

# ÁLGEBRA RELACIONAL DE BASE DE DATOS

# Álgebra Relacional

- Un *álgebra* es un sistema matemático constituido por
  - Operandos: objetos (valores o variables) desde los cuales nuevos objetos pueden ser contruidos.
  - Operadores: símbolos que denotan procedimientos para construir nuevos objetos desde objetos dados.
- El *álgebra relacional* es un álgebra en la cual
  - Sus operandos son relaciones (instancias) o variables que representan relaciones.
  - Sus operadores están diseñados para hacer la tareas más comunes que se necesitan para manipular relaciones en una base de datos.
- El resultado es que el álgebra relacional se puede utilizar como un lenguaje de consulta.
- En la práctica el álgebra relacional debe ser *extendida* para abarcar la mayor parte de las tareas reales que se hacen con los datos.
- Estudiaremos en detalle los operadores clásicos.
- El álgebra relacional es un lenguaje de consulta procedimental. En este tipo de lenguaje,el usuario da instrucciones al sistema para que realice una serie de procedimientos u operaciones en la base de datos para calcular un resultado final.

El SQL, es un lenguaje declarativo de acceso a base de datos relacionales que permite especificar diversos tipos de operaciones en ellas. Una de sus características es el manejo del álgebra y el cálculo relacional que permiten efectuar consultas con el fin de recuperar, de forma sencilla información de bases de datos, así como hacer cambios en ellas.

## Tablas

Estas serán las tablas que usaremos en la mayoría de ejemplos

### *Empleado*

<u>nombre</u>	<i>sueldo</i>	<i>cod_dept</i>	<i>fecha_ing</i>
Torres	\$ 1.200.000	A1	01/01/2004
Soto	\$ 500.000	A2	01/01/2003
Pérez	\$ 300.000	A2	01/10/2003
Figueroa	\$ 600.000	A1	01/03/2002
Salas	\$ 1.500.000	A1	01/01/2002
Ríos	\$ 2.000.000	A3	01/06/2002
Campos	\$ 800.000	A2	01/11/2003
Venegas	\$ 600.000	A1	01/06/2002
Carcamo	\$ 500.000	A2	01/04/2003
Gonzalez	\$ 2.000.000	A3	01/10/2002

### *Departamento*

<i>nombre</i>	<u><i>cod_dept</i></u>	<i>fecha_creac</i>
Informática	A1	01/03/2002
Marketing	A2	01/01/2002
Ventas	A3	01/01/2001
Recursos Humanos	A4	01/01/2003

## Selección

- Operador de selección  $\sigma$ , selecciona un subconjunto de las tuplas de una relación.
- Tuplas seleccionadas son las que satisfacen cierto predicado lógico  $P$ . El predicado puede depender de los atributos de la relación y de valores constantes.
- El operador  $\sigma$  toma una relación como argumento y el resultado es una nueva relación.
- Sintaxis:

$$\sigma P(r)$$

- Ejemplo: Seleccionar los datos del empleado Soto:

$$\sigma_{\text{nombre}=\text{Soto}}(\text{empleado})$$

<u>nombre</u>	<i>sueldo</i>	<i>cod_dept</i>	<i>fecha_ing</i>
Soto	\$ 500.000	A2	01/01/2003

- Ejemplo: Los datos de los empleados con sueldo  $\geq$  \$500.000 que ingresaron durante el 2003:

$$\sigma_{\text{sueldo} \geq 500000 \wedge (\text{fecha\_ing} \geq 01/01/2003 \wedge \text{fecha\_ing} \leq 31/12/2003)}(\text{empleado})$$

<u>nombre</u>	<i>sueldo</i>	<i>cod_dept</i>	<i>fecha_ing</i>
Soto	\$ 500.000	A2	01/01/2003
Campos	\$ 800.000	A2	01/11/2003
Carcamo	\$ 500.000	A2	01/04/2003

## Proyección

- Operador de proyección  $\pi$ , proyecta una relación sobre un subconjunto de sus atributos.
- El operador  $\pi$  toma una relación como argumento y el resultado es una nueva relación.
- Sintaxis:

$$\pi_A(r)$$

donde  $A$  representa el conjunto de atributos sobre los que la relación  $r$  se proyectará.

- Ejemplo: obtener los nombres de los distintos departamentos

$$\pi_{nombre}(departamento)$$

nombre
Informática
Marketing
Ventas
Recursos Humanos

- Obtener los montos de sueldo de los empleados:

$$\pi_{sueldo}(empleado)$$

sueldo
\$ 1.200.000
\$ 500.000
\$ 300.000
\$ 600.000
\$ 1.500.000
\$ 2.000.000
\$ 800.000

se eliminan los repetidos! una relación es un conjunto.

# Composición de Operaciones

- El resultado de cada operación es una nueva relación  $\Rightarrow$  se pueden aplicar operadores a los resultados de aplicaciones previas.

- Por ejemplo:

$$\pi_A(\sigma_P(r))$$

$$\sigma_P(\pi_A(r))$$

$$\sigma_{P_1}(\sigma_{P_2}(r))$$

- Ejemplo: Obtener los nombres de los empleados que ganan más de \$1.000.000.

$$\pi_{nombre}(\sigma_{sueldo > 1000000}(empleado))$$

<u>nombre</u>
Torres
Salas
Ríos
Gonzalez

- Ejemplo: Obtener el sueldo y la fecha de ingreso de Soto:

$$\pi_{sueldo, fech\_ing}(\sigma_{nombre=Soto}(empleado))$$

<u>sueldo</u>	<u>fecha_ing</u>
\$ 500.000	01/01/2003

# Unión

- Dado que las relaciones son conjuntos de tuplas, se pueden realizar las operaciones usuales de conjuntos como la unión.

- Sintaxis:

$$r_1 \cup r_2$$

- Se deben hacer ciertas restricciones para realizar la unión:
  - Ambas relaciones deben tener el mismo número de atributos.
  - El dominio del atributo  $i$ -ésimo de cada relación debe coincidir.
- Ejemplo: Obtener los nombres de los empleados que ganan más de \$1.500.000 o que trabajan en el departamento con código A1.

$$\pi_{nombre}(\sigma_{sueldo > 1500000}(empleado) \cup \sigma_{cod\_dept=A1}(empleado))$$

<u>nombre</u>
Torres
Figueroa
Salas
Ríos
Venegas
Gonzalez

# Diferencia

- También se puede usar la diferencia de conjuntos, las tuplas que están en una relación pero no en la otra.

- Sintaxis:

$$r_1 - r_2$$

- Para poder realizar la diferencia se deben cumplir las mismas restricciones que para la unión
- Ejemplo:

$$\pi_{nombre}(alumno) - \pi_{nombre}(\sigma_{carrera=Bioinformatica}(alumno))$$

Resulta en una relación que contiene a todos los nombres de los alumnos excepto de los alumnos de la carrera de Bioinformática.



## Producto Cartesiano

- Representa al producto cartesiano usual de conjuntos.
- Combina tuplas de cualesquiera dos (o más) relaciones, hace la combinación de todos con todos.
- Si las relaciones a operar tienen  $N$  y  $M$  tuplas de  $n$  y  $m$  componentes respectivamente, la relación resultante del producto cartesiano tiene  $N \times M$  tuplas de  $n+m$  componentes.
- Sintaxis:

$$r_1 \times r_2$$

- No hay restricciones a los dominios de las relaciones similares a las anteriores operaciones.
- Nos permite *reunir* datos de dos relaciones distintas.
- Cuidado con los nombres repetidos! se deben renombrar ciertos atributos para no tener problemas.

$$\text{departamento} \times \text{departamento}$$

<i>nombre</i>	<i>cd</i>	<i>fec creac</i>	<i>nombre</i>	<i>cd</i>	<i>fec creac</i>
Informática	A1	01/03/2002	Informática	A1	01/03/2002
Informática	A1	01/03/2002	Marketing	A2	01/01/2002
Informática	A1	01/03/2002	Ventas	A3	01/01/2001
Informática	A1	01/03/2002	Rec. Hum.	A4	01/01/2003
Marketing	A2	01/01/2002	Informática	A1	01/03/2002
Marketing	A2	01/01/2002	Marketing	A2	01/01/2002
Marketing	A2	01/01/2002	Ventas	A3	01/01/2001
Marketing	A2	01/01/2002	Rec. Hum.	A4	01/01/2003
Ventas	A3	01/01/2001	Informática	A1	01/03/2002
Ventas	A3	01/01/2001	Marketing	A2	01/01/2002
Ventas	A3	01/01/2001	Ventas	A3	01/01/2001
Ventas	A3	01/01/2001	Rec. Hum.	A4	01/01/2003
Rec. Hum.	A4	01/01/2003	Informática	A1	01/03/2002
Rec. Hum.	A4	01/01/2003	Marketing	A2	01/01/2002
Rec. Hum.	A4	01/01/2003	Ventas	A3	01/01/2001
Rec. Hum.	A4	01/01/2003	Rec. Hum.	A4	01/01/2003

# Producto Cartesiano: Ejemplos

- Obtener el nombre del departamento en el que Soto trabaja: Primero hacemos la unión cartesiana igualando un atributo

$$\sigma_{depto.cod\_depto=emp.cod\_depto}(empleado \times departamento)$$

Ahora podemos hacer la selección y proyección

$$\pi_{depto.nombre}(\sigma_{emp.nombre=Soto}(\sigma_{depto.cod\_depto=emp.cod\_depto}(empleado \times departamento)))$$

<u>departamento.nombre</u>
Marketing

# Renombre

- A veces necesitamos obtener información uniendo datos de la misma tabla.
- Por ejemplo, obtener los nombres de todos los empleados que ingresaron después que Soto a la empresa.
- Primer intento:

$$empleado \times (\sigma_{nombre=Soto}(empleado))$$

- ¿Cómo nos referimos a una u otra instancia de la tabla *empleado*?
- El operador de renombre  $\rho$  soluciona el problema.
- El operador  $\rho$  toma una relación y entrega la misma relación pero con otro nombre, podemos referirnos a distintas instancias de la misma relación.

- Sintaxis:

$$\rho_X(r)$$

- Volviendo al ejemplo, obtener los nombres de todos los empleados que ingresaron después que Soto a la empresa, segundo intento:

$$empleado \times (\sigma_{nombre=Soto}(\rho_{empleado2}(empleado)))$$

- Ahora podemos hacer:

$$\pi_{empleado.nombre}(\sigma_{empleado.fecha\_ing > empleado2.fecha\_ing} (empleado \times (\sigma_{nombre=Soto}(\rho_{empleado2}(empleado))))$$

## Formalización de expresiones

- El álgebra relacional es un lenguaje de expresiones. Toda expresión se puede generar a partir de las siguientes reglas:
  - Toda relación de la base de datos es una expresión.
  - Si  $E_1$  y  $E_2$  son expresiones entonces las siguientes también son expresiones:
    - $E_1 \cup E_2$
    - $E_1 - E_2$
    - $E_1 \times E_2$
    - $\sigma_P(E_1)$  donde  $P$  es un predicado con atributos de  $E_1$
    - $\pi_A(E_1)$  donde  $A$  es una lista de atributos de  $E_1$
    - $\rho_x(E_1)$  donde  $x$  es el nuevo nombre de la relación  $E_1$ .
- Cada expresión generada por las reglas anteriores tiene como resultado una relación.

# Operaciones Adicionales, Intersección

- Las operaciones anteriores son suficientes para definir toda el algebra relacional.
- Algunas consultas habituales son complejas de realizar con combinaciones de operaciones simples, usamos algunos operadores adicionales.

## Intersección

- La intersección usual de conjuntos.
- Sintaxis:

$$r_1 \cap r_2$$

- Se deben cumplir las mismas restricciones que en la unión y diferencia, los atributos de la relaciones involucradas deben tener los mismos dominios.
- La intersección se puede crear a partir de la diferencia:

$$r_1 \cap r_2 = r_1 - (r_1 - r_2)$$

- Ejemplo:

$$\pi_{nombre}(alumno\ preg) \cap \pi_{nombre}(alumno\ posg)$$

resulta en una relación que contiene los nombres de todos los alumnos que son simultáneamente de pre y posgrado.

# Reunión Natural (Natural Join)

- Hace un producto cartesiano de sus dos argumentos y realiza una selección forzando la igualdad de atributos que aparecen en ambas relaciones. Serían la clave primaria y foránea de ambas relaciones.
- Elimina repetidos (como toda operación de conjuntos).

- Sintaxis:

$$r_1 \bowtie r_2$$

Expresada en operaciones básicas quedaría

$$R \bowtie S = \Pi_{A_1, A_2 \dots A_n}(\sigma_{\theta}(R \times S))$$

donde la condición  $\theta$  es la igualdad Clave Primaria = Clave Externa (o Foránea), y la proyección elimina la columna duplicada (clave externa).

- Ejemplo: listar todos los empleados y el nombre del departamento en el que trabajan

$$\pi_{emp.nombre, dept.nombre}(empleado \bowtie departamento)$$

<i>empleado.nombre</i>	<i>departamento.nombre</i>
Torres	Informática
Soto	Marketing
Pérez	Marketing
Figueroa	Informática
Salas	Informática
Ríos	Ventas
Campos	Marketing
Venegas	Informática
Carcamo	Marketing
Gonzalez	Ventas

# Join en General

- Una forma más general de hacer Join es especificando una *propiedad de reunión*.
- Se hace entonces un producto cartesiano de las dos relaciones y se realiza una selección forzando una propiedad más general que la igualdad de atributos que aparecen en ambas relaciones.

- Sintaxis:

$$r_1 \bowtie_P r_2$$

donde  $P$  es la propiedad de reunión.

- Ejemplo: listar todos los pares de nombres de empleados y departamentos tales que el empleado ingresó a la empresa en una fecha anterior a la de creación del departamento

$\pi_{emp.nombre, depto.nombre}(empleado \bowtie_{(fecha\_ing < fecha\_creac)}$   
departamento)

# Relaciones Temporales

- A veces las consultas se hacen muy extensas.
- Una forma de simplificarlas es usando relaciones temporales y asignándoles expresiones para usar después:
- $r \leftarrow E$  asigna la expresión de álgebra relaciones  $E$  a la nueva relación  $r$ .
- Ejemplo:

$temp \leftarrow \sigma_{sueldo > 500000}(empleado \bowtie departamento)$

$\pi_{empleado.nombre, departamento.nombre}(temp)$

- El último resultado son los nombres de empleados y el departamento en el que trabajan tales que el sueldo del empleado es mayor a \$500.000.