos

- consultas que pueden devolver, potencialmente, infinitos valores
- errores en la expresión (lo que queremos obtener no se consigue con la fórmula propuesta)

Por tanto

- debemos asegurarnos de preguntar siempre por valores que sí están almacenados en la BD
- Los valores "desconocidos" no deben alterar el resultado esperado de la consulta
- Necesitamos una prueba sencilla para saber si la fórmula tiene aplicación en bases de datos relacionales

Por qué una implicación con $\forall$ pero no con $\exists$

...	...	
21	BC6	$\mathit{falso} \rightarrow ? = \mathit{cierto}$
22	BC7	$\mathit{falso} \rightarrow ? = \mathit{cierto}$
23	BC8	$\mathit{falso} \rightarrow ? = \mathit{cierto}$
24	BC9	$\mathit{falso} \rightarrow ? = \mathit{cierto}$
25	**BD1**	$\mathit{CIERTO} \rightarrow ? = ?$
26	BD2	$\mathit{falso} \rightarrow ? = \mathit{cierto}$
27	BD3	$\mathit{falso} \rightarrow ? = \mathit{cierto}$
...	...	

La única asignatura con aprobados es BD1

$$\forall x(APRUEBA(asig : x) \rightarrow x = \text{`}BD1\text{'})$$

Por qué una implicación con $\forall$ pero no con $\exists$

...	...	
21	BC6	$falso \rightarrow ? = cierto$
22	BC7	$falso \rightarrow ? = cierto$
23	BC8	$falso \rightarrow ? = cierto$
24	BC9	$falso \rightarrow ? = cierto$
25	**BD1**	$CIERTO \rightarrow ? = ?$
26	BD2	$falso \rightarrow ? = cierto$
27	BD3	$falso \rightarrow ? = cierto$
...	...	

La única asignatura con aprobados es BD1

$$\forall x(APRUEBA(asig : x) \rightarrow x = \text{`}BD1\text{`})$$

la fórmula sólo comprueba los datos que **sí** están en APRUEBA

Por qué una implicación con ∀ pero no con ∃

...	...	
21	BC6	$falso \wedge ? = falso$
22	BC7	$falso \wedge ? = falso$
23	BC8	$falso \wedge ? = falso$
24	BC9	$falso \wedge ? = falso$
25	**BD1**	$CIERTO \wedge ? = ?$
26	BD2	$falso \wedge ? = falso$
27	BD3	$falso \wedge ? = falso$
...	...	

Hay al menos un aprobado en BD1

$$\exists x (APRUEBA(asig : x) \wedge x = \text{`}BD1\text{`})$$

Por qué una implicación con $\forall$ pero no con $\exists$

...	...	
21	BC6	$falso \wedge ? = falso$
22	BC7	$falso \wedge ? = falso$
23	BC8	$falso \wedge ? = falso$
24	BC9	$falso \wedge ? = falso$
25	**BD1**	$CIERTO \wedge ? = ?$
26	BD2	$falso \wedge ? = falso$
27	BD3	$falso \wedge ? = falso$
...	...	

Hay al menos un aprobado en BD1

$$\exists x(APRUEBA(asig : x) \wedge x = \text{‘}BD1\text{‘})$$

la fórmula sólo comprueba los datos que **sí** están en APRUEBA

Prueba de las fórmulas seguras

Para toda v, variable en la fórmula

- Determinar la subfórmula afectada = G
  - Si v es libre, G=toda la fórmula
  - Si $\exists v$ o $\forall v$, G=alcance del cuantificador
- Analizar v
  - Si v es libre, un valor de v desconocido **no debe hacer cierta G**
  - Si v está bajo el alcance de un cuantificador existencial ($\exists$), un valor de v desconocido **no debe hacer cierta G**
  - Si v está bajo el alcance de un cuantificador universal ($\forall$), un valor de v desconocido **no debe hacer falsa G**

Prueba de las fórmulas seguras

dni de los alumnos desconocidos

$\neg ALUMNO(dni : x)$

Prueba de las fórmulas seguras

dni de los alumnos desconocidos

$\neg ALUMNO(dni : x)$

1. variable x: libre

Prueba de las fórmulas seguras

dni de los alumnos desconocidos

$\neg ALUMNO(dni : x)$

1. variable x: libre
2. subfórmula G = toda la fórmula

Prueba de las fórmulas seguras

dni de los alumnos desconocidos

$\neg ALUMNO(dni : x)$

1. variable x: libre
2. subfórmula G = toda la fórmula
3. hipótesis: un $x \notin Dom(G)$ debe hacer falsa G

Prueba de las fórmulas seguras

dni de los alumnos desconocidos

$\neg ALUMNO(dni : x)$

1. variable x: libre
2. subfórmula G = toda la fórmula
3. hipótesis: un $x \notin Dom(G)$ debe hacer falsa G
4. demostración: si x desconocido entonces
   $ALUMNO(dni : x) = FALSO$
   pero $\neg FALSO = CIERTO$

Prueba de las fórmulas seguras

dni de los alumnos desconocidos

$\neg ALUMNO(dni : x)$

1. variable x: libre
2. subfórmula G = toda la fórmula
3. hipótesis: un $x \notin Dom(G)$ debe hacer falsa G
4. demostración: si x desconocido entonces
   $ALUMNO(dni : x) = FALSO$
   pero $\neg FALSO = CIERTO$
5. conclusión: la formula **NO es segura**
   Valores **NO** almacenados en la tabla ALUMNO saldrían en el resultado

Prueba de las fórmulas seguras

BasesDatos2 tiene al menos un aprobado

$\exists x(ASIGNATURA(cod : x, nombre : \text{`}BasesDatos2\text{`})$
$\rightarrow APRUEBA(asig : x))$

Prueba de las fórmulas seguras

BasesDatos2 tiene al menos un aprobado

$\exists x(ASIGNATURA(cod : x, nombre : \text{‘}BasesDatos2\text{‘})$
$\rightarrow APRUEBA(asig : x))$

1. variable x: ligada por un $\exists$

Prueba de las fórmulas seguras

BasesDatos2 tiene al menos un aprobado

$\exists x(ASIGNATURA(cod : x, nombre : `BasesDatos2`)$
$\rightarrow APRUEBA(asig : x))$

1. variable x: ligada por un $\exists$
2. subfórmula $G = ASIGNATURA(cod : x, nombre : `BasesDatos2`)$
   $\rightarrow APRUEBA(asig : x)$

Prueba de las fórmulas seguras

BasesDatos2 tiene al menos un aprobado

$\exists x(ASIGNATURA(cod : x, nombre : \text{‘}BasesDatos2\text{‘})$
$\rightarrow APRUEBA(asig : x))$

1. variable x: ligada por un $\exists$
2. subfórmula $G = ASIGNATURA(cod : x, nombre : \text{‘}BasesDatos2\text{‘})$ $\rightarrow APRUEBA(asig : x)$
3. hipótesis: un $x \notin Dom(G)$ debe hacer falsa G

Prueba de las fórmulas seguras

BasesDatos2 tiene al menos un aprobado

$\exists x(ASIGNATURA(cod : x, nombre : `BasesDatos2`)$
$\rightarrow APRUEBA(asig : x))$

1. variable x: ligada por un $\exists$
2. subfórmula $G = ASIGNATURA(cod : x, nombre : `BasesDatos2`)$ $\rightarrow APRUEBA(asig : x)$
3. hipótesis: un $x \notin Dom(G)$ debe hacer falsa G
4. demostración: si $x \notin Dom(G)$ entonces $ASIGNATURA(cod : x, nombre : `BasesDatos2`) = FALSO$

Prueba de las fórmulas seguras

BasesDatos2 tiene al menos un aprobado

$\exists x(ASIGNATURA(cod : x, nombre : \text{`}BasesDatos2\text{`})$
$\rightarrow APRUEBA(asig : x))$

1. variable x: ligada por un $\exists$
2. subfórmula $G = ASIGNATURA(cod : x, nombre : \text{`}BasesDatos2\text{`})$
$\rightarrow APRUEBA(asig : x)$
3. hipótesis: un $x \notin Dom(G)$ debe hacer falsa G
4. demostración: si $x \notin Dom(G)$ entonces
$ASIGNATURA(cod : x, nombre : \text{`}BasesDatos2\text{`}) = FALSO$
pero $FALSO \rightarrow ? = CIERTO$

Prueba de las fórmulas seguras

BasesDatos2 tiene al menos un aprobado

$\exists x(ASIGNATURA(cod : x, nombre : `BasesDatos2`)$
$\rightarrow APRUEBA(asig : x))$

1. variable x: ligada por un $\exists$
2. subfórmula $G = ASIGNATURA(cod : x, nombre : `BasesDatos2`)$ $\rightarrow APRUEBA(asig : x)$
3. hipótesis: un $x \notin Dom(G)$ debe hacer falsa G
4. demostración: si $x \notin Dom(G)$ entonces $ASIGNATURA(cod : x, nombre : `BasesDatos2`) = FALSO$ pero $FALSO \rightarrow ? = CIERTO$
5. conclusión: la formula **NO es segura**
   Valores **NO** almacenados en la tabla ASIGNATURA hacen CIERTA la fórmula

Prueba de las fórmulas seguras

La única asignatura con aprobados es BD1

$$\forall x(APRUEBA(asig : x) \wedge x = `BD1`)$$

Prueba de las fórmulas seguras

La única asignatura con aprobados es BD1

$\forall x(APRUEBA(asig : x) \land x = `BD1`))$

1. variable x: ligada por un $\forall$

Prueba de las fórmulas seguras

La única asignatura con aprobados es BD1

$\forall x(APRUEBA(asig : x) \wedge x = `BD1`))$

1. variable x: ligada por un $\forall$
2. subfórmula $G = APRUEBA(asig : x) \wedge x = `BD1`$

Prueba de las fórmulas seguras

La única asignatura con aprobados es BD1

$\forall x(APRUEBA(asig : x) \land x = \text{‘}BD1\text{‘}))$

1. variable x: ligada por un $\forall$
2. subfórmula $G = APRUEBA(asig : x) \land x = \text{‘}BD1\text{‘}$
3. hipótesis: un $x \notin Dom(G)$ debe hacer cierta G

Prueba de las fórmulas seguras

La única asignatura con aprobados es BD1

$\forall x(APRUEBA(asig : x) \wedge x = `BD1`))$

1. variable x: ligada por un $\forall$
2. subfórmula $G = APRUEBA(asig : x) \wedge x = `BD1`$
3. hipótesis: un $x \notin Dom(G)$ debe hacer cierta G
4. demostración: si $x \notin Dom(G)$ entonces $APRUEBA(asig : x) = FALSO$

Prueba de las fórmulas seguras

La única asignatura con aprobados es BD1

$\forall x(APRUEBA(asig: x) \wedge x = `BD1`))$

1. variable x: ligada por un $\forall$
2. subfórmula $G = APRUEBA(asig: x) \wedge x = `BD1`$
3. hipótesis: un $x \notin Dom(G)$ debe hacer cierta G
4. demostración: si $x \notin Dom(G)$ entonces $APRUEBA(asig: x) = FALSO$ pero $FALSO \wedge ? = FALSO$

Prueba de las fórmulas seguras

La única asignatura con aprobados es BD1

$\forall x(APRUEBA(asig : x) \wedge x = \text{`}BD1\text{`})$

1. variable x: ligada por un $\forall$
2. subfórmula $G = APRUEBA(asig : x) \wedge x = \text{`}BD1\text{`}$
3. hipótesis: un $x \notin Dom(G)$ debe hacer cierta G
4. demostración: si $x \notin Dom(G)$ entonces
   $APRUEBA(asig : x) = FALSO$
   pero $FALSO \wedge ? = FALSO$
5. conclusión: la formula **NO es segura**
   Valores **NO** almacenados en la tabla APRUEBA hacen FALSA la fórmula

Quiero consultar una BDR con cálculo relacional

En el Modelo Relacional se puede usar el cálculo de predicados de primer orden (CPPO) porque una BDR siempre puede verse como una **interpretación** (I) de un **lenguaje de primer orden** (LPO).

El esquema
: de la BD induce un **lenguaje** con símbolos de constante, de predicado y variables, además de los símbolos usuales de conectivas lógicas, puntuación, etc. El lenguaje no cambia con el tiempo.

El estado
: de la BD es una **interpretación** de ese lenguaje: los valores de cada símbolo de constante y de cada símbolo de predicado. Cada estado de BD es una interpretación distinta, cambia con el tiempo.

Además

- si defino un conjunto de fórmulas como restricciones de integridad (desde integridad referencial hasta un "alumno debe matricularse de al menos dos asignaturas") todas las interpretaciones han de ser **modelo** para ese conjunto de fórmulas (que todas se evalúen a CIERTO en la interpretación)
- y puesto que estamos trabajando con BB.DD. no podemos plantear cualquier fórmula, sólo podemos utilizar **fórmulas seguras**.

Los apuntes

El cálculo relacional como otro lenguaje para el modelo

Los apuntes

- La base teórica en la que se basa la definición de los cálculos relacionales.
- Cuáles son los símbolos del lenguaje
- Cómo escribir fórmulas

Los apuntes

- Otra forma de describir una BDR
- El lenguaje lo define el esquema
- La interpretación la define el estado

Los apuntes

- El MR sólo admite ciertas expresiones
- Que las preguntas a la BD sólo trabajen con los datos almacenados

Los apuntes

- La adaptación de todo lo anterior al tipo de variable a utilizar: la tupla o el dominio
- Se modifica la forma de escribir fórmulas

Los ideas

- Una BDR se puede describir con CPPO
  - Relaciones $\longrightarrow$ Predicados
  - Valores $\longrightarrow$ Constantes
  - Consultas $\longrightarrow$ Fórmulas abiertas
  - Restricciones de integridad $\longrightarrow$ Fórmulas cerradas

Los ideas

- Una Fórmula Bien Formada es una fórmula "bien escrita"
  - Los términos son símbolos de variable, símbolos de constante, y símbolos de función (que no vamos a utilizar)
  - Los átomos son símbolos de predicado cuyos argumentos son términos
  - Las FBF son combinaciones de átomos y términos siguiendo unas reglas estrictas
    - $\forall x \forall y F$ sí, $\forall xyF$ **no**, $\forall x, yF$ **no**, $x, y \forall F$ **no**,$(\forall x)F$ **no**
    - $P(x, y)$ sí, $P(x, Q(y))$ **no**

Los ideas

- El cálculo relacional introduce
  - en tuplas, nuevos términos: $x.dni$ ...
  - en tuplas, nuevos átomos para comparación: $x.dni = \text{`21`}$ $x.sueldo < y.sueldo$ ...
  - en dominios, modificaciones a los átomos: $P(dni : x)$ ...
  - en dominios, nuevos átomos para comparación: $x = \text{`21`}$ $x < y$ ...

Los ideas

- La descripción de una BDR se hace
  1. Definiendo el lenguaje L(A,F) a partir del esquema
     - A es el alfabeto (símbolos)
     - F las reglas para escribir FBF's (términos → átomos → FBF)
  2. Definiendo la interpretación I(D,K,H,E)
     - D, universo de discurso, todos los valores posibles (de cualquier dominio)
     - K, dar valor a los símbolos de constante
     - E, asignar el contenido de las tablas a los símbolos de predicado
     - H, no se utiliza en MR

Los ideas

- El concepto de modelo es especialmente importante en el MR
  - Sea un conjunto T de fórmulas que expresan restricciones de integridad
    - referencial
    - existencia
    - correspondencia entre clases
    - ...
  - Que una I(D,K,H,E) sea modelo para T significa que la BD es correcta, cumple con todas las restricciones
    - si no es modelo es que alguna de las fórmulas cerradas de T es falsa en esa interpretación

Concluyendo

- Podemos consultar una BDR con CPPO porque ésta siempre se puede describir (representar) como una interpretación de un LPO
- Para cualquier BDR y para cualquier estado de ésta siempre podremos decir que
  - Se define un símbolo de constante para cada valor de cada dominio
  - Se define un símbolo de predicado para cada relación en el esquema
  - El Universo de Discurso es la unión de todos los dominios en el esquema
  - A cada símbolo de constante se le asigna su correspondiente valor del Universo de Discurso
  - A cada símbolo de predicado se le asigna la extensión de su correspondiente relación en el esquema