

Master Universitario en Inteligencia Artificial Aplicada

Bases de Datos e Infraestructuras

uc3m

Universidad
Carlos III
de Madrid



0.- Introducción

1.- Diseño de BB.DD. Relacionales (estática)

2.- Operación de BB.DD. Relacionales (dinámica básica)

3.- SQL para consultas analíticas

4.- Otros elementos de las BBDD Relacionales

Estructura del Tema

- 1.- Fundamentos Teóricos: Álgebra Relacional
- 2.- Actualizando datos mediante **SQL**: el LMD
- 3.- Consultar datos mediante **SQL**: la instrucción **SELECT**

- Una vez creada la BD, para poder utilizarla es necesario otro tipo de lenguaje de datos: el lenguaje de manipulación de datos (**LMD**)
- Existen dos tipos de LMD: algebraico (como SQL) y predicativo
- El **Álgebra Relacional** es un lenguaje algebraico (conjunto de operadores) que permiten describir transformaciones de los datos que el diseñador dispondrá para alcanzar un objetivo.
- El SGBD (p.e. Oracle) tendrá una copia (**workspace**) de los datos sobre la que aplicará esas operaciones para alcanzar el resultado deseado (como una receta de cocina).
- Los “ingredientes” son **miembros de un conjunto...**

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 Alatriste	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)

¿Y los libros escritos por Dumas o Pérez-Reverte?

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)

Si la proyección fuera por autor ¿cuántas tuplas tendría la nueva relación?

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 Q el conjunto de los títulos de los libros escritos por Dumas

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

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

$$\rho_Q (\pi_{\text{título}} A)$$

$$Q \equiv \pi_{\text{título}} A$$

*: para tablas temporales de mayor persistencia (sesión) se usa la asignación \leftarrow

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

Aventuras

<i>Signat.</i>	<i>Título</i>	<i>Autor</i>
Dum-1	Los Tres Mosqueteros	Dumas
Per-1	El Capitán Alatriste	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



Aventuras \cup *Novelas*

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

Libros de aventuras o 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 Alatriste	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 Alatriste	P-Reverte



Aventuras \cap *Novelas*

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

Novelas de aventuras

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

Aventuras

<i>Signat.</i>	<i>Título</i>	<i>Autor</i>
Dum-1	Los Tres Mosqueteros	Dumas
Per-1	El Capitán Alatriste	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 Alatriste	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

Libros de aventuras que no se consideran novelas

Producto cartesiano: tuplas de ambas en todas las combinaciones

Comidas

<i>Especialidad</i>	<i>Región</i>
Gazpacho	Andalucía
Cocido	Castilla
Pote	Galicia

Vinos

<i>Denominación</i>
Rioja
Rueda



Comidas x Vinos

<i>Especialidad</i>	<i>Región</i>	<i>Denominación</i>
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

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

<i>Nombre</i>	<i>Edad</i>
Fulano	29
Mengano	18
Zutano	9

Películas

<i>Título</i>	<i>Calificación</i>
Matrix	18
Sólo tu	0



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

<i>Nombre</i>	<i>Edad</i>	<i>Título</i>	<i>Calificación</i>
Fulano	29	Matrix	18
Fulano	29	Sólo tu	0
Mengano	18	Matrix	18
Mengano	18	Sólo tu	0
Zutano	9	Sólo tu	0

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

Empleados

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

Coches

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

Nombre de los empleados que son dueños de algún coche

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



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



<i>Nombre</i>
Fulano
Zutano

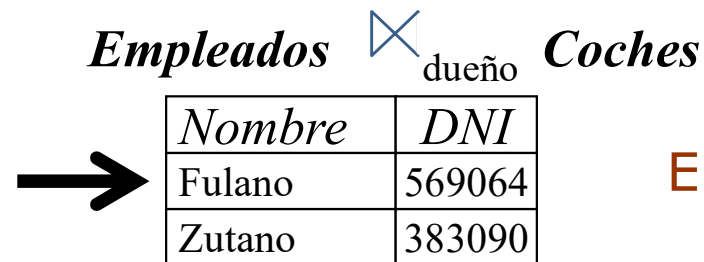
- observar que se elimina la columna repetida

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

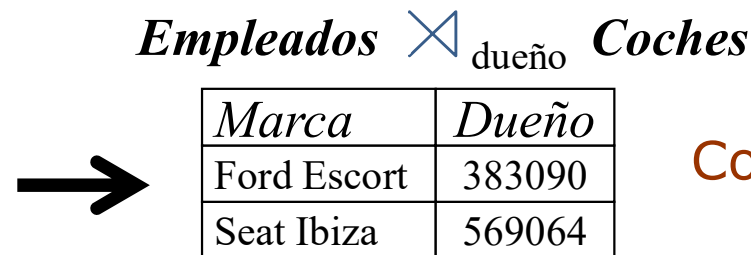
<i>Nombre</i>	<i>DNI</i>
Fulano	569064
Mengano	434848
Zutano	383090



Empleados con coche

Coches

<i>Marca</i>	<i>Dueño</i>
Ford Escort	383090
Seat Ibiza	569064
Honda Civic	868015



Coches de nuestros empleados

Notaciones: igual que en la combinación natural

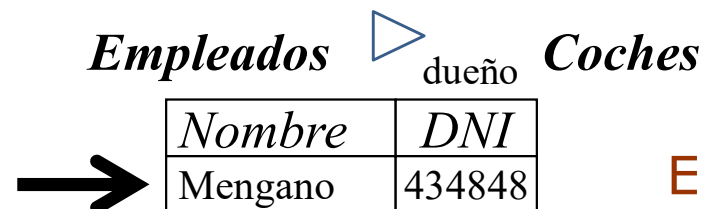
Equivalencia: $\ltimes \equiv \pi \rtimes$

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

Notación: left antijoin: \triangleright right antijoin: \triangleleft

Empleados

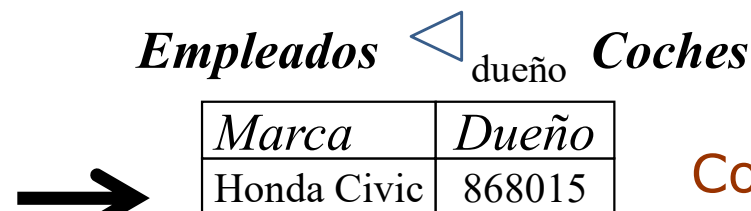
<i>Nombre</i>	<i>DNI</i>
Fulano	569064
Mengano	434848
Zutano	383090



Empleados sin coche

Coches

<i>Marca</i>	<i>Dueño</i>
Ford Escort	383090
Seat Ibiza	569064
Honda Civic	868015



Coches que no son de nuestros empleados

Notaciones: igual que en la combinación natural

Equivalencia: $A \triangleright B \equiv A - (A \bowtie B)$

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 ω).

left outer join: $]^*$ ó  **right outer join:** $^*[$ ó  **full outer join:** $]^*[$ ó 

Empleados

Nombre	DNI
Fulano	569064
Mengano	434848
Zutano	383090

Coches

Marca	Dueño
Ford Escort	383090
Seat Ibiza	569064
Honda Civic	868015

Empleados y sus coches,
en su caso

Empleados  *Coches*



Nombre	DNI	Marca
Fulano	569064	Seat Ibiza
Zutano	383090	Ford Escort
Mengano	434848	NULL

Coches con su dueño si es
uno de nuestros empleados.

Empleados  *Coches*



Marca	Dueño	Nombre
Ford Escort	383090	Zutano
Seat Ibiza	569064	Fulano
Honda Civic	868015	NULL

Empleados y coches,
asociados como proceda

Empleados  *Coches*



Nombre	DNI	Marca
Fulano	569064	Seat Ibiza
Zutano	383090	Ford Escort
Mengano	434848	NULL
NULL	868015	Honda Civic

P

El **renombrado** es como *utilizar vistas temporales*, es decir, *workspaces* temporales **constantes** auxiliares que facilitan la consulta principal. Así, mediante el renombrado, se puede aplicar la máxima de computación acuñada por Julio César: *'Divide y Vencerás'*. *Las consultas anidadas son costosas, y frecuentemente pueden ser evitadas* gracias al renombrado.



Θ

*

La **combinación** (*join*) se deriva de dos operadores primitivos ($\sigma \times$), pero al combinar se realizan al mismo tiempo. Imagina que tienes 100 lectores y 20 libros calificados por edades, y necesitas listar todas las combinaciones posibles (200). ¿Tú escribirías las 2000 posibles y luego tacharías las 1800 incorrectas? ¿O sólo escribirías directamente las correctas?



Orden: conjunto ordenado (lista) es el resultado de aplicar un orden ($\text{ORDER BY}_{\text{orden}}$ ó \top_{orden}) sobre una relación. Si *orden* omite la especificación de orden (sólo explicita atributos) se sobreentiende el orden \leq sobre números y el orden lexicográfico sobre caracteres.

Si se opera una lista ordenada se obtiene una relación (sin orden).

Sobre una lista ordenada se pueden aplicar funciones (analíticas) de agregación: *first*, *last*, y *rank(value)*, y la pseudocolumna *numrow*.

Personas

<i>Nombre</i>	<i>Edad</i>	<i>Nacionalidad</i>
Fulano	29	Español
Mengano	49	Español
John Doe	73	Inglés
Smith	14	Inglés
Zutano	3	Español
Pelancejo	25	Español

$\pi_{\text{first, last, rank (29)}}$ **order by** *edad* (*Personas*)



<i>first</i>	<i>last</i>	<i>rank(29)</i>
3	73	4

• *orden inverso:* $\text{ORDER BY}^{\text{DESC}}$ ó \top

Las instrucciones SQL3 de manipulación pueden operar de tres modos:

1. Interactivo (proporcionando instrucciones SQL directamente).
2. SQL embebido: instrucciones imbuidas en lenguaje anfitrión (C, JAVA,...).
3. Módulos: llamadas explícitas a procedimientos desde procesos externos.

Operaciones de Actualización

- Inserción de tuplas (INSERT)
- Borrado de tuplas (DELETE)
- Modificación de tuplas (UPDATE)

Operaciones de Recuperación

- Consulta o *Query* (SELECT)

Inserción directa de valores:

```
INSERT INTO <nombre_de_la_tabla>
    [ (<nombre_columna1>, ..., <nombre_columnaN>)]
VALUES (<valor1>, ..., <valorN>) [, (row2...) [...]];
```

Inserción masiva (desde otra tabla):

```
INSERT INTO <nombre_de_la_tabla>
    [ (<nombre_columna1>, ..., <nombre_columnaN>)]
SELECT ... ;
```

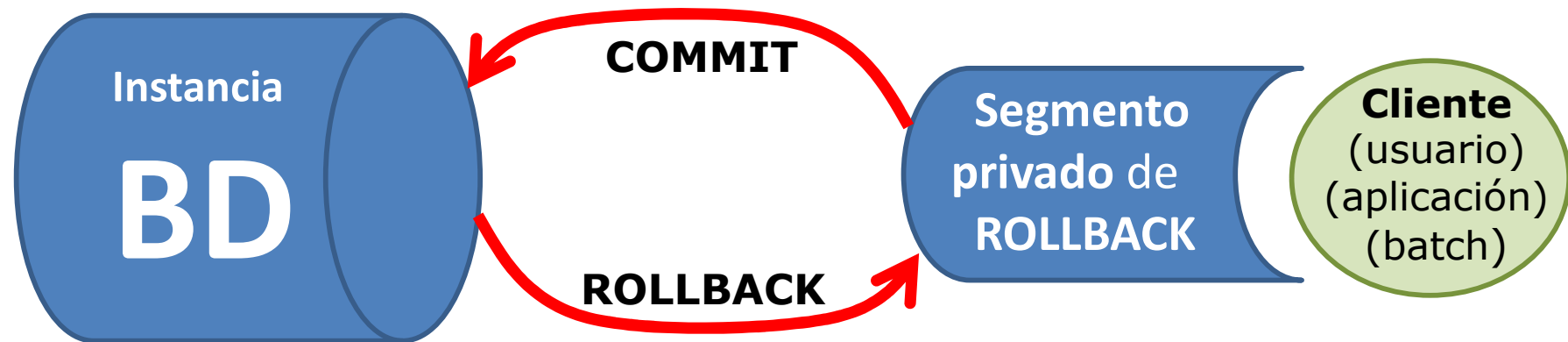
```
DELETE [FROM] <tabla> [<alias>]  
[WHERE <condicion>]
```

```
DELETE stocks;  
/* ¿peligroso? ¡No tanto como DROP o TRUNCATE! */  
DELETE FROM stocks WHERE f_caducidad<SYSDATE;
```

```
UPDATE <tabla> SET  
{ <columna> = {<expresion>|<subquery>}  
| (<columna> [{,<columna>}])=<subquery>}  
[WHERE <condicion>]
```

```
UPDATE persons SET age=25 WHERE name='Javi';  
UPDATE stock a SET price = (SELECT a.price*b.change  
                             FROM euro_quote b  
                             WHERE b.name='peseta');
```

- **Transacción**: conjunto de instrucciones de **actualización** que deben ser llevadas a cabo de modo atómico (como conjunto, “o todo o nada”)
- **Instrucciones**: COMMIT (realizar) y ROLLBACK (deshacer)
 - COMMIT [WORK]
 - ROLLBACK [WORK] [TO [SAVEPOINT] <savepoint>]
 - SAVEPOINT <savepoint>



[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**]] ;

- la proyección se refleja en la <lista de selección>
Lista de datos (del workspace) separados por comas.
- Puede ser todo el área de trabajo (*) o bien incluir:
 - **atributos** del esquema de relación del área de trabajo
 - **pseudo-columnas**, como ROWNUM y table.ROWID,...
 - **constantes** (como 1 o 'X') y **variables ligadas** (:NEW, ...)
 - **funciones**: aplicadas sobre lo anterior (o nularias)
 - aritméticas (+, -, ...), strings (||, SUBSTR, ...), codificación (CASE, NVL, ...), conversión (TO_CHAR, ...), sistema (SYSDATE, USER, ...), ...
 - de agregación (reciben un colectivo y devuelven un solo valor)
 - funciones compiladas (de usuario o procedentes de paquetes)
- Admite renombrado (alias): ... *elemento* [AS] *alias* ...

- **workspace**: es una tabla temporal (vinculada al origen)
- La cláusula FROM define el *área de trabajo* (que es una tabla)
 - Puede componerse de una tabla base, o varias tablas combinadas.
 - Dado que el área de trabajo es una tabla, un caso particular de tabla en la cláusula FROM es otro área de trabajo, es decir, una consulta (subquery).
 - Admite renombrado (obligatorio en *self-joins*): ... *tabla alias* ...
 - Existen diversas combinaciones:
 - **Combinación Elemental: el Producto Cartesiano**

... FROM Gente CROSS JOIN Clientes... \equiv FROM Gente, Clientes...

➤ Consultas: clausula FROM y combinación natural

- **Combinación General JOIN:** todas las columnas de ambas tablas

```
... FROM people [INNER] JOIN clients  
      [ ON (<col_a>=<col_b> [AND...] ) ]
```

- sin especificación: equijoin por todos los pares de columnas que reciben el mismo nombre en ambas tablas.
- con cláusula ON: solo las tuplas que cumplan la expresión (de =, !=, <, >, ...).

- **Combinación Natural:** no duplica las columnas incluidas en la igualdad

```
... FROM X NATURAL [INNER] JOIN Y...
```

Combinación Natural por pares de columnas que se llamen igual en ambas tablas

```
... FROM X [INNER] JOIN Y USING (<columns>) ...
```

Combinación Natural por pares de columnas especificados (que se llaman igual...)

- **Combinación sin pérdidas, Combinación Externa (outer):**

```
...FROM Gente {LEFT|RIGHT|FULL} [OUTER] JOIN Clientes  
      [USING <columnas> | ON <col_a>=<col_b> [AND...]]
```

➤ Mismo uso que la (inner) JOIN; 'outer' es opcional (recomendable por claridad)

- ... FROM X LEFT OUTER JOIN Y...

Combinación Externa por la izquierda (respeta todas las tuplas de la primera tabla)

- ... FROM X RIGHT JOIN Y...

Combinación Externa por la derecha (respeta todas las tuplas de la segunda tabla)

- ... FROM X FULL OUTER JOIN Y...

Combinación Externa completa (mantiene todas las tuplas de ambas tablas)

- la **expresión condicional** (**WHERE**) puede ser...
- una comparación de expresiones (`=`, `!=`, `<`, `>`, `<=`, `>=`)
 - los operandos pueden ser valores, expresiones, o incluso una **subquery** (!)
 - comprobación de inclusion test (en un rango):
`<expresión> [NOT] BETWEEN <expresión> AND <expresión>`
 - comprobación de valor nulo: `<expresión> IS [NOT] NULL`
 - test de semejanza (*patrón*): `<expresión_char> [NOT] LIKE <patrón>`
 - expresión lógica: {NOT, AND, OR} a partir de otras expresiones condicionales
 - test de existencia: `EXISTS subquery`
 - test de inclusión (en un conjunto dado, o en una **subquery**):
`<expresión> [NOT] IN {<expresión_list> | subquery}`

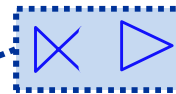
SUBQUERIES en la cláusula WHERE: pueden ser **ineficientes**

- EXISTS se detiene en el primer encaje, pero se ejecuta **anidado** siempre.
- IN puede optimizarse (no ejecutar anidado); conviene moverlo a la cl. WITH
- NOT IN puede usarse en anti-combinaciones (si se hace de modo eficiente)

[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]] ;

right

left





- Para practicar SQL, mira esta página web:
 - sqlzoo.net
- Apps (en Android / IOS)
 - SoloLearn
 - SQL Practice PRO (exercises)
 - Pocket PLSQL, SQL Tutorial, Learn SQL queries,