

# Teórico 3

## Álgebra Relacional



# Modelo Relacional

- El modelo de datos relacional representa la base de datos como un conjunto de tablas. Aunque las tablas son un conjunto simple e intuitivo, existe una correspondencia directa entre el concepto de una tabla y el concepto matemático de Relación.
- Una Base de Datos relacional consiste de un conjunto de tablas , que tienen asignado un nombre único. Una fila en una tabla representa una relación entre un conjunto de valores. Una tabla es un conjunto de estas relaciones.
- Dada una tabla Personas con atributos DNI, Dir, NyAp; y suponiendo que  $D_1$ ,  $D_2$ ,  $D_3$  son los dominios de DNI, Dir NyAp respectivamente. Cada fila de Persona debe componerse por una tupla de tres Valores  $(V_1, V_2, V_3)$  , donde  $V_1 \in D_1$  ,  $V_2 \in D_2$  y  $V_3 \in D_3$ . En general Persona va a contener únicamente un subconjunto de todas las filas posibles. Por lo tanto Persona es un subconjunto de

$$\times_{i=1}^3 D_i$$

- Mas general, una tabla de n columnas debe ser un subconjunto de:

$$\times_{i=1}^n D_i$$

# Modelo Relacional Cont.

## **Notación:**

Cuando se desea definir los dominios se utiliza la siguiente notación (DNI: entero, NyAp: Cadena, Dir:Cadena)

## **Por ejemplo, si tenemos la tabla:**

articulo(cod\_articulo: Entero, Descripcion:Cadena, precio:Real)

Un tupla de articulo pertenece al producto cartesiano  
Entero X Cadena X Real

- Debido a que las tablas son básicamente Relaciones, se utilizan los términos matemáticos *Relación* y *tupla* en vez de *Tabla* y *Fila*.

En la siguiente figura se muestra una instancia de la Relación Personas con tres tuplas:

DNI	NyAp	Dir
23423444	Jorge Peres	Bs As 234
23212187	Marcos Juarez	San Martin 123
14256632	Roberto Cejas	San Juan 342

Sea la variable de tupla  $t$  la primer tupla de la relación notaremos  $t[\text{DNI}]$  el valor en  $t$  del atributo DNI, por lo tanto  $t[\text{DNI}] = 23423444$ .

- En Base de Datos debe diferenciarse entre *esquema* (diseño lógico) y una instancia que es la información que almacena la Base de Datos en un instante de tiempo.

# Claves

- Superclave: es un conjunto de atributos que permiten identificar unívocamente una tupla(fila) dentro de una relación (Tabla).

## **Formalmente:**

- Sea  $R$  un esquema de Relación, se dice que un subconjunto  $K$  de  $R$  es una superclave de  $R$  si para todo par de tuplas  $t_1, t_2 \in r(R)$  y  $t_1 \neq t_2$  entonces  $t_1[k] \neq t_2[k]$

- Clave candidata, una clave candidata en un conjunto de atributos que es superclave y no tiene ningún subconjunto propio que sea superclave.

## **Formalmente:**

- Sea  $R$  un esquema de Relación. Se dice que un subconjunto  $K$  de  $R$  es una clave candidata de  $R$  si,  $K$  es superclave y no existe  $SK \subset K$  tal que  $SK$  es superclave.

- Clave primaria: es cualquiera de la claves candidatas de la relación, la elección se realiza por la importancia de los atributos en el dominio modelado.

# Ejemplos de Claves

- Consideremos la siguiente tabla:

Alumno(DNI, #alumno, nombre, dirección, fecha)

Superclaves: {DNI}, {#alumno}, {DNI, #alumno}, {DNI, nombre}, {DNI, #alumno, dirección, fecha}, etc..

Claves candidatas: {DNI}, {#alumno}

Clave primaria: {DNI}

# Lenguajes de Consulta

Un lenguaje de consulta es un lenguaje a través del cual se solicita información a la base de datos. Estos lenguajes están divididos en:

- *Procedimentales* el usuario le da instrucciones al sistema de cómo obtener la información deseada.
- *No procedimentales* el usuario describe que información deseada, sin dar un procedimiento concreto.

Dentro de los procedimentales se encuentra el álgebra relacional y dentro de los no procedimentales se encuentran el calculo relacional de Tuplas y de Dominio.

# Álgebra Relacional

El álgebra relacional es un lenguaje de consulta procedimental. Posee un conjunto de operadores que toman como parámetros una o dos relaciones y devuelven otra, estos operadores los podemos clasificar en:

## *Operadores Básicos*

- *Selección*
- *Proyección*
- *Unión*
- *Diferencia*
- *Producto Cartesiano*
- *Renombramiento*

## *Operadores Derivados*

- *Intersección*
  - *Join*
  - *División*
  - $\Theta$  *Join*
- Se forman combinando los operadores básicos

Asignación



# Notación

- Las letras mayúsculas R,S,T, ... denotan esquemas de relaciones (estructura)
- Las letras minúsculas r,s,t,... denotan instancias (datos concretos) de las relaciones cuyos esquemas son R,S,T,... respectivamente.

# Selección

$\sigma_{\text{predicado}}(r)$

*Es un operador unario.*

- *Define una relación con los mismos atributos que  $R$  y que contiene solo aquellas filas de  $r$  que satisfacen la condición especificada (predicado).*

**Ejemplo:** Consideremos la siguiente tabla

ingeniero

#Emp	Nombre	Edad
12	Juan	26
21	Manuel	40
233	Maria	50

a) *Dar los ingenieros de mas de 35 años*

$\sigma_{\text{Edad} \geq 35}$  (ingeniero)

#Emp	Nombre	Edad
21	Manuel	40
233	Maria	50

b) *Dar los datos del ingeniero cuyo número de empleado es 21*

$\sigma_{\text{\#Emp}=21}$  (ingeniero)

#Emp	Nombre	Edad
21	Manuel	40

c) *Devolver los ingenieros menores de 21 Años*

$\sigma_{\text{Edad} < 21}$  (ingeniero)

#Emp	Nombre	Edad
------	--------	------

# Proyección

$$\Pi_{Atr1, \dots, Atrn}(r)$$

- *Es un operador unario.*
- *Define una relación que contiene un subconjunto de la columnas de  $R$  con los valores de los atributos especificados, eliminando filas duplicadas en el resultado.*

**Ejemplo:** Dada la siguiente relación

ingeniero

#Emp	Nombre	Edad
12	Juan	26
21	Manuel	40
233	Maria	50
455	Juan	42

a) *Dar nombre y edad de los ingenieros*

$\Pi_{\text{Nombre,Edad}}$  (ingeniero)

Nombre	Edad
Juan	26
Manuel	40
Maria	50
Juan	42

b) *Dar el nombre de los ingenieros*

$\Pi_{\text{Nombre}}$  (ingeniero)

Nombre
Juan
Manuel
Maria

# Propiedad de Unión Compatible

Dos relaciones  $R$  y  $S$  tienen la propiedad de unión compatible si:

- $R$  y  $S$  tienen el mismo número de atributos (aridad).
- Los dominios de los  $i$ -ésimos atributos de  $R$  y  $S$  son iguales para todo  $i$ .

# Unión

$r \cup s$

- La unión de dos relaciones  $r$  y  $s$ , es otra relación que contiene todas las tuplas que están en  $r$ , o en  $s$ , o en ambas, eliminándose las tuplas duplicadas.
- $R$  y  $S$  deben ser **unión-compatible**, es decir, definidas sobre el mismo conjunto de atributos.

**Ejemplo:** Dada las siguiente relaciones:

ingeniero

#Emp	Nombre	Edad
12	Juan	26
21	Manuel	40
233	Maria	50

jefe

#Emp	Nombre	Edad
12	Juan	26
245	Marcos	45

a) Devolver todo el personal, es decir, ingenieros y jefes.

ingeniero  $\cup$  jefe

#Emp	Nombre	Edad
12	Juan	26
21	Manuel	40
233	Maria	50
245	Marcos	45



# Diferencia

***r - s***

- *La diferencia de dos relaciones r y s, es otra relación que contiene las tuplas que están en la relación r, pero no están en s.*
- *R y S deben ser unión-compatible.*

**Ej.:** Dada las siguiente relaciones:

ingeniero

#Emp	Nombre	Edad
12	Juan	26
21	Manuel	40
233	Maria	50

jefe

#Emp	Nombre	Edad
12	Juan	26
245	Marcos	45

a) Devolver los Ingenieros que no son jefes.

ingeniero - jefe

#Emp	Nombre	Edad
21	Manuel	40
233	Maria	50

b) Devolver los Jefes que no son Ingenieros.

jefe - ingeniero

#Emp	Nombre	Edad
245	Marcos	45

# Producto Cartesiano

***r x s***

- *Define una relación que es la concatenación de cada una de las tuplas (filas) de la relación r con cada una de las tuplas (filas) de la relación s.*

**Ejemplo:** Dada las siguientes relaciones:

ingeniero

#Emp	Nombre	#Dep
201	Juan	1001
202	Manuel	1002
203	Carlos	1001

departamento

#Dep	Nombre
1001	Electricidad
1002	Mecanica

proyecto

#Pro	Descripcion
501	Motor Hibrido
502	Transformador

a) Devolver los ingenieros junto a todos los departamentos de la empresa:

ingeniero X departamento

#Emp	Ing.Nombre	Ing.#Dep	Depa.#Dep	Depa.Nombre
201	Juan	1001	1001	Electricidad
201	Juan	1001	1002	Mecánica
202	Manuel	1002	1001	Electricidad
202	Manuel	1002	1002	Mecánica
203	Carlos	1001	1001	Electricidad
203	Carlos	1001	1002	Mecánica

b) Devolver los departamentos junto a todos proyectos de la empresa.

departamento X proyecto

#Dep	Nombre	#Pro	Descripcion
1001	Electricidad	501	Motor Hibrido
1001	Electricidad	502	Transformador
1002	Mecanica	501	Motor Hibrido
1002	Mecanica	502	Transformador

# Renombramiento

$\rho_x(E)$

- Define un nuevo nombre (x) para el resultado de una expresión (E).
- Se puede utilizar para “cruzar” información de una misma tabla.

**Ejemplo:** Dada la siguiente relación

empleado

#Emp	Nombre	#Emp_Jefe
500	Juan	null
501	Manuel	500
502	Pedro	500
503	Carlos	null
504	Maria	503

a) Listar todos los empleados incluyendo el nombre de su jefe.

$\Pi_{\text{empleado.\#Emp, empleado.\#Nombre, jefe.Nombre}} \left( \right.$   
 $\left. \sigma_{\text{empleado.\#Empl\_jefe = jefe.\#Emp}} (\text{empleado} \times \rho_{\text{jefe}} (\text{empleado})) \right)$

<b>empleado.#Em</b>		
$\varepsilon$		
$\varepsilon$		
$\varepsilon$		

# Definición Formal del Álgebra Relacional

Las expresiones fundamentales del álgebra relacional se componen de:

- Una relación de la Base de Datos.
- Una relación constante.

Las expresiones generales del álgebra relacional se construyen a partir de subexpresiones menores. Sean  $E1$  y  $E2$  expresiones del álgebra relacional. Las siguientes son expresiones del álgebra:

- $E1 \cup E2$
- $E1 - E2$
- $E1 \times E2$
- $\sigma_p(E1)$ , donde  $p$  es un predicado sobre los atributos de  $E1$ .
- $\pi_S(E1)$ , donde  $S$  es una lista de atributos de  $E1$ .
- $\rho_x(E1)$ , donde  $x$  es el nuevo nombre del resultado de  $E1$ .

# Operadores no Básicos





# Intersección

$r \cap s$

- Define una relación que contiene el conjunto de todas las filas que están tanto en la relación  $r$  como en  $s$ .
- $R$  y  $S$  deben ser unión-compatible.
- Equivalencia con operadores básicos.

$$r \cap s = r - (r - s)$$

**Ej.:** Dada las siguiente relaciones

ingeniero

#Emp	Nombre	Edad
12	Juan	26
21	Manuel	40
233	Maria	50

jefe

#Emp	Nombre	Edad
12	Juan	26
245	Marcos	45

a) Encontrar los ingenieros que son jefes.

ingeniero  $\cap$  jefe

#Emp	Nombre	Edad
12	Juan	26

# Join

# Natural Join (Unión Natural)

$r \bowtie s$  ó  $r * s$

- El resultado es una relación con los atributos de ambas relaciones y se obtiene combinando las tuplas de ambas relaciones que tengan el mismo valor en los atributos comunes.
- Normalmente la operación de join se realiza entre los atributos comunes de dos tablas que corresponden a la clave primaria de una tabla y la clave foránea correspondiente de la otra tabla.
- Método
  - Se realiza el producto cartesiano  $r \times s$ .
  - Se seleccionan aquellas filas del producto cartesiano donde los atributos comunes tengan el mismo valor.
  - Se elimina del resultado una ocurrencia (columna) de cada uno de los atributos comunes.
- Equivalencia con operadores básicos

$$r \bowtie s = \Pi_{R \cup S} \sigma_{r.A1 = s.A1 \wedge r.A2 = s.A2 \wedge \dots \wedge r.An = s.An} (r \times s)$$

$$\text{Donde } R \cap S = \{A1, A2, \dots, An\}$$



# Outer Join

- Es una variante del join en la que se intenta mantener toda la información de los operandos, incluso para aquellas filas que no participan en el join.
- Se “rellenan con valores nulos” las tuplas que no tienen correspondencia en el join.
- Tiene tres variantes:
  - Left ( $\sqcup\bowtie$ ): se tienen en cuenta todas las filas del primer operando.
  - Right ( $\bowtie\sqcup$ ): se tienen en cuenta todas las filas del segundo operando.
  - Full ( $\sqcup\bowtie\sqcup$ ): se tienen en cuenta todas las filas de ambos operandos.

# Ejemplos de los diferentes tipos de Join

Consideremos las siguientes relaciones:

ingeniero

#Emp	Nombre	#Dep
201	Juan	1001
202	Manuel	1002
203	Carlos	1001
204	Ariel	1004

departamento

#Dep	Descripcion
1001	Electricidad
1002	Mecanica
1003	Mantenimiento

- a) Ingenieros con sus departamentos (nombre incluido), sólo deberá mostrar los ingenieros que tengan departamento y sólo los departamentos que tengan algún Ingeniero.

ingeniero  $\bowtie$  departamento

#Emp	Nombre	#Dep	Descripcion
201	Juan	1001	Electricidad
202	Manuel	1002	Mecanica
203	Carlos	1001	Electricidad

- b) Devolver todos los ingenieros con sus departamentos, si hay ingenieros cuyo departamento no exista, los debe listar también.

ingeniero  $\bowtie$  departamento

#Emp	Nombre	#Dep	Descripcion
201	Juan	1001	Electricidad
202	Manuel	1002	Mecanica
203	Carlos	1001	Electricidad
204	Ariel	1004	null

- c) Listar todos los ingenieros con sus departamentos, sólo deberá mostrar los ingenieros que tengan departamento, si los departamentos no tienen ingenieros los deberá mostrar.

ingeniero  $\ltimes$  departamento

#Emp	Nombre	#Dep	Descripcion
201	Juan	1001	Electricidad
202	Manuel	1002	Mecanica
203	Carlos	1001	Electricidad
null	null	1003	Mantenimiento

- a) Listar todos los ingenieros con sus departamentos, si hay ingenieros cuyo departamento no exista, los debe listar también, igualmente si hay departamentos que no tengan ingenieros también deben listarse.

ingeniero  departamento

#Emp	Nombre	#Dep	Descripcion
201	Juan	1001	Electricidad
202	Manuel	1002	Mecanica
203	Carlos	1001	Electricidad
204	Ariel	1004	null
null	null	1003	Mantenimiento



# División

***r / s***

- Define una relación sobre el conjunto de atributos  $C = R-S$ , incluido en la relación  $R$ , y que contiene el conjunto de valores de  $C$ , que en las tuplas de  $r$  están combinadas con cada una de las tuplas de  $s$ .
- Se debe cumplir que  $S \subset R$ .
- Equivalencia con operadores básicos:  
$$r / s = \Pi_{R-S}(r) - \Pi_{R-S}((\Pi_{R-S}(r) \times s) - r)$$

**Ej.:** Dadas las siguientes relaciones:

trabaja

#Emp	#Proy
201	501
201	502
202	501
202	502
202	503
203	501
203	503

proyecto

#Proy
501
502

a) Obtener los empleados que trabajan en todos los proyectos

trabaja / proyecto

#Empl
201
202

# Asignación

A veces es conveniente escribir una expresión del álgebra relacional por partes utilizando la asignación a una variable de relación temporal.

Como ejemplo se muestra la operación de división expresada en pasos:

$$\text{aux1} \leftarrow \Pi_{R-S}(r)$$

$$\text{aux2} \leftarrow \Pi_{R-S}((\Pi_{R-S}(r) \times s) - r)$$

$$\text{resultado} \leftarrow \text{aux1} - \text{aux2}$$

# $\theta$ Join

$$r \bowtie_p s \text{ ó } r^*_p s$$

- Define una relación que contiene las tuplas que satisfacen el predicado  $p$  en el producto cartesiano de  $r$  y  $s$ .
- El predicado no tiene por qué definirse sobre atributos comunes.
- Equivalencia con operadores básicos

$$r \bowtie_p s = \sigma_p(r \times s)$$

# Cálculo Relacional

- La Lógica de Primer Orden puede pensarse como un lenguaje de consulta de dos formas:
  - Cálculo relacional de tuplas.
  - Cálculo relacional de dominio.
- La diferencia es el nivel en el que son utilizadas las variables
  - Nivel de Tuplas.
  - Nivel de Atributo para los Dominios.

# Cálculo Relacional de Tuplas

- Lenguaje de Consulta no Procedimental
  - Describe información deseada sin dar un proceso específico para obtener esa información.

- Consultas de la forma

$$\{ t \mid P( t) \}$$

- Conjunto de tuplas  $t$  tal que  $P(\text{Predicado})$  es verdadero en  $t$ .

# Ejemplo de Cálculo Relacional de Tuplas

Consideremos el siguiente esquema de base de datos de un banco:

- sucursal (nombre\_sucursal, sucursal-ciudad, activos)
- cuenta (número\_cuenta, nombre\_sucursal, saldo, número\_cliente)
- préstamo (número\_prestamo, nombre\_sucursal, importe, número\_cliente)
- Cliente (número\_cliente, apellido, nombre, DNI)

# Ejemplo del Banco: Consultas

- Encontrar el nombre de la sucursal, el número de préstamo, el importe y el número de cliente de los préstamos mayores a \$ 12.000

$$\{ t \mid t \in \text{préstamo} \wedge t[\text{importe}] > 12.000 \}$$

- Encontrar el número de préstamo de todos los préstamos superiores a \$ 12.000

$$\{ t \mid \exists p \in \text{préstamo} ( t[\text{número}] = p[\text{número}] \wedge p[\text{importe}] > 12.000 ) \}$$

- En el ejemplo, la variable  $t$  se define únicamente para el atributo número, por lo tanto ese es su esquema



# Cálculo Relacional de Tuplas (CRT)

$$\{ t \mid P( t) \}$$

## Notación

- $t$  : una *variable* de tupla
- $t[A]$  : el *valor* de la tupla  $t$  en el atributo  $A$ .
- $t \in r$  : la tupla  $t$  está en la relación  $r$ .
- $P$  : fórmula similar a aquellas del cálculo de predicados

# Fórmula del Cálculo de Predicados

- Conjunto de cuantificadores

$$\exists t \in r (Q(t))$$

- Existe una tupla  $t$  en la relación  $r$  tal que el predicado  $Q(t)$  es verdadero

$$\forall t \in r (Q(t))$$

- $Q(t)$  es verdadero para todas las tuplas  $t$  que pertenecen a la relación  $r$ .

# Ejemplo del Banco: Consultas

- Encontrar el número de todos los clientes del banco que tienen concedido un préstamo, una cuenta o ambas cosas:

$$\{ t \mid \exists c \in \text{cuenta} ( t [\text{número\_cliente}] = c [\text{número\_cliente}] ) \\ \vee \exists p \in \text{prestamo} ( t [\text{número\_cliente}] = p [\text{número\_cliente}] ) \}$$

- Encontrar el número de los clientes del banco que tienen concedido un préstamo y tienen una cuenta:

$$\{ t \mid \exists c \in \text{cuenta} ( t [\text{número\_cliente}] = c [\text{número\_cliente}] ) \\ \wedge \exists p \in \text{prestamo} ( t [\text{número\_cliente}] = p [\text{número\_cliente}] ) \}$$

# Ejemplo del Banco: Consultas(sigue)

Encontrar el apellido y nombre de todos los clientes del banco que tienen concedido un préstamo en la sucursal Neuquén:

$$\{ t \mid \exists c \in \text{cliente} ( t [\text{apellido}] = c [\text{apellido}] \\ \wedge t [\text{nombre}] = c [\text{nombre}] \\ \wedge \exists p \in \text{préstamo} ( p [\text{nombre-sucursal}] = \text{“Neuquén”} \\ \wedge c [\text{número\_cliente}] = p [\text{número\_cliente}] )) \}$$

# Definición formal (CRT) (I)

- Definición formal del Cálculo de tuplas
  - Una expresión del Cálculo de tuplas tiene la forma:

$$\{ t \mid P(t) \}$$

– tiene:

- *P es una fórmula donde pueden aparecer varias variables de tupla*
- *t ∈ (variable de tupla libre)*
  - ∃ s ∈ o ∀ s ∈ (s, variable de tupla ligada)*
- *Las fórmulas se componen de átomos*

# Definición formal (CRT) (II)

**Los átomos tienen una de la siguientes formas:**

–  $s \in r$

- $s$  variable de tupla y  $r$  relación.

–  $s[x] \Theta u[y]$ ,

- $s, u$  variables de tupla;
- $x, y$  atributos sobre  $s$  y  $u$  respectivamente;
- $\Theta$  operador ( $>$ ,  $<$ ,  $>=$ ,  $=$ ,  $<>$ , etc)

–  $s[x] \Theta c$

- $c$  constante.

# Definición formal (CRT) (III)

Las fórmulas se construyen a partir de átomos:

- Un átomo es una fórmula
- Si  $P_1$  es fórmula  $\Rightarrow \neg P_1$  es fórmula
- Si  $P_1$  y  $P_2$  son fórmulas  $\Rightarrow$

$P_1 \vee P_2, P_1 \wedge P_2, P_1 \Rightarrow P_2$  son fórmulas

- $P_1(s)$  es fórmula que contiene una variable de tupla libre  $s$  y  $r$  es una relación  $\Rightarrow$

$\forall s \in r(P_1(s))$  y  $\exists s \in r(P_1(s))$  son fórmulas

# Definición formal (CRT) (IV)

- Igual que en el álgebra, se pueden escribir expresiones equivalentes que no son idénticas

$$P_1 \wedge P_2 \equiv \neg (\neg P_1 \vee \neg P_2)$$

$$\forall t \in r(P_1(t)) \equiv \neg \exists t \in r(\neg P_1(t))$$

$$P_1 \Rightarrow P_2 \equiv \neg P_1 \vee P_2$$



# Seguridad de expresiones

Las expresiones del cálculo relacional de tuplas pueden generar relaciones infinitas, por ejemplo:

$\{ t / \sim ( t \in \text{préstamo} ) \}$  infinito

(todas las tuplas que no están definidas en la tabla, pero que se pueden formar a partir del dominio de los atributos).

**Definición:** Una expresión de la forma  $\{t|P(t)\}$  es segura, si todos los valores que aparecen en el resultado son valores de  $\text{Dom}(P)$ . Donde  $\text{Dom}(P)$  es el conjunto de todos los valores que  $P$  hace referencia

# Cálculo Relacional de Dominios

- Se utilizan variables de dominio que toman valores del dominio de un atributo (en lugar de tuplas completas)

## Definición formal:

Las expresiones son de la forma:

$\{ \langle x_1, \dots, x_n \rangle / P(x_1, \dots, x_n) \};$

Donde  $x_1, \dots, x_n$  variables de dominio y  $P$  representa una fórmula compuestas por átomos.

- Átomos definidos como el CRT
- Fórmulas definidas como el CRT

# Ejemplos de CRD

- Averiguar el nombre de la sucursal, el numero de préstamo , numero de cliente y el importe de los prestamos superiores a \$200000:

$$\{ \langle np, ns, i, nc \rangle \mid \langle np, ns, i, nc \rangle \in \text{préstamo} \wedge i > 200000 \}$$

- Averiguar los números de préstamo de los prestamos superiores a \$200000:

$$\{ \langle np \rangle \mid \exists ns, i, nc (\langle np, ns, i, nc \rangle \in \text{préstamo} \wedge i > 200000) \}$$

- Averiguar el nombre de los clientes que tienen concedido un préstamo en la sucursal de Córdoba y dar el importe también:

$$\{ \langle n, i \rangle \mid \exists nc, a, d (\langle nc, a, n, d \rangle \in \text{cliente} \wedge \exists np, ns (\langle np, ns, i, nc \rangle \in \text{préstamo} \wedge ns = \text{“Córdoba”}) ) \}$$