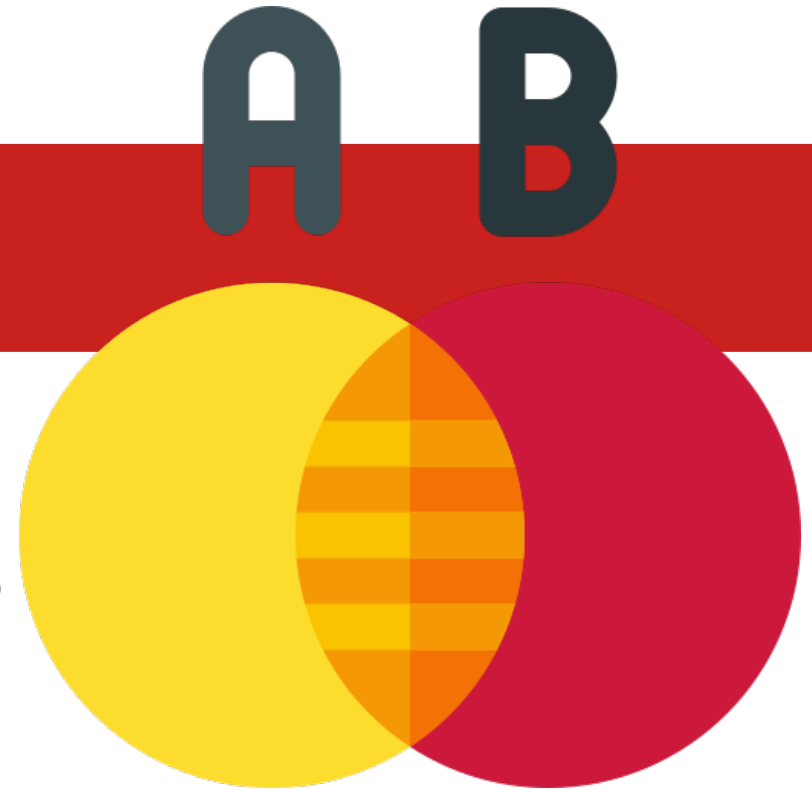


Bases de Datos I

Unidad VI

Lenguajes de consulta en el modelo relacional y el álgebra relacional



¿Qué es una consulta?

- **Una consulta intenta responder una pregunta sobre un hecho en particular, que puede o no existir en la base de datos.**
- **Por ejemplo, en una base de datos que contiene datos acerca de automotores:**
 - ¿Cuales son los automotores de marca Ford, que se dieron de alta en el año 2014?

Preguntas y respuestas

¿Cómo preguntamos?

- De igual forma que para darle instrucciones a una computadora, codificamos en un lenguaje de programación.
- Para realizar una consulta a nuestra base de datos, debemos codificarla en un lenguaje de consulta.

¿Cómo se nos responde?

- El resultado de ejecutar una consulta en una base de datos relacional es, ni más ni menos, que una relación.

Lenguajes de consulta

- **Dentro del modelo relacional existen varios lenguajes de consulta.**
- **Podemos categorizarlos utilizando distintos criterios:**
 - **Por la forma en que operan**
 - **Por su naturaleza**

Por la forma en que operan

- **Procedimentales o procedurales**
 - Requieren que se detalle la forma en que se debe obtener la respuesta (Cómo).
Por ejemplo, el álgebra relacional.
- **No procedimentales o no procedurales**
 - Se debe indicar que se busca, pero no como hacerlo (Qué).
Por ejemplo, el cálculo relacional, tanto de tuplas como de dominios.

Por su naturaleza

- **Formales o puros**
 - **Son lenguajes teóricos, basados en fundamentos matemáticos, raramente utilizados en sistemas para realizar consultas en SGBDs.**
Por ejemplo, el álgebra y el cálculo relacional.
- **De implementación**
 - **Son lenguajes utilizados para realizar consultas en SGBD.**
Generalmente están basados en lenguajes formales, pero ofrecen extensiones para brindar mayor simplicidad o potencia.
Por ejemplo, SQL.

¿Que vamos a hacer nosotros?

- **En la primera parte de los lenguajes de consulta, vamos a ver los lenguajes formales del modelo relacional.**
 - **Vamos a ver y practicar con álgebra relacional.**
 - **Vamos a mencionar el cálculo relacional.**
- **En la segunda parte, vamos a trabajar con bases de datos reales.**
 - **Vamos a trabajar con el lenguaje de manipulación y consulta más utilizado: SQL.**

Compatibilidad

- **Es posible redefinir cualquier consulta expresada mediante álgebra relacional con cálculos relaciones de tuplas o dominios.**
- **Además, cualquier expresión segura de estos cálculos puede expresarse también mediante álgebra relacional.**
- **Nos ayuda a abstraernos del SGBD relacional.**

¿Qué es el Álgebra Relacional?

- **Es un conjunto de operaciones que describen paso a paso (procedimental) la forma de transformar una serie de relaciones en otras, para así obtener la respuesta de una consulta.**
- **Cada operación de álgebra relacional toma una serie de relaciones de entrada y genera una relación de salida.**

¿Cómo funciona el Álgebra Relacional?

- **Estamos acostumbrados en aritmética a operar con números.**
- **Existe un conjunto de operadores u operaciones, que podemos combinar y formar expresiones.**
- **Cada operador u operación aplicada sobre uno o más números, da como resultados un número.**
 - 1) $5 + (10 * 2) - 12 =$
 - 2) $5 + 20 - 12 =$
 - 3) $25 - 12 =$
 - 4) 13
- **En álgebra, hemos trabajado de igual forma con vectores y matrices.**

¿Cómo funciona el Álgebra Relacional?

- **El álgebra relacional es similar.**
- **Existe un conjunto de operadores u operaciones, unarias y binarias.**
- **Operan sobre relaciones (conjuntos de tuplas).**
- **El resultado de aplicar un operador u operación, es una relación.**
- **Pueden combinarse para formar expresiones y utilizar ().**

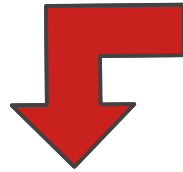
Operaciones básicas

- **Operaciones unarias:**
 - **Proyección (π)**
 - **Selección (σ)**
 - **Renombramiento (ρ)**
- **Operaciones binarias:**
 - **Producto cartesiano (\times)**
 - **Unión (\cup)**
 - **Diferencia ($-$)**

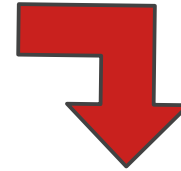
Proyección π (pi)

- Sea $R(a_1, a_2, \dots a_n)$ una relación de grado n .
- Definida como: $\pi_{a_1, a_2, \dots a_n}(R)$
- La proyección $\pi_{a_1, a_2, \dots a_k}(R)$, produce como resultado una relación de grado k , a partir de la relación R , eliminando los atributos no proyectados.
- Si el resultado contiene tuplas duplicadas, estas se suprimen.

Proyección π (π): Ejemplo



VEHÍCULO			
patente	marca	modelo	color
MBO34L	Ford	Ka	verde
LDA75K	Toyota	Corolla XL	blanco
ADA89A	Fiat	Siena	gris
LBF78G	Toyota	Corolla XL	blanco
XSA67D	Ford	Ka	rojo



π patente, marca (VEHÍCULO)

VEHÍCULO	
patente	marca
MBO34L	Ford
LDA75K	Toyota
ADA89A	Fiat
LBF78G	Toyota
XSA67D	Ford

π marca, modelo, color (VEHÍCULO)

VEHÍCULO		
marca	modelo	color
Ford	Ka	verde
Toyota	Corolla XL	blanco
Fiat	Siena	gris
Toyota	Corolla XL	blanco
Ford	Ka	rojo

Selección σ (sigma)

- Sea $R(a_1, a_2, \dots a_n)$ una relación de grado n .
- Definida como: $\sigma_p(R) = \{t / t \in R \wedge p(t)\}$
- La selección σ predicado(R), produce como resultado una relación de grado n , con las tuplas de R que cumplen con el predicado.
- El predicado es una expresión lógica que puede contener los operadores $=, <, >, \neq, \leq, \geq, \neg, \wedge, \vee$.

Selección σ (sigma): Ejemplo

- **Dada la relación**

persona(apellido, nombre)

- **Y su extensión**

**{(perez, jorge),(perez, nadia),
(martinez, maria), (martinez, alberto)}**

- **Un ejemplo de proyección:**

σ apellido='perez'(persona)

El resultado es {(perez, jorge),(perez, nadia)}

Selección σ (sigma): Ejemplo

VEHÍCULO		
marca	modelo	color
Ford	Ka	verde
Toyota	Corolla XL	blanco
Fiat	Siena	gris
Toyota	Corolla XL	blanco
Ford	Ka	rojo

$\sigma_{\text{marca}='Ford'}$ VEHÍCULO



VEHÍCULO		
marca	modelo	color
Ford	Ka	verde
Ford	Ka	rojo

VEHÍCULO		
marca	modelo	color
Ford	Ka	verde
Toyota	Corolla XL	blanco
Fiat	Siena	gris
Toyota	Corolla XL	blanco
Ford	Ka	rojo

$\sigma_{\text{marca}='Ford' \wedge \text{color}='rojo'}$ VEHÍCULO



VEHÍCULO		
marca	modelo	color
Ford	Ka	rojo

Componiendo expresiones

- La idea es ir componiendo operaciones a los resultados previos, formando expresiones, que respondan la pregunta planteada.
- La evaluación es de derecha a izquierda, pudiendo agrupar con paréntesis.

Ejemplo: obtener el nombre de la personas de apellido 'Perez':

Π nombre(σ apellido='Perez')(PERSONA))

Ejemplo: esquema

- **cliente(dni, nombre, calle, ciudad)**
- **sucursal(nombre, ciudad)**
- **cuenta(numero, sucursal, saldo)**
- **prestamo(numero, sucursal, monto)**
- **cliente-cuenta(cliente, cuenta)**
- **cliente-prestamo(cliente, prestamo)**

Ejemplo: consultas

- **Obtener todos los préstamos de más de \$120.000:**
- **Obtener el número de préstamo, para todos los préstamos de un monto superior a \$150.000:**

Ejemplo: consultas

- **Obtener todos los préstamos de más de \$120.000:**
 $\sigma \text{ monto} > 120000(\text{prestamo})$
- **Obtener el número de préstamo, para todos los préstamos de un monto superior a \$150.000:**
 $\Pi \text{ numero } (\sigma \text{ monto} > 150000 (\text{prestamo}))$

Renombramiento ρ (rho)

- **El operador de renombramiento, puede utilizarse para renombrar dos elementos:**
 - **Renombramiento de relaciones**
 - **Renombramiento de atributos**

Renombramiento ρ (rho) de relaciones

- Sea $R(a_1, a_2, \dots a_n)$ una relación de grado n .

- Definida como: $\rho_x(R)$

- Para renombrar la relación R se utiliza ρ nuevo (R), lo que produce como resultado una relación como R (tanto en su esquema como en su contenido) pero de nombre “nuevo”.

Renombramiento ρ (rho) de atributos

- Sea $R(a_1, a_2, \dots a_n)$ una relación de grado n .
- Definida como: $\rho_{x_1 \leftarrow a_1, x_2 \leftarrow a_2, \dots x_n \leftarrow a_n}(R)$
- Para renombrar atributos de la relación R se utiliza $\rho_{n_1 \leftarrow a_1}(R)$, lo que produce como resultado una relación como R (tanto en su esquema como en su contenido) pero con el atributo a_1 renombrado a n_1 .

Ejemplo: Renombramiento ρ (rho)

- **Dada la relación**

persona(apellido, nombre)

- **Renombrando la relación**

ρ empleado(persona)

El resultado es empleado(apellido, nombre)

- **Renombrando atributos:**

ρ gracia \leftarrow nombre(persona)

El resultado es persona(apellido, gracia)

Producto cartesiano \times

- Sea $R(a_1, a_2, \dots a_n)$ una relación de grado n , y $Q(b_1, b_2, \dots b_m)$ una relación de grado m .
- Definido como: $R \times Q = \{ts / t \in R \wedge s \in Q\}$
- El producto cartesiano $R \times Q$, es una relación de grado $n+m$ formada por todas las posibles tuplas en las que los m primeros elementos surgen de una tupla de R y los n últimos de una tupla de Q .

Ejemplo: Producto cartesiano \times

- **Dada las relaciones**

persona(documento)

mascota(nombre)

- **Y sus extensiones**

persona = {(27024277), (25743553)}

mascota = {(bobi), (mancha)}

- **persona \times mascota:**

- **El esquema resultante resultado(documento, nombre)**
- **Y el contenido {(27024277, bobi), (27024277, mancha), (25743553, bobi), (25743553, mancha)}**

Ejemplo: Producto cartesiano ×

R	
patente	marca
ADA89A	Fiat
LBF78G	Toyota
XSA67D	Ford

×



Q		
marca	modelo	color
Fiat	Siena	gris
Toyota	Corolla XL	blanco
Ford	Ka	rojo

VEHÍCULO				
patente	marcaR	marcaQ	modelo	color
ADA89A	Fiat	Fiat	Siena	gris
ADA89A	Fiat	Toyota	Corolla XL	blanco
ADA89A	Fiat	Ford	Ka	rojo
LBF78G	Toyota	Fiat	Siena	gris
LBF78G	Toyota	Toyota	Corolla XL	blanco
LBF78G	Toyota	Ford	Ka	rojo
XSA67D	Ford	Fiat	Siena	gris
XSA67D	Ford	Toyota	Corolla XL	blanco
XSA67D	Ford	Ford	Ka	rojo

Utilidad del producto cartesiano \times

- El operador productor cartesiano (\times), en combinación con el operador de selección (σ) nos permite realizar reuniones de relaciones.
- Las reuniones nos permiten combinar la extensión de dos relaciones de forma lateral.
- Es principalmente utilizada para combinar una relación con una clave foránea, y la relación a la que referencia.

Ejemplo: uso del producto cartesiano para reunión

- Dada las relaciones

persona(documento, nombre)

mascota(apodo, dueño)

- Y sus extensiones

persona = {(1, pepe), (2, pedro), (3, maria)}

mascota = {(bobi, 1), (laica, 2)}

- $\sigma_{\text{documento=dueño}}(\text{persona} \times \text{mascota})$:

Y el resultado {(1, pepe, bobi, 1), (2, pedro, laica, 2)}

Ejercicios

- **Obtener el nombre de los clientes que viven en la misma ciudad que Gómez:**
- **Obtener el dni de los clientes que tienen una cuenta en alguna sucursal de la ciudad de Río Grande:**

Ejercicios

- **Obtener el nombre de los clientes que viven en la misma ciudad que Gómez:**

$\Pi \text{ cliente.nombre}(\sigma \text{ cliente.ciudad} = \text{DireccionGomez.ciudad}(\text{cliente} \times \rho \text{ DireccionGomez} (\Pi \text{ ciudad}(\sigma \text{ nombre} = \text{'Gomez'}(\text{cliente}))))))$

- **Obtener el dni de los clientes que tienen una cuenta en alguna sucursal de la ciudad de Río Grande:**

$\Pi \text{ cliente}(\sigma \text{ cliente-cuenta.cuenta} = \text{R2.numero}(\text{cliente-cuenta} \times \rho \text{ R2}((\Pi \text{ numero}(\sigma \text{ cuenta.sucursal} = \text{R1.nombre}(\text{cuenta} \times \rho \text{ R1}((\Pi \text{ nombre}(\sigma \text{ nombre} = \text{'Río Grande'}(\text{sucursal}))))))))))$

Unión \cup

- Sea $R(a_1, a_2, \dots a_n)$ una relación de grado n , y $Q(b_1, b_2, \dots b_n)$ una relación de grado n .
- Definida como:
$$R \cup Q = \{t / t \in R \vee t \in Q\}$$

La unión de $R \cup Q$, es una relación con las tuplas de R y las tuplas de Q .

- **Para que la unión sea posible las relaciones deben ser compatibles.**
- **Se debe tener en cuenta que si el resultado de la unión contiene tuplas duplicadas, estas se suprimen.**

Ejemplo: Unión \cup

- Dada las relaciones

persona(apellido, nombre)

empleado(apellido, nombre)

- Y sus extensiones

persona = {(perez, jorge), (perez, nadia)}

empleado = {(martinez, maria), (perez, nadia)}

- **persona \cup empleado:**

El resultado es {(perez, jorge), (perez, nadia), (martinez, maria)}

Ejemplo: Unión \cup

EMPLEADO	
CÉDULA	NOMBRE
9.644.667	Pedro Pérez
10.133.212	Gabriel Mendoza
11.332.334	Luís Colina
12.562.884	Andrés Rojas
20.126.112	Gilberto Zapata



PERSONA	
CÉDULA	NOMBRE
8.347.223	Héctor Redondo
9.644.667	Pedro Pérez
10.133.212	Gabriel Mendoza
11.332.334	Luís Colina
12.123.231	Diego Dávila

CÉDULA	NOMBRE
8.347.223	Héctor Redondo
9.644.667	Pedro Pérez
10.133.212	Gabriel Mendoza
11.332.334	Luís Colina
12.123.231	Diego Dávila
12.562.884	Andrés Rojas
20.126.112	Gilberto Zapata

Diferencia -

- Sea $R(a_1, a_2, \dots a_n)$ una relación de grado n , y $Q(b_1, b_2, \dots b_n)$ una relación de grado n .
- Definida como:
$$R - Q = \{t / t \in R \wedge t \notin Q\}$$
- La diferencia de $R - Q$, es el conjunto de tuplas de R que no pertenecen a Q .

Para que la diferencia sea posible las relaciones deben ser compatibles.

Ejemplo: Diferencia -

- Dada las relaciones

persona(apellido, nombre)

empleado(apellido, nombre)

- Y sus extensiones

persona = {(perez, jorge), (perez, nadia)}

empleado = {(martinez, maria), (perez, nadia)}

- **persona - empleado:**

El resultado es {(perez, jorge)}

Ejercicios

- **Mostrar el dni de los clientes que tienen, una cuenta, un préstamo o ambos productos bancarios:**
- **Obtener el dni de los clientes que no tienen préstamo en ninguna sucursal del Banco:**

Solución

- **Mostrar el dni de los clientes que tienen, una cuenta, un préstamo o ambos productos bancarios:**

$\Pi \text{ cliente}(\text{cliente-prestamo}) \cup \Pi \text{ cliente}(\text{cliente-cuenta})$

- **Obtener el dni de los clientes que no tienen préstamo en ninguna sucursal del Banco:**

$\Pi \text{ dni}(\text{cliente}) - \Pi \text{ cliente}(\text{cliente-prestamo})$

Operaciones adicionales

- **Estos operadores no añaden ninguna funcionalidad nueva al álgebra relacional, pero facilitan la construcción de consultas (pueden ser contruidos combinando los operadores básicos):**
 - **Asignación ($=$, \leftarrow)**
 - **Intersección (\cap)**
 - **División ($/$)**
 - **Reunión natural (\bowtie)**
 - **Reunión externa (\ltimes , \ltimesl , \ltimesr)**

Asignación =, \leftarrow

- Sea $R(a_1, a_2, \dots a_n)$ una relación de grado n .

- Definida como: $R \leftarrow Q \vee R = Q$

- La operación de asignación $A \leftarrow R$, produce como resultado una relación como R (tanto en su esquema como en su contenido) pero con el nombre A .
- Es utilizada para almacenar los resultados temporales de los cálculos en relaciones nombradas.

Ejemplo: Asignación =, \leftarrow

- **Se utiliza principalmente para descomponer una consulta compleja, en varias pequeñas.**

$\Pi \text{ saldo}(\text{cuenta}) - \Pi \text{ cuenta.saldo}(\sigma \text{ cuenta.saldo} < d.\text{saldo} (\text{cuenta} \times \rho d (\text{cuenta})))$

- **Obtener el valor del mayor saldo entre las cuentas del banco:**

$d \leftarrow \text{cuenta}$

$r1 \leftarrow \sigma \text{ cuenta.saldo} < d.\text{saldo} (\text{cuenta} \times d)$

$\Pi \text{ saldo}(\text{cuenta}) - \Pi \text{ saldo}(r1)$

Intersección \cap

- Sea $R(a_1, a_2, \dots a_n)$ una relación de grado n , y $Q(b_1, b_2, \dots b_n)$ una relación de grado n .
- Definida como:
$$R \cap Q = \{t / t \in R \wedge t \in Q\}$$
- La intersección de $R \cap Q$, es una relación con las tuplas que pertenecen tanto a R como a Q .
- Para que la intersección sea posible las relaciones deben ser compatibles.
- Se debe tener en cuenta que si el resultado de la intersección contiene tuplas duplicadas, estas se suprimen.

Ejemplo: Intersección \cap

- **Dada las relaciones**

persona(apellido, nombre)

empleado(apellido, nombre)

- **Y sus extensiones**

persona = {(perez, jorge), (perez, nadia)}

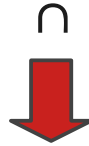
empleado = {(martinez, maria), (perez, nadia)}

- **persona \cap empleado:**

El resultado es {(perez, nadia)}

Ejemplo: Intersección \cap

EMPLEADO	
CÉDULA	NOMBRE
9.644.667	Pedro Pérez
10.133.212	Gabriel Mendoza
11.332.334	Luís Colina
12.562.884	Andrés Rojas
20.126.112	Gilberto Zapata



PERSONA	
CÉDULA	NOMBRE
8.347.223	Héctor Redondo
9.644.667	Pedro Pérez
10.133.212	Gabriel Mendoza
11.332.334	Luís Colina
12.123.231	Diego Dávila

CÉDULA	NOMBRE
9.644.667	Pedro Pérez
10.133.212	Gabriel Mendoza
11.332.334	Luís Colina

División /

- Dadas dos relaciones $R(x, y)$ y $S(y)$ donde el dominio de y en R y S , es el mismo.
- Definida como: $R \div S = \{ t / t \in \pi_{R'-S'}(R) \wedge \forall u \in S, tu \in R \}$
- El operador división R / S retorna todos los distintos valores de x tales que para todo valor “ y ” en S existe una tupla (x,y) en R .
- Esta operación se utiliza para encontrar la respuesta de consultas que incluyen la frase “para todos”.

Ejemplo: División /

- **Dada las relaciones**
 alumno(nombre, materia)
 materia(materia)
- **Y sus extensiones**
 - **alumno = {(jorge,BDI),(jorge,BDII),(nadia,BDI)}**
 materia = {(BDI), (BDII)}
- **alumno / materia:**
 El resultado es {(jorge)}

Ejemplo: División /

R	
DNI	POYECTO
12.345.678	PROY1
12.345.678	PROY2
66.688.444	PROY3
45.345.345	PROY1
45.345.345	PROY2
33.444.555	PROY2
33.444.555	PROY3
33.444.555	PROY10
33.444.555	PROY20



S
PROYECTO
PROY1
PROY2

DNI
12.345.678
45.345.345

Reunión natural \bowtie

- Sea A y B relaciones con esquemas A' y B' respectivamente, entonces, $A \bowtie B$ devuelve una relación con esquema $A' \cup B'$ teniendo en cuenta que:
 - Dado cada par de tuplas t_a de A y t_b de B .
 - Si t_a y t_b tienen los mismos valores en cada atributo de $A' \cap B'$, se añade la tupla t como resultado, donde:
 - t tiene los mismos valores que t_a en A
 - t tiene los mismos valores que t_b en B

Reunión natural ⋈

- En términos de los operadores del álgebra relacional el producto natural se define como:

$$R \bowtie S = \pi_{a_1, a_2, \dots, a_n} \left(\sigma_{\theta} (R \times S) \right)$$

- Las columnas por la que se hace la reunión de ambas tablas deben tener el mismo nombre y ser del mismo tipo.
- Si no hay atributos en común entre las relaciones, la reunión natural es igual a un producto cartesiano.

Ejemplo: Reunión natural \bowtie

- Dada las relaciones

persona(documento, nombre)

mascota(apodo, documento)

- Y sus extensiones

persona = {(1, jorge), (2, nadia), (3, pepe)}

mascota = {(boby, 1), (toy, 2), (laica, nulo)}

- **persona \bowtie mascota:**

- El resultado es **{(1, jorge, boby), (2, nadia, toy)}**

Ejemplo: Reunión natural ☒

MARCA	
CodigoMarca	NombreMarca
T1	Toyota
F1	Ford
F2	Fiat

VEHÍCULO			
Patente	CodigoMarca	Modelo	Color
MNJ321	T1	Corolla	Blanco
KLP654	T1	Etios	Rojo
DSR159	F1	Ka	Azul
EER133	F2	Siena	Gris

Patente	CodigoMarca	NombreMarca	Modelo	Color
MNJ321	T1	Toyota	Corolla	Blanco
KLP654	T1	Toyota	Etios	Rojo
DSR159	F1	Ford	Ka	Azul
EER133	F2	Fiat	Siena	Gris

Ejemplo: Reunión natural ☒

PROFESOR		
DNI	NOMBRE	DPTO
6.274.445	José Méndez	1
7.422.114	Juan Zapata	1
8.347.223	Héctor Redondo	2
9.644.667	Pedro Pérez	2
11.332.334	Luís Colina	
12.123.231	Diego Dávila	3



DEPARTAMENTO		
CÓDIGO	NOMBRE_D	DNI
1	Computación	6.241.445
2	Investigación	
3	Control	12.123.231

PROFESOR				
DNI	NOMBRE	DPTO	CÓDIGO	NOMBRE_D
6.274.445	José Méndez	1	1	Computación
12.123.231	Diego Dávila	3	3	Control

Reunión condicional \bowtie_{θ}

- Una reunión theta (θ -Join) o producto condicional o producto con condición de dos relaciones es equivalente a:

$$R \bowtie_{\theta} S = \sigma_{\theta} (R \times S)$$

- Donde la condición θ es libre. Y si θ es una igualdad se denomina EquiJoin.

Ejemplo: Reunión condicional \bowtie_{θ}

- Dada las relaciones

docentes(documento, nombre, apellido, localidad)

ciudades(codigo_postal, nombre)

- Y sus extensiones

**docentes = {(11222333, Sandra, Guzmán, 9410),
(22333444, Roberto, Bustos, 9420)}**

ciudades = {(9410, Ushuaia), (9420, Río Grande)}

- **docentes \bowtie_{θ} ciudades:**

- El resultado es **{(11222333, Sandra, Guzmán, 9410, Ushuaia),
(22333444, Roberto, Bustos, 9420, Río Grande)}**

Ejemplo: Reunión condicional ☒

PROFESOR		
DNI	NOMBRE	DPTO
6.274.445	José Méndez	1
7.422.114	Juan Zapata	1
8.347.223	Héctor Redondo	2
9.644.667	Pedro Pérez	2
11.332.334	Luís Colina	
12.123.231	Diego Dávila	3



DEPARTAMENTO		
CÓDIGO	NOMBRE_D	DNI
1	Computación	6.241.445
2	Investigación	
3	Control	12.123.231

☒ PRFESOR.DPTO = DEPARTAMENTO.CODIGO

PROFESOR			
DNI	NOMBRE	CÓDIGO	NOMBRE_D
6.274.445	José Méndez	1	Computación
7.422.114	Juan Zapata	1	Computación
8.347.223	Héctor Redondo	2	Investigación
9.644.667	Pedro Pérez	2	Investigación
12.123.231	Diego Dávila	3	Control

Reunión externa

- **La reunión externa es una forma ampliada de la reunión natural que evita la pérdida de información. Esta operación muestra en la relación resultado las tuplas de una relación que no tienen contraparte en la otra con el valor nulo en los demás atributos. Existe tres formas de realizar reunión externa:**
 - **Reunión externa izquierda (\bowtie)**
 - **Reunión externa derecha (\ltimes)**
 - **Reunión externa completa (\ltimes)**

Reunión externa izquierda \bowtie

- **La reunión externa izquierda, es una forma ampliada de la reunión natural.**
- **Las tuplas de la relación del lado izquierdo, forman parte del conjunto resultado, por más que no estén vinculadas con ninguna tupla del lado derecho.**
- **Los atributos del lado derecho se completan con valores nulos.**

Ejemplo: Reunión externa izquierda ⋈

- Dada las relaciones

persona(documento, nombre)

mascota(apodo, documento)

- Y sus extensiones

persona = {(1, jorge), (2, nadia), (3, pepe)}

mascota = {(boby, 1), (toy, 2), (laica, *nulo*)}

- **persona ⋈ mascota:**

- El resultado es **{(1, jorge, boby), (2, nadia, toy), (3, pepe, *nulo*)}**

Ejemplo: Reunión externa izquierda ⌘

PROFESOR		
DNI	NOMBRE	CÓDIGO_DPTO
6.274.445	José Méndez	1
7.422.114	Juan Zapata	1
8.347.223	Héctor Redondo	2
9.644.667	Pedro Pérez	2
11.332.334	Luís Colina	NULO
12.123.231	Diego Dávila	3



DEPARTAMENTO		
CÓDIGO	NOMBRE_D	DNI
1	Computación	6.241.445
2	Investigación	NULO
3	Control	12.123.231

PROFESOR					
DNI	NOMBRE	CÓDIGO_DPTO	CÓDIGO	NOMBRE_D	DNI
6.274.445	José Méndez	1	1	Computación	6.274.445
7.422.114	Juan Zapata	1	NULO	NULO	NULO
8.347.223	Héctor Redondo	2	NULO	NULO	NULO
9.644.667	Pedro Pérez	2	NULO	NULO	NULO
11.332.334	Luís Colina		NULO	NULO	NULO
12.123.231	Diego Dávila	3	3	Control	12.123.231

Reunión externa derecha \bowtie

- **La reunión externa derecha, es una forma ampliada de la reunión natural.**
- **Las tuplas de la relación del lado derecho, forman parte del conjunto resultado, por más que no estén vinculadas con ninguna tupla del lado izquierdo.**
- **Los atributos del lado izquierdo se completan con valores nulos.**

Ejemplo: Reunión externa derecha \bowtie

- Dada las relaciones

persona(documento, nombre)

mascota(apodo, documento)

- Y sus extensiones

persona = {(1, jorge), (2, nadia), (3, pepe)}

mascota = {(boby, 1), (toy, 2), (laica, nulo)}

- **persona \bowtie mascota:**

- El resultado es **{(1, jorge, boby), (2, nadia, toy), (nulo, nulo, laica)}**

Ejemplo: Reunión externa derecha ⌋

PROFESOR		
DNI	NOMBRE	CÓDIGO_DPTO
6.274.445	José Méndez	1
7.422.114	Juan Zapata	1
8.347.223	Héctor Redondo	2
9.644.667	Pedro Pérez	2
11.332.334	Luís Colina	NULO
12.123.231	Diego Dávila	3



DEPARTAMENTO		
CÓDIGO	NOMBRE_D	DNI
1	Computación	6.241.445
2	Investigación	NULO
3	Control	12.123.231

PROFESOR					
DNI	NOMBRE	CÓDIGO_DPTO	CÓDIGO	NOMBRE_D	DNI
6.274.445	José Méndez	1	1	Computación	6.274.445
NULO	NULO	NULO	NULO	Investigación	NULO
12.123.231	Diego Dávila	3	3	Control	12.123.231

Reunión externa completa \bowtie

- **La reunión externa completa, es una forma ampliada de la reunión natural.**
- **Las tuplas de ambas relaciones, forman parte del conjunto resultado, por más que no estén vinculadas con ninguna tupla de la otra relación.**
- **Los atributos de aquellas tuplas que no estén vinculadas, se completan con valores nulos.**

Ejemplo: Reunión externa derecha \bowtie

- Dada las relaciones

persona(documento, nombre)

mascota(apodo, documento)

- Y sus extensiones

persona = {(1, jorge), (2, nadia), (3, pepe)}

mascota = {(boby, 1), (toy, 2), (laica, nulo)}

- **persona \bowtie mascota:**

- El resultado es **{(1, jorge, boby), (2, nadia, toy), (3, pepe, nulo), (nulo, nulo, laica)}**

Ejemplo: Reunión externa derecha ⌘

DEPARTAMENTO		
CÓDIGO	NOMBRE_D	DNI
1	Computación	6.241.445
2	Investigación	NULO
3	Control	12.123.231



PROFESOR		
DNI	NOMBRE	CÓDIGO_DPTO
6.274.445	José Méndez	1
7.422.114	Juan Zapata	1
8.347.223	Héctor Redondo	2
9.644.667	Pedro Pérez	2
11.332.334	Luís Colina	NULO
12.123.231	Diego Dávila	3

PROFESOR					
CÓDIGO	NOMBRE_D	DNI	DNI	NOMBRE	CÓDIGO_DPTO
1	Computación	6.274.445	6.274.445	José Méndez	1
NULO	NULO	NULO	7.422.114	Juan Zapata	1
NULO	NULO	NULO	8.347.223	Héctor Redondo	2
NULO	NULO	NULO	9.644.667	Pedro Pérez	2
NULO	NULO	NULO	11.332.334	Luís Colina	
2	Investigación	NULO	NULO	NULO	NULO
3	Control	12.123.231	12.123.231	Diego Dávila	3

Ejemplo: Esquema

- Utilizaremos el siguiente esquema de base de datos para ejercitar:

cliente(dni, nombre, calle, ciudad)

sucursal(nombre, ciudad)

cuenta(número, sucursal, saldo)

préstamo(número, sucursal, monto)

cliente-cuenta(cliente, cuenta)

cliente-préstamo(cliente, prestamo)

Ejemplo: Ejercicios

- **Clientes que tienen una cuenta en (por lo menos) las sucursales “Ush01” y “RG02”:**

Ejemplo: Ejercicios

- **Clientes que tienen una cuenta en (por lo menos) las sucursales “Ush01” y “RG02”:**

$R1 \leftarrow \rho \text{ cuenta} \leftarrow \text{numero}(\text{cuenta})$

$R2 \leftarrow \rho \text{ cliente} \leftarrow \text{dni}, \text{ncliente} \leftarrow \text{nombre}(\text{cliente})$

$R3 \leftarrow \sigma \text{ nombre} = \text{“Ush01”}(\text{cliente-cuenta} \bowtie R1)$

$R4 \leftarrow \sigma \text{ nombre} = \text{“RG02”}(\text{cliente-cuenta} \bowtie R1)$

$R5 \leftarrow \Pi \text{ cliente}(R3 \bowtie R2)$

$R6 \leftarrow \Pi \text{ cliente}(R4 \bowtie R2)$

$\text{Resultado} \leftarrow R5 \cap R6$

Ejemplo: Ejercicios

- **Clientes con cuentas en todas las sucursales de la ciudad de Ushuaia:**
- **Encontrar todos los clientes que tienen un préstamo con el banco y mostrar la ciudad de residencia:**

Ejemplo: Ejercicios

- **Clientes con cuentas en todas las sucursales de la ciudad de Ushuaia:**
 - $R1 \leftarrow \rho \text{ cuenta} \leftarrow \text{numero}, \text{nombre} \leftarrow \text{sucursal}(\text{cuenta})$
 $R2 \leftarrow \Pi \text{ cliente, nombre}(\text{cliente-cuenta} \bowtie R1)$
 $R3 \leftarrow \Pi \text{ nombre}(\sigma \text{ ciudad} = \text{"Ush"}(\text{sucursal}))$
 $\text{Resultado} \leftarrow R2 / R3$
- **Encontrar todos los clientes que tienen un préstamo con el banco y mostrar la ciudad de residencia:**
 - $R1 \leftarrow \rho \text{ cliente} \leftarrow \text{dni}(\text{cliente})$
 $\Pi \text{ nombre, ciudad}(\text{cliente-prestamo} \bowtie R1)$

Ejemplo: Ejercicios

- **Mostrar el capital y el nombre de todas las sucursales que tienen clientes con cuentas y viven en Trelew.**

Ejemplo: Ejercicios

- **Mostrar el capital y el nombre de todas las sucursales que tienen clientes con cuentas y viven en Trelew.**

$R1 \leftarrow \Pi \text{ dni}(\sigma \text{ ciudad} = \text{"Trelew"}(\text{cliente}))$

$R2 \leftarrow \rho \text{ cliente} \leftarrow \text{dni}, \text{ncliente} \leftarrow \text{nombre}(R1)$

$R3 \leftarrow \rho \text{ sucursal} \leftarrow \text{numero}(\text{sucursal})$

$R4 \leftarrow R2 \bowtie \text{cliente-cuenta} \bowtie R3$

$\text{Resultados} \leftarrow \Pi \text{ nombre, saldo}(R4)$

¿Cómo modificamos el conjunto de tuplas?

- **Hasta ahora se ha centrado la atención en la consulta de la base de datos.**
- **Vamos a abordar la manera de añadir, modificar y eliminar datos de la base de datos.**
- **Las modificaciones de la base de datos se expresan mediante la operación asignación.**

Inserción

- Para insertar datos en una relación hay que especificar la tupla que se va a insertar o escribir una consulta cuyo resultado sea el conjunto de tuplas que se van a insertar.
- Evidentemente, el valor de los atributos de las tuplas insertadas debe ser miembro del dominio de cada atributo.
- De manera parecida, las tuplas insertadas deben tener el grado correcta.
- El álgebra relacional expresa las inserciones mediante:

$$R \leftarrow R \cup E$$

Inserción: Ejemplo

- **Dada la relación**
persona(apellido, nombre)
- **Y su extensión**
persona = {(perez, jorge), (perez, nadia)}
- **Realizamos la inserción:**
 - **persona \leftarrow persona \cup {(martinez, maria)}**

Modificación

- Se puede usar el operador proyección generalizada para llevar a cabo la modificación de algunos atributos de todas las tuplas de la relación:

$$R \leftarrow \Pi f_1, f_2, \dots, f_n(R)$$

- Si se desea seleccionar varias tuplas de r y actualizar sólo esas tuplas, se puede usar la expresión siguiente, donde P denota la condición de selección que escoge las tuplas que hay que actualizar:

$$R \leftarrow \Pi f_1, f_2, \dots, f_n(\sigma p(R)) \cup (R - \sigma p(R))$$

Modificación: Ejemplo

- Dada la relación

persona(apellido, nombre)

- Y su extensión

persona = {(perez, jorge), (perez, nadia)}

- Realizamos la modificación:

persona $\leftarrow \Pi$ f1(apellido), f2(nombre)(persona)

Eliminación

- Las solicitudes de borrado se expresan básicamente igual que las consultas.
- Sin embargo, en lugar de mostrar las tuplas al usuario, se eliminan de la base de datos las tuplas seleccionadas.
- Sólo se pueden borrar tuplas enteras; no se pueden borrar valores de atributos concretos.
- En el álgebra relacional los borrados se expresan mediante:

$$R \leftarrow R - E$$

Eliminación: Ejemplo

- **Dada la relación**
persona(apellido, nombre)
- **Y su extensión**
persona = {(perez, jorge), (perez, nadia)}
- **Realizamos la eliminación:**
 - **persona \leftarrow persona - {(perez, jorge)}**

Bibliografía

- **Elmasri R. y Navathe Sh. (2000). Sistemas de Bases de Datos. Conceptos Fundamentales. Segunda Edición. Addison Wesley Longman de México, S. A.**