

Cálculo Relacional

• **Temas:**

- Introducción al Cálculo Relacional (CR).
- Cálculo Relacional de Tuplas (CRT).
- Equivalencia entre CR y AR.

Cálculo Relacional

- **Referencia:**

- Capítulo 9 del [Elmasri-Navathe].

Cálculo Relacional - Introducción

♦ De que se trata ?

- Familia de lenguajes de consulta sobre Mod Relacional.
- Son equivalentes al Algebra Relacional.
- Basado en fórmulas de Lógica de 1er orden para definir conjuntos
- Una consulta en CR es una especificación de conjunto por comprensión de un conjunto de tuplas.

♦ Dos Sublenguajes:

- Cálculo de Tuplas: variables de tipo tupla, es decir, el universo está formado por tuplas.
- Cálculo de Dominios: variables por valor de atributo, es decir, el universo está formado por valores individuales.
- Sólo se va a trabajar con el primero de estos lenguajes.

El **cálculo relacional** es una familia de lenguajes de consulta sobre el Modelo Relacional, que es equivalente al álgebra relacional pero, se basa en lógica de primer orden. De esta forma, una consulta es la definición por comprensión del conjunto de tuplas que tienen se desean en el resultado.

Hay dos sublenguajes básicos: Cálculo de Tuplas en donde el universo son tuplas y Cálculo de Dominios en donde el universo es la unión de todos los dominios posibles.

En este curso se trabajará solamente con Cálculo de Tuplas.

Cálculo de Tuplas

• Ejemplo.

- Dado el siguiente Esquema Relacional:
 - FABS (#f, Nombre, Direcc) ,
 - PRODS (#p, desc) ,
 - VENTAS(#f, #p, precio)
- Qué ventas fueron por un precio mayor de \$50.000.
- Expresión en CRT:
 - $\{t / \text{VENTAS}(t) \wedge t.\text{precio} > 50000\}$
 - donde: VENTAS(t) especifica que t pertenece a VENTAS.
 - t.precio referencia al atributo PRECIO de la variable de tupla t.
- Resultado:
 - tuplas t de VENTAS con t.precio > 50000.

Dadas las siguientes tablas (ya usadas en Algebra):

FABS (#f, Nombre, Direcc) ,

PRODS (#p, desc) ,

VENTAS(#f, #p, precio)

Se desea saber qué ventas fueron por un precio mayor de 50.000 pesos.

En cálculo de tuplas, la expresión sería la siguiente:

$\{t / \text{VENTAS}(t) \wedge t.\text{precio} > 50000\}$

En esta expresión **Ventas** es un símbolo de predicado tal que su interpretación es la instancia actual de la tabla Ventas. El símbolo **t** es una variable (libre) y **t.precio** es una expresión que representa el valor del atributo **precio** de la variable **t**.

El resultado de esta consulta son las tuplas de la tabla ventas con el precio mayor que 50.000.

Cálculo de Tuplas

- **Refinando las fórmulas ...**

- Si se quieren algunos de los atributos ...
 - Se especifica la estructura del resultado.
 - Esto es equivalente a una Proyección del AR.

- Por ejemplo:

$\{ \langle t.\text{Nombre}, t.\text{Direc} \rangle / \text{FABS}(t) \text{ and } t.\#f > 5 \}$

En caso que la tupla resultado no sea directamente una tupla de una tabla, en la expresión del lado izquierdo de la barra se puede construir la tupla que se necesite.

En el caso del ejemplo, se toman algunos atributos de una misma variable, pero eso no es imprescindible, se pueden tomar atributos de las variables libres que convenga.

Cálculo de Tuplas

• Resumiendo ...

- Informalmente, en una expresión de CRT se necesita especificar la siguiente información :
 - Una tupla que caracteriza el resultado.
 - Se puede especificar la estructura de dicha tupla (sus atributos).
 - Una condición que caracteriza las tuplas resultado en términos de lógica de primer orden.
 - Hay ciertos “cuidados” para escribir fórmulas que realmente describan los conjuntos que interesan.
 - EJ: Para cada variable t ligada, debe aparecer en el alcance de su cuantificación, al menos una expresión $R(t)$ donde R es una tabla de la base.

Informalmente, una expresión en CRT es definición por comprensión del conjunto de tuplas que se desean en el resultado.

Esta definición por comprensión tiene su forma más típica.:

- Describe conjunto por lo que usa los caracteres $\{ \}$.
- Tiene una primera parte en donde se describe la estructura de las tuplas del resultado.
- La segunda parte es una condición en lógica de primer orden.

Cálculo de Tuplas - Intuición

- **Una expresión de CRT tiene la forma:**

$$\{ \langle x_1.A_1, \dots, x_n.A_m \rangle / \varphi(x_1, \dots, x_n) \}$$

- siendo φ una fórmula de 1er orden, donde:
 - Los predicados corresponden a relaciones (tablas) o a comparaciones entre atributos y/o valores.
 - Las variables corresponden a tuplas.
 - La única función válida es la selección de atributos:
(o sea, no hay funciones del usuario).
- El universo de las estructuras que se usan para interpretar las fórmulas es el conjunto de todas las tuplas posibles.

Así, una expresión en Cálculo Relacional de Tuplas tiene la siguiente forma:

$$\{ \langle x_1, \dots, x_n.A_m \rangle / \varphi(x_1, \dots, x_n) \}$$

donde:

φ es una fórmula de primer orden donde las únicas variables libres son x_1, \dots, x_n

Los predicados base de esa fórmula son o bien relaciones (tablas) o bien comparaciones entre atributos y/o valores.

La única función válida es la selección de atributos.

El universo de las estructuras que se usan para interpretar las fórmulas, es el conjunto de todas las tuplas posibles. Esto es el conjunto de todas las tuplas posibles que se pueden construir en base a los dominios.

Esto último hace que las variables siempre se deban pensar como tuplas.

Cálculo de Tuplas - Sintaxis

- **Las expresiones del cálculo son de la forma:**
 - $\{ \langle t_1, \dots, t_n \rangle / \varphi(x_1, \dots, x_n) \}$ donde:
 - t_i son términos de la forma $x_j.A_k$ o bien de la forma c_i . Aquí, x_i es una variable libre de φ y A_k un atributo de una tabla para la que x_i representa una tupla y c_i es una constante de algún dominio (número, string, fecha, etc.)
 - φ es una fórmula donde las x_i que aparecen a la izquierda de la barra son exactamente las variables libres de φ .
($FV(\varphi) = \{x_1, \dots, x_m\}$) .
- **Hay que definir el lenguaje de las fórmulas φ válidas.**
 - Para esto hay que definir dos lenguajes: Los términos y las fórmulas propiamente dichas.

Las expresiones son de la forma presentada en la figura, sin embargo falta definir el lenguaje de los términos y el lenguaje de las fórmulas.

Cálculo de Tuplas - Sintaxis

• **Términos:**

1. x_i es un término, si x_i es una variable.
2. c_i es un término, si c_i es una constante de algún dominio.
3. $x_i.A_j$ es un término, si x_i es una variable y A_j es un nombre de atributo.
4. Estos son todos los términos. (no hay más funciones que la selección de atributos).

• **Hay dos tipos de términos:**

- Atómicos: c_i o $x_i.A_j$
- No atómicos: x_i (Variables)

Los términos o bien son variables x_i (intuitivamente se puede pensar que representan tuplas), o bien constantes de algún dominio, o bien selección de atributos de las tuplas. Observar que:

- Dado que las constantes son de los dominios, necesariamente son valores atómicos.
- Los atributos también representan valores atómicos.

Cálculo de Tuplas - Sintaxis

• La fórmulas pueden ser:

- $t_i <op> t_j$ donde t_i y t_j son términos atómicos ($x_i.A_k$ o c_j). y $<op> \in \{=, <, >, \neq, \dots\}$.
- $P_i(x_i)$ es una fórmula si P_i es el nombre de una tabla y x_i es una variable.
- $(\phi <BinConn> \psi)$ es una fórmula si ϕ y ψ son fórmulas y $<BinConn> \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$.
- $(\neg \phi)$ es una fórmula si ϕ es un fórmula.
- $(\exists x_i \phi)$ y $(\forall x_i \phi)$ son fórmulas si ϕ es una fórmula x_i es una variable.
- Estas son todas las fórmulas.

A las fórmulas las podemos definir como un lenguaje inductivo de la siguiente forma:

•Cláusulas Base:

- O bien comparaciones de atributos y valores o entre atributos.
- O bien símbolos de predicados unarios (que corresponderán a las tablas en la interpretación).

•Cláusulas inductivas:

- O bien dos fórmulas conectadas por un conectivo binario.
- O bien negaciones de fórmulas.
- O bien fórmulas con cuantificadores.

Cálculo de Tuplas – Semántica

- La expresión $\{ \langle x_1.A_k, \dots, x_n.A_j \rangle / \varphi \}$ define una relación tal que:
 - El esquema tiene los n atributos A_k, \dots, A_j
 - La instancia es la formada por todas las tuplas t del **universo** que tienen aridad n , tales que $\varphi(t)$ es verdadera.
- **Observaciones:**
 - La tabla del resultado tiene exactamente n atributos con los nombres A_k, \dots, A_j exactamente en ese orden.

Una expresión en cálculo de tuplas, define una relación con esquema indicado por el lado izquierdo de la expresión y la instancia formada por todas las tuplas del universo tal que la fórmula (la parte derecha de la expresión) es verdadera.

Recordar que $\varphi(t) = \varphi[t/x]$. La variable x puede verse como el vector de todas las variables libres de la fórmula.

Cálculo de Tuplas – Semántica

- **Las fórmulas son verdaderas si cumplen:**
 - **ϕ de la forma $P(t)$ es verdadera**, si y solo si la tupla t pertenece a la tabla P en la base.
 - **ϕ de la forma $t_i <op> t_j$ es verdadera**, si y solo si se cumple la relación especificada por $<op>$ entre los valores asociados a t_i y t_j en el momento de la consulta. (Recordar que t_i y t_j son términos atómicos: representan valores de atributos o constantes de un dominio.)
 - **El resto de las fórmulas se evalúan según la semántica clásica de la lógica de primer orden.**

Para comprender la semántica hay que comprender cómo se construye la interpretación (la estructura) de acuerdo a la base de datos. Esto es muy simple: cada símbolo de predicado de base está asociado a una tabla de la base.

De esta forma, una fórmula de la forma $P(t)$ es verdadera cuando la tupla t está en la instancia actual de la tabla asociada a P .

Una comparación es verdadera de acuerdo con la relación asociada a esa comparación ($<$, $>$, $=$, etc.).

Las fórmulas complejas se evalúan igual que en la lógica clásica de primer orden.

Cálculo de Tuplas – Universo.

- **El universo siempre es el conjunto de todas las tuplas que es posible formar con todos los dominios de los atributos.**
 - Es un conjunto infinito.
 - Contiene a tuplas de cualquier aridad.
 - Contiene a tuplas con cualquier dominio en cualquier posición.
 - Se deben tener en cuenta estas características a la hora de escribir las consultas.

El universo de la estructura sobre la que se evalúan las expresiones está formado por TODAS LAS TUPLAS QUE ES POSIBLE CONSTRUIR, no sólo con las que están en la base.

Esto hace que el universo:

- Sea un conjunto infinito.
- Contenga a tuplas de cualquier aridad.
- Contenga tuplas con cualquier dominio atómico en cualquier posición.

Hay que tener en cuenta estas propiedades a la hora de escribir consultas en Cálculo.

Cálculo de Tuplas - Ejemplos

- **Fabs(#f, Nombre, Direcc), Prods(#p, Desc), Ventas(#f, #p, Precio).**
- **Consulta 1:**
 - Dar nombre y la dirección de fabricantes que hicieron ventas por mas de \$1000.
$$\{ \langle t.nombre, t.Direcc \rangle \mid FABS(t) \wedge \exists v (VENTAS(v) \wedge v.\#f = t.\#f \wedge v.precio > 1000) \}$$
 - Observaciones:
 - Se asumen las reglas de eliminación de paréntesis de la lógica de primer orden.
 - Las únicas variables libres en una expresión del cálculo relacional deberán ser las que aparezcan a la izquierda de la barra (/).
 - En la consulta, **t** es la única variable libre, y **v** es ligada.

Considerando la realidad de los fabricantes, productos y ventas se puede plantear un primer ejemplo:

“Dar el nombre y la dirección de los fabricantes que hicieron ventas por más de \$1000”

En la imagen se puede ver la solución de esta consulta. Observar que **t** es la única variable libre en

$FABS(t) \wedge \exists v (VENTAS(v) \wedge v.\#f = t.\#f \wedge v.precio > 1000)$

Esto hace que **t** sea la única variable que puede aparecer en el constructor de tuplas del resultado (el lado izquierdo izquierdo de la expresión) .

Cálculo de Tuplas - Ejemplos

• Consulta 2:

- $\text{Fabs}(\#f, \text{Nombre}, \text{Direcc}), \text{Prods}(\#p, \text{Desc}), \text{Ventas}(\#f, \#p, \text{Precio})$.
- Dar parejas de numero de fabricante que vendieron los mismos productos.

$$\{ \langle v1.\#f, v2.\#f \rangle / \text{VENTAS}(v1) \wedge \text{VENTAS}(v2) \wedge v1.\#p = v2.\#p \wedge v1.\#f > v2.\#f \}$$

• Observaciones:

- Dos o más variables de tupla de una consulta pueden abarcar la misma relación. En este caso v1 y v2. Esto permite evitar el uso de posiciones.

La segunda consulta pide:

“Dar las parejas de número de fabricante que vendieron los mismos productos”.

Para construir la consulta, conviene analizar el significado del texto de lo pedido: si son fabricantes que vendieron el mismo producto, dado que son fabricantes distintos, deben haber dos tuplas en ventas (una para cada fabricante) que tengan el mismo producto. Por otro lado, si encontramos una pareja $\langle a, b \rangle$ de fabricantes, entonces no es necesario incluir la pareja $\langle b, a \rangle$, al menos en este caso particular. Por eso es que se pide que el primer número de fabricante sea mayor que el segundo. Con esto además, se evitan las parejas $\langle a, a \rangle$ o sea, de un fabricante consigo mismo.

Observar que a pesar de que se utilizan dos ocurrencias de la relación ventas, no es necesario utilizar posiciones porque se usan dos variables distintas.

Cálculo de Tuplas - Ejemplos

• Consulta 3:

- Dar los números de fabricantes, la descripción de los productos que venden y a que precio lo hacen.

$$\{ \langle t1.\#f, t2.desc, t1.precio \rangle / VENTAS(t1) \wedge PRODS(t2) \wedge t1.\#p=t2.\#p \}$$

• Consulta 4:

- Dar los nombres de los fabricantes que vendieron productos con descripción d1.

$$\{ f.nombre / FABS(f) \wedge \exists p.(PRODS(p) \wedge p.desc='d1' \wedge \exists v.(VENTAS(v) \wedge f.\#f=v.\#f \wedge p.\#p=v.\#p)) \}$$

En la consulta 3 se puede ver que se usan del lado izquierdo (en el constructor del resultado) **todas las variables libres de la fórmula.**

Si hay variables de las que no se necesita tomar ningún atributo, entonces deben estar ligadas como se ve en la consulta 4.

Cálculo de Tuplas - Ejemplos

• Consulta 5:

- Dar los nombres de los fabricantes que venden todos los productos.

~~$\{ f.\text{nombre} / \text{FABS}(f) \wedge \forall p. (\text{PRODS}(p) \wedge \exists v. (\text{VENTAS}(v) \wedge f.\#f=v.\#f \wedge p.\#p=v.\#p)) \}$~~

- Observar que:
 - p es cualquier tupla del universo.
 - El universo es infinito.
 - La consulta no es correcta.

Para resolver

“Dar los nombres de los fabricantes que venden todos los productos”

El primer intento puede ser:

$\{ f.\text{nombre} / \text{FABS}(f) \wedge \forall p. (\text{PRODS}(p) \wedge \exists v. (\text{VENTAS}(v) \wedge f.\#f=v.\#f \wedge p.\#p=v.\#p)) \}$

Sin embargo, esto es **INCORRECTO**.

Para ver el error, hay que tener en cuenta que $\forall p.\phi(p)$ se interpreta como “para cualquier elemento del universo, se cumple ϕ ”. Esto hace que lo que se esté diciendo en esa expresión sea

“deme los nombres de los fabricantes tal que tienen una venta para cualquier tupla del universo y además todas las tuplas del universo están en **prods**”

Para leerlo de esa forma, recordar que \forall distribuye frente a \wedge . Para observar el error, recordar que el universo contiene **TODAS LAS TUPLAS POSIBLES QUE SE PUEDEN CONSTRUIR**, lo que claramente es mucho más que el contenido de la tabla **PRODS** ! (El universo están todas las tuplas de Prods pero también las de cualquier otra tabla !!! Y para todas ellas se pide que estén en prods).

Esto hace que el resultado de esta consulta sea vacío siempre.

Cálculo de Tuplas - Ejemplos

• Consulta 5:

- Dar los nombres de los fabricantes que venden todos los productos.
 - El lenguaje **NO PERMITE** escribir algo así:
$$\{ \text{f.nombre} / \text{f} \in \text{FABS} \wedge \forall \text{p} \in \text{PRODS.} \exists \text{v} \in \text{VENTAS.} (\text{f.\#f}=\text{v.\#f} \wedge \text{p.\#p}=\text{v.\#p}) \}$$
- Sin embargo, se pueden escribir fórmulas equivalentes mediante las siguientes traducciones (Relativización):
 - $\exists t \in P. \varphi \equiv \exists t. (P(t) \wedge \varphi)$
 - $\forall t \in P. \varphi \equiv \forall t. (P(t) \rightarrow \varphi)$

Siguiendo con la Consulta 5, lo que se querría escribir es algo de esta forma:

$$\{ \text{f.nombre} / \text{f} \in \text{FABS} \wedge \forall \text{p} \in \text{PRODS.} \exists \text{v} \in \text{VENTAS.} (\text{f.\#f}=\text{v.\#f} \wedge \text{p.\#p}=\text{v.\#p}) \}$$

Sin embargo, la definición del lenguaje no permite utilizar cuantificadores relativizados, es decir, en donde se indica de qué subconjunto del universo pueden tomar valores las variables.

Pero se pueden escribir fórmulas equivalentes en el lenguaje definido teniendo en cuenta las siguientes definiciones:

- $\exists t \in P. \varphi \equiv \exists t. (P(t) \wedge \varphi)$
- $\forall t \in P. \varphi \equiv \forall t. (P(t) \rightarrow \varphi)$

Cálculo de Tuplas - Ejemplos

• Consulta 5:

- Dar los nombres de los fabricantes que venden todos los productos.

$$\{ f.\text{nombre} / \text{FABS}(f) \wedge \forall p.(\text{PRODS}(p) \rightarrow \exists v.(\text{VENTAS}(v) \wedge f.\#f=v.\#f \wedge p.\#p=v.\#p)) \}$$

- Observacion:

- Pensamos relativizado (con \in), escribimos con la forma sin relativizar (se escribe con la traducción).

De esta forma, la consulta 5 se puede escribir de la siguiente forma:

$$\{ f.\text{nombre} / \text{FABS}(f) \wedge \forall p.(\text{PRODS}(p) \rightarrow \exists v.(\text{VENTAS}(v) \wedge f.\#f=v.\#f \wedge p.\#p=v.\#p)) \}$$

De esta forma, se puede pensar “relativizado” o sea limitando las variables de los cuantificadores a conjuntos, pero escribir en el lenguaje definido.

Conviene además, pensar que sucede si la tabla PRODS es vacía: el implica es trivialmente verdadero con lo que el paratodo se hace verdadero para cualquier fabricante. El resultado de la consulta en ese caso, son todos los nombres de fabricantes.

Devolver ese resultado para la consulta pedida puede ser discutible. Sin embargo, se puede pensar que al no haber productos no debería haber ventas por lo que efectivamente, los fabricantes que vendieron todos los productos existentes son todos.

Cálculo de Tuplas – Ejemplos

• Consulta 6:

- Dar los nombres de fabricantes que vendieron todos los productos con descripcion = 'd3'.

```
{ t.Nombre / FABS(t) ∧  
  ( (∀p)(PRODS(p) ∧ p.desc='d3' →  
    (∃v)( VENTAS(v) ∧ p.#p=v.#p ∧ t.#f=v.#f)) )  
}
```

- Que resultado debe dar la consulta si no hay productos con descripcion = 'd3' ?

- En este caso debería dar vacío, por lo que la consulta anterior es incorrecta.

```
{ t.Nombre / FABS(t) and  
  ( (∀p)(PRODS(p) and p.desc='d3' →  
    (∃v)( VENTAS(v) and p.#p=v.#p and t.#f=v.#f)) )  
    and (∃q)(PRODS(q) and q.desc='d3')  
}
```

- Hay que analizarlo en cada caso concreto !!!!.

El caso del antecedente que falla es claramente un problema en algunos casos.

La consulta 6 pide

“Dar los nombres de los fabricantes que vendieron todos los productos con descripción d3”.

La expresión en CRT que resolvería esa consulta sería:

```
{ t.Nombre / FABS(t) ∧  
  ( (∀p)(PRODS(p) ∧ p.desc='d3' →  
    (∃v)( VENTAS(v) ∧ p.#p=v.#p ∧ t.#f=v.#f)) )  
}
```

Sin embargo, si no hay productos con descripción 'd3' entonces debería dar vacío porque no puede haber ningún vendedor que lo haya vendido. Esto hace que la consulta sea claramente incorrecta.

Aún así, la solución es bastante simple: agrega un existencial con el antecedente del implica. Esto deja la consulta de la siguiente forma:

```
{ t.Nombre / FABS(t) and  
  ( (∀p)(PRODS(p) and p.desc='t3' →  
    (∃v)( VENTAS(v) and p.#p=v.#p and t.#f=v.#f)) )  
    and (∃q)(PRODS(q) and q.desc='t3')  
}
```

Cálculo de Tuplas – Fórmulas Inseguras

• Fórmulas inseguras:

• Qué son ?

- Fórmulas que permiten resultados infinitos en consultas CRT.

• Cómo puede pasar esto ?

- En la medida de que el universo de valores es infinito, pueden construirse consultas CRT que generen resultados infinitos.

• Por ejemplo:

- $\{ t / \neg \text{FABS}(t) \}$: Es insegura ya que retorna las tuplas del universo infinito que no pertenecen al conjunto finito de Fabricantes.
- $\{ t / 1=1 \}$: Esta consulta no respeta la sintaxis presentada ya que la variable t no está libre en la expresión. Si fuera posible escribirla, entonces sería insegura dado que t tomaría como valores todo el universo.

Una **fórmula insegura** es una fórmula que construye resultados infinitos. Esto es posible dado que el universo es infinito y las variables toman valores de ese universo.

En la figura se ven dos consultas que tienen claramente fórmulas inseguras, generando de esa forma, resultados infinitos.

Cálculo de Tuplas - Fórmulas Seguras

• **Dominio de una Expresión CRT:**

- Es el conjunto de todos los valores que, aparecen como valores constantes en la expresión o bien existen en cualquier tupla de las relaciones a las que se hace referencia en la expresión.

Para resolver el problema de las fórmulas inseguras y escribir sólo fórmulas seguras, es necesario definir primero de dónde se desea que tomen valores las variables en una consulta para luego, limitar de alguna forma las expresiones para garantizar que tomen valores de ese conjunto.

De esta forma, se define el **Dominio de una Expresión en Cálculo Relacional de Tuplas**: Es el conjunto de todos los valores que, aparecen como constantes en la expresión o bien existen en alguna tupla de las relaciones a las que se hace referencia en la expresión.

Observar que esta definición es paramétrica en la expresión . O sea, sobre la misma base, dos expresiones distintas pueden tener dominios distintos.

Cálculo de Tuplas - Fórmulas Seguras

- **Fórmulas seguras:**

- Se dice que una expresión CRT es **segura** si todos los valores de su resultado pertenecen al dominio de la expresión.

Una expresión en CRT es segura si todos los valores del su resultado pertenecen al dominio de la expresión.

Observar que con esto se está garantizando que sólo se devuelven en el resultado tuplas de la base.

Cálculo de Tuplas - Fórmulas Seguras

♦ Ejemplo:

- ♦ $\{v.\#f / \neg \text{VENTAS}(v) \}$
 - ♦ Es **insegura** porque el resultado incluye valores que no están en el Dominio de la expresión (formado por las tuplas de VENTAS).
- ♦ $\{t.\#f / \text{FABS}(t) \wedge \neg(\exists v)(\text{VENTAS}(v) \text{ and } v.\#f=t.\#f) \}$
 - ♦ Es **segura**, porque el resultado solo incluye valores de FABS.
- ♦ $\{ t / (\exists f) (\text{FABS}(f)) \}$
 - ♦ Es **insegura** porque el resultado incluye valores que no están en el Dominio de la expresión (formado por las tuplas de FABS).

La primer consulta es insegura porque el resultado incluye valores que no están en el dominio de la expresión. Ese dominio está formado sólo por las tuplas de la tabla ventas y la expresión pide sólo tuplas que NO ESTAN en ventas.

La segunda consulta es segura. Los valores del resultado, son tomados de fabs.

La tercer consulta, es insegura, porque devuelve valores del universo que no so tomados de ninguna tabla. De acuerdo a las condiciones que se vieron, esta consulta no se debería escribir nunca dado que la variable que aparece en el constructor no es una variable libre de la expresión. Si se pudiera escribir, devolvería todo el universo si hay alguna tupla en fabs y vacío cuando no hay tuplas en fabs.

Cálculo de Tuplas - Fórmulas Seguras

- Hay algunos criterios mas o menos generales para chequear si la fórmula es segura. Entre otros:

- Pensar en la fórmula de la consulta, traducida con \wedge, \vee y \neg .
- Si la fórmula tiene la forma $\phi_1 \vee \dots \vee \phi_n$, **en cada ϕ_i** deben aparecer **todas** las variables libres en un predicado **no negado**.
- Si la fórmula es de la forma $\phi_1 \wedge \dots \wedge \phi_n$, cada variable debe aparecer **en al menos una ϕ_i** en un predicado no negado.
- Recordar las asociaciones de \exists con \vee y \forall con \wedge y razonar de forma similar.
- Ej: $\{ f / \neg \text{FABS}(f) \}$,
 $\{ f / \exists p(\text{FABS}(f) \rightarrow \text{PRODS}(p) \wedge p.\#f = f.\#f) \}$,
 $\{ t.\#f / \neg \text{FABS}(t) \vee \neg \text{VENTAS}(t) \}$

Son todas expresiones a evitar.

Se pueden definir reglas sintácticas de cierta complejidad que garantizan la seguridad de la fórmula. Aquí se presentan algunos criterios que nos permiten decidir si la fórmula que se escribe es segura aunque no sea exactamente “reglas sintácticas”.

- Sin importar los operadores que se utilicen en la fórmula, la fórmula conviene leerla como si estuviera escrita sólo con \wedge, \vee y \neg .
- Si la fórmula tiene la forma $\phi_1 \vee \dots \vee \phi_n$, **en cada ϕ_i** deben aparecer **todas** las variables libres en un predicado **no negado**. Observar que cuando se hace un \vee se están tomando las tuplas que satisfacen las dos partes de ese operador o sea, la unión de las tuplas que satisfacen cada parte. La idea es que se tomen de la base sin importar a qué lado del operador satisfacen.
- Si la fórmula es de la forma $\phi_1 \wedge \dots \wedge \phi_n$, cada variable debe aparecer **en al menos una ϕ_i** en un predicado no negado. La lógica de esto es que las tuplas que se consideran aquí, son las que satisfacen cada parte del operador \wedge simultáneamente. Esto hace que si aparece en un predicado base no negado, se tome que un tupla que está en la base y el resto son condiciones que esa tupla debe cumplir para aparecer en el resultado.
- Recordar las asociaciones de \exists con \vee y \forall con \wedge y razonar de forma similar a la anterior.

Cálculo Vs Algebra

- **Qué consultas se pueden resolver en CRT Seguro?**

- Todas las resolubles en Algebra Relacional.

- **Cómo se expresan :**

- Selección:

- Se expresan las comparaciones sobre los atributos.

- Ventas por mas de \$100 y menos de \$ 5000.

$\{t \mid \text{VENTAS}(t) \wedge t.\text{precio} > 100 \wedge t.\text{precio} < 5000\}$

- Proyección:

- Se estructura la tupla resultado como se desea.

- Nombre y dirección de fabricantes.

$\{ \langle t.\text{Nombre}, t.\text{Direcc} \rangle \mid \text{FABS}(t) \}$

Cuando se define el modelo relacional, el mismo se define con el Algebra y el Cálculo relacional y se demuestra que son lenguajes equivalentes. A los lenguajes que tienen al menos el mismo poder expresivo del Algebra o del Cálculo se les llama **Relacionalmente completos**.

En las imágenes se pueden ver con ejemplos, la forma en que se pueden escribir los operadores en Algebra

Cálculo Vs Algebra

- Union:

- Se utiliza el operador “or”.
 - Nombre y dirección de fabricantes llamados Ana o Pepe.
 $\{t.Nombre, t.Direcc \mid FABS(t) \wedge (t.Nombre='Ana' \vee t.Nombre='Pepe') \}$

- Join:

- Se utiliza el cuantificador existencial.
 - Dar nombre y la dirección de fabricantes que hicieron alguna venta.
 $\{t.Nombre, t.DIRECCIÓN \mid FABS(t) \wedge (\exists v)(VENTAS(v) \wedge v.\#f=t.\#f) \}$

Cálculo Vs Algebra

- Resta:

- Se utiliza el “not”.
 - Nombre y dirección de fabricantes que no vendieron nada.

$$\{t.Nombre, t.Direcc / FABS(t) \wedge \\ \neg (\exists v)(VENTAS(v) \wedge v.\#f = t.\#f) \}$$

- Otro ejemplo ...

- Dar descripción de productos no vendidos por Ana.

$$\{t.Descrip / PRODS(t) \wedge \\ \neg (\exists v)(\exists f)(VENTAS(v) \wedge FABS(f) \\ \wedge v.\#p = t.\#p \text{ and } v.\#f = f.\#f \\ \wedge f.Nombre = 'Ana') \}$$

Conclusiones

- **Se presentó el Cálculo Relacional:**
 - Familia de lenguajes relacionales basados en lógica de 1er orden.
 - Incluye dos sub-lenguajes:
 - Cálculo de Tuplas.
 - Cálculo de Dominios.
- **El CR ha tenido una gran influencia:**
 - Lenguajes para BDs basados en Lógica.
 - En lenguajes implementados:
 - Sub consultas del SQL.

Se presentó el Cálculo Relacional, una familia de lenguajes de consulta sobre el modelo relacional basados en la lógica de primer orden. Esta familia incluye dos lenguajes: el cálculo relacional de tuplas y el cálculo relacional de dominios.

Son lenguajes que han tenido gran influencia en la implementación de otros lenguajes como DATALOG y SQL. Observar que la estructura de una consulta SQL básica equivalente a la de ciertas fórmulas del cálculo:

```
Select  $x_1.A_1, \dots, x_n.A_m$ 
  From  $P_1 x_1, \dots, P_n x_n$ 
 Where  $\phi(x_1, \dots, x_n)$ 
 $\equiv \{ \langle x_1.A_1, \dots, x_n.A_m \rangle / P_1(x_1) \wedge \dots \wedge P_n(x_n) \wedge \phi(x_1, \dots, x_n) \}$ 
```

Observar que la estructura de la consulta en cálculo anterior es segura si ϕ es segura, dado que todos los valores de las variables libres surgen de las tablas de la base.

Conclusiones

• El CR es equivalente al AR:

- En realidad las expresiones seguras del CR son equivalentes al AR.
- Esto se demuestra en el Teorema de Codd.
 - $\cup(R, S) = \{t / R(t) \vee S(t)\}$
 - $-(R, S) = \{t / R(t) \wedge \neg S(t)\}$
 - $\Pi_{A_1 \dots A_n}(R) = \{t.A_1, \dots, t.A_n / R(t)\}$
 - $\sigma_\phi(R) = \{t / R(t) \wedge \phi(t)\}$
 - $\text{Join}_\phi(R, S) = \{t / (\exists r^n)(\exists s^m)(R(r) \wedge S(s) \wedge \phi(r, s) \wedge t.A_1=r.A_1 \wedge \dots t.A_{n+m}=s.A_m)\}$

Equivalencias lógicas útiles

• Entre cuantificadores:

$$• (\forall x)(P(x)) \leftrightarrow \neg (\exists x)(\neg (P(x)))$$

$$• (\exists x)(P(x)) \leftrightarrow \neg (\forall x)(\neg (P(x)))$$

• Entre conectivos:

$$• \neg(x \vee y) \leftrightarrow (\neg x) \wedge (\neg y)$$

$$• \neg(x \wedge y) \leftrightarrow (\neg x) \vee (\neg y)$$

$$• x \rightarrow y \leftrightarrow \neg x \vee y.$$

Las equivalencias de la figura pueden ser particularmente útiles a la hora de analizar la seguridad de una fórmula. Son exactamente las mismas que se estudian en Lógica.