

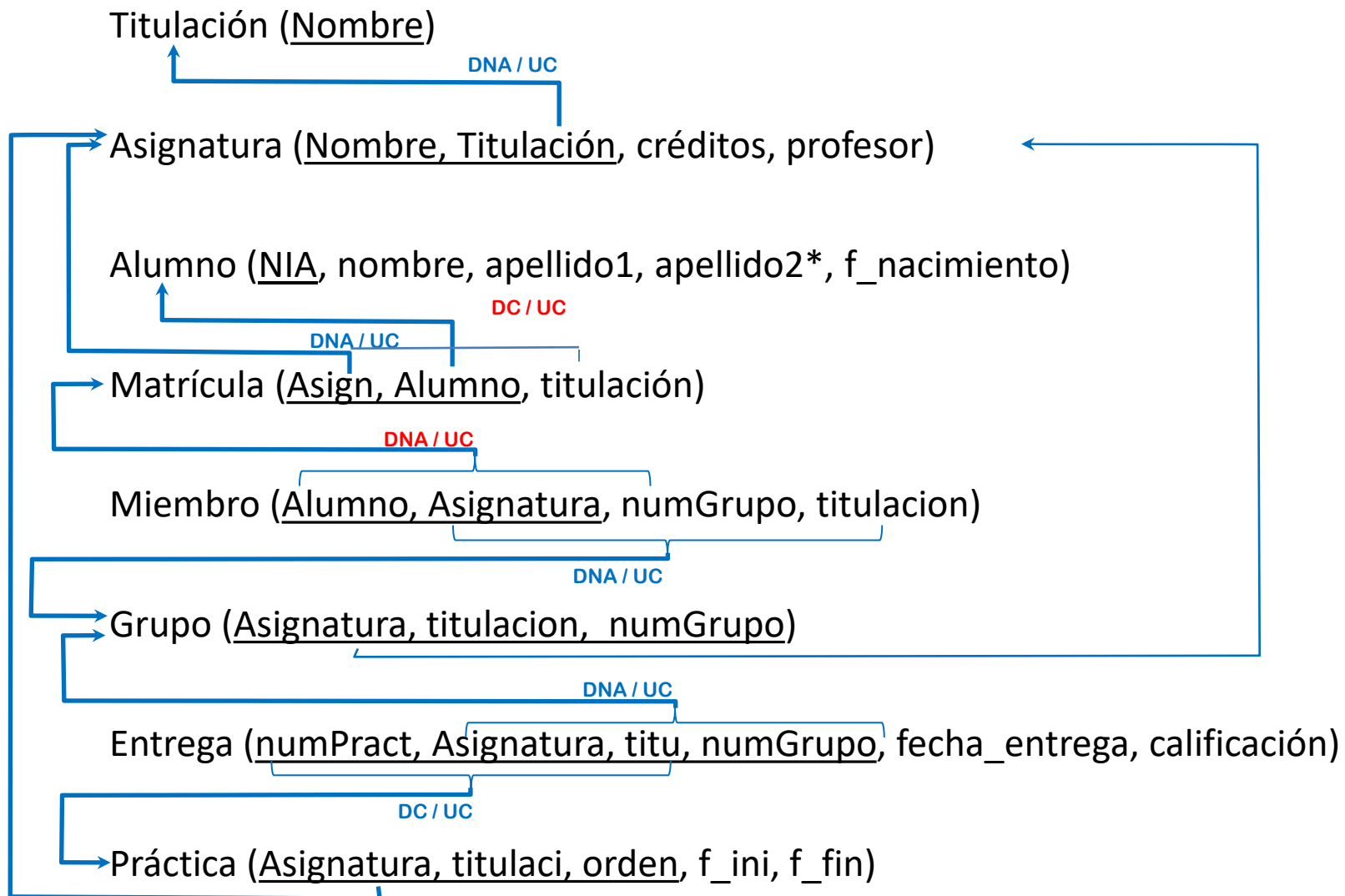
## 3.1. Estrategias de Resolución

## 3.2. del Álgebra Relacional al SQL

## 3.2. Ejemplos

- **Dividir** problemas complejos en varios más sencillos (establecer otras subconsultas como pasos intermedios para alcanzar el resultado final)
- Analizar **ubicación** de los datos (qué tablas contienen las claves de búsqueda y cuáles las claves a proyectar).
- Establecer una **ruta** (navegación) entre las tablas involucradas en la consulta (el Álgebra es un lenguaje *navegacional*).
- Descubrir la necesidad de **agrupar** tablas cuando se necesitan datos agregados.
- Diferenciar agregación de datos de comparación de tuplas (*agrupar y ordenar* son recursos diferentes).

# Diseño Relacional: Ejemplo



**Selección**: escogemos las tuplas que cumplan una condición

Notación:  $\sigma_{\text{predicado}}$  (Relación)

Libros escritos por Dumas

**Libros**

| <i>Signat.</i> | <i>Título</i>           | <i>Autor</i> |
|----------------|-------------------------|--------------|
| Dum-1          | Los Tres Mosqueteros    | Dumas        |
| Per-1          | El Capitán Alatraste    | P-Reverte    |
| Asi-1          | El Fin de la Eternidad  | Asimov       |
| Dum-2          | El Conde de Montecristo | Dumas        |



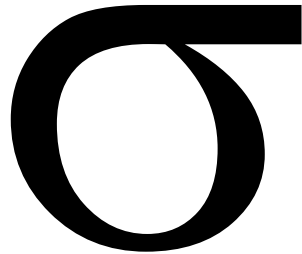
$\sigma_{\text{autor}='Dumas'}$  (**Libros**)

| <i>Signat.</i> | <i>Título</i>           | <i>Autor</i> |
|----------------|-------------------------|--------------|
| Dum-1          | Los Tres Mosqueteros    | Dumas        |
| Dum-2          | El Conde de Montecristo | Dumas        |

(se escogen filas)

```
SELECT * FROM Libros WHERE autor='Dumas';
```

### Tu turno:



- Estudiantes senior (mayores de 65)
- Cursos en los que se ha matriculado el estudiante 1234
- Estudiantes que celebran hoy su cumpleaños
- Prácticas expiradas
- Cursos con el término 'Big Data' en su título

# Tema 3P.2: Algebra Relacional

## Operadores Unarios

- Estudiantes senior (mayores de 65)

$\sigma_{(\text{SYSDATE}-f\_nacimiento)/365.2422 > 65}$  (Alumno)

- Cursos en los que se ha matriculado el estudiante 1234

$\sigma_{\text{alumno}=1234}$  (Matrícula)

- Estudiantes que celebran hoy su cumpleaños

$\sigma_{\text{to\_char}(\text{SYSDATE}, 'DD/MM')=\text{to\_char}(f\_nacimiento, 'DD/MM')}$  (Alumno)

- Prácticas expiradas

$\sigma_{f\_fin > \text{SYSDATE}}$  (Práctica)

- Cursos con el término 'Big Data' en su título

$\sigma_{\text{nombre LIKE } \% \text{Big Data} \%}$  (Asignatura)

**Proyección**: subconjunto del esquema relación

Notación:  $\pi_{\text{atrib1, atrib2..}}(\text{Relación})$

**Libros**

| <i>Signat.</i> | <i>Título</i>           | <i>Autor</i> |
|----------------|-------------------------|--------------|
| Dum-1          | Los Tres Mosqueteros    | Dumas        |
| Per-1          | El Capitán Alatraste    | P-Reverte    |
| Asi-1          | El Fin de la Eternidad  | Asimov       |
| Dum-2          | El Conde de Montecristo | Dumas        |



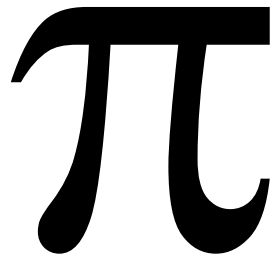
$\pi_{\text{título, autor}}(\text{Libros})$

| <i>Título</i>           | <i>Autor</i> |
|-------------------------|--------------|
| Los Tres Mosqueteros    | Dumas        |
| El Capitán Alatraste    | P-Reverte    |
| El Fin de la Eternidad  | Asimov       |
| El Conde de Montecristo | Dumas        |

(se escogen columnas)

```
SELECT distinct título, autor FROM libros;
```

## Tu turno:



- Nombre, apellido y NIA de los estudiantes
- Asignaturas del Grado en Informática
- Máximo número de días dados para realizar cada práctica.
- Edad de cada estudiante
- Número total de asignaturas impartidas en Grados de Ingeniería



# Tema 3P.2: Algebra Relacional

## Operadores Unarios

- Nombre, apellido y NIA de los estudiantes

$\pi_{\text{nombre, apellido, NIA}}(\text{Alumno})$

- Asignaturas del Grado en Informática

$\pi_{\text{Nombre}}(\sigma_{\text{titulación}='Grado en Informática'}(\text{Asignatura}))$

- Máximo número de días dados para realizar cada práct.

$\pi_{\text{asignatura, titulación, orden, (f_{fin}-f_{ini}) \text{ as días}}}(\text{Práctica})$

- Edad de cada estudiante

$\pi_{\text{NIA, (SYSDATE-f_{nacim})/365.2422 \text{ as edad}}}(\text{Alumno})$

- Número total de asignaturas impartidas en Grados de Ingeniería

$\pi_{\text{COUNT('x') as \#cursos}}(\sigma_{\text{titulación LIKE '%Ingeniería\%'}}(\text{Asignatura}))$

# Tema 3P.2: Álgebra. Definición de relación Renombrado: $\rho$

**Renombrado**: asigna el resultado de una expresión a un símbolo (vista o relación temporal con existencia limitada a la consulta definida)\*.

Notación:  $\rho_{\text{símbolo}} (\text{Expresión})$ , o también  $S \equiv \text{Expresión}$

Sea Aux el conjunto de libros escrito por Dumas, coger solo los títulos.

$\rho_{\text{Aux}} (\sigma_{\text{autor}='Dumas'} (\text{libros}) )$

$\rho_Q (\pi_{\text{título}} \text{Aux})$

```
WITH Aux AS SELECT * FROM libros WHERE autor='Dumas'
SELECT titulo FROM Aux;
```

$\rho$

- Nombre de los asignaturas de la titulación Grado en Informática
- Asignaturas con alumnos matriculados

## Tema 3P.2: Álgebra. Definición de relación Renombrado: $\rho$

- Nombre de los asignaturas de la titulación  
Grado en Informática

$$\rho_{ASG} (\pi_{\text{nombre}}(\sigma_{\text{titulación}='Grado en Informática'}(Asignaturas)))$$

$$ASG \equiv \pi_{\text{nombre}}(\sigma_{\text{titulación}='Grado en Informática'}(Asignaturas))$$

- Asignaturas con alumnos matriculados

$$ASG\_ALM \equiv \pi_{\text{asignatura,titulación}}(Matrícula)$$

$$\rho_{ASG\_ALMN} (\pi_{\text{asignatura,titulación}} Matrícula)$$

**Unión:** todas las tuplas de ambas relaciones (compatibles), eliminándose todas las tuplas repetidas

Libros de aventuras o novelas      *Adventuras*  $\cup$  *Novelas*

## *Adventuras*

| <i>Signat.</i> | <i>Título</i>           | <i>Autor</i> |
|----------------|-------------------------|--------------|
| Dum-1          | Los Tres Mosqueteros    | Dumas        |
| Per-1          | El Capitán Alatraste    | P-Reverte    |
| Dum-2          | El Conde de Montecristo | Dumas        |

## *Novelas*

| <i>Signat.</i> | <i>Título</i>          | <i>Autor</i> |
|----------------|------------------------|--------------|
| Ner-1          | Veinte poemas de amor  | Neruda       |
| Asi-1          | El Fin de la Eternidad | Asimov       |



## *Adventuras* $\cup$ *Novelas*

| <i>Signat.</i> | <i>Título</i>           | <i>Autor</i> |
|----------------|-------------------------|--------------|
| Dum-1          | Los Tres Mosqueteros    | Dumas        |
| Per-1          | El Capitán Alatraste    | P-Reverte    |
| Asi-1          | El Fin de la Eternidad  | Asimov       |
| Dum-2          | El Conde de Montecristo | Dumas        |
| Ner-1          | Veinte poemas de amor   | Neruda       |

```
SELECT * FROM Aventuras
UNION
SELECT * FROM Novelas;
```

**Intersección**: todas las tuplas que estén en ambas (compatibles)

***Aventuras***

| <i>Signat.</i> | <i>Título</i>           | <i>Autor</i> |
|----------------|-------------------------|--------------|
| Dum-1          | Los Tres Mosqueteros    | Dumas        |
| Per-1          | El Capitán Alatraste    | P-Reverte    |
| Dum-2          | El Conde de Montecristo | Dumas        |

***Novelas***

| <i>Signat.</i> | <i>Título</i>         | <i>Autor</i> |
|----------------|-----------------------|--------------|
| Ner-1          | Veinte poemas de amor | Neruda       |
| Per-1          | El Capitán Alatraste  | P-Reverte    |



***Aventuras*  $\cap$  *Novelas***

| <i>Signat.</i> | <i>Título</i>        | <i>Autor</i> |
|----------------|----------------------|--------------|
| Per-1          | El Capitán Alatraste | P-Reverte    |

**Novelas de aventuras**

```
SELECT * FROM Aventuras
INTERSECT
SELECT * FROM Novelas;
```

**Diferencia**: tuplas que aparecen en la primera y no en la otra (compatibles)

Libros de aventuras que no se consideran novelas

***Aventuras***

| <i>Signat.</i> | <i>Título</i>           | <i>Autor</i> |
|----------------|-------------------------|--------------|
| Dum-1          | Los Tres Mosqueteros    | Dumas        |
| Per-1          | El Capitán Alatraste    | P-Reverte    |
| Dum-2          | El Conde de Montecristo | Dumas        |

***Novelas***

| <i>Signat.</i> | <i>Título</i>         | <i>Autor</i> |
|----------------|-----------------------|--------------|
| Ner-1          | Veinte poemas de amor | Neruda       |
| Per-1          | El Capitán Alatraste  | P-Reverte    |

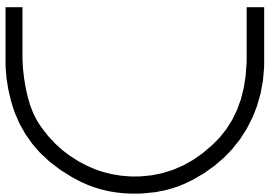


***Aventuras - Novelas***

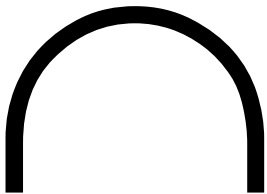
| <i>Signat.</i> | <i>Título</i>           | <i>Autor</i> |
|----------------|-------------------------|--------------|
| Dum-1          | Los Tres Mosqueteros    | Dumas        |
| Dum-2          | El Conde de Montecristo | Dumas        |

```
SELECT * FROM Aventuras
MINUS
SELECT * FROM Novelas;
```

Tu turno:



- Listado de profesores y alumnos



- Asignaturas sin prácticas



- Profesores que son alumnos

- Listado de profesores y alumnos

$$\pi_{\text{profesor}}(\textit{Asignatura}) \cup \pi_{\text{nombre||apellido1||apellido2}}(\textit{Alumno})$$

- Asignaturas que no tienen prácticas

$$\pi_{\text{nombre,titulación}}(\textit{Asignatura}) - \pi_{\text{asignatura,titulación}}(\textit{Práctica})$$

- Profesores que son alumnos

$$\pi_{\text{profesor}}(\textit{Asignatura}) \cap \pi_{\text{nombre||apellido1||apellido2}}(\textit{Alumno})$$



**Producto cartesiano**: tuplas de ambas en todas las combinaciones

## Comidas

| Especialidad | Región    |
|--------------|-----------|
| Gazpacho     | Andalucía |
| Cocido       | Castilla  |
| Pote         | Galicia   |

## Vinos

| Denominación |
|--------------|
| Rioja        |
| Rueda        |



## Comidas × Vinos

| Especialidad | Región    | Denominación |
|--------------|-----------|--------------|
| Gazpacho     | Andalucía | Rioja        |
| Gazpacho     | Andalucía | Rueda        |
| Cocido       | Castilla  | Rioja        |
| Cocido       | Castilla  | Rueda        |
| Pote         | Galicia   | Rioja        |
| Pote         | Galicia   | Rueda        |

Carta de comidas con los distintos vinos que se ofrecen

```
SELECT * FROM Comidas CROSS JOIN Vinos;  
SELECT * FROM Comidas, Vinos;
```

**Combinación** (simple join or **inner join**): tuplas del producto cartesiano que cumplen una expresión condicional genérica. La condición lleva operadores de comparación (=, >, <..)

¿Qué películas pueden ver los clientes del cine?

**Cientes**

| Nombre  | Edad |
|---------|------|
| Fulano  | 29   |
| Mengano | 18   |
| Zutano  | 9    |

**Películas**

| Título  | Calificación |
|---------|--------------|
| Matrix  | 18           |
| Sólo tu | 0            |



**Cientes**  $\theta_{\text{Edad} \geq \text{Calificación}}$  **Películas**

| Nombre  | Edad | Título  | Calificación |
|---------|------|---------|--------------|
| Fulano  | 29   | Matrix  | 18           |
| Fulano  | 29   | Sólo tu | 0            |
| Mengano | 18   | Matrix  | 18           |
| Mengano | 18   | Sólo tu | 0            |
| Zutano  | 9    | Sólo tu | 0            |

```
SELECT * FROM Cientes JOIN Películas
ON (Cientes.edad>=calificación);
```

**Combinación Natural (equijoin)**: Caso particular del operador combinación donde la comparación es de igualdad. Notación: \* ó

***Empleados***

| <i>Nombre</i> | <i>DNI</i> |
|---------------|------------|
| Fulano        | 569064     |
| Mengano       | 434848     |
| Zutano        | 383090     |

Nombre de los empleados que son dueños de algún coche

$Empleados *_{\text{dueño}} Coches$        $\Pi_{\text{nombre}} (Empleados *_{\text{dueño}} Coches)$

***Coches***

| <i>Marca</i> | <i>Dueño</i> |
|--------------|--------------|
| Ford Escort  | 383090       |
| Seat Ibiza   | 569064       |



| <i>Nombre</i> | <i>DNI</i> | <i>Marca</i> |
|---------------|------------|--------------|
| Fulano        | 569064     | Seat Ibiza   |
| Zutano        | 383090     | Ford Escort  |



| <i>Nombre</i> |
|---------------|
| Fulano        |
| Zutano        |

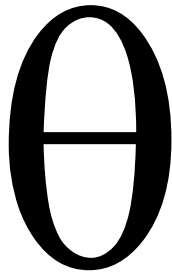
- observar que se elimina la columna repetida

```
SELECT Nombre FROM Clientes NATURAL JOIN Coches
USING (DNI, Dueño);
```

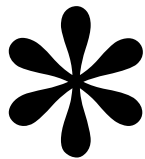
Tu turno:



- Posibles grupos en la asignatura de Ficheros y BD



- Grupos con entregas fuera de plazo.



- Listado de alumnos que han entregado alguna práctica (nombre y primer apellido)

- Posibles grupos en la asignatura de Ficheros y BD

$$\Pi_{\text{Alumno}} (\sigma_{\text{asignatura}='Ficheros y BD'} (\text{Matrícula})) \times \Pi_{\text{Alumno}} (\sigma_{\text{asignatura}='Ficheros y BD'} (\text{Matrícula}))$$

$$\Pi_{\text{Alumno}} (\sigma_{\text{asignatura}='Ficheros y BD'} (\text{Matrícula})) \theta_{\text{Alumno} \neq \text{Alumno}} \Pi_{\text{Alumno}} (\sigma_{\text{asignatura}='Ficheros y BD'} (\text{Matrícula}))$$

- Grupos con entregas fuera de plazo.

$$\Pi_{\text{Asignatura, titulación, num_grupo}} (\text{Entrega} \theta_{\text{fecha_entrega NOT BETWEEN f\_ini AND F\_fin}} \text{Práctica})$$

- Listado de alumnos que han entregado alguna práctica (nombre y primer apellido)

$$\Pi_{\text{nombre, apellido}} ((\text{Entrega} *_{\text{asignatura, titulación, num_grupo}} \text{Miembro}) *_{\text{alumno}} \text{Alumno})$$

**Agrupación:** formación de grupos según un conjunto de atributos al cual se le aplica una función de agregación.

Notación:  $\pi_{\text{proyección}} \sigma_{\text{selección}} \mathbf{G}_{\text{criterio}}(\text{relación})$   
 /\* también se admite GROUP BY en vez de  $\mathbf{G}$  \*/

## Personas

| Nombre    | Edad | Nacionalidad |
|-----------|------|--------------|
| Fulano    | 29   | Español      |
| Mengano   | 49   | Español      |
| John Doe  | 73   | Inglés       |
| Smith     | 14   | Inglés       |
| Zutano    | 3    | Español      |
| Pelancejo | 25   | Español      |

$\pi_{\text{Count('x')}} \mathbf{Group\ by}_{\text{Nacionalidad}}(\text{Personas})$



| Nacionalidad | Count |
|--------------|-------|
| Español      | 4     |
| Inglés       | 2     |

Número de personas por nacionalidad

```
SELECT Nacionalidad, count('x')
FROM Personas
GROUP BY Nacionalidad;
```

Funciones de agrupación: Count(), Sum(), Avg(), Min(), Max()

# Tema 3P.2: Álgebra Relacional Extendida. Conjuntos Ordenados

**Orden:** conjunto ordenado (lista) es el resultado de aplicar un orden ( $\text{ORDER BY}_{\text{orden}}$  ó  $\top_{\text{orden}}$ ) sobre una relación. Sobre una lista ordenada se pueden aplicar funciones (analíticas) de agregación: *first*, *last*, y *rank(value)*, y la pseudocolumna *numrow*.

Cual es la posición de John Doe entre los clientes ordenados alfabéticamente?

## Clients

| Name      | Age | Nationality |
|-----------|-----|-------------|
| Fulano    | 29  | Spanish     |
| Mengano   | 49  | Spanish     |
| John Doe  | 73  | English     |
| Smith     | 14  | English     |
| Zutano    | 3   | Spanish     |
| Pelancejo | 25  | Spanish     |

$\pi_{\text{rank}('John Doe')} \top_{\text{age}}(\text{Clients})$



| <i>rank('John Doe')</i> |
|-------------------------|
| 6                       |

```
SELECT rank('John Doe')
       within group (ORDER BY Name) "position"
FROM Clients;
```

**Tu turno:**

G,

T

- Asignaturas con menos de 15 alumnos
- Nota media y máxima por práctica, y nota media 'global' (media de todas las prácticas de cada asignatura).
- Asignatura con mayor número de prácticas
- Nombre y apellido del estudiante con mayor calificación por asignatura.



# uc3m Tema 3P.2: Álgebra Rel. – otros operadores

- Asignaturas con menos de 15 alumnos

$\pi_{\text{asignatura,titulación}} \sigma_{\text{COUNT('X')} < 15} ( \text{G}_{\text{asignatura,titulación}} \text{Matrícula} )$

- Nota media y máxima por práctica, y media 'global' (media de todas las prácticas de cada asignatura).

$\pi_{\text{asignatura,titulación,num\_pract,AVG(calificación)} \text{ as media } \text{G}_{\text{asignatura,titulación,num\_pract}} (\text{Entrega})$

$\cup \pi_{\text{asignatura,titulación,'Global',AVG(calificación)} \text{ as media } \text{G}_{\text{asignatura,titulación}} (\text{Entrega})$

- Asignatura con mayor número de prácticas

$\pi_{\text{asignatura,titulación}} \sigma_{\text{first}(1)} \top_{\text{COUNT('X')}} ( \text{G}_{\text{asignatura,titulación}} \text{Práctica} )$

- Nombre y apellido del estudiante con mayor calificación por asignatura.

$\rho_{\text{CAL}} ( \pi_{\text{asignatura,titulación,num\_grupo,SUM(calificación)} \text{ as nota } ( \text{G}_{\text{asignatura,titulación,numgrupo}} \text{Entrega} ) )$

$\rho_{\text{GRP}} ( \pi_{\text{asignatura,titulación,MAX(nota)} \text{ as nota } ( \text{G}_{\text{asignatura,titulación}} \text{CAL} ) )$

$\rho_{\text{Q}} ( \pi_{\text{asignatura,titulación,nombre,apellido,nota} ( \text{Alumno} * \text{Miembro} * \text{CAL} * \text{GRP} )$

# Tema 3P.2: Álgebra Relacional Extendida: División

**División:** el cociente lo forman todas las tuplas que concatenadas con cada tupla del divisor estén contenidas en el dividendo.

Notación:

$$A \div B \equiv \pi_{\text{esq}(A)-\text{esq}(B)} A - \pi_{\text{esq}(A)-\text{esq}(B)} ((\pi_{\text{esq}(A)-\text{esq}(B)} A \times B) - A)$$

## Currícula

| Candidato  | Habilidad  |
|------------|------------|
| Pepe Pérez | Lenguaje C |
| Pepe Pérez | Java       |
| José Gómez | Lenguaje C |
| José Gómez | SQL        |
| Fulano Pi  | SQL        |
| Fulano Pi  | Java       |
| Fulano Pi  | Lenguaje C |

Candidatos que cumplen el perfil 'Analista/Des'

$$\text{Currícula} \div \pi_{\text{requisito}} (\sigma_{\text{perfil}='Analista/Des'} \text{Puestos})$$

```
SELECT candidato FROM currícula
MINUS
SELECT candidato
  FROM (SELECT candidato, requisito
        FROM currícula, puesto
        MINUS
        SELECT * FROM currícula);
```

## Puestos

| Perfil       | Requisito  |
|--------------|------------|
| Analista/Des | Lenguaje C |
| Analista/Des | SQL        |



| Candidato  |
|------------|
| José Gómez |
| Fulano Pi  |

# Tema 3P.2: Álgebra Relacional Extendida: Semi-Join

**Semi-Combinación**: igual que cualquier combinación, pero sólo se toman las columnas del operando izquierdo ( $|*$ ) o derecho ( $*|$ ).

*Notación*: left semijoin:  $|*$  ó  $\ltimes$  right semijoin:  $*|$  ó  $\rtimes$

**Empleados**

| Nombre  | DNI    |
|---------|--------|
| Fulano  | 569064 |
| Mengano | 434848 |
| Zutano  | 383090 |

**Empleados**  $\ltimes$  **Coches**  
dueño



| Nombre | DNI    |
|--------|--------|
| Fulano | 569064 |
| Zutano | 383090 |

**Empleados con coche**

**Coches**

| Marca       | Dueño  |
|-------------|--------|
| Ford Escort | 383090 |
| Seat Ibiza  | 569064 |

```
SELECT *
FROM empleados NATURAL JOIN
(SELECT dueño FROM coche);
```

```
SELECT * FROM empleados
WHERE DNI IN
(SELECT dueño FROM coche);
```

# Tema 3P.2: Álgebra Relacional Extendida: Anti-Join

**Anti-Combinación**: igual que la semi combinación, pero las tuplas que se incluyen son las que no cumplen la condición definida.

***Empleados***

| Nombre  | DNI    |
|---------|--------|
| Fulano  | 569064 |
| Mengano | 434848 |
| Zutano  | 383090 |

***Empleados***  dueño ***Coches***



| Nombre  | DNI    |
|---------|--------|
| Mengano | 434848 |

**Empleados sin coche**

***Coches***

| Marca       | Dueño  |
|-------------|--------|
| Ford Escort | 383090 |
| Seat Ibiza  | 569064 |
| Honda Civic | 383090 |

```
SELECT * FROM empleados
MINUS
SELECT * FROM empleados NATURAL JOIN
    (SELECT dueño FROM coche);
```

```
SELECT * FROM empleados
WHERE DNI NOT IN
    (SELECT dueño FROM coche);
```

# Tema 3P.2: Álgebra Relacional Extendida: Outer Join

**Combinación externa:** extensión de la combinación, que incluye las tuplas que no encajan de la relación izquierda/derecha/ambas. Las columnas que no aplican, adoptan el valor nulo (NULL ó  $\omega$ ).

**left outer join:**  $]^* \text{ ó } \bowtie$  **right outer join:**  $^*[ \text{ ó } \bowtie$  **full outer join:**  $]^*[ \text{ ó } \bowtie$

Empleados y sus coches,  
en su caso

*Empleados*  $\bowtie_{\text{dueño}}$  *Coches*

```
SELECT *
  FROM empleados
 LEFT OUTER JOIN coches;
```

Coches con su dueño si es  
uno de nuestros empleados.

*Empleados*  $\bowtie_{\text{dueño}}$  *Coches*

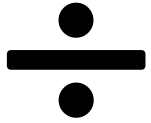
```
SELECT *
  FROM empleados RIGHT
 OUTER JOIN coches;
```

Empleados y coches,  
asociados como proceda

*Empleados*  $\bowtie_{\text{dueño}}$  *Coches*

```
SELECT *
  FROM empleados
 FULL OUTER JOIN coches;
```

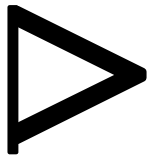
**Tu turno:**



- Estudiantes en evaluación continua (su calificación es mayor de cero en cada una de las prácticas planteadas).



- Profesores de asignaturas con prácticas.



- Nombre completo de los estudiantes que no están matriculados



- Estudiantes con sus asignaturas, tanto los matriculados como los que no lo están.

## uc3m Item 3P.2: Álgebra Rel. – otros operadores

- Estudiantes en evaluación continua (su calificación es mayor de cero en cada práctica).

$$\rho_{\text{CONT}} \left( (\sigma_{\text{calificacion} > 0} \text{Entregas}) \div (\pi_{\text{asignatura, titulación, orden}} \text{Práctica}) \right)$$

$$\rho_Q \left( \pi_{\text{asignatura, titulación, NIA}} \text{Miembro} * \text{CONT} \right)$$

- Profesores de asignaturas con prácticas.

$$\pi_{\text{profesor}} (\text{Asignatura} \bowtie_{\text{asignatura, titulación}} \pi_{\text{asignatura, titulación}} (\text{Práctica}))$$

- Nombre completo de estudiantes no matriculados

$$\pi_{\text{nombre, apellido}} (\text{Alumno} \triangleright \pi_{\text{alumno}} (\text{Matrícula}))$$

- Estudiantes con sus asignaturas, tanto los que estén matriculados como los que no lo están.

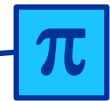
$$\pi_{\text{NIA, asignatura, titulación}} (\text{Alumno} \Join \pi_{\text{alumno}} (\text{Matrícula}))$$

[WITH

<símbolo> **AS** <subquery>  
[, <símbolo> AS <subquery> ... ] ]



**SELECT** [ALL|DISTINCT] <lista de selección>



**FROM** <cláusula de origen>



[WHERE <condición> ]



[GROUP BY <expresión> [HAVING <condcn>]]

[{UNION|UNION ALL|MINUS|INTERSECT} <query>]

[ORDER BY <expresión> [ASC|DESC]] ;





[WITH

1

<símbolo> **AS** <subquery>  
[, <símbolo> AS <subquery> ... ] ]

**SELECT** [**ALL|DISTINCT**] <lista de selección>

7

2

**FROM** <cláusula de origen>

3

[**WHERE** <condición> ]

4

[**GROUP BY** <expresión> [**HAVING** <condcn>]]

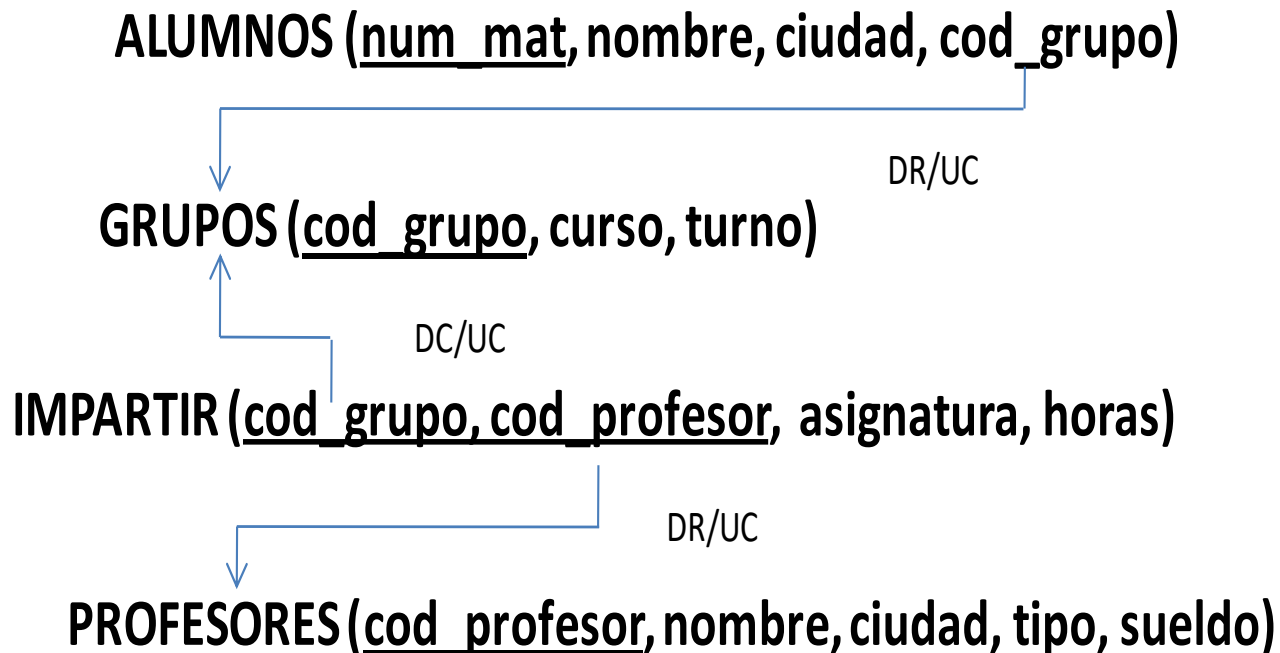
5

[{**UNION|UNION ALL|MINUS|INTERSECT**} <query>]

6

[**ORDER BY** <expresión> [**ASC|DESC**]] ;

## ➤ Consultas (ejemplos)



- Nombre y ciudad de los alumnos matriculados en el grupo cuyo código es 12

| ALUMNOS |                |          |           |
|---------|----------------|----------|-----------|
| num_mat | nombre         | ciudad   | cod_grupo |
| 0001    | Juan García    | leganés  | 11        |
| 0002    | Ana Pérez      | alcorcón | 11        |
| 0003    | Isabel Sánchez | leganés  | 12        |
| 0004    | José López     | leganés  | 12        |

(a) ¿Cuál sería el resultado de la consulta?

(b) ¿Cómo se expresaría en álgebra?

(c) ¿Cuál sería su equivalente en SQL?

$$\pi_{\text{nombre, ciudad}}(\sigma_{\text{cod\_grupo} = '12'}(\text{alumnos}))$$

```
SELECT nombre, ciudad  
FROM alumnos WHERE cod_grupo = '12';
```

➤ Ciudad de los alumnos del grupo 12

| ALUMNOS |                |          |           |
|---------|----------------|----------|-----------|
| num_mat | nombre         | ciudad   | cod_grupo |
| 0001    | Juan García    | leganés  | 11        |
| 0002    | Ana Pérez      | alcorcón | 11        |
| 0003    | Isabel Sánchez | leganés  | 12        |
| 0004    | José López     | leganés  | 12        |

(a) ¿Cuál sería el resultado de la consulta?

(b) ¿Cómo se expresaría en álgebra?

(c) ¿Cuál sería su equivalente en SQL?

$$\pi_{\text{ciudad}}(\sigma_{\text{cod\_grupo} = '12'}(\text{alumnos}))$$

```
SELECT DISTINCT ciudad
FROM alumnos WHERE cod_grupo = '12';
```

➤ Nombre de los alumnos y profesores que viven en Leganés

| ALUMNOS |                |          |           |
|---------|----------------|----------|-----------|
| num_mat | nombre         | ciudad   | cod_grupo |
| 0001    | Juan García    | leganés  | 11        |
| 0002    | Ana Pérez      | alcorcón | 11        |
| 0003    | Isabel Sánchez | leganés  | 12        |
| 0004    | José López     | leganés  | 12        |

(a) ¿Cuál sería el resultado de la consulta?

(b) ¿Cómo se expresaría en álgebra?

(c) ¿Cuál sería su equivalente en SQL?

| PROFESORES   |                |         |      |        |
|--------------|----------------|---------|------|--------|
| cod_profesor | nombre         | ciudad  | tipo | sueldo |
| 001          | Sonia Pérez    | madrid  | AU   | 1500   |
| 002          | Sofía Hernán   | leganés | AU   | 1500   |
| 003          | Víctor Álvarez | leganés | TUI  | 2200   |

$$(\pi_{\text{nombre}}(\sigma_{\text{ciudad}='Leganés'}(\text{alumnos}))) \cup (\pi_{\text{nombre}}(\sigma_{\text{ciudad}='Leganés'}(\text{profesores})))$$

```
SELECT nombre FROM alumnos WHERE ciudad ='Leganés'
```

```
UNION
```

```
SELECT nombre FROM profesores WHERE ciudad ='Leganés';
```

- Nombre de los profesores que en la actualidad no imparten ninguna asignatura

| IMPARTIR  |              |             |       |
|-----------|--------------|-------------|-------|
| cod_grupo | cod_profesor | asignatura  | horas |
| 11        | 003          | Inglés      | 2     |
| 12        | 003          | Inglés      | 2     |
| 11        | 001          | Informática | 3     |
| 12        | 001          | Informática | 3     |

(a) ¿Cuál sería el resultado de la consulta?

(b) ¿Cómo se expresaría en álgebra?

(c) ¿Cuál sería su equivalente en SQL?

| PROFESORES   |                |         |      |       |
|--------------|----------------|---------|------|-------|
| cod_profesor | nombre         | ciudad  | tipo | suelo |
| 001          | Sonia Pérez    | madrid  | AU   | 1500  |
| 002          | Sofía Hernán   | leganés | AU   | 1500  |
| 003          | Víctor Álvarez | leganés | TUI  | 2200  |

| GRUPOS    |       |       |
|-----------|-------|-------|
| cod_grupo | curso | turno |
| 11        | 1     | M     |
| 12        | 1     | T     |

$$\pi_{\text{nombre}}(\text{profesores} * (\pi_{\text{cod\_profesor}}(\text{profesores}) - \pi_{\text{cod\_profesor}}(\text{impartir})))$$

SELECT nombre

FROM profesores NATURAL JOIN

(SELECT cod\_profesor FROM profesores

MINUS

SELECT cod\_profesor FROM impartir);

➤ Nombre de los alumnos matriculados en el turno de mañana

| ALUMNOS |                |          |           |
|---------|----------------|----------|-----------|
| num_mat | nombre         | ciudad   | cod_grupo |
| 0001    | Juan García    | leganés  | 11        |
| 0002    | Ana Pérez      | alcorcón | 11        |
| 0003    | Isabel Sánchez | leganés  | 12        |
| 0004    | José López     | leganés  | 12        |

(a) ¿Cuál sería el resultado de la consulta?

(b) ¿Cómo se expresaría en álgebra?

(c) ¿Cuál sería su equivalente en SQL?

| GRUPOS    |       |       |
|-----------|-------|-------|
| cod_grupo | curso | turno |
| 11        | 1     | M     |
| 12        | 1     | T     |

$$\pi_{\text{nombre}}(\text{alumnos} * (\pi_{\text{cod\_grupo}}(\sigma_{\text{turno}='M'}(\text{grupos}))))$$

```
SELECT nombre
FROM alumnos NATURAL JOIN
      (SELECT cod_grupo FROM grupos WHERE turno='M');
```

| GRUPOS    |       |       |
|-----------|-------|-------|
| cod_grupo | curso | turno |
| 11        | 1     | M     |
| 12        | 1     | T     |

(a) ¿Cuál sería el resultado de la consulta?

(b) ¿Cómo se expresaría en álgebra?

(c) ¿Cuál sería su equivalente en SQL?

- Número de grupos que existen en cada curso

$\pi_{\text{curso}, \text{count}} (\text{GROUP BY}_{\text{curso}} \text{ grupos})$

SELECT curso, count('x') FROM grupos GROUP BY curso;

- Cursos que solo tienen un grupo de mañana

$\pi_{\text{curso}} (\sigma_{\text{count}=1} (\text{GROUP BY}_{\text{curso}} (\sigma_{\text{turno}='M'} (\text{grupos})))$

SELECT curso

FROM grupos

WHERE turno='M'

GROUP BY curso HAVING count('x')=1;

- Cursos que solo tienen un grupo y además es de mañana

$\pi_{\text{curso}} (\sigma_{\text{count}=1 \wedge \text{MIN}(\text{turno}='M')} \text{ GROUP BY}_{\text{curso}} (\text{grupos}))$

SELECT curso FROM grupos GROUP BY curso HAVING count('x')=1 AND MIN(turno)='M';