

Bases de datos - Facultad de Ingeniería de Sistemas
 Est. Ing. Luis Felipe Narváez Gómez Cod: 3120905

Algebra Relacional.

El algebra relacional derivado a las bases de datos, es un lenguaje de consulta procedimental que se puede englobar en las operaciones de relacion entre conjuntos, matrices o entidades.

¿De que consta?

¿Que operaciones se pueden Realizar?

Son un conjunto de operaciones que toman como "IN" una o dos relaciones y producen como resultado una nueva relación.

- Selección
- Proyección
- Union
- Diferencia
- Producto Cartesiano
- Renombramiento
- Intersección
- Reunion Natural
- División
- Asignación

• Operación de Selección

- Selecciona tuplas que satisfacen a un predicado dado.
- Su letra es el sigma griego σ
- El predicado de la selección aparece como subíndice del sigma σ
- La relación del argumento se da entre parentesis.

| numero-prestamo | nombre-sucursal | importe |
|-----------------|-----------------|---------|
| P-11 | Collada Morado | 1500 |
| P-12 | Centro | 1200 |
| P-13 | Campus | 1300 |

$\sigma_{\text{nombre-sucursal} = \text{«Campus»}} (\text{Prestamo})$

| numero-prestamo | nombre-sucursal | importe |
|-----------------|-----------------|---------|
| P-13 | Campus | 1300 |

SELECT nombre_sucursal FROM prestamo WHERE nombre_sucursal = 'Campus'

Se pueden utilizar las comparaciones como:

$=, \neq, <, \leq, >, \geq$ en el predicado de la selección.

Las mismas se pueden utilizar en combinación utilizando los símbolos conectivos (\wedge) o (\vee).

$\sigma_{\text{nombre_sucursal} = \text{«Campus»} \wedge \text{importe} > 1200}(\text{prestamo})$

Haz una selección en nombre sucursal igual nombre atributo que tambien nombre atributo sea mayor a valor

| nombre-prestamo | nombre-sucursal | importe |
|-----------------|-----------------|---------|
| P-13 | Campus | 1300. |

`SELECT nombre_sucursal FROM prestamo WHERE nombre_sucursal = "Campus"`

• Operación de Proyección `AND importe > 1200`

- Traer los datos de una tabla sin que todas las columnas se muestren.
- Esta es una operación Unaria, es decir que solo necesita de una relación, la cual devuelve su relación de argumentos, excluyendo algunos de esos argumentos.
- Este elimina las filas repetidas ya que son relaciones entre conjuntos.
- Los atributos que se quiere que aparezcan se escriben como subíndice.
- Se denominan con la letra griega π .

| id | nombre | apellido | cedula | deuda |
|----|----------|----------|--------|-------|
| 1 | Felipe | Narvaez | 1041 | 500 |
| 2 | Fernando | Martinez | 1042 | 600 |
| 3 | David | Moreno | 1043 | 700 |
| 4 | Alex | Galurza | 1044 | 100. |

Π nombre, deuda (Clientes)

| nombre | deuda |
|----------|-------|
| Felipe | 500 |
| Fernando | 600 |
| David | 700 |
| Alex | 100. |

SELECT nombre, deuda
FROM clientes

• Operación Union.

Se denota con la letra de union (de conjuntos " \cup "), se encarga de unir los datos de tablas con valores similares, descartando las filas repetidas.

Π nombre, id (Clientes)

| nombre | id |
|------------|-----|
| Isabela | 300 |
| Jorge Luis | 800 |
| Alberto | 500 |
| Jessica | 700 |

Π nombre, id (Proveedores)

| nombre | id |
|---------|-----|
| Gisel | 100 |
| Monica | 200 |
| Isabela | 300 |
| Alberto | 500 |

Π nombre, id (Clientes)

\cup

Π nombre, id (Proveedores)

SELECT * FROM clientes UNION SELECT * FROM proveedores

| nombre. | id |
|------------|------|
| Isabela | 300 |
| Jorge Luis | 800 |
| Alberto | 500 |
| Jessica | 700. |
| Gissel | 100 |
| Monica. | 200. |

• Intersección de Conjuntos

- Se denota con la letra " \cap " que en conjuntos significa intersección.
- Este opera sobre 2 o más tablas.
- Es necesario que todas tengan la misma estructura.
- Da como resultado una nueva tabla cuyo contenido son las filas comunes de las tablas originales descartando filas repetidas.

$\Pi_{id, nombre} (clientes)$

$\Pi_{id, nombre} (proveedores)$

SELECT id, nombre FROM clientes;

SELECT id, nombre FROM proveedores;

| clientes | |
|----------|------------|
| id | nombre |
| 300 | Isabela |
| 800 | Jorge Luis |
| 500 | Alberto |
| 700 | Jessica |

| Proveedores | |
|-------------|---------|
| id | nombre |
| 100 | Gissel |
| 200 | Monica |
| 300 | Isabela |
| 500 | Alberto |

$\Pi_{id, nombre} (clientes) \cap \Pi_{id, nombre} (proveedores)$

SELECT id, nombre FROM clientes INTERSECT
SELECT id, nombre FROM proveedores;

| clientes INTERSECCIÓN proveedores | |
|-----------------------------------|---------|
| id | nombre |
| 300 | Isabela |
| 500 | Alberto |

Operación Diferencia de Conjuntos.

- Esta denotada por el símbolo " $-$ " que en matemáticas, significa resta.
- Esta operación permite buscar las tuplas que estén en una relación pero no en la otra.
- Por ejemplo $r - s$ dará como resultado las tuplas que contiene r pero no están en s .

$\Pi_{id, nombre} (Clientes)$

SELECT id, nombre FROM Clientes;

| Clientes | |
|----------|-----------|
| id | nombre |
| 300 | Isabela |
| 800 | Jorge Los |
| 500 | Alberto |
| 700 | Jessica |

$\Pi_{id, nombre} (Proveedores)$

SELECT id, nombre FROM Proveedores;

| Proveedores | |
|-------------|---------|
| id | nombre |
| 100 | Gissel |
| 200 | Monica |
| 300 | Isabela |
| 500 | Alberto |

$\Pi_{id, nombre} (Clientes) - \Pi_{id, nombre} (Proveedores)$

SELECT id, nombre FROM Clientes EXCEPT SELECT id, nombre FROM Proveedores ;

| Clientes Diferencia Proveedores | |
|---------------------------------|-----------|
| id | nombre |
| 800 | Jorge Los |
| 700 | Jessica |

$\Pi_{id, nombre} (Proveedores) - \Pi_{id, nombre} (Clientes)$

SELECT id, nombre FROM Proveedores EXCEPT SELECT id, nombre FROM Clientes ;

| Proveedores DIFERENCIA Clientes | |
|---------------------------------|--------|
| id | nombre |
| 100 | Gissel |
| 200 | Monica |

• Producto Cartesiano.

- Se denota por el símbolo en aspa: "x" que en matemáticas significa multiplicación.
- Permite combinar información de cualesquiera dos relaciones.
- Se debe tener en cuenta que las relaciones entre $R1 \times R2$ se definen como subconjuntos del producto cartesiano de un conjunto de dominios.
- No es necesario que las tablas utilizadas tengan la misma estructura.
- El contenido de la tabla resultante es todas las posibles combinaciones de las filas de una de ambas tablas.

$$\sigma_* = \langle \langle * \rangle \rangle (\text{Tabla A})$$

SELECT * FROM TablaA;

Tabla A

| W | X |
|----|----|
| 1 | 23 |
| 78 | 32 |
| 67 | 5 |

$$\sigma_* = \langle \langle * \rangle \rangle (\text{Tabla B})$$

SELECT * FROM TablaB;

Tabla B

| Y | Z |
|----|-----|
| 15 | 320 |
| 7 | 5 |

$$\sigma_* = \langle \langle * \rangle \rangle (\text{Tabla A} \times \text{Tabla B})$$

SELECT * FROM TablaA, TablaB;

| Tabla A * Tabla B | | | |
|-------------------|----|----|-----|
| W | X | Y | Z |
| 1 | 23 | 15 | 320 |
| 1 | 23 | 7 | 5 |
| 78 | 32 | 15 | 320 |
| 78 | 32 | 7 | 5 |
| 67 | 5 | 15 | 320 |
| 67 | 5 | 7 | 5 |

• Operación de Reunión.

- Opera sobre 2 o más tablas, que posean estructuras distintas.
- Devuelve una tabla cuyo contenido es un conjunto de filas con las columnas deseadas provenientes de las diferentes tablas.
- Las filas resultantes están relacionadas bajo una condición.

| Tabla A | | Tabla B | | |
|---------|---------|---------|---------|---------|
| valor V | valor W | valor X | valor Y | valor Z |
| 1 | 23 | 5 | 15 | 320 |
| 78 | 32 | 78 | 37 | 5 |
| 67 | 5 | 1 | 33 | 3 |
| | | 78 | 5 | 404 |

$\Pi_{A.valorW, B.valorY} (TablaA \Join TablaB \Rightarrow A.valorV = B.valorX \ \& \ B.valorZ > 10)$

SELECT A.valorW, B.valorY FROM TablaA
 Join TablaB
 ON (A.valorV = B.valorX AND B.valorZ > 10)

| Reunión de A y B Tomando W e Y con X = V y Z > 10. | |
|---|---------|
| valor W | valor Y |
| 32 | 5 |

• Operación División.

- Opera sobre 2 Tablas.
- Al dividir B/A se obtiene una tabla cuyas columnas sean aquellas de la tabla B que no estén en la tabla A y cuyas Filas sean tales que cumplan con estar relacionadas con todas y cada una de las Filas de la tabla A.

| Tabla A |
|---------|
| Codigo |
| 1425 |
| 2000 |
| 3000 |

| Tabla B | |
|---------|---------|
| Codigo | Indice. |
| 1425 | 15%. |
| 2000 | 27%. |
| 3000 | 33%. |
| 2128 | 45%. |
| 2121 | 13%. |
| 2000 | 15%. |
| 3000 | 15%. |

En MySQL version 5.2 no existe una implementación directa del operador de división por lo que se debe implementar combinando condiciones y otros operadores que si estan disponibles.

```
SELECT DISTINCT X.Indice FROM bb X
WHERE NOT EXISTS (
```

```
SELECT Y.codigo X FROM aa = Y
WHERE NOT EXISTS (
```

```
SELECT Z.codigo FROM bb Z
WHERE Z.codigo = Y.codigo AND Z.Indice =
X.Indice.
)
```

)

| B Dividida A | |
|--------------|------|
| Indice | 15%. |

• Operación Asignación

- Opera sobre una única tabla
- Asigna nuevos valores a algunas columnas de algunas filas de la misma.

| Tabla A | |
|---------|--------|
| Código | Índice |
| 2000 | 27% |
| 3000 | 33% |
| 1425 | 15% |
| 2128 | 45% |
| 2121 | 13% |
| 2000 | 15% |
| 3000 | 15% |

UPDATE A SET índice = "100%" WHERE código > 2000

| Tabla A Modificado | |
|--------------------|--------|
| Código | Índice |
| 2000 | 27% |
| 3000 | 100% |
| 1425 | 15% |
| 2128 | 100% |
| 2121 | 100% |
| 2000 | 15% |
| 3000 | 100% |